

An Interview with Robert Delwood, Author of *The Secret Life of Word: A Professional Writer's Guide to Microsoft Word Automation*

By Robert Reynolds

This spring, I did an interview with Robert Delwood, who is a senior systems analyst in the NASA Johnson Space Center community, to talk about his new book, *The Secret Life of Word: A Professional Writer's Guide to Microsoft Word Automation*. A writer turned programmer, Delwood connects the worlds of writing and technology by writing Microsoft Office solutions to help people and teams automate their tasks, whether those tasks are simple or complex. This book is targeted toward technical and other professional writers, and gives the reader an in-depth look at the hidden capabilities of Word, and shows how to take advantage of these capabilities without being a programmer.

Thanks for participating in this interview. You are a writer-turned-programmer. How did that come about?

I'm a writer by education (I have a Journalism degree) and a programmer by vocation (I have been programming since seventh grade). The

convergence of the two was incidental. After I'd been tech writing for a few years, at one position I needed some additional tools, so I ended up creating them myself (with the ROM-based BASICA if anyone remembers that). My management liked the tools and encouraged me to continue. I then programmed tools for my writing group and later ended up as an architect in the newly emerging desktop group. I found out Microsoft actually has a title combining the two, programmer-writer. Most see this as a writer among programmers but the truth is we'll have to start seeing it as programmer among writers if we're to get any real tools.

What was your intended audience for the book? The private/public sector? Universities? Individuals? Is the actual audience different from what you expected?

The working title was "Word Automation for Technical Writers." I expanded the scope to power

Interview continued on page 5

Interview continued from page 4

users. There's nothing for dummies here; rather it's for experienced users who already know the interface and who want to get more from sequencing a series of tasks, or take advantage of the automation. Therefore, it's really aimed at information or knowledge workers.

What inspired you to write this book? Was there a demonstrated need for sharing your knowledge that wasn't already written elsewhere?

I believe tech writers are the most put upon group in that they have the least software support. They may have a few tools but given a large table of error codes to reformat, ensure document consistency, reviewing production logs, or finding an important bug fix from a database, it's mostly a manual or brute force effort. Moreover, many of the challenges are unique to individual writers, and often as onetime events. No programming group is going to lend development support for that. In short, if writers want any tools, they'll have to write them themselves.

There are a lot of how-to books teaching the user interface (which tech writers don't need), and a lot more about Word programming (which tech writers don't want to know), so obviously there's a gap right in the middle. There are even fewer instructional books about using automation to reduce the tediousness of the tasks. My book hopefully addresses these issues:

1. Automation is usually seen as programming world, which scares tech writers off, and in doing so, it loses the ability to help with everyday tasks, and:
2. Automation is so much more than programming. Macros aside, for starters there

are Building Blocks, import/export, linking, smart tags, and fields. The longest chapter is about Find and Replace, since it can do such much more than people think.

With all this in mind, I wanted to explain automation without the programming.

What are some examples of tasks that people can do with your book? In other words, what are the takeaways?

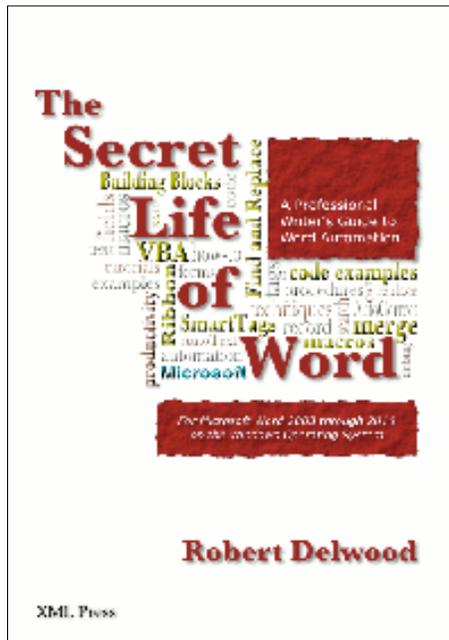
The most important takeaway is to change the way you think about automation. My mantra is: **If you do anything twice, automate it.**

That means you're not only likely to do it several times again, but also your team members might need it. Whatever the task, there's probably a way to make it faster/better/more accurate. Another takeaway is that **automation doesn't have to be complex.** Take the F4 key, or Repeat Last Command. Everyone knows it repeats the last simple command, such as bolding text but it also repeats all the options selected from a dialog box. For instance, in the Paragraph dialog if you select an Alignment, Spacing, Line spacing, Keep with Next, and

Page break before, all those will be applied with F4. Further, if you wait to do all similar formatting at one time, you can get significant speed and quality improvements. Of course, Word usually gives you three ways of doing everything, so you could also record a macro if you needed to use that formatting combination at different times.

What would you say are the most commonly used automated tasks in Word?

Word does such a great job at disguising automation! Page numbers and hyperlinks are two forms no one considers as automation and yet they're just a convenient way of placing pre-



Interview continued on page 6

Interview continued from page 5

formatted fields. Recording macros is another obvious way although many still don't use them as much as they could. Although Find and Replace by itself is not automation, there is an astonishing unused capability. You can transform Word's ugly HTML into clean XHTML in a few steps.

Can you walk me through the building blocks automation, for those of us not familiar with building blocks? How would this benefit folks in the technical communication industry?

Building blocks (BB) are another type of non-programming automation, and are new since Word 2007. They represent pre-designed document parts. Using them, you can quickly format entire reports. For instance, using only the predefined BBs you can use the Austere style to add headers, footers, cover page, and even sidebars and drop quotes, to end up with a consistent look and feel for the document. Although this sounds like a template or AutoText, BBs give you more flexibility in formatting, placing and moving items; a template wouldn't give you a sidebar or drop quote complete with a content control. Additionally, a BB can be several pages, not just one page or one item, also complete with formatting. That means for the same report, different departments such as legal, marketing, or development can have consistent formatting but different customized inserts; again, something templates can't do.

Upon first glance, the examples in your book, in particular the if statements and loops, may be difficult for people with no programming skills. How do you minimize the intimidation factor?

This book isn't about programming, and it won't make you a programmer. With that said, code does form the backbone of the book but it's intended to be pasted into Word's macro editor and run. This cookbook approach hopefully encourages reader to try it. Each example gives you enough background that you can change what need to. For more extensive changes, nearby or other example may get closer to what you want to do. Ultimately, you

may want to learn Visual Basic for Applications (VBA) formally.

A commonly-asked question is if recording macros are so powerful, why would you want to write or modify code? Simply because recorded macros can't do everything. You can't record looping (or going through multiple cases at one time), or decision making (testing for and then responding to specific cases). Moreover, that level of customization may not be enough. For instance, you can't record a macro to convert the selected text into a hyperlink. The Insert Hyperlink dialog requires you to type the target, making it specific to that one link. To make it more general but using the selected text as the link target, you'd have to write your own:

```
Sub InsertHyperLink()
    ActiveDocument.Hyperlinks.Add
    Anchor:=Selection.Range, _
        Address:=Selection.Text,
    ScreenTip:=Selection.Text, _
        TextToDisplay:=Selection.Text,
    Target:="_blank"
End Sub
```

The book, available at Barnes and Noble and Amazon, has over 200 examples of these Word automation capabilities, and dozens of detailed procedures. Many of these examples are freely available for download. That and other information about Robert's book can be found at: xmlpress.net/publications/word-secrets

Robert E. Reynolds is now in his fourth year as a technical writer (and first as a DITA processor) after wearing hats as a software tester and landscape designer/environmental planner. He currently works at EDIFECS in Bellevue, Washington, and he can be reached at rer22@cornell.edu.