# MALT
MORE ALT-KEYS FOR THE PERFECTS

## USER REFERENCE GUIDE

10309 Parkwood Drive
Kensington, Maryland 20895

Version 2.2

# MALT
# USER REFERENCE GUIDE

## INTRODUCTION

MALT (More Alt-Keys for the Perfects) is a memory resident utility (TSR) that adds 90 new keys for running Alt-like macros in DOS versions of WordPerfect and other DOS programs in the WordPerfect family. To avoid conflicts with other TSRs, MALT does not rely on the traditional shift keys (Shift/Alt/Ctrl). Instead, MALT uses the CapsLock key as an auxiliary shift key. MALT uses a mere 480 bytes of RAM, can be unloaded from memory, and does not interfere with the normal operation of CapsLock.

MALT also offers a library feature. Many sub-macros may be combined in one library macro called MALTLIB.WPM. Any of the macros in the library may be accessed by using the key combination ScrollLock + key, thereby effectively adding an additional 90 keys to play back macros.

MALT works with WordPerfect DOS 4.2, 5.0, 5.1, and 6.0, WordPerfect Presentations for DOS, LetterPerfect, WordPerfect Works, and should work with other DOS programs in the WordPerfect family, too. It is compatible with WordPerfect Office Shell 3.x and 4.0 and WordPerfect's Repeat Performance. Users of WordPerfect 6.0 will appreciate MALT because WordPerfect appropriates so many Ctrl and Alt keys for its own purposes. MALT should also be particularly helpful for users of WordPerfect Works and LetterPerfect (neither of which has a configurable keyboard), as well as anyone else with a shortage of keys, including notebook users and those without an enhanced keyboard. Finally, MALT can be used at the DOS command line. (See the section entitled "Other Uses for MALT" for further information on this feature.)

Because you are probably eager to learn how to use MALT, I shall have only a few words to say about registration at the beginning of this document. MALT is user supported software. It is not public domain. In brief, if you are using MALT for personal use, you need to pay a $15 registration fee. MALT may not be used in business without a license. More information concerning registration and licensing may be found at the end of this document in the section entitled "REGISTRATION/ LICENSE/ COPYRIGHT". Registration information is also available on demand by typing "MALT /R" at the DOS command line.

## INSTALLATION

Copy MALT.COM to any convenient directory on your hard drive. If you are using WordPerfect 6.0, copy the file MALTLIB.60 to your macros/keyboards/button bar directory and rename it MALTLIB.WPM. If you are using WordPerfect 5.1, copy the file MALTLIB.51 to your macro/keyboard directory and rename it MALTLIB.WPM. If--
and only if--you use a key other than Alt-F10 to run a macro or Ctrl-F10 to define a macro, then copy the file MALTCFG.EXE to the same directory as MALT.COM and read the section entitled "Configuring MALT".

MALT must be loaded into memory before you run WordPerfect. Simply type MALT at the DOS command line. If you use WordPerfect Shell, MALT must be loaded before Shell. MALT may be loaded HIGH or LOW, from the command line or in a batch file, while running from DOS, DESQview, OS/2, or Windows. See the section entitled "Loading and Unloading MALT" for detailed instructions.

## MALT-KEYS

Veteran WordPerfect users should feel at home using CapsLock as an auxiliary shift key, since, on an enhanced keyboard, it is located in the same position that the Ctrl key used to occupy. MALT-keys operate analogously to Alt-keys. To define a MALT-key macro first press **Macro Define** (Ctrl-F10), then hold down CapsLock and press the key to be defined. For example, to define MALTP, hold down CapsLock and press **P**. In WordPerfect 5.1, you may enter a brief description or press **Enter** to define the macro without a description. Type the keystrokes you wish to record then press **Macro Define** (Ctrl-F10) to end defining the macro.

MALT also has a *Quick Define* feature. To use *Quick Define*, hold down the Shift key and the CapsLock key simultaneously, then press the key you wish to define. For example, to *Quick Define* MALTP, hold down Shift and CapsLock together, then press **P**. You still must press **Macro Define** (Ctrl-F10) to end macro definition. To go directly into the macro editing window without recording, press **Home** before pressing *Quick Define*. Please read the section entitled "Compatibility and Keyboard Peculiarities", if you have trouble with the *Quick Define* feature.

Running a MALT-key macro is as easy as running an Alt-key macro: simply hold down CapsLock and press a key. MALT will then tell WordPerfect to run a macro with the name "MALT" plus the corresponding key name. Thus, holding down CapsLock and pressing **P** will cause WordPerfect to run the macro MALTP.WPM, just as holding down Alt and **P** will run ALTP.WPM.

Note: For the sake of convenience, this document refers to macros as having a "WPM" extension. Macros in LetterPerfect, of course, have an "LPM" extension, those in ED have an "EDM" extension, and so forth. Similarly, discussions about WordPerfect apply equally to LetterPerfect and the other Perfects. As a corollary, bear in mind that you need load only one copy of MALT if you switch between WordPerfect and one of the other Perfects. While you are in WordPerfect, MALT will work with "WPM" macros in your macro/keyboard directory. Should you switch to ED, MALT will work with "EDM" macros in that program's keyboard/macro directory. The same holds true for the other Perfects. The only proviso is that each program should be configured with the same **Macro Execute** and **Macro Define** keys. Finally, although MALT functions harmoniously with WPCorp programs running under WordPerfect Office Shell, it cannot be used to define or execute Shell macros.

Unlike Alt-keys, MALT-keys are not restricted to the letters of the alphabet. A MALT-key can be assigned to almost any key. Accordingly, space (MALTSP.WPM), insert (MALTIN.WPM), left (MALTLE.WPM), and en left (MALTXLE.WPM), are all perfectly acceptable MALT-keys. (The only exceptions are the obvious ones, the shift keys, Num and Scroll Lock, Print Screen and Pause.) This will add 90 new keys to an enhanced keyboard; 75 keys to a PC keyboard. Notice that MALT uses abbreviations for non-alphanumeric keys. A list of these abbreviations is included as an Appendix.

Existing macros may be converted to MALT-key macros by renaming them. For instance, if you have a macro called HEADER.WPM, you may convert it to MALT-H by renaming the macro file to MALTH.WPM. Remember, MALT-key macros must be in your keyboard/macro directory. For a complete list of possible MALT-key file names, please see the Appendix.

You should also be aware that in two small respects MALT-keys are not as capable as Alt-keys. Alt-keys may be run from anywhere within WordPerfect. MALT-keys, however, can only be run or defined in the same situations that conventional macros can be run or defined. Additionally, after running a MALT-key macro, CapsLock must be released before running another MALT-key. In contrast, you may hold down the Alt-key and press a key several times to run a series of macros or one macro several times. As a partial remedy, remember that, like any other macro, a MALT-key macro can be repeated by pressing the repeat key (ESCape in 5.1, Ctrl-R in 6.0) to input a repeat number, and then pressing the appropriate MALT-key. Most users will not find these to be serious inconveniences.

MALT does not interfere with the normal operation of CapsLock, and with each use of a MALT-key, CapsLock is restored to its prior state. Thus if CapsLock was off when you pressed the MALT-key, it will stay off; if it was on, it will stay on.

## LIBRARY FEATURE

MALT's library feature allows you to assemble up to 90 sub-macros in one comprehensive macro called MALTLIB.WPM. Any of the library macros in MALTLIB can be accessed with a single keystroke, using ScrollLock as an auxiliary shift key. Among the advantages of a library macro are the conservation of disk space and the capacity for sharing subroutines among library macros.

The library feature can best be explained by comparison to the more conventional menu macros. A menu macro solicits the user to press one or more keys, then takes an action based on user input. Typically, using a menu macro involves at least two steps: (1) calling the macro, then (2) pressing one or more keys to select a choice offered by the menu. MALT's library feature automates this process. Pressing ScrollLock + key causes MALT to call MALTLIB.WPM and to execute one of the sub-macros in the library based on the key pressed with ScrollLock. Thus, pressing ScrollLock + C causes MALT to call MALTLIB.WPM and to execute choice "C"--all with one keystroke.

Preparing a library macro requires only a rudimentary knowledge of macro programming. By referring to the accompanying sample library macros, and by observing the following simple rules, most macro users should not have any difficulty in creating a library macro.

The structure of a library macro depends on the version of using WordPerfect you are using. Instructions follow for creating a library macro for use in versions 5.x and 6.0, as well as for converting an existing 5.x macro to 6.0.

### *WordPerfect 5.0 and 5.1*

(1) The library macro must be named MALTLIB.WPM and should be located in the

same directory as your other macros.

(2) The library macro must begin with a **{TEXT}** command. (Note: Global macro instructions such as **{ON ERROR}** may precede the **{TEXT}** command. An example of this is shown in the sample library.)

The **{TEXT}** command has two parts. The first part is the name of a variable; the second is an optional message. With WordPerfect 5.0 the variable name must be a number between 1 and 9. In 5.1, the variable name may be any combination of letters. In the accompanying sample library macro, the variable is given the name "KeyName".

When a library macro is executed using ScrollLock + key, the name of the key is assigned to the variable named in the **{TEXT}** command. Thus, if you press ScrollLock + "C", the variable KeyName will equal "C". Press ScrollLock + F6, and the variable KeyName will equal "F6".

(3) Follow the **{TEXT}** command with a **{CALL}** or **{GO}** command in the following form:

```
{CALL}{VARIABLE}KeyName~~
```

or

```
{GO}{VARIABLE}KeyName~~
```

where KeyName is the name of the variable assigned in **{TEXT}** command. (Note: If the call command is used, it should be followed by a **{RETURN}** or **{QUIT}** to indicate termination of macro execution.)

(4) Each sub-macro must be preceded by a **{LABEL}** command followed by the name of the key associated with the sub-macro. The label names must follow the naming conventions for keys described in the Appendix. For example, the sub-macro associated with the "C" key would be preceded by the command **{LABEL}**C~. Likewise, the sub-macro associated with F6 would be preceded by the command **{LABEL}**F6~. You do not need labels for keys that you do not intend to use.

Let us suppose you have a simple macro that avoids WordPerfect's annoying habit of always asking for confirmation that you wish to replace an existing file each time you save it. The macro looks like this:

```
{Save}
{IF}"{SYSTEM}name~"!=""~
        {Enter}y
{END IF}
```

You would like to incorporate this macro in your library for use with F-10

First, you consult the appendix to find that the abbreviation for F-10 is F0. Thus the label for the sub-macro will be **{LABEL}**F0~. The entire sub-macro will read:

```
{LABEL}F0~
{Save}
{IF}"{SYSTEM}name~"!=""~
    {Enter}y
{END IF}
{RETURN}
```

(5) Each sub-macro must end with a command that terminates macro execution, such as **{QUIT}** or **{RETURN}**. Advanced users will prefer **{RETURN}**, because it permits a sub-macro to be called by another sub-macro.

You may also use the sample macro as a template to create your own Library macro. To add a sub-macro insert a new **{LABEL}** command followed by the MALT-key abbreviation that corresponds to the key you wish to define. For example, to add a sub-macro to be executed in conjunction with F1, begin by inserting **{LABEL}**F1~. Now insert the contents of the sub-macro and finish with a **{RETURN}** command. The whole thing would look like this:

```
{LABEL}F1~
[Your macro goes here]
{RETURN}
```

You may safely remove any of the sub-macros from the sample MALTLIB macro. Each sub-macro begins with a **{LABEL}** command and ends with a **{RETURN}** command. To use a defined key for another macro of your own simply delete the portion between these two commands. If you do not wish to use a key for another macro, you may also delete the **{LABEL}** and **{RETURN}** commands. The liberal comments in MALTLIB.WPM may also be deleted to free up more space.

The easiest way to create a library macro for WordPerfect 5.1 is with my program MPE4WP (Macro Programming Environment for WordPerfect), which was selected by *PC Magazine* as one of the best word processing add-in products. MPE4WP gives users of 5.1 the ability to create and edit macros on the WordPerfect editing screen. MPE4WP provides much of the same flexibility to edit macros on-screen that is built into 6.0, but without the overhead.

## *WordPerfect 6.0*

(1) The library macro must be named MALTLIB.WPM and should be located in the same directory as your other macros.

(2) The library macro must begin with a **GETSTRING** command. (Note: Global macro instructions such as **DISPLAY ONERROR, ONCANCEL** may precede the **GETSTRING** command. The sample library includes an example of this usage.)

> The **GETSTRING** command is the vehicle by which MALT tells WordPerfect which key was pressed in combination with ScrollLock. (Note that the **GETSTRING** command has four permissible parameters, but MALTLIB requires

only the variable parameter.) In the accompanying sample, the name of the key is assigned to the variable "KeyName". Thus, if you press ScrollLock + "C", the variable KeyName will equal "C". Press ScrollLock + F6, and the variable KeyName will equal "F6".

(3) Follow the **GETSTRING** command with a **SWITCH** command in the following form:

**SWITCH**(KeyName)

where KeyName is the name of the variable assigned in the **GETSTRING** command.

**SWITCH** corresponds loosely to the **{CASE}** command in the WordPerfect 5.x macro language. A complete switch statement includes the following commands: **SWITCH**, **CASEOF**, **CONTINUE** (optional), **DEFAULT** (optional), and **ENDSWITCH**.

The variable KeyName (in parenthesis after the **SWITCH** command) is compared with a series of **CASEOF** statements. If a match is found, then the commands following the matching **CASEOF** statement are executed up to the next **CASEOF**, **DEFAULT**, or **ENDSWITCH** instruction, whichever comes first. It is these commands immediately following the **CASEOF** instruction that form the sub-macro to be executed when you press ScrollLock + key. MALTLIB runs these instructions then jumps to the command following the **ENDSWITCH** instruction.

If none of the **CASEOF** expressions match the **SWITCH** expression, the macro executes the commands between **DEFAULT** and **ENDSWITCH**. In the sample macro the **DEFAULT** is to place a brief error message on the screen. When a **DEFAULT** command is not included and there is no matching **CASEOF** statement, the macro jumps to the commands following **ENDSWITCH**.

(4) Each sub-macro must be preceded by a **CASEOF** instruction consisting of three parts. First the word **CASEOF**. Second, the name of the key associated with the sub-macro in quotation marks. Third, a colon. The key names must be *upper case* and must follow the naming conventions for keys described in the Appendix. For example, the sub-macro associated with the "C" key would be preceded by the command

**CASEOF** "C":

Similarly, the sub-macro associated with F6 would be preceded by the command

**CASEOF** "F6":

You do not need **CASEOF** commands for keys that you do not intend to use.

Let us suppose you have a simple macro that transposes the word on which the cursor is resting with the word on its right. The macro looks like this:

```
BlockOn(WordMode!)
CutAndPaste
PosWordNext
MoveModeEnd
```

You would like to incorporate this macro in your library for use with F-10

First, you consult the appendix to find that the abbreviation for F-10 is F0. Thus, the case for the sub-macro will be

**CASEOF** "F0":

The entire sub-macro will read:

```
CASEOF "F0":
BlockOn(WordMode!)
CutAndPaste
PosWordNext
MoveModeEnd
```

You will probably wish to use the sample macro as a template to create your own library macro. A comment in the example points where new sub-macros may be inserted.

### *Converting from 5.1 to 6.0*

Before converting your 5.1 library macro you should be aware of three important changes to the macro language. First, there are new commands. Second, macros are now result oriented rather than keystroke oriented. Finally, many tricks required to make a 5.1 macro perform acceptably are no longer necessary and no longer work in 6.0.

These observations are by way of warning that the process of converting any 5.1 macro to 6.0 may be difficult. Many problems arise from the need to think differently when programming 6.0 macros. Sometimes, it may be preferable to start again from scratch rather than coerce a 5.1 macro to 6.0 format.

The following instructions assume you are converting a simple 5.1 library macro in which each sub-macro is completely contained between a **{LABEL}** and **{RETURN}** command. Those of you skilful enough to have written more complicated macros will recognize where additional steps are required to preserve your macros.

Please review the preceding section concerning the creation of a WordPerfect 6.0 library macro. You will note some similarities between the structure of a WordPerfect 5.1 library macro and a 6.0 macro. Each has a series of macros sandwiched between an opening and a closing sequence. There is no need to worry about converting the opening and closing sequences. That work has been done for you in the accompanying template.

The sub-macros in a 5.1 library are contained between **{LABEL}** and **{RETURN}** commands. In contrast, the sub-macros in a 6.0 library begin with a **CASEOF** instruction, and do not contain an explicit end instruction corresponding to the **{RETURN}** command. A 6.0 sub-macro ends where the next **CASEOF** instruction begins, or, in the case of the last sub-macro, a **DEFAULT** or **ENDSWITCH** instruction.

Retrieve the file MALTLIB.60 into WordPerfect. After acquainting yourself with the structure of a 6.0 library macro, you will modify the sample to form a skeleton for your own library macro. Edit MALTLIB.60 by deleting any of the sub-macro examples that are not useful to you. Remember each sub-macro begins with a **CASEOF** instruction. Be sure that none of the key names associated with the remaining **CASEOF** statements duplicates any keys that you wish to preserve in your existing library macro.

Now convert your 5.1 macro to the new format using the MCV (Macro Convert) program that accompanies WordPerfect 6.0. Retrieve the converted macro into a new window. Edit the converted macro by deleting (a) the beginning portion up to the first **LABEL** instruction associated with your first sub-macro, and (b) the final portion after the **RETURN** command of your last sub-macro. You will now be left with a series of sub-macros, each beginning with a **LABEL** command and ending with a **RETURN** command. WordPerfect will have noted where it encountered problems in the conversion process.

Each **LABEL** command that introduces a sub-macro should be replaced with a **CASEOF** command. Notice that in 6.0 the **LABEL** command takes the form **LABEL**(*Key Abbreviation*). The *Key Abbreviation* will be the abbreviation of the key associated with the sub-macro. Be sure to observe the correct syntax of the **CASEOF** command. It consists of three parts: (a) the

**CASEOF** command, (b) the *Key Abbreviation*, which must be capitalized and in quotes, and (c) a colon. All **RETURN** commands should be deleted.

This generalized description makes the process sound much more complicated than it really is. An example should clarify matters. A macro associated with the "F6" key in 5.1 will look like this after having been converted by MCV:

```
LABEL(F6)
[Sub-macro commands]
RETURN
```

For the 6.0 library it should look like this:

```
CASEOF "F6" :
[Sub-macro commands]
```

This is the simple part of the conversion process. You can perform it using WordPerfect's **Replace** feature. The more complicated part pertains to the actual sub-macro commands. In some cases MCV will have flagged problems in the conversion. For guidance on converting the actual sub-macros, I must refer you to the WordPerfect documentation (what there is of it) and to WordPerfect technical support. The easiest way to experiment with a sub-macro is to assign it temporarily to a MALT-key. Once you have it working properly, add the **CASEOF** instruction and insert it in your library.

Those of you using my program MPE4WP will also find it helpful to convert your old macro to text using M2T. You may then load the text file in a window for comparison with the new macro.

After you have successfully converted your sub-macros, copy them to MALTLIB.60. They should be inserted immediately after the **SWITCH** instruction. There is a comment to guide you to the correct insertion point.

Now, save MALTLIB.60 as MALTLIB.WPM and put it in your **Macros/Keyboards/Button Bar** directory. When you save the macro, WordPerfect should display the message "Compiling Macro" with a bar graph showing the progress of compilation. If this message did not appear, take the following steps. Exit from MALTLIB.WPM. Press Ctrl-F10, type "MALTLIB", and choose "Edit Macro". WordPerfect will again retrieve MALTLIB to the editing screen. Now save the macro again. This should force WordPerfect to compile your macro.

### *Sample Library Macro*

To try out the sample library macro included with this package, copy either MALTLIB.51 (for WordPerfect 5.1) or MALTLIB.60 (for WordPerfect 6.0) to the same directory as your other macros and rename the file as MALTLIB.WPM. The sample is intended to illustrate the structure of a library macro. Many of the sub-macros could just as well be mapped to keyboard file. One of MALT's virtues, however, is independence from keyboard files. The Memorandum sub-macro (ScrollLock + M) is more ambitious than the other examples. The sample library macro also contains comments for purposes of explanation. It may be desirable to limit comments in your library macro, in order to reduce macro size. Please note that the sample macro for use with WordPerfect 5.x contains a few commands that are unique to the 5.1 macro language.

The sample library macro contains the following sub-macros. Some macros in sample for WordPerfect 6.0 operate differently. The differences are noted in brackets.

C

Center justify
D

Page

Make header A
I

Italics
K

Page

Change case of letter under cursor
L

Page

Left/Right Margins
[Margin dialog]
M

Memorandum (Modify to include your own name in the **FROM:** field)
R

Flush right
S

Page

Set tabs
T

Page

Top/Bottom margins
[Margin dialog]
V

View document (Preview)
Z

Paper size
F6

Page

Paragraph [counter] numbering (levels 1 to 4)
CR

Page

Page

Advance right
LE
XLE

Page

Advance left
UP
XUP

Page

Advance up
DO
XDO

Advance down

In the interest of candor, it is worth acknowledging that ScrollLock is not the most felicitous choice for a shift key. On the other hand, it is no worse placed than the track balls on many new notebook computers. Unfortunately, the other key choices are either worse, or more or less in use. Please let me know if you have a suggestion for an alternative. Incidentally, by its nature the library feature is not amenable to a *Quick Define* operation. Those who manage to press ScrollLock, Shift, and a third key together will find the effort unavailing.

## LOADING AND UNLOADING MALT

MALT must be loaded into memory before you run WordPerfect. If you run WordPerfect or another program in the Perfect family from WordPerfect Office Shell or from WordPerfect Works Shell, then MALT must be installed before Shell. To load MALT into memory, type **MALT** at the DOS command line. If MALT is not on your path or in the default directory, you must include the full path name. MALT then becomes memory resident for use with WordPerfect or other members of the Perfect family that are run afterwards. MALT will refuse to load a second copy of itself, if it finds that it has previously been loaded into memory.

Unloading MALT is equally simple. To unload MALT from memory type **MALT /U**. If you have loaded any TSRs after MALT, be sure to unload them first. Since MALT uses such a small amount of memory, you may not wish to go to the trouble of unloading it.

The most practical way to load and unload MALT is with a batch file that also starts WordPerfect. A simple batch file would look like this:

```
MALT
WP
MALT /U
```

Similarly, if you start WordPerfect from your AUTOEXEC.BAT file, you should edit the lines immediately before and after the line that starts WordPerfect according to the preceding instructions. A simple AUTOEXEC.BAT file might look like this:

```
PROMPT $p$g
PATH C:\DOS;C:\WP;C:\UTILITY
MALT
WP
MALT /U
```

Of course, you will add full path names if MALT is not on your path. Incidentally, MALT releases its copy of the environment, so it does not matter where in your batch file you place commands that go into the environment.

On a 386SX, 386DX or 486 computer with an appropriate memory manager, such as QEMM, DR DOS's MemoryMAX, or DOS 5+ HIMEM.SYS and EMM386.EXE, MALT may also be loaded high using the appropriate LOADHI command.

To use MALT with WordPerfect while running in a DOS session under OS/2 2.1, add MALT to the batch file that is run at the beginning of the WordPerfect session.

DESQview and Windows users may find it makes sense to load MALT in high memory before running DESQview or Windows. Alternatively, MALT may also be loaded in a batch file that loads WordPerfect in a DESQview window or Windows DOS session.

In rare circumstances, MALT may refuse to load itself into memory in the mistaken belief that it is already resident. This is most likely to occur in a multi-tasking or task switching environment under the following circumstances. MALT is loaded with WordPerfect, but is not explicitly unloaded after the session is over. When you next try to run a WordPerfect session, MALT thinks it is still installed.

If this occurs, you may use the "/F" switch to force MALT to install itself, even though it erroneously believes it is already resident. Use the "/F" switch only if you are certain that MALT is not resident. If this situation arises frequently, you may wish to use the command "MALT /F" in the batch file that installs MALT. Do this only if you are certain that duplicate copies of MALT cannot be installed. This would be the case if you are running WordPerfect and MALT in a DESQview window, or in a DOS session under OS/2 or Windows.

Should you manage to install two copies of MALT, by using the "/ F" switch, all MALT-keys will be interpreted as MALT-F11. If this happens, uninstall one copy by typing "MALT /U" at the DOS command line.

## CONFIGURING MALT

In order for MALT to work properly, it must know what keys you are using for the **Macro Execute** or **Macro Define** functions. If you have not remapped these two keys from their respective default assignments of Alt-F10 and Ctrl-F10, you do not need to read this section, and you do not need to configure MALT.

If you have changed the **Macro Execute** key from Alt-F10, or if you have changed the **Macro Define** key from Ctrl-F10, you must use the MALTCFG program to configure MALT. To run the configuration program, type **MALTCFG** at the command line. To force MALTCFG to run in monochrome, type **MALTCFG /M**. It really is not necessary to read this section to run MALTCFG, since the program is self-explanatory.

MALTCFG first looks in its own directory, then in the default directory (if the two are different), for a copy of MALT.COM. It will then verify that the copy of MALT.COM it has found is the correct version. MALTCFG will not recognize MALT.COM if it has been compressed by PKLite, LZE or a similar utility. MALT.COM may be compressed, however, *after* it has been configured. (Note: MALT is now so small that PKLite will not compress it. Moreover, on a standard hard disk, there would be no benefit to compression, because MALT uses only one cluster.) If the configuration program cannot find MALT.COM, you will have an opportunity to tell MALTCFG where it is located. Likewise, MALTCFG will allow you to configure a copy of MALT.COM in a different directory.

If you encounter problems configuring MALT on a compressed drive (Stacker, DoubleSpace, etc.), then copy MALT.COM to a floppy disk. Tell MALTCFG to configure A:MALT.COM (or B:MALT.COM). When configuration is complete, copy MALT.COM back to your hard drive.

MALTCFG will next ask you for the appropriate key assignments for the **Macro Execute** or **Macro Define** functions. Simply press the appropriate key then press **Enter** to confirm each choice. MALTCFG recognizes the same keys as WordPerfect in the same manner as WordPerfect, so if you press an unrecognized key like Ctrl-3 or Ctrl-/, nothing will happen. Similarly, pressing Ctrl-Space is recognized as Space, the same as it is in WordPerfect. MALTCFG will query you, if it questions the propriety of one of your choices.

After you have confirmed your choices are correct, MALTCFG will configure a new copy of MALT.COM. The old copy of MALT.COM will be renamed MALT.CO!. Since the same copy of MALT.COM may be re-configured any number of times, you may safely delete MALT.CO!.

## OTHER USES FOR MALT

A thoughtful user has pointed out to me that MALT can also be used at the DOS command line to run batch files. Suppose, for example, that you have a batch file called BACK.BAT for backing up new files. You could rename the file MALTB.BAT. Pressing MALT-B at the DOS command line would now cause MALTB.BAT to execute. You should be aware, however, that MALT may not recognize all the keys on the enhanced keyboard while at the command line. Thus, MALT may not recognize F11 and F12, or distinguish between the keys on the cursor pad and their counterparts on the numeric keypad.

## USING MALT WITH MPE4WP

If you are writing macros of any length or complexity for WordPerfect 5.1, you are, or should be, using my program MPE4WP. To use MPE4WP with MALT, you need only rename the file macros that are distributed with the program. For example, {CREATE}.WPM would be renamed MALTC.WPM. Pressing MALT-C (CapsLock and C) will now run the *Create* macro. Continue to use the MPE4WP keyboard for those macros that only exist as key macros.

## COMPATIBILITY AND KEYBOARD PECULIARITIES

MALT has been tested on IBM-compatible systems under DOS versions 3.x and 5.0, QEMM 6.0 (with and without Stealth enabled), DESQview 2.4, DR DOS 6.0, OS/2 2.1, and Windows 3.1. If MALT does not operate as described, the most likely explanation is a TSR conflict. MALT should be compatible with other TSRs because it uses a different hot key arrangement and because it does not trap interrupt 9. My experience has verified this, although it is impossible for me to have tested it with all TSRs. Please let me know if you encounter any difficulties.

The standard advice when encountering TSR conflicts is first to unload all TSRs, then load MALT to determine if it works by itself. If so, the problem is a TSR conflict. To try and isolate the offender, re-load the other TSRs one at a time. The other advice for remedying TSR conflicts is to try changing the order in which the programs are loaded. Since MALT only works with WordPerfect (and other members of the WPCorp family), it makes sense to load it immediately before loading WordPerfect (or the other program) so it can be unloaded immediately after you exit the WPCorp program. But MALT really does not care where it is loaded, so long as it is loaded before the WPCorp program and Shell (if you use Shell). If you spend most of your day in WordPerfect, you probably do not care where you load MALT.

Another type of compatibility problem may arise with certain keyboards. Most programs that manipulate the keyboard warn about the Tandy keyboard. Not having access to a Tandy keyboard, I cannot verify whether there is an incompatibility with MALT.

I am aware of one incompatibility that may arise when using the *Quick Define* feature. My Northgate keyboard does not recognize certain keystrokes when more than one than shift key is held down simultaneously. For example, it recognizes CapsLock-Shift-T and CapsLock-Shift-I, but not CapsLock-Shift-Y or CapsLock-Shift-U. (Incidentally, this anomaly is not unique to CapsLock, but also occurs with other shift key pairs.) If you encounter this problem, use the old-fashioned way of defining macros: first press **Macro Define** (Ctrl-F10) then press the MALT-key, e.g., CapsLock-Y. This problem is a failing of the keyboard and cannot be remedied by MALT or any other program.

MALT always restores each of CapsLock and ScrollLock to the same state it was before you hit a MALT-key or MALTLIB-key, but in certain circumstances you might believe MALT has erred. For example, if you have a PC keyboard with a CapsLock light, the light may change even though CapsLock is set correctly. (This idiosyncrasy does not generally occur with enhanced keyboards.) If you check the "pos" on the WordPerfect status line, however, you can verify the correct state of CapsLock. If CapsLock is off, "pos" is lower-case; if CapsLock is on "POS" will be upper-case. Another peculiarity occurs in WordPerfect 5.0. If you accidentally hit a MALT-key for which a macro has not been defined in WordPerfect 5.0, "pos" will switch from lower to upper, or from upper to lower, as the case may be, and will not switch back until the next time you hit **Enter**. Aside these oddities and indications to the contrary, it bears repeating that MALT will always properly reset CapsLock and ScrollLock to their prior states.

If you use a memory map utility to determine how MALT is loaded into memory, do not be concerned to find that MALT is listed as unknown. This happens because MALT, like a good citizen, releases its copy of the environment.

## PROGRAM NOTES

Beginning with version 1.3, MALT has been written entirely in assembly language. MALTCFG is written in Microsoft Professional Development System (PDS) BASIC 7.10, using routines contained in Crescent Software's QuickPak Professional, and linked with Crescent's P.D.Q.

I drew on several reference sources detailing the operation of the keyboard and the ROM BIOS while preparing MALT and MALTCFG. Of these, by far the most instructive was the article by Michael J. Mefford entitled "An In-depth Exploration of the PC Keyboard and its Interrupt Service Routines", which appeared in the May 1990 issue of *Microsoft Systems Journal* (Volume 5, Number 3). His method for detecting the presence of an enhanced keyboard is incorporated in MALTCFG. Also very useful were the pair of articles by Barry Simon (someone for whom I have a deep admiration because of his work on CTRLALT and STACKEY) entitled "Learning Your Way Around the Keyboard under DOS", which appeared in the December 25, 1990 and January 15, 1991 issues of *PC Magazine* (Volume 9, Number 22 and Volume 10, Number 1). His program BIOSDATA, which was distributed with the articles, was especially helpful in deciphering the operation of the PC keyboard BIOS.

Two other informative resources were *DOS Programmer's Reference (2nd Edition)*, by Terry Dettmann and Jim Kyle, and *The New Peter Norton Programmer's Guide to the IBM PC & PS/2*, by Peter Norton and Richard Wilton. I also referred frequently to *Advanced MS-DOS Programming (Second Edition)*, by Ray Duncan, but not for keyboard information.

Surprisingly, only the Norton book contained even a partial list of the codes generated by the ROM BIOS for legal keys on the enhanced keyboard. To remedy this deficiency, I have prepared a program called SCANASC for use by the programming community (or anyone else who happens to be curious). SCANASC reports the scan and ASCII codes for all keys interpreted by the BIOS keyboard services in hex and decimal and identifies whether the key is available only on the enhanced keyboard. SCANASC will optionally prepare a customized list of the values for all keys pressed during a session for later reference. There is no charge for the program and it is available in the Programming Forum of PCMagnet where it is called SCANAS.ZIP. (Notice the absence of the final "C" in the root name. CompuServe only permits six-letter root names.)

I am indebted to *PC Magazine*, Crescent Software, and Ethan Winer's book *BASIC Techniques and Utilities* for information that permitted the translation of MALT from BASIC to assembly language. MALT 2.0 and above benefitted from the insights gained by study of *Zen of Assembly Language, Volume I, Knowledge* by Michael Abrash. This book is not only a compendium of information about assembly language programming, but also a treatise on how to think. Credit for much of the reduction in MALT's memory usage since Version 1.3 goes to Ralf Brown whose RBkeyswap is a fascinating collection of tricks.

## STANDARD DISCLAIMERS

Most personal computer users realize that only the blandest warranties accompany software. This software is no different from any other in that respect. Accordingly, there are no warranties of merchantability or fitness for a particular purpose, nor will the author be responsible for any damages (incidental, consequential, or otherwise) in excess of the purchase price. I have tested MALT and MALTCFG, and I believe, they will work as described. If you encounter problems, please let me know. I may be reached at the address given below or on CompuServe (ID No. 76170,1627).

## REGISTRATION/LICENSE/COPYRIGHT

MALT is a user supported program. It is not public domain. If you are using MALT for personal use, you should pay a $15 registration fee. MALT has deliberately been priced low to encourage registration. A combined registration for MALT and my other program MPE4WP is $30. If you have already registered MPE4WP, you may register MALT for an additional $10. MALT may not be used in business without a license. For information about securing a license to use MALT in business, please see the section entitled "Business Use". Should you misplace this documentation, registration information may be obtained on demand by typing "MALT /R" at the DOS command line.

## HOW TO REGISTER

To register MALT for personal use, please send your $15 check or money order ($30 for a combined registration with MPE4WP) to:

Michael H. Shacter
10309 Parkwood Drive
Kensington, Maryland 20895

In exchange, you will receive an acknowledgement and at least $15 (or $30) worth of satisfaction. (Payment for orders outside the U.S.A. must by check or money order drawn on a U.S. bank or by international money order. I know this is a nuisance, and I am sorry.) Please note that you will not be sent a disk, only a registration number. If you absolutely must have a copy of MALT on disk, please send an additional $5 ($10 for foreign orders). Please (a) specify 5¼ or 3½ disk size, (b) include 5% sales tax if you are a Maryland resident, and (c) be patient. For your convenience, a registration form is included with this package. Please see the file REGISTER.FRM.

As of the date of this release, MALT may be found on CompuServe in library 3 (Macros/Merges) of the WPUSERS forum, where the most current version is always known as MALT.ZIP. There is no cost for registered users of prior versions to upgrade to the current version (aside from the cost, if any, of securing a physical copy). Registered users may secure the most current version of my programs from me for a $5 shipping and handling fee. Remember to specify disk size.

## BUSINESS USE

A license is required to use MALT in business, after a reasonable trial period. Business use includes use by governmental entities, non-profit organizations, and any other use that is not strictly personal. You may obtain a license by writing to me at the above address. The license fee will depend on the number of MALT users in your business. Discounts are available for multiple copies. Please feel free to submit a reasonable proposal, including any special requirements for customization you may have. Remember, if you are using MALT to make money, I am too, and in this respect, my expectation of a living is superior to yours.

## OTHER RESTRICTIONS

All authorized users are granted a limited license to make copies of MALT without charge, subject to the restrictions contained elsewhere in this document as well these:

(a) MALT must be distributed in absolutely unmodified form, including MALT, MALTCFG, the sample library macro, the registration form and this documentation.

(b) For-profit use of MALT without a license is prohibited.

(c) MALT may not be included or bundled with any other product for any reason.

Not-for-profit user's groups, such as the Capital PC User Group, are permitted to charge a small fee for materials, handling, postage, and general overhead. No other organization is permitted to charge for distribution of copies of MALT.

Electronic bulletin board system operators are encouraged to post MALT on their bulletin board systems for downloading by their users, if the above conditions are met, and if no special fee is necessary to access the MALT files (a general fee to access the BBS is acceptable). Likewise, all authorized users are encouraged to upload to MALT to BBS meeting these requirements.

# APPENDIX

## NAMING MALT-KEY MACROS

MALT uses a set of abbreviations to identify non-alphanumeric MALT-keys. Most people will not care about the file names of their MALT-key macros so long as each MALT-key is consistently associated with a macro file. To assemble a library macro, however, it is essential to know key names. This appendix describes the MALT naming strategy. The abbreviations, while logical, are necessarily terse because of DOS naming conventions (only eight letters) and because of the need for MALT to conserve memory (longer names mean increased memory requirements).

Alphanumeric MALT-key macros follow the naming convention set by WordPerfect: "MALT" plus the letter or number. Thus, MALT-P is associated with the file MALTP.WPM, and, by extension, MALT-4 is associated with the file MALT4.WPM. Punctuation keys cannot be disposed of so easily because most punctuation marks are illegal in file names. For instance, MALT=.WPM would not work because the equal sign may not be included in a file name. In general, most punctuation keys, cursor positioning keys, and function keys use a two-letter abbreviation following the name MALT. For example, MALT-= is associated with the file MATLEQ.WPM and MALT-F1 with MALTF1.WPM. To distinguish between keys on the numeric keypad and identically named keys on the enhanced keyboard's cursor pad, MALT adds an "X" between "MALT" and the key name in the latter case. Accordingly, the file associated with MALT-PgDn on the numeric keypad is MALTPD.WPM and MALT-en PgDn on the enhanced keyboard is MALTXPD.WPM.

The following table lists all non-alphanumeric MALT-keys and their associated abbreviations. Keys preceded by an asterisk are only available on enhanced keyboards.

| KEY | ABBR | KEY | ABBR |
|---|---|---|---|
| ESC | ES | | |
| - | - | | |
| = | EQ | | |
| Backspace | BS | | |
| Tab | TA | | |
| [ | { | | |
| ] | } | | |
| Enter | CR | | |
| *Num Enter | NCR | | |
| ' | ' | | |
| ` | ~ | | |
| \ | BA | | |
| , | CO | | |
| . | PE | | |
| / | SL | | |
| *Num / | NSL | | |
| Num * | NA | | |
| Space | SP | | |
| F1 | F1 | | |
| F2 | F2 | | |
| F3 | F3 | | |
| F4 | F4 | | |
| F5 | F5 | | |
| F6 | F6 | | |
| F7 | F7 | | |
| F8 | F8 | | |
| F9 | F9 | | |

| | |
|---|---|
| F10 | F0 |
| *F11 | F11 |
| *F12 | F12 |
| Insert | IN |
| Delete | DE |
| Home | HO |
| End | ED |
| PageUp | PU |
| PageDown | PD |
| Up | UP |
| Down | DO |
| Right | RT |
| Left | LE |
| Num - | NM |
| Num + | NP |
| *Num 5 | N5 |
| *en Insert | XIN |
| *en Delete | XDE |
| *en Home | XHO |
| *en End | XED |
| *en PageUp | XPU |
| *en PageDown | XPD |
| *en Up | XUP |
| *en Down | XDO |
| *en Right | XRT |
| *en Left | XLE |