

Use a Fading Page Border
WordPerfect Magazine
By Elden Nelson

I like my documents to stand up and shout. Even when I don't have anything important to say, I want everybody who sees my work to think it's mission critical.

To really call attention to my documents, I put a border at the top and bottom margins of a page. Not just a dull border with one solid color: my borders fade from black to white or from white to black.

To accomplish this, I've created the included FADEBLAK.WPM macro to give a unique look to just about any kind of document (see figure below).

FADEBLAK.WPM relies on the {FOR} macro command to build this border. You'll find the {FOR} command useful in other macros you create as well. A little later in this article, you'll learn what the {FOR} command is and how to use it.

Using the Macro

You can run the macro from either a blank document screen or from an existing document. Run the macro by pressing Macro (Alt-F10), typing "fadeblak," then pressing (Enter).

The message *Please Wait* appears in the lower left corner of the screen for a moment, then disappears. Since the macro uses graphics lines, you won't be able to see the border you've created until you print your document or view it using View Document – (Shift-F7), 6.

From here, just go about the business of creating or editing your document. *Note: This macro uses Header B and Footer B. Therefore, you shouldn't run this macro in documents that already have a Header B or Footer B defined.*

Understanding the {FOR} command

This macro creates a fade-to-black border effect by putting varying dark lines just above each other. You could manually determine how dark each line should be and where each line needs to be placed, but it would be tedious work. After all, the lines become darker at a steady rate, so there should be a way to have the macro do the math.

That's where the {FOR} command comes in. Sometimes, you'll want part of a macro to run a certain number of times using a certain set of numbers. In the FADEBLAK.WPM macro, for example, you first want to create a 5 percent shaded line. Next, you'll create a 20 percent shaded line, then a 35 percent shaded line, then 50 percent, then 65 percent. As you see, the progression is 5, 20, 35, 50, 65. The macro needs to count from 5 to 65 in increments of 15.

It's this kind of repetition that the {FOR} command is good for. It lets you run a series of commands a specified number of times. You indicate which number the {FOR} starts with, stops with and at what increment it counts each time through the loop. You can tell the {FOR} to count forward or backward, 1 at a time, 3 at a time or more. For instance, you can have the {FOR} command count from 5 to 65 or 65 to 5.

The {FOR} command looks like this: {FOR}Variable Name~First Number~Last Number~Increment~. The first part, "Variable Name," is the name of a variable. This variable is assigned a different number each time it performs the loop. For example, if the {FOR} loop is counting from 1 to 5, the variable is assigned 1 the first time, 2 the second time, and so on, right up to when the variable is assigned 5 for the loop's last pass.

The second part, "First Number," tells the {FOR} command the first number the variable will be assigned or the number that you want the counting to start with. If you want the {FOR} loop to count backward from 65 to 5, like in line 21 of FADEBLAK.WPM, the second parameter will be 65.

The "Last Number" parameter says what the last number will be in the count. If the macro is counting from 1 to 10, this third parameter will be 10. If the macro is counting from 65 down to 5, this parameter will be 5.

The last parameter in the {FOR} command is "Increment." This tells the macro whether to count forward or backward and whether to skip any numbers. If you want the macro to count up one number at a time, like 1, 2, 3, 4, you'd use 1 for this parameter. This means the macro will count by 1 each time the loop runs. If you want the macro to count backward, like 4, 3, 2, 1, you'd use a negative number for this parameter: -1.

This parameter also lets you make the macro skip numbers while it counts. If, for example, you wanted the macro to count every other number, like 1, 3, 5, 7, you would make this parameter 2, or -2 if you were counting backward. In the same way, if you wanted the macro to count every third number, you would make this parameter 3, so the macro would count like 1, 4, 7, 10. So, on line 7 of the macro, it's counting up by 15s (5, 20, 35).

The {FOR} command can't be by itself, though. After {FOR}, there should always be a series of commands you want to run a certain number of times, followed by an {END FOR} command. Each time the {FOR} command counts a number, the series of commands between {FOR} and {END FOR} runs.

When the macro hits the {END FOR} command, it jumps back to the {FOR} command, which goes to the next number in the series. The macro then goes over the series of commands again. This continues until the {FOR} has gone through all the numbers in the series. When this happens, the macro continues with the commands below the {END FOR}.

Understanding the Macro

Now let's see how the {FOR} command works in the FADEBLAK.WPM macro.

Lines 1-3 go to the top of the document and set half-inch top and bottom margins so there'll be room for the borders. Next, lines 4-6 go into the Footer B edit screen and set half-inch left and right margins in the footer. This means that the border will extend further to the left and right than the text in the document. Line 6 assigns variable Height the value 25. This value determines the vertical placement for the line.

The job of the {FOR} loop in lines 7-14 is to create a series of lines, one on top of another, each increasingly dark. The {FOR} command on line 7 means that this {FOR} loop counts from 5 to 65,

stopping at every 15th number. Line 8 goes into the Graphics: Horizontal Line menu. Line 9 increases the value of variable Height by 5, and line 10 uses that value to set the vertical position of the line being made. Because the value of variable Height is increased by 5 each time through the {FOR} loop, each line is 5 points lower than the previous line. Line 11 sets the thickness of the current line to 5 points as well, so there's no white space between lines.

In line 12, variable Shade, which is assigned in the {FOR} statement on line 7, specifies how dark the current line should be. Since variable Shade increases by 15 each pass of the {FOR} loop, each line is darker than the previous, giving a finished product of 5 shaded lines. These are the first 5 lines in the bottom border.

Once the {FOR} loop is finished, the macro goes to the command after {END FOR} (line 14) and continues on from there. Here, line 15 creates another line. This one can't be in the {FOR} loop, since it doesn't fit in with the progression of the shading. Line 16 then exits to the Format: Page screen.

Lines 17-20 do pretty much the same thing as lines 4-6. They go into the Header B edit screen, set half-inch left and right margins and set the initial value for variable Height, which specifies the height of the lines. Line 19 creates the first line in the header, since the shading of this line doesn't fit in with the {FOR} loop that's coming up.

Lines 21-28 work almost like lines 7-14. The big difference is that the {FOR} command is counting backward, from 65 to 5, instead of 5 to 65. This means that the lines in the header start out dark and become lighter as they work down. Once this {FOR} loop is finished, the macro goes to line 29, which leaves the Header B edit screen and then returns you to the document.

