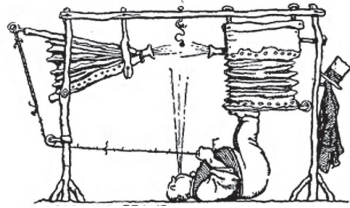




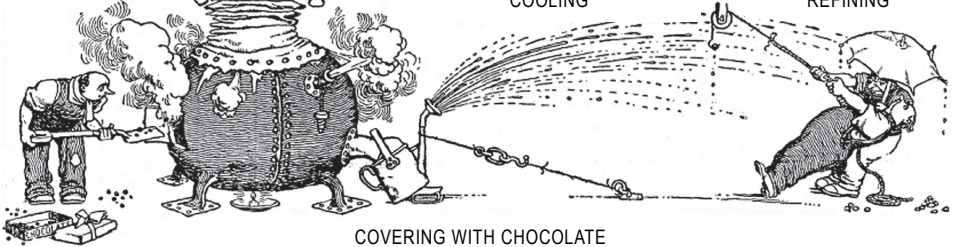
SHAPING



COOLING

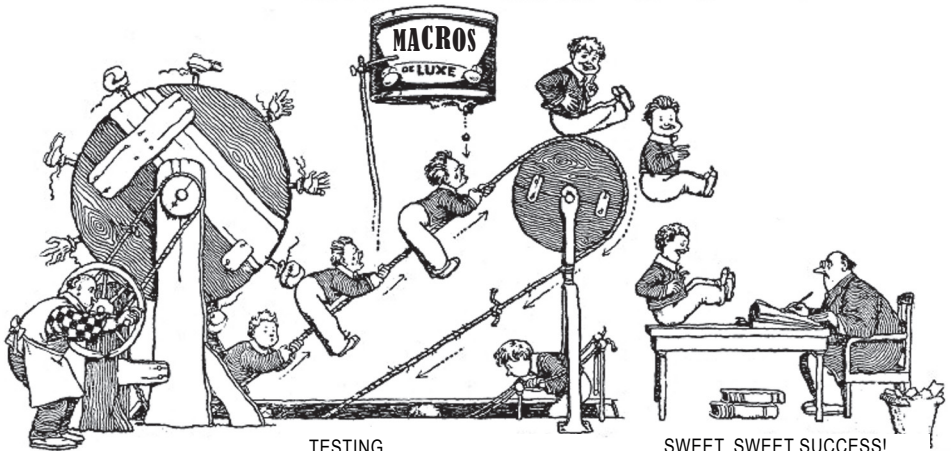


REFINING



COVERING WITH CHOCOLATE

# Macro Cookbook for Microsoft Word



TESTING

SWEET, SWEET SUCCESS!

# Jack Lyon

# Macro Cookbook

*for Microsoft Word*

Jack Lyon

THE EDITORIUM

**Please read this:** The reader (that's you) assumes the entire risk as to the use of this book and the information therein. I don't anticipate any problems, but all computers and Microsoft Word installations are different, and I can't be held responsible for what might happen with yours. To avoid problems, be sure to *back up your files* before trying anything in this book; then you'll have something to go back to if anything goes wrong. As the lawyers say, this book is provided for informational purposes only and without a warranty of any kind, either express or implied, including but not limited to implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement.

ISBN 978-1-4341-0332-1

Copyright © 2012 by The Editorium, LLC  
All rights reserved  
Printed in the United States of America

Cover illustrations (here slightly revised) are by William Heath Robinson, from an ad for Mackintosh's Toffee in the London *Daily Mail*, October 1, 1921.

The Editorium  
3907 Marsha Drive  
West Valley City, UT 84128  
[www.editorium.com](http://www.editorium.com)  
[editor@editorium.com](mailto:editor@editorium.com)

The names of any programs or companies mentioned in this book (including mine) are acknowledged as trademarks of their owners. The Editorium™ is a trademark of The Editorium, LLC.

The Editorium is not affiliated with Microsoft Corporation.

# Contents

<b>Contents</b>	<b>v</b>
<b>Preface</b>	<b>vii</b>
<b>Introduction</b>	<b>ix</b>
<b>1 Understanding Macros</b>	<b>1</b>
Recording a Macro . . . . .	5
Looking at the Macro . . . . .	7
Running the Macro . . . . .	8
Word Happens . . . . .	9
<b>2 Looping Macros</b>	<b>11</b>
Do and Repeat . . . . .	14
Other Looping Commands . . . . .	15
<b>3 Decisions, Decisions</b>	<b>19</b>
“Returning” Information . . . . .	21
The Powers-Wilson Doctrine . . . . .	25
<b>4 Using “Found” Macros</b>	<b>26</b>
Places to Get Macros . . . . .	26
Adding a Macro to Microsoft Word . . . . .	28
Running a Macro . . . . .	29
<b>5 Adapting “Found” Macros</b>	<b>30</b>
Clarifying Macro Structure . . . . .	31
Understanding and Adapting Macro Commands . . . . .	33
A Real-World Example . . . . .	40
In Summary . . . . .	49

<b>6</b>	<b>Basic “Ingredients”</b>	<b>50</b>
	Moving . . . . .	50
	Selecting . . . . .	51
	Moving <i>and</i> Selecting . . . . .	52
	Inserting . . . . .	52
	Deleting . . . . .	52
	Using Bookmarks . . . . .	53
	Using Revisions . . . . .	53
	Checking . . . . .	54
	Displaying Messages . . . . .	54
	Finding and Replacing . . . . .	55
	Running Macros . . . . .	58
<b>7</b>	<b>Secret “Ingredients”</b>	<b>59</b>
	Looking Under the Hood . . . . .	60
	Overriding a Word Command . . . . .	61
	Word Happens—Again . . . . .	63
	Listing Word’s Commands . . . . .	65
<b>8</b>	<b>Storing Macros</b>	<b>66</b>
	Creating Your Own Template . . . . .	67
	Adding a Template Automatically . . . . .	69
	Making Macros Accessible . . . . .	70
<b>9</b>	<b>Additional Examples and Practice</b>	<b>71</b>
	Recording a Simple Macro . . . . .	71
	Recording a Find-and-Replace Macro . . . . .	74
	Recording a Complex Macro . . . . .	77
	Automatically Selecting the Quotations . . . . .	83
	Running the Macros with Another Macro . . . . .	87
	The Power of Macros . . . . .	88
	<b>Index</b>	<b>89</b>

# Preface

This book includes lots of macro commands, but you probably don't want to type them into Microsoft Word in order to use them. For that reason, I've made the macros available in a Word document that you can download here:

[www.editorium.com/cookbookcommands.zip](http://www.editorium.com/cookbookcommands.zip)

Some people worry that running a macro will hurt their computer, and there definitely are some nasty macro viruses floating around out there. You'll find more information here:

<http://support.microsoft.com/kb/187243>  
<http://support.microsoft.com/kb/211800>

However, macros you record or create usually aren't dangerous unless you use them to try something stupid. ("Gee, I wonder if I can modify this macro to wipe out my hard drive.") Nevertheless, you should experiment with macros only on junk documents; don't use any documents you're not willing to have destroyed. And if you do mess something up, have the presence of mind to close your document without saving it. Most important of all, be sure to *back up your files* before using them with anything in this book; then you'll have something to go back to if something goes wrong.

This book was written primarily with Word 2010 and 2011 in mind. However, if you're using an earlier version of Word, such as 2003 or 2004, you can still use this book; you'll just need to be aware of a few things:

- In Word 2007, instead of clicking the “File” menu, you'll click the round Office button at the top left of your Word window.
- In earlier versions of Word (except Word 2007), you'll need to click the “Tools” menu to manage templates, get to macro commands, and change Word's options<sup>1</sup> (such as file locations).
- In earlier versions of Word (except Word 2007), template names end with a “.dot” extension rather than the new “.dotm” in Word 2010 and 2011.

---

1. To change options on a Macintosh, click the “Word” menu and then “Preferences.”

# Introduction

Understanding macros in Microsoft Word isn't particularly difficult. What's difficult is learning to program in Visual Basic for Applications (VBA), which is *not* the same thing. Programming in VBA means understanding and using Word's "Object Model," including its objects, properties, and methods (huh?), and that's a big undertaking. If you're interested in that, the Microsoft Word MVP website provides an excellent tutorial to get you started:

<http://word.mvps.org/FAQs/MacrosVBA/VBABasicsIn15Mins.htm>

There are some excellent books as well:

Guy Hart-Davis, *Mastering VBA*, ISBN 978-0782144369

Lee Hudspeth, Timothy-James Lee, and Woody Leonhard, *Word 97 Annoyances*, ISBN 978-1565923089

Allen L. Wyatt, *Word VBA Guidebook*, ISBN 978-1613590300

The point of *this* book, however, is to help you understand, create, and use Microsoft Word macros *without* learning to program in VBA.<sup>2</sup> In fact, if you can do what

---

2. This book includes some material from my newsletter *Editorium Update*. However, I've adapted and modified that material for use here in a more structured context. For that reason, I hope you'll read and work through the material even if you've seen some of it before.



you need without learning to program, so much the better. There are actually several levels in the hierarchy of macro use, from simple to complex:

1. Record and run the procedures you usually do manually.
2. Find and run macros other people have created.
3. Adapt those macros for your own use.
4. Write your own macros from scratch.

But I'll let you in on a secret: Usually only programmers write macros from scratch. The vast majority of Word users simply (1) record and run the procedures they do most often, or (2) use macros other people have created. Many of those macros are free; people give them away just to be helpful. Others aren't free but may be well worth buying; I consider the macros I sell at the Editorium ([www.editorium.com](http://www.editorium.com)) to be in that category. If you charge \$50 an hour for your services, and you can buy a \$30 macro that saves you an hour, you've just made \$20 without doing any work. I'll take as many of those hours as I can get! But that's not all: Every time you use that already-paid-for macro to save an hour in the future, you'll make your usual \$50 without doing any work!<sup>3</sup> So what is that macro really worth? A lot more than you might realize.

Of course, you may not be able to find an existing macro that does what you need. If that's the case, the next level in the hierarchy is to adapt existing macros—something that's not as difficult as you might think. If you want to go

---

3. This assumes, of course, that you're charging by the job, the page, or the word (which you should be) rather than by the hour. If you charge by the hour, you're limiting the amount of money you can make, because no matter how efficient you are, you have only so many hours to sell. If, on the other hand, you charge by the job, the page, or the word, your hourly income grows every time you figure out a way to work more efficiently. That's what macros are all about.

further than that, you can learn to write your own macros from scratch—in other words, learn to program. In this book, however, I just want to show you how to use macros *without* learning to program. Is that really possible? Let's find out.

## Chapter 1

# Understanding Macros

A macro is like a cake—it includes certain ingredients added in a certain order to get a certain result. For example:

### Ingredients

- 1 $\frac{3}{4}$  cups all-purpose flour
- 2 cups white sugar
- $\frac{3}{4}$  cup unsweetened cocoa powder
- 2 teaspoons baking soda
- 1 teaspoon baking powder
- 1 teaspoon salt
- 2 eggs
- 1 cup buttermilk
- 1 cup vegetable oil
- 1 teaspoon vanilla extract

## Directions

1. Preheat oven to 350 degrees F (175 degrees C).
2. Grease and flour two 9-inch round cake pans or one 9-by-13 inch pan.
3. In large bowl combine flour, sugar, cocoa, baking soda, baking powder, and salt.
4. Add eggs, buttermilk, oil, and vanilla.
5. Beat for 2 minutes on medium speed. (Batter may be thin.)
6. Pour into prepared pans.
7. Bake at 350 degrees F (175 degrees C) for 30 to 40 minutes, or until toothpick inserted into center of cake comes out clean.
8. Cool for 10 minutes, then remove from pans and finish cooling on a wire rack.
9. Fill and frost as desired.

Now let's look at some *macro* ingredients. What do you suppose this one does?

```
Selection.Font.Bold = True
```

Yep. It makes the currently selected text turn bold. How about this one?

```
Selection.MoveDown Unit:=wdParagraph, Count:=1
```

That's a little tougher, but really not so bad. Evidently it moves the selection (cursor position) down one paragraph. Want to move down two paragraphs? Try this:

```
Selection.MoveDown Unit:=wdParagraph, Count:=2
```

How about moving down two lines?

```
Selection.MoveDown Unit:=wdLine, Count:=2
```

How about selecting the two lines as you move down?

```
Selection.MoveDown Unit:=wdLine, Count:=1, _
Extend:=wdExtend1
```

Now, you probably wouldn't be able to write down those "ingredients" without looking them up—and neither would I. But (for the most part) we can *understand* them when we see them. That's an important point, because if we can understand them, we can use them and combine them in various ways to get the results we seek. For example, let's say we want to select the next paragraph *and* make it bold. Here's the recipe, with just two ingredients:

```
Selection.MoveDown Unit:=wdParagraph, Count:=1, _
Extend:=wdExtend
Selection.Font.Bold = True
```

Here's a recipe that's a little more complex. See if you can figure out what it does:

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "supercede"
    .Replacement.Text = "supersede"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
```

---

1. The lowline \_ character at the end of the first line (and elsewhere in this book) just breaks up the macro command if it gets too long for easy reading. You can delete it and merge the two lines if you want to.

```
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

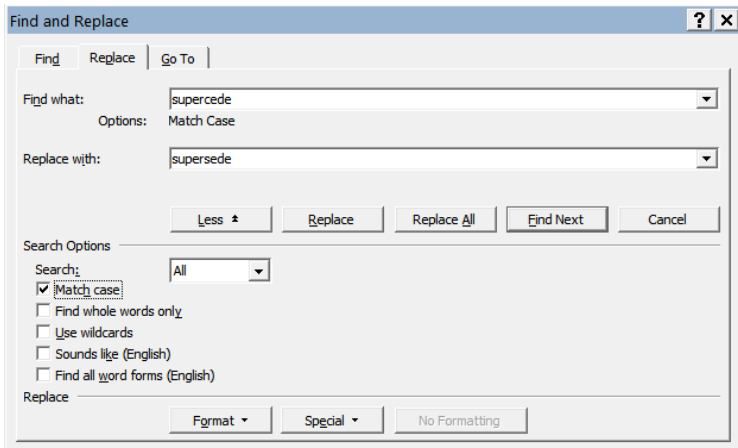
You may not understand the specific purpose of each line, but you can probably tell that the macro finds every instance of *supercede* in a selection and replaces it with the correctly spelled *supersede*. Now consider this line from the macro:

```
.MatchCase = False
```

What would happen if we changed it to this?

```
.MatchCase = True
```

Yes, the macro would find every instance of *supercede* but would ignore a capitalized *Supercede*. Hmmm. That must mean “.MatchCase” corresponds to the “Match Case” option in Word’s Find and Replace dialog:



And so it does!

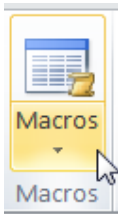
## RECORDING A MACRO

So there's a correspondence between Word's features and Word macros—not too surprising. The question is, how can you find out what the macro is for any particular feature? The answer is, you *record* a macro while *using* the feature. Try this in Microsoft Word—no, seriously, try it:

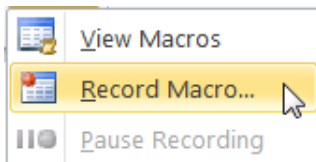
1. Click the “View” menu at the top of Word's ribbon interface.



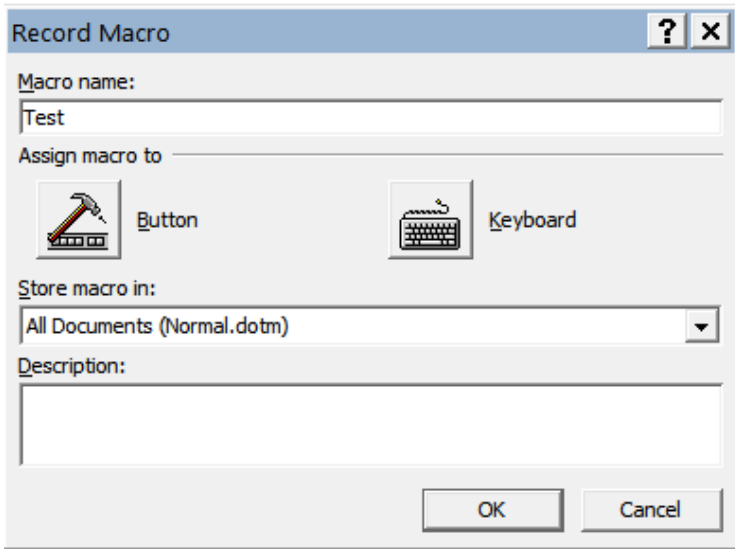
2. Click the little arrow under the “Macros” button (on the right).



3. Click “Record Macro.”



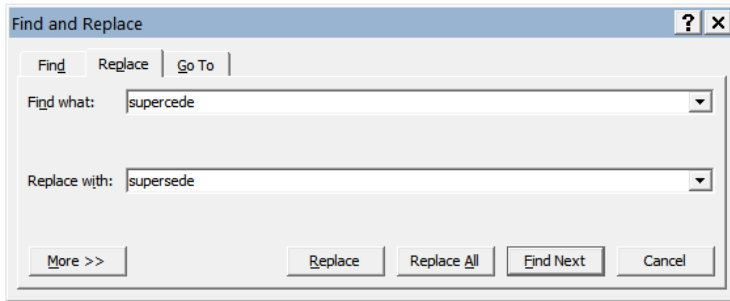
4. Give your macro a name (“Test” will do) and click the “OK” button.



You’ve started the recorder, and Word will now record your actions. Now do this:

1. Press CTRL + H to bring up Word’s Find and Replace dialog.
2. In the “Find What” box, type *supercede*.
3. In the “Replace With” box, type *supersede*.
4. Click the “Replace All” button.
5. To finish up, click the “OK” button and then the “Close” button.





Word has replaced every instance of *supercede* with *supersede*. But you've been recording that, remember? So now turn off the recorder:

1. Click the View menu at the top of Word's ribbon interface.
2. Click the little arrow under the "Macros" button.
3. Click "Stop Recording."

## LOOKING AT THE MACRO

Finally, take a look at the macro you just recorded. To do so, press ALT + F11. The VBA editor will open, and you'll see your macro at the bottom of the window. It will look like this:

```
Selection.Find.ClearFormatting
Selection.Find.Replacement.ClearFormatting
With Selection.Find
    .Text = "supercede"
    .Replacement.Text = "supersede"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
```

```
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
```

In other words, it will look exactly like the macro we were examining earlier! So, in a way, the macro *is* the feature, and the feature *is* the macro—two sides of the same coin.

## **RUNNING THE MACRO**

Note that you don't have to understand or even look at a macro to use it. In fact, if there are certain repetitive procedures you use over and over again, you can record those procedures (just as you did above) and then play them back as needed, saving yourself a lot of time and trouble. Here's how to run a macro you've already recorded:

1. Click the "View" menu at the top of Word's ribbon interface.
2. Click the "Macros" button.
3. Click "View Macros."
4. Select the macro by name ("Test" in this case).
5. Click the "Run" button.

The macro will play, reproducing the steps you took manually while you were recording it. And for many tasks, that may be all you need.

Any time you have repetitious tasks to perform, record those tasks and then play them back. Let the computer do the work—that's what it's for.

# Index

- accessibility of macros, 70
- adding macros to Microsoft Word, 28–29
- add-in templates
  - activating, 69
  - adding, automatically, 69–70
  - creating, 67–69
- all caps, fixing, in body text, 47–49
  
- Beverley, Paul, 27
- bookmark commands, 53
  
- checking commands, 54
- commands
  - accessing list of, 65
  - bookmark commands, 53
  - checking commands, 54
  - deleting commands, 52–53
  - find commands, 55–56
  - in hidden Word features, 60–61
  - inserting commands, 52
  - message box commands, 54–55
  - moving and selecting commands, 51–52
  - moving commands, 50–51
  - overriding, 61–64
  - replace commands, 57–58
  - revision commands, 53–54
  - run macro commands, 58
  - selection commands, 51–52
- complex macros, recording, 77–88
- cursor, commands for, 50–51

- decision-making macros
  - overview of, 19–21
  - and Powers-Wilson Doctrine, 25
  - and returned information, 21–25
- deleting commands, 52–53
- Do-While loop, 15, 21
  
- Electric Editors, 27
- “ephemeralization,” 88
  
- features. *See* hidden features; Word features, and recording macros
- find-and-replace macros, recording, 74–77
- find commands, 55–56
- For Each/Next loop, 16
- For/Next loop, 16
- “found” macros
  - adapting, 30–31
  - adding, to Microsoft Word, 28–29
  - cautions regarding, 26
  - and clarifying macro structure, 31–32
  - running, 29
  - sources for, 26–28
  - and understanding and adapting macro commands, 33–49
- Fuller, R. Buckminster, 88
  
- GoTo loop, 17–18
  
- hidden features
  - and listing Word commands, 65
  - and overriding Word commands, 61–64
  - viewing, 59–60
  - viewing underlying commands for, 60–61
  
- inserting commands, 52
  
- looping macros
  - additional looping commands, 15–18
  - and decision-making macros, 21
  - steps for, 11–14
  - versatility in, 14–15

*Macros for Writers and Editors* (Beverly), 27  
message box commands, 54–55  
moving and selecting commands, 52  
moving commands, 50–51

Normal template, 66

Powers, Hilary, 27  
Powers-Wilson Doctrine, 25

recorded macros  
  problems with, 9–10  
  running, 8, 74  
  viewing, 7–8

recording macros  
  looping macros and, 11  
  and overriding Word commands, 63–64  
  and Powers-Wilson Doctrine, 25  
  problems in, 9–10  
  recording complex macros, 77–88  
  recording find-and-replace macros, 74–77  
  steps for, 5–7, 71–74

repeat commands. *See* looping macros

replace commands, 57–58

returned information, for decision-making macros, 21–25

revision commands, 53–54

running macros, 8, 29, 58

selection commands, 51–52, 83–87

storing macros  
  and automatically adding templates, 69–70  
  cautions regarding, 66  
  creating templates and, 67–69  
  and macro accessibility, 70  
  structure, clarifying macro, 31–32

templates  
  activating, 69  
  adding, automatically, 69–70  
  creating, 67–69

Title Case macros, 40–47

- understanding macros
  - and adapting macro commands, 33–49
  - clarifying macro structure and, 31–32
  - looking at recorded macros, 7–8
  - overview of, 1–4
  - recording macros, 5–7
  - running macros, 8
  
- variables, 24n2
  
- While/Wend loop, 16
- With/End With loop, 17
- Word features, and recording macros, 5–7, 8. *See also* hidden features
- “Word-VBA Code Samples for Newbies & Others” (Zawawi), 50
- Wyatt, Allen, 27, 30
  
- Zawawi, JoJo, 50