

Find Overused Words
WordPerfect Magazine
By Elden Nelson

When I was a writing tutor in college, I must have seen just about every possible writing mistake ever imagined – along with a few I would never have imagined. There was one problem that everybody (including me) seemed to have, though. People have favorite words and phrases and tend to overuse them. For example, I really have to watch how often I use the phrase "for example." See what I mean? Some people have to be careful of transition words like "however" and "therefore." If you're writing a document on a certain topic, it's likely that you'll type that topic more often than you need.

But, how do you avoid overusing words? After all, most people don't notice the problem in their own writing. The solution is in the attached macro, which lets you quickly find out exactly how often you've used a certain word or phrase in your current document. You just run the included OVERUSED.WPM macro, type the word or phrase you might be overusing and let the macro do the rest. It'll report exactly how many times you've used that phrase in your document.

What's a Loop?

In macros, "looping" is when a certain part of a macro is repeated. Why would you want to do that? Let's use the OVERUSED.WPM macro to demonstrate. The macro needs to find each instance of the text you specify. It does this by going to the top of the document (line 5), then using the Search feature (line 7) to find that word you might be overusing.

The {Search} command on line 7 only finds the word once, though. There may be many more instances in your document. How many more? Of course, you don't know that – it's the macro's job to find out. So, the macro needs to use that command over and over again, until all the occurrences of your overused word are found.

How do you tell the macro that you need to use that {Search} command several times? It's a two-step process. First, you put a {LABEL} command before the part of the macro you want to repeat, in this case, line 6, just before the line that performs the search. The {LABEL} command is a lot like a bookmark; it marks a place the macro can go to later.

You can have several {LABEL} commands in a macro. However, it's very important that they all be unique. You make each {LABEL} command unique by giving each a different name called the "Label Name." This name goes between the {LABEL} command and the tilde (~). In line 6, I used "Check" as my Label Name.

The second part of creating a loop is to put a {GO} command after the part of the macro you want to repeat (see line 10). This {GO} command should use the same Label Name as the {LABEL} command that starts the loop. This {GO} command takes the macro back to the {LABEL} command. The macro then continues on to line 7, which searches for the word or phrase you want to check. The macro goes through lines 8 and 9 (we'll talk about them in a minute), comes to the {GO}Check~ command again, goes back to the {LABEL}Check~ command again, and so on and so on.

The one thing that you need to be careful of when looping macros is that if you don't watch out, the loop could run forever – or at least until you cancel the macro. That's why you need to give the macro some way to tell when the loop should stop and go on to something else.

Here's how the OVERUSED macro solves the problem of looping forever. Take a good look at line 4. This line gives the macro a standing order: "As soon as a search fails, go to the {LABEL}Report~ command." So, when the search looks for your text and doesn't find it, the macro jumps to the {LABEL}Report~ command on line 11. This command is outside the loop on lines 6-10, so the macro continues down through line 14, where the macro ends.

What about Variables?

So far, we've covered how the macro uses looping. But the macro doesn't need to just find each occurrence of the word, it needs to count each instance, too. This macro uses a variable called "Found" to keep track of how many times your word or phrase is found.

Let's track the progress of this variable through the macro.

The first time you see variable Found in the macro is in the {ASSIGN} Found~0~ command on line 2. {ASSIGN} is a command used to give variables a certain value. The word between the command name and the first tilde is the variable name, the number before the second tilde is the value that is assigned to the variable. In this case, variable Found now contains the number 0. This ensures that variable Found starts out at 0. Think of this {ASSIGN} command as pressing the "reset" button for the variable.

The next time you see variable Found is in line 8: {ASSIGN} Found~{VARIABLE}Found~+1~. At this point, line 7 has just found an occurrence of a specific word or phrase, so the {ASSIGN} command increases the value of variable Found by 1. The first time the macro comes across this command, variable Found is assigned the value 1. The next time this command is encountered (remember, this command is part of the loop), variable Found has the value 1. Since the macro adds 1 to this, variable Found now has the value of 2. So, variable Found keeps track of how many times your word or phrase has been located.

When the macro finally leaves the loop, variable Found is in the {CHAR} command on lines 12-14. This command uses variable Found to show you how many times your word has been found in the document.

Using the Macro

To use the macro, press Macro (Alt-F10), type "overused," then press (Enter). A prompt appears, asking you to type the word or phrase you want to check. Type your text, then press (Enter).

Note: Since it uses the Search feature to find words and phrases, the macro is subject to the same rules as Search. So, when you type the word or phrase you want to count, type it in all lowercase letters so that the macro can find instances containing both uppercase and lowercase text.

The macro takes a moment while it counts the instances of that word or phrase in your document, then displays a prompt at the bottom of the screen, reporting the macro's results. Press any key to make the prompt disappear.