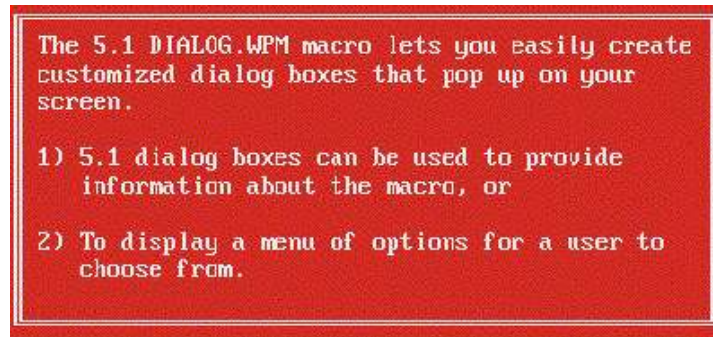


Building Dialog Boxes
WordPerfect Magazine
By Allen Biehl, Macros by Karen Baldwin-Porter and Roger Gagon

Dialog boxes are a great addition to any macro. They attract more attention than a one-line prompt displayed at the bottom of the screen. They pop up in front of a user, demanding to be noticed (see the figure below).



They are not, however, the easiest things to create. They require a collection of commands only a true macro guru can decipher. The included DIALOG.WPM macro helps make the task of creating dialog boxes much easier.

The macro is designed to be run from within the Macro Editor using an (Alt-letter) combination. This lets you create your dialog box prompts without having to know the (Alt-letter) combinations for all of the line-drawing characters. It also simplifies the sizing and positioning of the box. Therefore, be sure to rename the macro to one of these combinations, e.g., ALTB.WPM.

Note: The macro discussed in this article only helps you build the dialog box. It doesn't create an entire macro for you. To effectively use the dialog boxes in macros you create, you'll need to have a good understanding of macro programming.

Using the Dialog Box Generator

Go to the Macro Editor by pressing (Home), Macro Define (Ctrl-F10), typing the name of a macro you wish to create or edit and pressing (Enter) twice.

In the Macro Editor, with your cursor at the position where you need the dialog box codes, press the Alt-letter combination to which you renamed the macro, e.g., (Alt-B).

The macro first asks whether you wish to use a {PROMPT}, {CHAR} or {TEXT} command to display the dialog box. The {PROMPT} command simply displays a message and then accepts no input. The {CHAR} command lets someone type one character in response to your prompt, such as Y for Yes or N for No. The {TEXT} command asks someone to enter a string of information. Press the number corresponding to the command that fits your needs.

Next, if you chose {CHAR} or {TEXT}, the macro asks you to enter a variable name. This variable is used to store the response typed by the user of your macro.

Continue following the prompts. The macro asks you to enter the horizontal and vertical positions of the top left corner, followed by the width of the dialog box. Finally, you're prompted to enter the text you'd like to see in the dialog box. You're limited to four characters less than the width of the dialog box to allow room for the two border characters and two spaces. For example, if your dialog box is 40 characters wide, you're limited to 36 characters. A ruler above the typing area helps you gauge the length of your text.

After you type a message and press (Enter), you're asked if you want to enter another line. Press (Y) Yes if you'd like another line or (N) No if you don't. If you answer Yes, you're guided through the same process for entering text. Keep answering (Y) Yes for each new row you want to create. *Note: Keep an eye on the number following the second {NTOK} command on each line. This number represents the vertical position of your dialog box. If the number ever reaches 23, you'd better answer (N) No because your dialog box only displays up to line 24 and the macro still needs to add the bottom border to your dialog box.*

When you're done, just answer (N) No and the macro creates the bottom border, leaving you in the Macro Editor ready to complete your macro.

How the Macro Works

The first thing the macro does is check to make sure you're in the Macro Editor. If you're not, the macro tells you so (line 2), then quits on line 3.

Lines 6-7 ask what kind of prompt command you want. If you respond with either a (C) {CHAR} or a (T) {TEXT}, lines 10-11 execute, letting you enter a variable name.

Line 13 inserts the command you chose – {PROMPT}, {CHAR} or {TEXT} – into the Macro Editor. If a variable name was entered, line 14 inserts the variable name followed by a tilde (~).

Lines 15-16 ask for the desired position of the box, then lines 17-19 obtain the width of the dialog box. Line 20 calls the AddNtok subroutine (lines 44-46), which enters the proper position commands.

Line 21 adds the left corner border character (f), uses a {FOR} loop to create the top border of the dialog box (), then concludes with the right corner border character (f). Later on (line 41), a similar loop is used to create the bottom border.

Lines 22-38 form the loop that allows you to enter text into your dialog box. Lines 26-29 create the prompt and ruler you can use to enter the text for each line of the dialog box. Lines 30-32 make sure you haven't entered too much text on a line. Lines 33-35 finish off the line by padding with spaces and inserting the right border character ([Set 6,40]).

Lines 36-37 ask if you want to enter more lines of text. If you answer (Y) Yes, you're sent back to the top of the loop (line 22). Otherwise, the macro continues on and lines 39-41 create the bottom border.

Finally, lines 42-43 position the cursor in the bottom left corner of the screen, in case you'll be entering any text when this dialog box displays on the screen.