

Contents

[Overview](#) [Quick Start](#) [Revision History](#)

MACRO MANIA COMMANDS:

The 2 items in **boldface** are [important commands](#) used most often.

Activate	Beep
Branch	ChDir
Clipboard_Set	Clipboard_SetFile
Clipboard_Restore	Clipboard_Save
End	Exit
GetInput	Macro
Minimize	Msg_Ok
Msg_OkCancel	Open
Pause	PlayWav
Rem	Repeat
Repeat_Prompt	Restore
Run	Send
SendInput	SendNow
WinExit	

TRICKS/EXAMPLES/MISCELLANEOUS:

[Important Tricks and Techniques](#)

[A Simple Example](#)
[Scheduling Macros](#)

REGISTRATION AND TECHNICAL SUPPORT:

[Contacting the Author for Technical Support](#)
[About the Author](#)
[Benefits of Registering this Program](#)
[How to Register this Program](#)

Overview

Description:

Macro Mania is a powerful, yet very easy-to-use macro program that works with *any* windows program! It enables you to run programs (or switch to currently running programs) and then send keystrokes to that program with one simple command invoked from the *Macro Mania* command center, just as if you had manually started the program and typed in the keys yourself. Additionally, it can be told to switch back and forth between programs to transport data and you can easily schedule macros to run unattended or to use as a convenient reminder program.

If you are like many of the *Macro Maniacs* that continue to emerge, you will find yourself creating macros for all kinds of tasks that require repetitive keystrokes. Some of them will be permanent, crucial parts of your daily activities and others will be quick, temporary macros that still manage to save you a lot of time and keystrokes. *Macro Mania* continues to receive positive comments from professional software reviewers and numerous Windows users alike, you will not be alone <smile>.

An example of the *many uses for Macro Mania*:

I have a contact management program that stores peoples names, addresses, e-mail addresses, etc. Occasionally I want to send an *e-mail newsletter* to certain people in my list. Because I do not want to have to maintain two lists (one in my e-mail program and one in my regular contact management program), and I want to send a message to most, but not all of the people in my list, I need a way to use my contact management program to select the people I want, then have it send e-mail messages to only those people I have selected.

Using *Macro Mania*, I can have it send the keystrokes to my contact management program that filters the list for me and moves to the first entry in the filtered list. Then it starts my e-mail program, sends the keystrokes I would use to compose a new message, automatically plugs the names and e-mail addresses, switches over and captures the text of a letter I have already typed out in Windows Notepad, switches back to the e-mail program and pastes the contents of the letter, saves the message, then moves on to the next entry in my database to do the same thing. With one click of the mouse I have maneuvered among three different Windows programs, moving information from one program to the other, and have sent dozens of keystrokes! What would take a good *cut-and-paste artist* several minutes (or hours depending on the number of entries) to normally do now takes only a few seconds!!!

-ENJOY!!!

Activate

Description:

Brings an already running copy of a program to the front of your Windows environment. (See [Starting Programs](#) for more discussion.)

Syntax 1:

```
ACTIVATE [Title Bar]
```

Title Bar is the exact caption that is found at the top of the programs windows (like the *Macro Mania Help* title bar shown above).

Syntax 2:

```
Activate2 [Title Bar Text]
```

Title Bar Text is any text within a title bar. This is useful where only a partial match is needed or desired. While only a partial match is needed, be sure to use the same capitalization that appears in the partial string that you wish to find.

Example 1:

```
ACTIVATE NOTEPAD - thefile.txt
```

Example 2:

```
ACTIVATE2 Notepad
```

Notes:

An error occurs when you use the Activate command and the Title Bar is incorrect or the program is not already running. Activate2 is a powerful extension of the Activate command when the Title Bar changes (as when it also has the name of an open file).

It does not change whether it is maximized or minimized. If you are not sure if the macro might be activating a program that has been minimized, one trick is to send *Alt-space, R* using the SEND command (translated as *SEND %{}R*). This usually causes the program to **Restore** to a *normal* window if it is minimized. (See [manipulate the state of windows](#) for more discussion.)

The Activate command, though still useful, is generally not needed since the [Run](#) command can also pull currently running programs to the front and, if not found, automatically start the program for you.

Beep

Description:

Sends a beep to any normal PC Speaker.

Syntax:

Beep

Notes:

Useful for bringing attention to the computer. Particularly useful to put at the end of long macros or immediately before a pause to indicate you may begin typing.

ChDir

Description:

Like the DOS command, this changes the working directory to the directory you specify. Issue this command if the program you wish to run using the [Run](#) command needs to find support files. (Issuing this command before the RUN command is like specifying the *Working Directory* when you create a program item in the Windows Program Manager.)

Syntax:

CHDIR [path]

path is a string expression that identifies which directory becomes the new default directory. This argument must contain fewer than 128 characters and has the following full syntax:

CHDIR [drive:] [\]directory[\directory] . . .

The argument drive is an optional drive specification; the argument directory is a directory name. If you omit drive, CHDIR changes the default directory on the current drive.

Notes:

CHDIR may be abbreviated with *CD*.

End

Description:

Causes the Macro Mania program to completely end.

Syntax:

END

Notes:

This command is especially useful if you wish to run a macro, perhaps from another program, then want Macro Mania to stop running entirely. To simply exit an individual macro, see the [Exit](#) command.

Exit

Description:

Causes the macro to exit and quit running.

Syntax:

EXIT

Notes:

This command is especially useful for debugging your macros or dynamically changing where the macro should end without having to comment a lot of lines or delete part of a macro script you may want to use later. For debugging, if you are not sure where your macro is not working correctly, place this command a few lines above where you think the error is happening. Then gradually move it down one line at a time until the line with the error is identified.

To end the Macro Mania program entirely, see the [End](#) command.

GetInput

Description:

Gives a prompt in the middle of a macro, allowing for input of small amounts of dynamic data that can later be sent with the [SendInput](#) command.

Syntax:

GetInput [Prompt]

Prompt is any text (approximately 255 characters maximum) that you want displayed when the Input Box is shown. (Usually a description of what should be typed in the box.)

Notes:

This command is useful for getting file names, search strings, and other data that may change each time a macro is run. Once you use the GetInput command, it is necessary to use either the Run or Activate command and later a SendInput command to send the information that was put into the Input Box. If no input is put into the box or the cancel button is selected, the macro asks if you wish to abort the macro. Yes causes the macro to stop, no causes the macro to continue without any input. (When the SendInput command is encountered without any text to be sent, it will ask a similar question.)

Example:

```
GETINPUT Enter the file to open.  
RUN notepad.exe  
SEND %Fo  
SENDINPUT  
SEND {enter}
```


Macro

Description:

Runs another macro you have programmed, allowing you to chain macros together.

Syntax:

MACRO [n]

n is a number indicating the number of the macro you wish to run. Refer to the number displayed in the "Add/Edit Macros" window.

Example:

MACRO 3

Notes:

Macro Mania prevents you from using the MACRO command to call itself, otherwise it would get into a continuous loop. It is possible to call a macro that, in turn, calls the macro that called it, so be sure you refer to the correct macro number and avoid getting yourself into such a continuous loop.

The macro command essentially allows you to create subroutines within your macros. By putting repeat commands in certain macros, you can also create nested loops, etc. too. If you only plan to use a macro as a subroutine from another macro, you can turn its visible property to off from where you select the icons for buttons, preventing a macro from showing up as a button on the main Macro Mania screen.

Important Note: While adding new macros will not affect the position of your macros since new macros are simply appended to your macro list, deleting macros in front of existing macros causes them to be resequenced by -1. For example, you have 10 macros and delete the 7th, now the 8th becomes the 7th, the 9th becomes the 8th, and so on. Macro Mania will *automatically* make the adjustments for you in your other macros when needed.

Minimize

Description:

Causes the Macro Mania command center to minimize itself (become an icon). This is useful if you wish to see what is happening as the macro executes, or plan to immediately doing something manually in a program after a macro is finished.

Syntax:

MINIMIZE

Notes:

This command is more helpful if used at the beginning of a macro, before a Run or Activate command has been issued. Use the [Restore](#) command to return the window to its *Normal* size.

See [Important Notes for Windows 95](#)

Pause

Description:

Use this command to pause a macros execution. Useful if you wish to give yourself time to manually type in some keystrokes or otherwise wish to prevent the macro script from immediately continuing.

Syntax:

PAUSE [n] <show>

n is a number indicating the number of seconds you wish the macro to pause for. The optional word *SHOW* may follow the command to indicate that a small box counting down the number of seconds remaining should be displayed. The display box may be moved, and an option to immediately resume or cancel the macro is also available when the *SHOW* parameter is used.

Example:

PAUSE 3 show

[Important Notes for Windows 95](#)

Rem

Description:

Use before a command or text so that line is ignored when the macro is executed.

Syntax:

REM <command> or <text>

command or *text* is either a command you do not want to be used in your macro or some text, such as a comment within the macro to help you better identify what the macro is intended to do.

Example:

REM Go to the bottom of the notepad.

Notes:

You may also use a single quote in place of the Rem statement.

Repeat

Description:

Causes a macro to repeat.

Syntax:

```
REPEAT [n]
```

n is a number indicating the number of times the macro should repeat.

Example:

```
REPEAT 3
```

Notes:

This command should be put on the first line of your macros to help you remember its value. (You may also wish to include its value as part of your macro description.)

The repeat option can also be set dynamically by setting the [Temporary Repeat Value](#) or using the [Repeat Prompt](#) command.

See [Stopping a Macro](#) for strategies to help you stop macros if needed.

Restore

Description:

Use after issuing the [Minimize](#) command to make the Macro Mania window normal again.

Syntax:

RESTORE

Notes:

Using the restore command also puts the focus back to Macro Mania. Any keystrokes that are sent with the Send command will be sent to **Macro Mania** unless a Run or Activate command is used to put focus back to another program.

See [Important Notes for Windows 95](#)

Run

Description:

Starts a program or brings an existing program to the front of the Windows environment. (See the notes section below for more details.)

Syntax:

RUN [program], <parameters>

program is the path and name of the executable part of the program. If the path is not included, the current directory, Windows subdirectory, and DOS path are searched for a copy of the program.

parameters are any valid parameters the program can accept when started.

Example:

The following example runs Windows Notepad, automatically opening the file *MYNOTES.TXT*.

```
RUN c:\windows\notepad.exe, mynotes.txt
```

Notes:

When the Run command is used, it searches your Windows environment and detects if a program is already running. If the program is found, it will simply pull the current instance of that program to the front, if it is not found it will automatically start a new copy of it for you.

Usually this is exactly what you want; however, there may be times where you would like to start a new copy of a program regardless of whether a copy is already running (this only works with smaller programs that allow themselves to be run more than once). For example, you may want a fresh instance of notepad, regardless of whether you are running another copy of notepad.

If this is the case, by simply leaving off the path of the program, Macro Mania will not find the previous instance of the program and will run a new copy of it. Note that if the program is not in the current directory, the Windows directory, or in the environment path, you may need to use the [CHDIR](#) immediately before the Run command so that the executable or its support files will be found.

If *program* is not found, an error occurs.

Send

Description:

Sends keystrokes to a program as if you had typed them manually. (Issue this command only after you have issued either the [activate](#) command or [run](#) command.)

Syntax:

SEND [keystrokes]

keystrokes are the keys you would press if you were sitting in a program and typing them manually.

If you want to represent more than one character, append each additional character to the one preceding it. To represent the letters x, y, and z, use *xyz*.

You will be using the Send command a lot. Once you master how to use the Send command to its full potential, you can do virtually anything with your macros! You may wish to print this section for easy reference.

Syntax Details:

The plus sign (+), caret (^), percent sign (%), tilde (~), and parentheses () have special meanings to the Send command. To specify one of these characters, enclose it inside braces. For example, to specify the plus sign, use {+}. Brackets ([]) have no special meaning to the Send command, but you must enclose them in braces as well, because in other programs brackets do have special meaning. To send brace characters, use {{}} and {}.

To specify characters that are not displayed when you press a key (such as Enter or Tab) and keys that represent actions rather than characters, use the codes shown below:

<u>Key</u>	<u>Code</u>
Backspace	{BACKSPACE} or {BS} or {BKSP}
Break	{BREAK}
Caps Lock	{CAPSLOCK}
Clear	{CLEAR}
Del	{DELETE} or {DEL}
Down Arrow	{DOWN}
End	{END}
Enter	{ENTER} or ~
Esc	{ESCAPE} or {ESC}
Help	{HELP}
Home	{HOME}
Ins	{INSERT}
Left Arrow	{LEFT}
Num Lock	{NUMLOCK}
Page Down	{PGDN}
Page Up	{PGUP}
Print Screen	{PRTSC}
Right Arrow	{RIGHT}
Scroll Lock	{SCROLLLOCK}
Tab	{TAB}
Up Arrow	{UP}
F1 - F16	{F1} - {F16}

SpaceBar { } << 1 space between the brackets

To specify keys combined with any combination of Shift, Ctrl, and Alt keys, precede the regular key code with one or more of the following codes:

<u>Key</u>	Code
Shift	+
Ctrl	^
Alt	%

To specify that Shift, Ctrl, and/or Alt should be held down while several other keys are pressed, enclose the keys code in parentheses. For example, to have the Shift key held down while E and C are pressed, use `+(EC)`. To have Shift held down while E is pressed, followed by C being pressed without Shift, use `+EC`.

To specify repeating keys, use the form `{key number}`; you must put a space between key and number. For example, `{LEFT 42}` means press the Left Arrow key 42 times; `{h 10}` means press h 10 times.

The macro will not continue until the SEND command is complete. This feature ensures that the macro does not get ahead of itself. If you are sending a lot of keystrokes, you may wish to break them up into several SEND commands so that the program that is receiving the keystrokes has time to process them before another group of keystrokes are sent.

Limitations: The Send command can not send keystrokes to a program that is not designed to run in Microsoft Windows. Send also can not send the Print Screen (PRTSC) key to any program.

See [Making Sure the Send Command works](#) for more discussion.

SendInput

Description:

Sends the most recently entered text retrieved from the [GetInput](#) command.

Syntax:

SendInput

Notes:

If no text is in the buffer from the GetInput command, the macro asks if you wish to abort the macro. Yes causes the macro to stop, no causes the macro to skip the current SendInput command. Like the Send command, an Activate or Run command must be issued before the SendInput command. A GetInput command should also precede the SendInput command.

Example:

```
GETINPUT Enter the file to open.  
RUN notepad.exe  
SEND %Fo  
SENDINPUT  
SEND {enter}
```

SendNow

Description:

Sends the current (system) date and/or time as specified with the *format* parameter. (Like the send command, this command should only be issued after you have issued either the [activate](#) command or [run](#) command.)

Syntax:

SENDNOW [format]

The following table shows the characters you can use to create user-defined date/time formats (examples are at the bottom):

Character Meaning

c	Send the date as ddddd and send the time as t t t t t, in that order. Only date information is sent if there is no fractional part to the date serial number; only time information is sent if there is no integer portion.
d	Send the day as a number without a leading zero (1-31).
dd	Send the day as a number with a leading zero (01-31).
ddd	Send the day as an abbreviation (Sun-Sat).
dddd	Send the day as a full name (Sunday-Saturday).
ddddd	Send a date serial number as a complete date (including day, month, and year) formatted according to the Short Date setting in the International section of the Windows Control Panel. The default Short Date format is m/d/yy.
dddddd	Send a date serial number as a complete date (including day, month, and year) formatted according to the Long Date setting in the International section of the Control Panel. The default Long Date format is mmmm dd, yyyy.
w	Send the day of the week as a number (1 for Sunday through 7 for Saturday.)
ww	Send the week of the year as a number (1-53).
m	Send the month as a number without a leading zero (1-12). If m immediately follows h or hh, the minute rather than the month is sent.
mm	Send the month as a number with a leading zero (01-12). If m immediately follows h or hh, the minute rather than the month is sent.
mmm	Send the month as an abbreviation (Jan-Dec).
mmmm	Send the month as a full month name (January-December).
q	Send the quarter of the year as a number (1-4).
y	Send the day of the year as a number (1-366).
yy	Send the year as a two-digit number (00-99).
yyyy	Send the year as a four-digit number (100-9999).
h	Send the hour as a number without leading zeros (0-23).
hh	Send the hour as a number with leading zeros (00-23).
n	Send the minute as a number without leading zeros (0-59).
nn	Send the minute as a number with leading zeros (00-59).

- s** Send the second as a number without leading zeros (0-59).
- ss** Send the second as a number with leading zeros (00-59).
- t t t t t** Send a time serial number as a complete time (including hour, minute, and second) formatted using the time separator defined by the Time Format in the International section of the Control Panel. A leading zero is sent if the Leading Zero option is selected and the time is before 10:00 A.M. or P.M. The default time format is h:mm:ss.
- AM/PM** Use the 12-hour clock and send an uppercase AM with any hour before noon; send an uppercase PM with any hour between noon and 11:59 PM.
- am/pm** Use the 12-hour clock and send a lowercase AM with any hour before noon; send a lowercase PM with any hour between noon and 11:59 PM.
- A/P** Use the 12-hour clock and send an uppercase A with any hour before noon; send an uppercase P with any hour between noon and 11:59 PM.
- a/p** Use the 12-hour clock and send a lowercase A with any hour before noon; send a lowercase P with any hour between noon and 11:59 PM.
- AMPM** Use the 12-hour clock and send the contents of the 1159 string (s1159) in the WIN.INI file with any hour before noon; send the contents of the 2359 string (s2359) with any hour between noon and 11:59 PM. AMPM can be either uppercase or lowercase, but the case of the string sent matches the string as it exists in the WIN.INI file. The default format is AM/PM.

The following are examples of date and time formats:

Format	What is sent if the current day is December 9, 1995 and the time is 8:50 pm.
m/d/yy	12/9/95
d-mmmm-yy	9-December-95
d-mmmm	9 December
mmm-yy	December 95
hh:mm AM/PM	08:50 PM
h:mm:ss a/p	8:50:35 p
h:mm	20:50
h:mm:ss	20:50:35
m/d/yy h:mm	12/9/95 20:50

A Simple Example

Maybe the best way to explain the easy syntax that Macro Mania uses is with an example. Here is an example of a simple macro that will start Windows Notepad and then type the message *Macro Mania is powerful, yet easy-to-use!*:

RUN Notepad.exe

SEND Macro Mania is powerful, yet easy-to-use!

Starting Programs

The first part of a macro needs to start the program that will receive the keystrokes sent from the macro. There are two commands you can use to start a program, either the [Run](#) command, or the [Activate](#) command.

The method used most often is the [Run](#) command to start programs since it can either find a currently running copy and bring it to the front, or run a new instance of the program if it is not found. Another option, though less useful, is to use the [Activate](#) command.

Tricks and Techniques

[Helpful Hints for Beginners](#)

[Making Sure Keystrokes Get Received](#)

[Important Notes for Windows 95](#)

[Recording Keystrokes](#)

[Launching Macros](#)

[Manipulating the State of Windows](#)

[Launch a Particular Macro at Startup](#)

[Adding Icons to the icon list](#)

[Activating Buttons](#)

[Combine a Search Feature with a Macro](#)

[Save Space in Your Macros](#)

[Mouse Movement](#)

[Sending a Space \(spacebar\)](#)

[General Keyboard Navigation](#)

[Stopping a Macro](#)

[Fast Access to Edit a Macro](#)

[Creating and Opening New Macro Files](#)

[Password Protecting Your Macros](#)

[Subroutines within macros \(looping, etc.\)](#)

[Making sure the SEND command works](#)

[Using Old Macro Files \(prior to v3.5\)](#)

[DOS Batch Files](#)

[Resquencing Macros](#)

How to Register

This program is distributed as [shareware](#), which means you must pay for it if you continue to use it. Although many people have suggested that I could charge much more, I have made the registration price low to encourage more registrations. Please do not think your registration is not important to me (your registration is important to me). In fact, there are several [benefits](#) I can offer you when you register.

Be sure to see [Getting the Latest Version...](#) to make sure you have the latest version of Macro Mania.

You must purchase a license for each PC you will use Macro Mania on (note that multiple licenses are available for a significantly discounted price):

1 copy = \$29.95 << need only **ONE** registered copy.
2 copies = \$20.00/copy << **most popular, very good deal.**
3 - 5 copies = \$15.00/copy << need more than 2, but less than 6.
6 or more copies = \$10.00/copy << best deal for site licenses.

For your convenience there are several ways to register:

1) If you have a valid Visa, Mastercard or Discover card, you can contact *NorthStar Solutions for orders only* via any of the following methods (please note that you are registering **Product 933** and how many copies you need when you contact them so they can immediately give you the appropriate [registration code](#)):

[INTERNET ORDERS](#)

Just point your Web Browser to
<http://nstarsolutions.com/85.htm>
and fill out the online order form--fast, easy and secure!

[PHONED ORDERS](#)

Calls are taken 10 am - 8 pm, EST, Monday thru Saturday.
1-800-699-6395 (From the U.S. only.)
1-803-699-6395

[FAXED ORDERS](#)

Available 24 hours. International & business orders are encouraged.
1-803-699-5465

[E-MAILED ORDERS](#)

CompuServe: **starmail**
America Online: **starmail**
Internet: **starmail@compuserve.com**

[MAILED ORDERS](#)

You may register with a check or money order.
Make them payable to "**NorthStar Solutions**" and send them to:
PO Box 25262, Columbia, SC 29224

Please provide (or be prepared to provide) the following information:

- * The program you are registering (Product 933).
- * Your mailing address.
- * Your Visa, MasterCard, or Discover # and its expiration date (if using credit card).
- * Your E-Mail address (so NorthStar Solutions can send you an E-Mail confirming your order and so I can contact you easily with any important follow-up information, upgrade announcements, etc.).

2) If you have a CompuServe account and would like to register online via CompuServe, you can use their [SWREG](#)

services to do so.

3) You may also register with a check or money order. Make them payable to **NorthStar Solutions** and send them to:

P.O. Box 25262
Columbia, SC 29224

(For fastest delivery of your [registration code](#), provide either an Email address (preferred) or a fax number where it can be sent. Otherwise one will be promptly sent via snail mail.)

"Shareware"

The essence of "user-supported" software (a.k.a. "Shareware") is to provide computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products (or improve existing ones). If you find this program useful and continue to use it after a reasonable trial period, you must make a registration payment to obtain a legal copy.

Contacting the Author for Technical Support

Technical support is available if you experience any problems using this program. Contact me using any of the methods listed below, and I will try to help. If any problem reported is found to be a significant "bug", I will try to fix it as soon as possible and send you an upgrade. (Please make sure you have read the Help I have provided with this program first. Since much of the way your macros behave is in the way you tell them to, make sure the macros are not merely doing what you asked <grin>.)

E-mail contact is preferred because it is fast, inexpensive, and accurate. I check Email *several* times a day and often respond very quickly (and in more detail, if needed) **via e-mail**:

CompuServe - **starmail**
America Online - **starmail**
Internet - **starmail@compuserve.com**

If you do not have an e-mail account, the second best way to get help is to fax a scenario of the problem you are having (the more details you can provide the better I can help you). I can be reached via fax 24 hours at **(803) 699-5465**. You can write to me at:

Jeff Camino
P.O. Box 25262
Columbia, SC 29224

If you have an absolute emergency, you can also call (803) 699-6395. I am usually available from 10 am to 8 pm, EST. **PLEASE DO NOT CALL DURING HOURS OUTSIDE THIS RANGE, AND USE EMAIL IF POSSIBLE.**

Run Command

Starts a program or brings an existing program to the front of the Windows environment.

Activate Command

Brings an already running copy of a program to the front of your Windows environment.

Restore Command

Use after issuing the MINIMIZE command to make the Macro Mania window a "normal" size.

Temporary Repeat Value

Set via the "Settings" pull-down menu in the main form, the Temporary Repeat Value will be set for ALL macros and overrides any internal REPEAT value you may have set using the REPEAT command. You should always make sure a macro is functioning the way you intended before setting any repeat value, temporary or otherwise.

Minimize Command

Causes the Macro Mania command center to minimize itself (become an icon). This is useful if you wish to see what is happening as the macro executes, or plan to immediately start doing something manually in a program after a macro is finished.

Activate Command

Brings an already running copy of a program to the front of your Windows environment.

Run Command

Starts a program or brings an existing program to the front of the Windows environment.

Quick Start

The easy-to-use commands and syntax for *Macro Mania* will have you creating useful macros in minimal time. The nice thing is that once you learn the [2 important commands](#) of *Macro Mania*, you will have the ability to create macros for *all* your Windows programs!

If you are like most people, you will find using the interface for *Macro Mania* is very intuitive. However, to get a quick start you will need to refer to the few commands that it uses and their syntax in this help section. For now, assume you can refer to them easy enough (because you can). To create a new macro follow these steps:

1. From the main *Macro Mania* form that is presented when you start up, select **Add/Edit Macros...** from the *File* pull-down menu, or press *Ctrl-Ins*.
2. From the new form that pops up, select the **Add** button.
3. Select **Select Icon** to associate an icon for your macro.
4. **Click once on the icon you want**, then select the **OK** button.
5. Move (**Tab**) to the description box, then **type in a description for your macro**.
6. Move (**Tab**) to the area where you type in the script of your macro.
7. Refer to the *Macro Mania Commands* in this help file for a list of commands available to you. In general, you will probably want to start with the *Run* command to start a program, then use the *Send* command to send keystrokes to your program(s). See [A Simple Example](#) for an easy example that you can copy to get acquainted with *Macro Mania*.
8. Once you have typed in your macro, select the **Save** button, which will prompt you to Test your macro.

Special Rules for the SEND Command

The plus sign (+), caret (^), percent sign (%), tilde (~), and parentheses () have special meanings to the Send command. To specify one of these characters, enclose it inside braces. For example, to specify the plus sign, use {+}. Brackets ([]) have no special meaning to the Send command, but you must enclose them in braces as well, because in other programs brackets do have special meaning. To send brace characters, use {{}} and {}.

SWREG

Please note that this option of registering will cost you \$2 more, or \$31.95 per copy, which helps me absorb the 15% of the registration price that CompuServe charges me when someone uses their SWREG service. Quite frankly, with a price as low as it is, I can not afford to take much more out of my small profit per registration.

So why do I offer SWREG as a means of registering? I realize that some of you may find it more convenient to use your CompuServe account to register programs online. I know some of you want to purchase a shareware program with your companys CompuServe account because there is a lot less paperwork involved and approving signatures to obtain, etc.

You can also purchase 6 or more site licenses via SWREG. The discounted price is \$10 per license (6 license minimum). If you want to order 6 or more licenses via SWREG, please Email me at starmail on CompuServe and I will send you special instructions to do so.

To register online via CompuServe, first **GO SWREG**, then select **Register Shareware** and move thru the menu-driven system. The registration ID for this program is: **13289**

Once you register online, an Email message will automatically be sent to me notifying me of your registration. I will then promptly (usually within 8 hours and often within only a couple hours if during the day or evening) send you an Email message with the [registration code](#).

Msg_Ok Command

Description:

Stops the macro from executing and displays a message. Execution is immediately resumed when the Ok button is selected.

Syntax:

MSG_OK [message]

[message] is any text you wish to have displayed in the message box.

Notes:

This command is useful for giving the user instructions and/or stopping a macro while another program is executing. If you need the message to be more interactive and allow the option of continuing with the macro or stopping the macro entirely, see the [Msg_OkCancel](#) command.

See [Important Notes for Windows 95](#)

Msg_OkCancel Command

Description:

Stops the macro from executing and displays a message with an OK and CANCEL button. If OK is selected, the macro continues, if CANCEL is selected the macro exits the macro.

Syntax:

MSG_OKCANCEL [message]

[message] is any text you wish to have displayed in the message box.

Notes:

The OK button is usually the default button (the button that is highlighted and therefore executed if the enter button or space bar is pressed). To make the CANCEL button the default, extend the command to **MSG_OKCANCEL_2**, making the 2nd button (CANCEL) the default button.

Use the [PAUSE](#) command to have a macro pause, but then continue automatically after a default time period has elapsed. The pause command can also display a RESUME or CANCEL button by using the SHOW parameter, but it can not display a custom message as with the MSG_OKCANCEL command.

See [Important Notes for Windows 95](#)

About the Author

How Macro Mania evolved:

I originally created Macro Mania for my own personal needs (I fully believe the computer should do as much work as possible <smile>). I soon realized that with a little polish that the program could be made useful to many others. The little polish ended up being *many* hours of programming, but also an addiction and enjoyment. I quickly found the hard part was not necessarily making it work (though that did not come without a lot of hours of testing), but rather making the interface easy for someone else to use. Judging from the positive comments I have received, I must have succeeded for the most part. Naturally my main incentive is getting a little monetary reward for the many hours of programming, testing, programming again, writing the help file, etc. <hint, hint, smile>

A little about me:

I am married with a beautiful 2 year old daughter and 2-month old son. They are both terrific and help me keep my sanity by forcing me to take programming breaks <smile>. So, if I use the cliché that *Baby needs a new pair of shoes, please send in your registration fee*, I am not kidding <smile>.

I am a College of Business graduate of Kansas State University (good football team and excellent school overall). I worked myself through college as a part-time computer consultant and as a computer operator for the Air National Guard. Before that I spent 4 years in the Air Force (right after high school) and was stationed in Japan for most of my enlistment.

Send me Email:

Please send me an Email if you find this program useful (even if you are not ready to register yet). I like to hear that my work has helped others, even if it is not making me rich <smile>, and it will encourage me to keep adding new features.

E-MAIL: **America Online: starmail**
CompuServe: starmail
Internet: starmail@compuserve.com

Scheduling Macros

The interface for scheduling a macro can be found on the Add/Edit Macros screen and should be easy enough to figure out. Just a couple notes to help you along:

Macro Mania must be running in order for this function to work. It also must be sitting at the regular, main icon window (the scheduler is not activated within the Add/Edit Window).

You may wish to use the PAUSE command with the show parameter as the first command in any macro that is scheduled so that you can be warned that it is about to begin and you will be able to let it resume or cancel it. At any rate, you will want to stop what you are doing while the macro executes so that you do not interfere with it or otherwise accidentally put the focus onto the wrong program and cause very bad things to occur.

Scheduling Times:

Macros scheduled to execute for a specific time interval (hourly, every half-hour, or every 15 minutes) will execute according to your system clock as outlined below:

Hourly: when the minute and second are at 00:00

Half Hourly: when the minute and second are either at 00:00 or 30:00

Every 15 Minutes: when the minute and second are either at 00:00 or 15:00 or 30:00 or 45:00.

Any schedule you set for a macro is saved so that the next time you start Macro Mania that schedule is followed. Use the optional date setting if you only want the macro to run once (rather than every day at the scheduled time).

You can also define a specific time for a macro to run. You are encouraged to use the spin buttons and click the AM/PM buttons accordingly. If you wish to type in the time yourself, do **not** use military time. That is, use the regular time and define the AM or PM setting with the appropriate option button. For example, to have a macro run at 3:10 PM, key in **3** in the hour box, **10** in the minute box, and then select the **PM** option button.

SCHEDULING DAYS:

The interface to schedule days is very easy-to-use (just point-and-click) and can be found by selecting the *day* button from the scheduling interface on the *Add/Edit Macros* screen. By default, if you only schedule the *time* portion of a macro, the macro will run every day for that time (or time interval) you set. However, you can have macros run on specific days of the week or specific dates by scheduling them to do so. If you select one or more days of the week (e.g. Monday, Tuesday, etc), the specific date (e.g. June 14, 1996) will be ignored. If you wish to have the macro run on a specific date, simply make sure there are no days of the week checked (quickly toggled by unselecting the ALL check box).

Benefits of Registering this Program

- 1) You will be given a [registration code](#) to make your current copy a fully-registered copy.
- 2) No more reminder screens (hurray!).
- 3) The opening screen disappears automatically (without you having to select the OK button).
- 4) Your name (or your company name) is displayed on the opening screen and in the About window. No more shamefully displaying that you are merely evaluating this program--you will be a full-fledged Macro Mania power user <grin>.
- 5) You will know in your heart that you have done the right thing (both legally and morally).
- 6) You will be notified via Email of new releases along with a brief description of new features and any bug fixes.
- 7) You will be eligible for a 50% discount on any upgrades to the latest, greatest version of Macro Mania. In other words you will be able to obtain a *registered* version with major new features for very little money. (See the section called [Revision History](#) for an overview of the new features and fixes that have accompanied each new version--you will see that I try to actively enhance Macro Mania, at least as much as time permits.)
- 8) Although technical support has not been needed much, registered users are naturally given more priority and attention regarding technical support.

Revision History

Be sure to see [Getting the Latest Version...](#) to make sure you have the latest version of Macro Mania.

>> VERSION 3.51 <<

Fixed a bug with scheduling a macro between midnight and 1:00 am--it used to not save a macro scheduled in that time range correctly.

Made Macro Mania smarter about saving the size and position of a branch Window when it exits. If it is minimized during the exit it will not save the minimized size and therefore cause the Windows to be virtually unusable. This only happened under unusual exit circumstance with branch Windows, but that has been fixed.

Added some discussion about [DOS batch files](#) to the help system.

Added the feature/option to have the first button beep when it gets the focus. This is especially useful if you like to tab through your macros from the keyboard and would like an audio reference of when you have just put focus on (or passed) the first button.

>> VERSION 3.5 <<

Added several commands that enable you to manipulate the contents of the Windows clipboard. Namely the [Clipboard Restore](#), [Clipboard Save](#), [Clipboard Set](#), and [Clipboard SetFile](#) commands.

Added the ability to [create](#) different macro files and then open them as needed. Useful if you want to categorize your macros (e.g. by user, functionality, what application they will be used with, etc.).

Added the [Open](#) command to open a different macro file from within a macro itself. Also added the ability to pass a macro file at startup and the ability to manually open another macro file via a selection on the pull-down menu.

Added the [Branch](#) command. This command allows you to specify another macro file which will pop up and allow you to interactively select an option. Once an option is selected, a macro will run and return to the original macro to execute further macro commands if needed.

Added the [Repeat Prompt](#) command to give more control and easier access regarding setting repeat values in a macro.

Added the ability to make a macro button invisible from the main screen in Macro Mania. This is useful when you do not normally select a macro directly (e.g. it is used via a [Macro](#) command in another macro, it is only executed at [startup](#), etc.)

Enhanced Macro Mania so that the [macro](#) command no longer has to be the last command of a macro. Execution will now return to a macro that launches another macro. You can now easily create [subroutines and loops](#) within macros.

To [quickly modify](#) a macro, you can now click once on the *right* mouse button to automatically go to the edit screen to edit the macro your mouse pointer is on. Especially helpful if you have several macros and forget their sequence number.

Added a feature that *automatically* searches and adjusts any macros that have a [macro](#) command which points to macros that get resequenced when you delete macros.

Added a few more bells and whistles on the Add/Edit screen. For example, F3 will take you to the box so you can jump to a macro quickly.

Added a [password feature](#) you can set at two levels. You can now password protect your macro file entirely, password protect the macros from being view or edited via Macro Mania, or password protect both.

Added to the discussion of the Run command, which mentions some not so obvious features that will be sure to interest most of you. (See the NOTES section of the [Run](#) command for details.)

>> **VERSION 3.1** <<

Removed the opening screen if Macro Mania is a registered copy and it is has been run minimized or with a parameter to have it run a specific macro at startup (an unregistered copy requires you to select OK at the opening screen). If Macro Mania is just being run normally (registered or not registered) a Splash Screen is displayed for just a couple seconds when Macro Mania first initializes.

Fixed a minor bug with the scheduler: if a date had been scheduled, one had to select *Not Scheduled*, save it, and then move to another macro or exit the macro setup for it to clear. (Now you only have to select *Not Scheduled*--the need to move to another macro or exit the macro setup is no longer necessary to unschedule a macro.)

Added the ability to schedule macros to run only on certain days of the week (e.g. Monday, Tuesday, etc.).

Added the [PlayWav](#) command so that you can play a wav file from a macro for added pizzaz or to otherwise bring attention to a macro. (Kind of a fun to play with too!)

Added the [WinExit](#) command which can either exit Windows, restart Windows, or restart the computer--fast and easy.

>> **VERSION 3.0** <<

Extended the ability of the [Activate](#) command to enable it to activate a program with only a partial match of its Title Bar. This took a lot of extra programming effort (and testing), but I could not ignore the cries of the masses <smile>.

Added the [GetInput](#) and [SendInput](#) commands. The GetInput command can be used to request small amounts of dynamic information (file names, search strings, etc.). The SendInput command will then send the information obtained from the GetInput command.

Reworked the examples that are installed. Based on the operating system being installed to (Windows 3.x or Windows 95), appropriate examples are installed to best demonstrate the usefulness of Macro Mania.

Enhanced the discussion of Windows 95 considerations in the Help System. (See the [Pause](#) or [Minimize](#) command for more details.)

>> **VERSION 2.2** <<

Added an optional date setting that can be used with the scheduler.

Added 2 more options to select from when toggling button sizes: extra small with icons and extra small without icons. (Now the footprint can be set to an even smaller size if needed.)

Added a new Installation routine (undetected change for most users).

>> **VERSION 2.1** <<

Due to popular demand, added the option to set/toggle the size of the buttons. The buttons can now be smaller (making the program's "footprint" much smaller).

Added the [Exit](#) command, which allows you to exit the macro anywhere within the macro.

Added the [End](#) command, which allows you to have "Macro Mania" easily end itself entirely. (For those of you who like to just run one macro, perhaps from another program, and then want "Macro Mania" to end.)

>> **VERSION 2.0** <<

Added a schedule feature enabling you to schedule macros for specific times or during any 15-minute, half-hourly, or hourly interval.

Added the [Msg_Ok](#) and [Msg_OkCancel](#) commands that you can use to stop the macro and display a message. You can use either a regular message with an "Ok" button or use an interactive message that displays an "Ok" and "Cancel" button which will resume or cancel the Macro, respectively.

Note: Combine the above two features for a handy alarm/tickler!

Removed the prompt "Exit Macro Mania - Yes/No" when the program is exited.

Added an option to expand the area where you Add/Edit the actual macro script.

Bug fix: when the window was resized sometimes a few icons would get totally "lost" and the window had to be resized again to "find" them.

Bug fix: when Macro Mania was started minimized the initial screen minimized, but then the main icon window screen would return to its normal state.

Bug fix: when the program was closed from an icon the Window's environment still thought it was running if the task manager or Alt-Tab sequence was used.

Made a few underlying enhancements to fine tune performance.

>> **VERSION 1.3** <<

Added the ability to launch a particular macro when "Macro Mania" is first loaded.

Optimized the speed at which macro buttons are adjusted when you resize the "Macro

Mania" screen.

>> **VERSION 1.2** <<

Fixed a problem that happens on some PC's when the PAUSE command is used.

Removed some duplicate or infrequently used icons that are installed. This keeps the distribution file small. (Search for "Icons" in the Help system to find out how you can easily add your own icons.) Still, there are 200 icons that come with Macro Mania!

Made sure the installation of new versions does not overwrite any macros from older versions unless the users "Okay's it". Too many happy users for me to risk messing that up. <smile>

>> **VERSION 1.1** <<

Fixed a small bug that caused the "Always On Top" characteristic to turn off when the program was minimized.

Added the [SendNow](#) Command, which allows you to send just about any conceivable combination/format of the current date and/or time to a program.

Added the [Beep](#) Command, which allows you to cause the PC to beep via its speaker. Useful when you need to bring attention to the PC, particularly after a long macro and/or when its time for you to type.

(Registration Code)

Given to you after you register your copy (or copies) of Macro Mania, this code will make your copy a fully-registered version.

This code is entered in the "Enter Registration Code" window which can be found under the Help pull-down menu on the main Macro Mania window (the window with all your macros and their corresponding buttons).

It is given immediately after you contact NorthStar Solutions with a valid Visa/MasterCard/Discover, or promptly Emailed to you after you use the SWREG services or send payment by mail.

Important Notes for Windows 95

In Windows versions prior to Windows 95, one could easily switch back and forth between programs by sending ALT-Tab (even programmatically via Macro Mania). In Windows 95, however, along with the calendar, a simple version of Write, a decent File Manager and more, they took this away from us. That is, Alt-Tab does not work with Windows 95--a limitation Microsoft recognizes, but does not intend to fix (they claim this behavior is intended--though it could be done easily--and without any problems--in Windows 3.x).

What this means is that some of the Macro Mania commands put focus onto Macro Mania and can not automatically put it back to the other application (I automatically send an Alt-Tab when these commands are used, but it just sits unaffected in Windows 95. Fortunately it does not react in some other, negative way--it just does nothing.) So, if you need to put focus back to an application (other than Macro Mania) you need to either do so manually (by clicking anywhere on one of that applications windows), or by explicitly doing so with either the [Run](#) command or the [Activate](#) command (using Run or Activate is a more reliable strategy). Other than the Alt-Tab sequence, all the other Send combinations test out thoroughly with all versions of Windows.

The commands affected by this quirk include: Minimize, Restore, Pause, Msg_OK, Msg_OKCancel, and Repeat_Prompt (in general, any command that makes Macro Mania, or part of it, to be pulled forward on the screen). Again, by simply using the Run or Activate command after using these commands and immediately before the Send command, the problem will be easily avoided.

PlayWav

Description:

Plays a *.wav file on any PC set up and equipped with a sound card.

Syntax:

PlayWav [path][wavfile]

where [path][wavfile] is the full path and *.wav file to be played.

Example:

```
PLAYWAV c:\windows\tada.wav
```

Notes:

If the wav file and/or path is not found, the macro will simply continue (without playing the wav file).

WinExit

Description:

Allows you to reboot the computer, restart Windows, or exit Windows.

Syntax:

WinExit [n]

where [n] is either 1, 2, or 3 as follows:

- 1 - reboots the computer
- 2 - restarts Windows
- 3 - exits Windows

Example:

WINEXIT 3

Notes:

Macro Mania gracefully shuts down all Windows programs as if you manually closed Windows before closing other programs. If a program can not be closed gracefully, such as a DOS prompt, then Windows will not exit and you will be prompted to close the active application before exiting.

Macro Mania is a Windows program and, as a result, Macro Mania is unloaded when this command is run. Therefore, there is no point to have any commands following the WinExit command in your macro, as it will never be executed. (You could, however, put Macro Mania in your Windows Startup Group and pass it a macro number as a parameter to have Macro Mania begin executing a macro immediately upon starting Windows. See [Launch a Particular Macro at Startup.](#))

It may be useful to have a [MSG_OKCANCEL](#) command right before the WinExit command so you have the opportunity to abort the macro if desired.

Manipulating the State of Windows:

A Windows program needs to be running either *Normal* or *Maximized* (not *Minimized* as an icon) for it to accept keystrokes. (When you think about it, this is true whether the keystrokes are being sent from Macro Mania, or you are working with a program and typing manually.)

With that in mind, the first set of keystrokes you may wish to send are keystrokes to either Normalize or Maximize a window (to ensure it is not Minimized as an icon and it is ready to accept keystrokes).

The state of most program windows (whether they are Maximized, Minimized, or Normal) can be manipulated by sending keystrokes. By sending *Alt-space* (translated as *SEND %{ }
}*) with the SEND command, you can pull up the control button of a window and then send the letter you wish to use to either Minimize (n), Maximize (x), or Restore (r) the window. If I am not making sense, press *Alt-space bar* now to see what I mean, then press *Esc* to get rid of the command button menu. In summary, once you use either the *RUN* command or the *ACTIVATE* command to bring a window to the front, you can use the send command to manipulate its state (which you will want to do if you think the window could be minimized and not able to accept keystrokes).

Normalize - SEND %{ }r

Maximize - SEND %{ }x

Minimize - SEND %{ }n

Since brackets are ignored by the SEND command, note that in the above syntax, the brackets are optional and used to help identify there is a space there.

Launch a Particular Macro at Startup:

You can launch a particular macro when Macro Mania is first started by following the command with the number of the macro you want it to run automatically. For example, to run macro 3 you would use: "[path]MACROM.EXE 3" to automatically start your 3rd macro (where [path] is the location of MACROM.EXE).

Note that while adding new macros will not affect the position of your macros since new macros are simply appended to your macro list, deleting macros in front of existing macros causes them to be resequenced by -1. For example, you have 10 macros and delete the 7th, now the 8th becomes the 7th, the 9th becomes the 8th, and so on. Just bear this in mind whenever you delete macros so you can make any needed adjustments.

Adding Icons to the icon list:

Macro Mania installs 200 icons you can choose from; however, if you want to add to the list of your choices, you can! Just copy any valid icon file(s) to the *icons* subdirectory beneath the subdirectory where you installed *Macro Mania*. (This is done automatically for you if you select an icon from a directory other than the default icon directory.)

There are many icon libraries available and you can probably locate one where you found this program. Be sure to find a library that has actual icons files (*.ico) that can be copied to the icon subdirectory, not icon libraries that are contained in DLLs--unless you have a utility that can extract the icon files from the DLL.

The Macro Mania Icon Selector can read up to 900 icons, so please do not get too carried away <smile>.

Activating Buttons

Most buttons in a program can be activated by sending a space or, perhaps more obvious and common, an {enter}; however, be sure that the focus is on the button you wish to activate. A better way to activate a button is to use its hotkey if it has one. Hotkeys are the Alt-letter combination that invokes that button and can be identified because the letter is underlined. (This discussion is true, at least, if the program follows normal Windows conventions.)

Combine the search feature of the program with a macro

For example, suppose you have 50 expressions that say $A = B$, but they are not all the same, some say $C = D$, some say $R = Y$, etc. Using the *Find* command of your program, you can search for the equal sign (=) and then send the keystrokes that you would do in order to move the value on the left of the equal sign to the right of the equal sign and vice versa.

Here is a macro that would do that, assuming you have already done a search for the equal sign and F3 causes a find next to occur.

```
REM Repeat this 49 times  
REPEAT 49  
REM Run (or activate) the program (in this case Visual Basic)  
RUN c:\vb\vb.exe  
REM Search for the next equal sign  
SEND {F3}  
REM Delete the equal sign and space, then cut what is on its right  
SEND {del 2}+{end}%Et  
REM Add the equal sign and paste back what was on the left  
SEND {home}%Ep=
```

Save Space in Your Macros

Do not use your macros to type every character of a particularly big group of text (several paragraphs). By using the [Clipboard Setfile](#) command, you can quickly put the contents of a large group of text into the clipboard and then paste it where you need. This will also improve performance for the macro in general, as pasting is faster than sending one character at a time with the Send command.

Mouse Movement

First, let me note a significant limitation with mouse movement: it relies on the exact position of a Window in order for it to be useful. If a window is resized or moved slightly, the macro becomes unusable (possibly even causing bad things to happen), and either the Window has to be repositioned to exactly where it was before or the macro has to be reprogrammed. (Macro Mania is intended to make tasks faster and easier, not vice versa!) The strength and intent of Macro Mania, therefore, is sending *keystrokes*. Since Macro Mania can send any keystroke, you should be able to reliably perform most, if not all, tasks by simply using keystrokes.

A workaround, however, for sending mouse movement under Win 95 is to use the "Accessibility Features", which then allows one to use arrow keys (keystrokes) to perform mouse movement. To access this:

- 1) Select "Settings" from the Win95 Task Bar
- 2) Select "Control Panel"
- 3) Select "Accessibility Options"
- 4) Select the "Mouse" folder
- 5) Select "Use Mousekeys"

Here's a short little macro that demonstrates that the arrow keys (down, up, right, left) now move the mouse:

```
RUN notepad.exe
SEND %{ }m
SEND {Down 20}{enter}
```

The above macro should move your notepad window down the screen a little.

I personally have not found a program that can not be completely manipulated with keystrokes (IMHO, it would have been a short-sighted programmer to create a program that requires the mouse. Perhaps drawing programs are the only exception to the rule that everything should be performable from the keyboard.). :) If you elect to try using the Accessibility Feature, I should mention I have not had enough experience to help you--Macro Mania is advertised to send keystrokes, not mouse movement; but I did want to mention this possible option. Some users have Emailed me and said it has worked nicely for them. :) Just one final note, you will want to take full advantage of Macro Mania being able to send repeated keystrokes without having to retype them for each keystroke (e.g. {down 20} will work the same as {down}{down}{down}....{down} 20 times.)

Getting the Latest Version...

As evident by the various [revisions](#), I am adding new features to Macro Mania as fast as time and ideas permit. Additionally, Macro Mania has been named a "Reviewer's Pick" by professional reviewers at Ziffnet, has been featured as "Program of the Week" from the Windows Users Group Network [WUGNET] on CompuServe, and it has received favorable mention in nationally recognized computer magazines such as *Info World* and *PC World*.

OBTAINING THE LATEST VERSION VIA COMPUSERVE:

Macro Mania can be located in the Windows Utilities Forum (**GO WINUTIL**). It is in the "Batch/Launch Tools (3)" library as MACROM.ZIP.

OBTAINING THE LATEST VERSION VIA AMERICA ONLINE:

Visit NorthStar Solutions Forum. **Keyword: NorthStar**

OBTAINING THE LATEST VERSION VIA THE INTERNET:

Point you Web Browser to: <http://nstarsolutions.com/mm.htm>

If you are unable to find the latest version (or do not want to have to hunt it down at registration time), I have provided a very convenient way for you to register the current copy you have, and get the latest version sent to you. This upgrade feature gets the latest version sent to you on diskette for only \$7 to U.S. destinations and \$9 for all other destinations. (This option is only available when you register through the company that processes the orders for us, NorthStar Solutions, because SWREG does not provide an easy way to do this--of course if you are using SWREG, you can find the latest version in the WINUTIL forum and download it yourself on CompuServe). I have no special delivery schedule for new versions, so you are encouraged to inquire about this upgrade option when registering.

Sending a space (spacebar)

Either of the following two syntax methods will send a space:

Syntax 1: Send << two spaces follow *Send*

Syntax 2: Send { }

Since brackets are ignored in the Send command, the second syntax method is helpful for identifying that a space is there and is probably a better way to send a space.

Stopping a macro

Once a macro is in motion, it is nearly impossible to stop it, which is why I always emphasize saving your work and testing your macros carefully. However, if you do have a macro that you might need to stop (like when you use the repeat command and are not sure of the exact number of times you need it to repeat, etc.) you can explicitly add a command in the macro which would allow you to stop the macro at certain points:

One option would be to place a [MSG_OKCancel](#) command in the macro. This would allow you the option to cancel it or press Enter to have it resume.

Another option would be to use the [Pause](#) command with the SHOW parameter and a small delay (2-3 seconds), then you would be able to cancel the macro, press resume, or it would count down in a couple of seconds and resume automatically.

(Important Commands)

Although Macro Mania sports a variety of commands to add power to your macros, there are really only 2 main commands (RUN and SEND) that you need to be familiar with to create useful macros fast and easy. As you become a more experienced *Macro Maniac*, you can add to your macro vocabulary and create more sophisticated macros. Also note that I have made the syntax as simple and straight-forward as possible, and syntax reference is just a few steps away in this help system.

Clipboard_Set

Description:

Sets the contents of the Windows clipboard with text you specify.

Syntax:

Clipboard_Set [text]

text is a single string of characters you want to put in the Windows clipboard.

Example:

CLIPBOARD_SET Let love and faithfulness never leave you; bind them around your neck, write them on the tablet of your heart. - *Proverbs 3:3*

Notes:

This command is useful for putting small amounts of simple text strings into the clipboard for future paste commands to other programs. However, using the *Send* command will accomplish this just as effectively, though the information may be sent slightly slower. If a large amount of text, or text which includes carriage returns, is needed, see the [Clipboard_SetFile](#) command for a better way to put such information in the clipboard.

If *text* is omitted, the contents of the clipboard is cleared.

The length of the *text* you wish to put in the clipboard can not exceed any limitations Windows clipboard has.

Clipboard_SetFile

Description:

Sets the contents of the Windows clipboard with the contents of the file you specify.

Syntax:

```
Clipboard_SetFile [file]
```

file is the full path and file name for a text file that you want put in the Windows clipboard.

Example:

```
CLIPBOARD_SETFILE c:\jeff\myfile.txt
```

Notes:

The file should be a pure text file. Unlike the [Clipboard_Set](#) command, this command enables you to also put text with Carriage Returns in the clipboard, since Carriage Returns are copied to the clipboard as they appear in the file.

For this command to work, the length of the file can not exceed 65,535 bytes. Of course, that is probably more than the limit for how much the clipboard can hold anyway. The length of the file can not exceed the limitations of the Windows clipboard for this command to work.

If the file is not found, an error occurs and the macro will terminate.

Clipboard_Restore

Description:

Restores the contents of the Windows clipboard which was saved with the [Clipboard_Save](#) command.

Syntax:

Clipboard_Restore

Notes:

This is useful if you do not want the clipboard commands of Macro Mania to permanently overwrite the contents of the Windows clipboard after a macro has finished executing. The Clipboard_Save and Clipboard_Restore commands must be used in the same macro for Macro Mania to associate them correctly.

The Clipboard_Save command must have been used prior to this command, or the contents of the clipboard will be cleared entirely.

Clipboard_Save

Description:

Saves the current contents of the Windows clipboard so that a [ClipBoard_Restore](#) command may be used later in a macro to restore the clipboard contents.

Syntax:

Clipboard_Save

Notes:

This is useful if you do not want the clipboard commands of Macro Mania to permanently overwrite the contents of the Windows clipboard after a macro has finished executing. The Clipboard_Save and Clipboard_Restore commands must be used in the same macro for Macro Mania to associate them correctly.

Repeat_Prompt

Description:

Prompts you to enter the number of times you want a macro to repeat. See the [repeat](#) command for more information about repeating macros.

Syntax:

Repeat_Prompt <text>

text is an optional parameter you can add to customize the prompt. If *text* is omitted, a generic prompt with the title of the macro will be used.

Example:

REPEAT_PROMPT Enter the number of times we should copy information from Program A to Program B.

Notes:

The prompt will only appear once (the *first* time it is encountered) during the execution of a macro. Like the repeat command, it is generally better to put this command near the top of your macros for easy reference. Unlike the repeat command, however, this prompt allows you to more dynamically set the number of times a macro executes if needed.

The repeat option can also be set dynamically by setting the [Temporary Repeat Value](#), though the Repeat_Prompt command is probably more useful and less dangerous. With the Repeat_Prompt command, the repeat value is retained only while the macro is running and you are reminded to set (or reset) the value as needed. Therefore the Repeat_Prompt command generally supercedes the previous way to set a temporary repeat value.

See [Stopping a Macro](#) for strategies to help you stop macros if needed.

See [Important Notes for Windows 95](#)

Branch

Description:

Allows you to specify another macro file which will pop up and allow you to interactively select an option. Once an option is selected, a macro will run and return to the original macro to execute further macro commands if needed.

Syntax:

Branch [file]

file is the name of the macro file (or branch file) that you want displayed.

Example:

BRANCH choices.mm

Notes:

Macro Mania will suspend all macro activity until an option is selected from a branch option box. Once an option is selected, the macro for that option is executed and control returns back to the original macro that called the branch. You can cancel a branch by selecting *Exit* from the *File* pull-down menu and the option to cancel or resume the main macro entirely will be presented.

You may nest branch commands from other branch option boxes. Although most Macro Mania commands may be used from a branch, which certainly includes the 3 [important commands](#), the [Open](#) command may not be used from a branch option box.

Use the Open command to open a different macro file entirely without just branching to it.

See [Creating New Macro Files](#), for more information about how to create branch files.

As you are testing your Branches, note that their positions and sizes can be manipulated and will automatically be stored for the next time that branch is used (the same feature that is applied to the main Macro Mania screen).

Creating New Macro Files

You may find it useful to group your macros into different categories (e.g. by function, user, application they are used for, etc.). Macro Mania allows you to do this by creating new macro files in addition to the default DEFAULT.MM file. Creating new macro files is also necessary if you wish to use the [Branch](#) command to provide a branch option box in your macros.

To create a new file, select *New* from the *File* pull-down menu on the screen where you Add/Edit your macros. Then just enter the name for the new file and Add/Edit your macros as needed. (Note: I recommend you use the MM extension for macro files to keep their extensions consistent. The *Open* feature when you manually select macro files to open will default to looking for all files that end with .MM.)

Although Macro Mania usually tries to locate and execute its default file, DEFAULT.MM, you can pass it the file as a parameter when first running *macrom.exe* for it to automatically open a different macro file when it first loads. For example, *macrom.exe c:\macrom\jeff.mm* will automatically open jeff.mm rather than DEFAULT.MM when Macro Mania first loads. (You can also use the [Open](#) command to have a macro open another macro file, or you can open another file manually by selecting Open from the File pull-down menu.)

Open

Description:

Opens another macro file for Macro Mania to use as its main macro file.

Syntax:

Open [file]

file is the name of the macro file that you want to use.

Example:

OPEN jeff.mm

Notes:

See [Creating New Macro Files](#) for more information.

Fast access to edit a macro

To quickly access the edit screen to edit a macro, just point to the macro you wish to edit from the main Macro Mania screen and click the *right* mouse button once. The Add/Edit screen will appear and the macro you clicked on will automatically be found and ready to modify.

If you have several macros and do not remember their sequence number, you can easily exit the Add/Edit Screen, find the next macro you wish to edit, and repeat this procedure.

General Keyboard Navigation

The following are some navigation tips that apply to most Windows programs. Since many Windows program(mer)s follow a set of rules about how a Windows application is supposed to behave, these rules will likely apply (though I obviously have no control if a program(mer) does not follow these conventions). While this is not a definitive discussion on the topic, I think it covers the most common keyboard conventions/shortcuts available in most Windows applications.

Tab - Moves from object to object in a general pattern (often left to right, top to bottom like one would read a book or in some other logical fashion). To move in the opposite direction, a Shift-Tab sequence can often be used. Naturally an exception to this is with large notepad entry fields where a tab needs to act like a tab (usually to quickly indent a certain number of spaces).

Left, Right, Up, Down Arrow Keys - Self explanatory

Ctrl-Left and Ctrl-Right - Usually allows you to jump a word at a time left or right, respectively.

Home and End - Move the cursor to the far left or far right of a field, respectively.

Ctrl-Home and Ctrl-End - In large notepad entry fields, move the cursor to the top, left corner or top, right corner respectively.

Holding the **Shift** key down while moving the cursor within a field has special meaning: it *highlights* the text it crosses similar to dragging your mouse pointer across text (allowing for subsequent cut, copy, and paste operations). Using the shift key with some of the conventions mentioned above, you can quickly highlight information within a field. For example, to highlight the entire portion of a large text field, you can first move to the top of the field (Ctrl-Home), then while holding the Shift key, move to the very bottom of the field (Ctrl-End). Translated into a macro where a ^ represents the Ctrl key and a + represents the Shift key, it would look like this:

```
Send ^{Home}+^{End}
```

Often letters that are underlined mean that the command can be quickly executed by holding down the Alt key and pressing that letter at the same time. For example, a button or menu that had the word Help can be invoked by holding down the Alt key and pressing H, translated as Send %H in macro syntax.

Other Conventions/Shortcuts:

F1 - Invoke the Help system

Alt-F4 - Quit a program

Ctrl-X - Cut

Ctrl-C - Copy

Ctrl-V - Paste

Shift-Insert - Another Paste shortcut

Del - Delete character to the right of the cursor (or a currently highlighted section)

BackSpace - Delete character to the left of the cursor (or a currently highlighted section)

PgUp - Scroll one screen up

PgDown - Scroll one screen down

Alt-Space - Display the control box of a Window

Shift-F10 - Display the Task List (Win 3.1) or Start Menu (Win 95)

Just a final note about keyboard shortcuts: programs often have their own shortcuts built into them, and you are encouraged to take advantage of them if you can. For example, in word processors, etc. Ctrl-B often means to toggle the **bold** attribute of the font, Ctrl-U to toggle underline, Ctrl-I for *italic*, etc.

Password Protecting Your Macros

To offer *some* security over your macros, you can define a password that must be used to either open a macro file, view/edit your macros via Macro Mania, or both. This password protection scheme is not hack proof in that someone with reasonable knowledge of computer files can view and/or edit your macros, but it does offer protection from casual nosiness or prying eyes.

Actually, the only reason it was added is because some have mentioned they would like to create macros for others without the others going in and messing them up, in which case you can set the password level to prevent someone from accessing the Add/Edit Screen of Macro Mania. To prevent anyone but people with the password to be able to open (and consequently run) macros, you can password protect the macro file entirely.

The password and level of use is set up via the *Set Password Options* found under the File pull-down menu on the Add/Edit Screen.

IMPORTANT NOTE: As mentioned above, this password protection scheme is not absolutely hack proof, but should offer some reasonable level of protection for preventing people from easily viewing, editing, and/or using your macros. However, if you use Macro Mania to send passwords, etc. you should note that someone with a little more than casual experience with Windows may be able to view your macros via a way other than using Macro Manias interface. If you think that information in your macros could become compromised, you should not hard code that information into your macros. Consider using the [GetInput](#) and [SendInput](#) commands in your macros to dynamically send password information, etc. from your macros.

Subroutines within macros (looping, etc.)

There may be times when you want only a certain portion of a macro to loop. That is, you have something that needs to be done repeatedly, but you do not want the whole macro to be done repeatedly. You can do this easily with Macro Mania by combining a couple of its features:

Create a separate macro that does what you need to be done repeatedly. Then by using the [Repeat](#) (or [Repeat Prompt](#)) command, you can control how many times that part repeats. You are essentially creating a looping subroutine. Then simply use the [Macro](#) command to call that repeating macro, and only that portion will repeat, not the entire macro. If you are concerned about the repeating macros getting in the way, etc. you do not need to be--just make them Invisible by checking that option in the Icon box when you set them up. That way they'll be available to your other macros, but not available (or in the way) as a selection from the Macro Mania main screen.

Also, since your macros are *automatically* adjusted when you delete macros, you will not have to worry about keeping track of their sequence, etc.--it is all done for you.

Making sure the SEND command works

As mentioned elsewhere in this help system (but it never seems to hurt to emphasize this again), it is imperative that the application that you want the [Send](#) command to send keystrokes to has the focus. Although you may have used a [Run](#) or [Activate](#) command previously in a macro, if other commands have been used since the Send command, it may be necessary to use the Run or Activate command again to put focus back to that application--it is better to overuse the Run or Activate command a little rather than risk sending keystrokes to the wrong application (or more likely to Macro Mania itself).

Using Old Macro Files (prior to v3.5)

If you have created macros with a version of Macro Mania prior to v3.5, your old macros are compatible with later versions, but you may wish to rename the macro files. In previous versions, the file containing your macros was called MACROM.LST; however, in version 3.5 and later, the default file is called DEFAULT.MM. The new naming convention has been implemented now that you can create more than one macro file. See [Creating New Macro Files](#) for more discussion.

DOS Batch Files

Sometimes there is nothing as good as a DOS batch file for accomplishing what you need to do. With batch files, DOS has a built in scripting language that allow you to copy, delete, and move files, subdirectories, etc. You can use batch files to check for the existance of files (if exist), work interactively (see the DOS choice command), run DOS programs, etc. This help section is not intended to be a batch file tutorial, but I do want to point out this overlooked resource--which can easily be used in combination with macros to accomplish such tasks.

I could have devoted some time and resources creating commands that would allow you to copy, delete, and move files (as well as a few other commands), but the effort is not worth it since we already have a powerful way to accomplish these tasks. Simply create your batch file and use the [Run](#) command to launch the batch file.

There is one issue you need to be aware of: in the multitasking environment of Windows a macro that has more macro commands that follow the RUN command may try to continue too soon (before the DOS batch file is complete). However, to keep this from happening just use one of the commands available in Macro Mania to either pause or stop the macro from continuing too soon (use a [PAUSE](#), a [MSG_OK](#), a [MSG_OKCANCEL](#), or a [Branch](#) command to keep the macro from continuing too soon).

What if keystrokes are not being received

I have found, on occasion, that a program needs time to "catch up" with Macro Mania. That is, sometimes it may be necessary to put in a short PAUSE command between SEND commands to give the program receiving keystrokes time to get ready to receive (more) keystrokes. Most of the time adding a PAUSE is not necessary, but if you experience that a program is not getting keystrokes (and you are sure you have made sure it has the focus in order to get keystrokes), then it may be necessary to add a few second delay before using a SEND command. An example being worth a thousand words:

```
RUN myapp.exe  
PAUSE 2  
SEND {down 3}+{end}%Ec
```

Most applications are smart enough to start getting the keystrokes from the keyboard buffer after they launch, but some seem to ignore the keyboard buffer and thus never see what Macro Mania sent. Also, on the same topic, you could use the strategy of breaking up any long SEND commands into smaller ones. This is effective because Macro Mania will not begin processing the next line until the previous SEND command is finished and accepted by the receiving program. The following example shows what I mean:

```
SEND abcdefg^G{tab}{up 4}qrstuv
```

is changed to

```
SEND abcdefg  
SEND ^G  
SEND {tab}{up 4}  
SEND rstuv
```

Helpful Hints for Beginners

Practice with a few simple macros:

Macro Mania is very powerful and relatively simple to use. The nice thing is that once you master it, you can use it for any windows program! However, you may wish to start off gradually, creating a few simple macros to get an understanding of the basic concept. Also note that while Macro Mania sports a host of commands and options, there are only two [important commands](#) that are the heart and soul of Macro Mania.

Get familiar with the SEND command:

The [Send](#) command is the core part of Macro Mania, offering a full host of options to ensure you can do virtually anything, send any keystroke, and be as productive as possible without a lot of repetitive typing. I recommend that if you print nothing else, you print the syntax of the *Send* command in this help screen so you will have easy access to the syntax while you create your macros. Be sure to pay careful attention to the [special rules](#) of the *Send* command.

Save your work:

Before executing a macro, especially a new one, save your work in all your running applications so that if a macro you write does something you did not quite expect, you can return to the original state of the program easily.

Manually perform the steps and write them down:

Before trying to create a macro, perform the task manually and carefully record every key that you press. Another way to write macros is to perform a few keystrokes, switch over to Macro Mania to record them into your macro, then switch back and perform a few more keystrokes, switch back, etc.

Debugging your macros:

Macro Mania sends keystrokes fast--sometimes too fast for you to detect where a macro may be doing something unexpected or missing a step. For debugging: if you are not sure where your macro is not working correctly, place the [exit](#) command a few lines above where you think the error is happening. Then gradually move it down one line at a time until the line with the error is identified. (It may also be useful to break your SEND commands into smaller ones for this purpose.)

Make macros that are easy to read and follow:

Macro Mania allows you to put in extra spaces, comments, capitalize its key words, and issue commands repeatedly--so try to make your macros easy to read. For example, while the following two macros do the same thing, one is obviously easier to follow than the other:

EXAMPLE 1:

```
MINIMIZE
RUN WRITE.EXE
send %Pc^B%CeJeff Camino{enter}P.O. Box 25262{enter}Columbia, SC 29224{enter
3}^{home}+{end}^I^{end}{F5}
```

EXAMPLE 2:

```
MINIMIZE
RUN WRITE.EXE
```

```
REM Center the text, make it bold, then enlarge the font
SEND %Pc^B%Ce
```

REM Type in the letter head
SEND Jeff Camino{ENTER}P.O. Box 25262{ENTER}Columbia, SC 29224{ENTER 3}

REM Go back and italicize my name
SEND ^{HOME}+{END}^I

REM Go to bottom, make text normal and left justify paragraph
SEND ^{END}{F5}%PI

Always test your macros:

Test your macros, especially before issuing the MACRO command that calls other macros, or the REPEAT command which causes the macro to repeat itself. If a macro is not working properly, you definitely want to find out BEFORE you have issued the macro several times.

Remember, once you master using Macro Mania, it is the only syntax you will need to remember to create macros for many, if not all, of your programs!!

Recording Keystrokes

Macro Mania does not record your keystrokes as you enter them. Still, translating your keystrokes into a macro is *very* easy and this should not present any problem. When I create a macro I either remember the keystrokes needed, or if it is a long sequence, I simply start editing the macro, then flip over to whatever application is getting the keystrokes and perform a few, flip back to the editor to write the macro reflecting what I have sent, etc. The process is really easy and takes very little time. Once you have done this a couple times you will appreciate how easy it is, and that macros can be edited, portions copied and reused for other macros, etc.

Launching Macros

To launch a macro, simply click once on the button or use the tab key on your keyboard to tab to the macro you want, then press Enter or Space to launch it. The description of the macro will be shown in the bottom bar of the main Macro Mania Window when that button is highlighted (not necessarily chosen). You should use descriptions to best identify what a macro does and you can further aid yourself by selecting icons to match.

Another option is to assign a *hotkey* to the button which can be done when you select the *Use a Hotkey* button at the bottom of where you select the icon/hotkey for the macro. While not as graphical as icons, hotkeys enable you to quickly access a macro button by enabling you to just press *Alt-character*. For example, if you assign the letter A, you can then just press Alt-A from your keyboard and the macro will be executed without you having to highlight it with the mouse or tab to it via the keyboard.

IMPORTANT NOTE: Macro Mania does not intercept keystrokes being sent to other programs and, therefore, needs to be activated in order to run a macro. I could have added a "keyboard hook" that would trap, interpret, and perform keystrokes being sent to every Windows program, but this is not an easy task and causes a great deal of overhead because, when you think about it, there is a program layered between you, your applications, and every keystroke you send (even keystrokes from Macro Mania itself). However, a quick Alt-Tab over to Macro Mania and then invoking a hot-key you have assigned is still many less keystrokes than usual if you have macros that do anything substantial.

Related Topics:

[Launch a Particular Macro at Startup](#)

[Adding Icons to the icon list](#)

Resquencing Macros

Currently there is no pretty interface to facilitate resequencing the order of macros. However, the file structure of the files where your macros are stored is in such a way that this is not too hard to do manually if you are so inclined. I do warn you that these procedures are provided for informational purposes only and if you use them, you do so at your own risk!

First, Exit Macro Mania.

Next, Make a backup of your *.MM files where Macro Mania was installed (e.g. copy *.MM *.BAK). If things go wrong, you will have a backup.

Now you are ready to edit your *.MM file (DEFAULT.MM is the default one that is initially opened by Macro Mania). You may use a text editor of your choice, but I recommend using DOS Edit or Windows Notepad. (Make sure that the Word Wrap is **off**). If word wrap is off, then each line starting at line 4 (below the warning about editing your macros directly) will be a single macro. The beginning of each line is the description of the macro and it is followed by other information related to the macro, including the macro itself eventually.

To change the order of a macro just highlight the entire line for that macro and then cut and paste it to the position where you want it. Once you are satisfied with your new sequence, save and exit the editor and restart Macro Mania. Your macros will now be in the order you saved them as.

IMPORTANT NOTES:

Any time you change the sequence of macros you may need to adjust any macros that use the macro command because they could end up pointing to the wrong macro if you are not careful.

Again, this is all done at your own risk--do not call me if you mess things up--and do not forget to make a backup of your macro files before editing them like this. Of course, it never hurts to have a backup of your macros at all times anyway.

