EDITING MACROS

How to Edit

In order to edit a macro, press [Ctrl F10] and enter the name of the macro to edit. Then press E to edit. If you get a message to describe the macro, you are creating a new macro and not editing an existing one. Press [F1] to cancel and try again, using the full path name of the macro file. Later, look for and delete any macro (a file ending in .WPM) that has a size of 57 - it is empty.

Look for an arrow <------- at the end of a line. It marks a place that I think you might want to edit. Some macros are designed to be edited. In those I have put the assignments to edit up at the top so they are even easier to find.

Editing the **{ASSIGN}** Statement

Many of the FREE Keyboard macros use advanced macro programming techniques that are difficult to understand or modify. However, most of the changes that you will want to make involve one of the simplest of the macro statements - the **{ASSIGN}** statement.

The **{ASSIGN}** statement is used to save text to a variable. Its form is **{ASSIGN}**Variable~Text~. The tilde character (~) marks the end of the name of the variable and the end of the text. Here is an example:

{ASSIGN}Author~Gabriel•Fineman~

This sets the variable "Author" to contain the words "Gabriel Fineman". The space is shown as a bold dot. To change this to "Joan Smith", first delete the old name without deleting the tilde. It will look like this:

{ASSIGN}Author~~

Then type in the new name. It will look like this:

{ASSIGN}Author~Joan•Smith~

---------- Less used statements ----------

The **{PROMPT}** Statement

The {PROMPT} statement is very much like the {ASSIGN} statement and just as easy to edit. Just be careful not do delete the tilde (~) at the end. Prompts are used to put information on the screen and you will want to change some of the messages in the Help files (usually in a macro ending with _H). For example, you will want to edit SHOW_H, the huge help file for the [Ctrl H] index macro. It consists mainly of a series of {Prompt} statements that describe each macro key. Just change the text and be careful not to delete a tilde (~). Out of Memory? Try

restarting WordPerfect and doing the edit immediately. If this does not work, try starting WP using the /w (workspace) switch or borrowing a copy of ED, the WP Office macro editor.

The **{CHAR}** Statement

The {CHAR} statement is really just prompt statement that also has a variable where the first character in reply to the prompt (like Y for yes or N for no) is kept. Feel free to change these also.

The **{CASE}** Statement

The {CASE} statement evaluates a variable and branches based on the data in the variable. The branches look like this:

Y~Yes~ y~Yes~ N~No~ n~No~

to branch to the {LABEL} "Yes" if the data is "Y" or "y" and to the {LABEL} "No" if the data is "N" or "n".

A typical series is to list choices in a {PROMPT} statement, get the user's selection into a variable using a {CHAR} statement, and then a {CASE} statement to screen the selection and branch to a routine to process the selection.

{^V} and **{^Q}**

In prompt statements you sometimes see various codes. Many are explained in the reference manual under Macro Message Display. I use {^V} a lot because it gives the same attribute (usually Bold) that the user has chosen to emphasize WordPerfect prompts. {^Q} turns off all attributes, like bolding. The ^ is short for {Ctrl} and you create a {^V} by pressing {Ctrl V} twice and {^Q} by pressing {Ctrl V}{Ctrl Q}. It is best to turn off the keyboard with {Ctrl 6} before entering any {Ctrl} codes since if you pick one that you have defined to a macro, a copy of the macro is inserted instead of the control key. You can turn the keyboard back on later by pressing {Ctrl 6} twice.

========== Advanced Macro Editing ==========

If you have had some experience in writing macros, you will notice that most of the FREE Keyboard macros are both complex and heavily commented. One of the best ways to learn how to write macros is by reading existing macros and understanding what they do.

I have tried to always tell you what each line does. The comments are written for someone who has had a four hour course in macro writing or has read the macro appendix of the WordPerfect manual but has had little hands on experience. You will have to keep the manual handy, but should be able to follow what is happening. When you come to a subroutine (call), I suggest that you assume that it does what its title says and read it later.

The macro editor only recognizes the first seven letters of a variable name and the first fifteen characters of a label. This does not stop us from using much longer descriptive names for

both. This is a major method of documentation.

The variables x, y and z are reserved for input and are assumed to never be saved but always available for reuse. The variables t, tt and ttt are also temporary variables that are used in the next line or two. Most of the macros clean up after themselves and release variables. Temporary variables are not released.