

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 is has a size of 57 - it is empty.

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~
```

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.

We 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.