



# Communicator

Special Supplement:  
50 years of change

Autumn 2018

Compression  
Platform

The skill of writing well will  
always matter

A history of word processors

Changes in print technology

25 years of online help

Memories of times gone by



# Macros: A brief history of time (saving)

**Mike Mee** summarises macros from the key-combo to a full programming language.



Mike Mee

## Introduction

My apologies to the late Stephen Hawking for misappropriating the title of his famous book. Mainly because it looked a lot better than my first attempt at a title, which was 'Macros, macros, and more macros!'.  
How long have macros been a part of our favourite word processors?

Which word processor was the first to allow its end-users to incorporate them? These and maybe a few other questions will hopefully be answered in my brief history of them.

**Note:** I have tried not to step on Neil Perlin's toes as his article covers the history of word processing software. To be honest, I think I have the easier task in this special supplement.

## Early word processors

This view depends on whether you class keyboard shortcuts as the forerunner to macros, or not. If you think they are two separate things, then skip this section and move on to Microsoft's first attempt with their WordBASIC.

It was only in and around the early 1980s that word processing moved from the big and bulky machines of old into the home computer range. Personally I remember typing up a piece of work for my English lesson using TasWord 2 on my ZX Spectrum. The only downside was printing it on a ZX Printer, which burnt the print onto special thermal paper. And this does not last very long.

Even Tasword 2 had key combinations in it (this was 1983 or so) and you could use them to invoke text centring and such like.

As the machines on the desk got bigger and better, and more features were added such as menu bars and dialog boxes, the ability to cut, copy and paste was included as standard in every word processor.

## Cut, copy and paste

My first encounter with cut, copy and paste was with the word processing program WordStar. I have vague memories of using WordStar in my 6th form via a Research Machines Nimbus with twin (the luxury!!) 8" floppy disks! The lecturer would boot up CP/M first and then we all took turns writing a mini-essay on it and saving it to another massive floppy disk.

After an apparently disastrous conversion of version 3.0 from CP/M to MS-DOS, WordStar version 4.0 was rewritten completely.

This new generation of word processor for IBM PCs (and early Macintosh machines) was heading towards the inclusion of a 'coding language' available to the end user so that they could add more functionality to the software.

There was a fairly badly received conversion of WordStar across to Windows but by then Word 2.0 was already on the market and, like them or loathe them, Microsoft did their



*The author George R.R. Martin (he of Game of Thrones fame) stated in an interview on a US chat-show that he still runs WordStar 4.0 on a DOS computer. He has a second 'modern' machine for emails, but keeps his DOS machine for his script-writing. Some habits are hard to break, it would seem.*



utmost to ensure that WordStar (and other word processors for Windows) did not last very long.

## WordPerfect for DOS and Windows

When I was discussing the premise of what this article was going to cover, I mentioned that I did not know of any macro recording options for DOS-based word processors.

Katherine (our illustrious editor) sent me a few examples of some training courses she used to run covering WordPerfect version 5.1 (for DOS). It had the option to create macros for simple tasks and they could also be chained up to run one after the other. This wasn't macro 'programming' as it is these days though but they were powerful enough to ensure the average user would get a speed boost in their everyday life.

I doffed my cap in Katherine's general direction for putting me right about WordPerfect as I had never used it. One of my HND in Computing lecturers swore by it but most of us (much younger) students swore at it. Our only real experience of a DOS-based word processor was Microsoft's Word 5.5. We used it for a few months, but then the college ensured that we too would also only use Windows-based software and we were all given access to Microsoft Word 2.0 for Windows.

The macro support was there but like WordPerfect, it wasn't a programming language as such. This all changed with the next major release of Microsoft Office in 1995.

## WordBASIC (Office 1995)

When Microsoft introduced its first attempt at a macro programming language that sat behind the Microsoft Office suite, it was with Office 95.

This was based on the BASIC language but it was a limited variant. There were 900 or so commands available in the library, but the language was fairly rigid in its syntax. Here is an example:

```
Sub MAIN
FormatFont .Name = "Arial", .Points = 10
Insert "Hello World"
End Sub
```

The above would insert 'Hello World' in Arial, (size 10) font at the current selection point in the document.

There are still websites out there that mourn the loss of WordBASIC and list some of the commands that did not 'make the upgrade' to Word 97 and the new VBA variant of BASIC.

### VBA (Office 1997 onwards)

VBA is an implementation of Microsoft's Visual Basic language: an event-driven programming language, which was (and still is) built into most Microsoft Office applications. It first appeared as an upgrade (well, it was actually a complete replacement) to WordBASIC in Microsoft Word 97 and has been upgraded in every version of Word since then.

There were a lot of changes between VBA and WordBASIC. VBA was object-based, whereas WordBASIC was not. The general coding style had also changed as VBA was based on Microsoft's then current development language, Visual Basic 6.

To demonstrate the changes, the previous WordBASIC example now looked like this in VBA:

```
Public Sub Main()
    With Selection.Font
        .Name = "Arial"
        .Size = 10
    End With
    Selection.TypeText Text:="Hello World"
End Sub
```

The object was the current selected item of text, or a paragraph or just a single word. VBA did not care, as it just looked at whatever was selected. Then it changed the font and size and printed "Hello World".

### Well, almost every version of Microsoft Word ...

However, there is one exception to the above and that is Word 365.

As that is all run online, you cannot add your own VBA macros. Only via the offline variant, which is usually Word 2016.

A recent contract role showed me how good, but also how annoyingly bad Word 365's non-existent support was for macros.

I was just glad I had Word 2016 on my laptop and I could 'work offline' with the documents and use my Word Toolbox just fine to fix the common errors that I found.

### OpenOffice

Apache's OpenOffice also allows the use of macros in its word processing package. It even allows you to import and convert your VBA code, although it will not be a 100% perfect conversion due to certain differences between the software packages.

You can find out more about how Open Office handles VBA code (in general) by visiting the following Wiki page: [https://wiki.openoffice.org/wiki/VBA\\_interoperability\\_in\\_OpenOffice](https://wiki.openoffice.org/wiki/VBA_interoperability_in_OpenOffice)

As a quick test of the VBA import, I tried to get OpenOffice (via LibreWord) to run my Word Toolbox. Suffice to say it would not work and I managed to crash the application! More tinkering and a slower import and test methodology might get me a bit further.

### The future?

It all depends what will happen with regards to macros within your favourite word processor.

Microsoft has confirmed that Word 2019 will still support VBA macros, and equally, as it pushes more and more towards moving everyone onto Office 365, there could be less and less chances to run VBA macros.

The way forward, it seems, is to convert your VBA macros into JavaScript (or VB.NET) and move them into Microsoft's new style of add-ins.

Then again, with Apache's OpenOffice slowly taking chunks out of Microsoft's sales, maybe the way forward is to move your development across to its word processor instead. The plus point is that your code will be run on more operating systems and not just Windows and Mac OS.

As and when, or if, I start down that path, I will be sure to let you all know! 

### References

Apache Open Office (2012) 'VBA interoperability in OpenOffice' [https://wiki.openoffice.org/wiki/VBA\\_interoperability\\_in\\_OpenOffice](https://wiki.openoffice.org/wiki/VBA_interoperability_in_OpenOffice) (accessed July 2018)

Hawking S (1988) *A brief history of time*. London. Bantam Dell Publishing Group.

Khaw C (2014) 'Game of Thrones author George R.R. Martin explains why he writes on a DOS machine' The Verge. [www.theverge.com/2014/5/14/5716232/george-r-r-martin-uses-dos-wordstar-to-write](http://www.theverge.com/2014/5/14/5716232/george-r-r-martin-uses-dos-wordstar-to-write) (accessed July 2018)

### Terminology

**BASIC** (an acronym for Beginner's All-purpose Symbolic Instruction Code) is a family of general-purpose, high-level programming languages whose design philosophy emphasises ease of use. <https://en.wikipedia.org/wiki/BASIC> (accessed July 2018)

**CP/M**, originally standing for Control Program/Monitor and later Control Program for Microcomputers,[3][4][5] is a mass-market operating system created for Intel 8080/85-based microcomputers. <https://en.wikipedia.org/wiki/CP/M> (accessed July 2018)

**Microsoft QuickBASIC** is an Integrated Development Environment and compiler for the BASIC programming language that was developed by Microsoft. <https://en.wikipedia.org/wiki/QuickBASIC> (accessed July 2018)

**Visual Basic for Applications (VBA)** is an implementation of Microsoft's event-driven programming language Visual Basic 6. [https://en.wikipedia.org/wiki/Visual\\_Basic\\_for\\_Applications](https://en.wikipedia.org/wiki/Visual_Basic_for_Applications) (accessed July 2018)

**WordBASIC** was a subset of Microsoft QuickBASIC customised for word-processing. It was replaced by Visual Basic for Applications (VBA) when Word 97 was released. Contrarily to VBA, WordBasic was not object-oriented but consisted of a flat list of approximately 900 commands. <https://en.wikipedia.org/wiki/WordBASIC> (accessed July 2018)

**Mike Mee MISTC** SC contract technical author

E: [mugukmail@gmail.com](mailto:mugukmail@gmail.com)

T: @Mug\_UK

W: [www.mikestoolbox.co.uk](http://www.mikestoolbox.co.uk) – my toolkit for Word 2007-2018