

QuickStart Tool

Microsoft Word Macro Language

Getting a good start in any new technology or programming language often depends on finding the best available information. The TechRepublic.com QuickStart Tools give you the information you need to quickly grasp the fundamentals of developing in a new IDE, using a new programming language, or working with a new technology.

Besides explaining the basics, the TechRepublic.com QuickStart Tool: Microsoft Word Macro Language shows common tasks, exposes strengths and weaknesses, demonstrates some of the best uses of the technology, and lists a variety of other online and offline resources that can help you build a solid foundation of practical knowledge.

Table of contents

Macro automation	3
Common tasks	4
Strengths	6
Weaknesses	6
Best uses	7
Online resources	7
Other resources	8
Additional resources	8
About TechRepublic.com	9

Macro automation

One of the benefits of computers in general, and word processors in particular, is the potential for automating repetitive tasks. Beyond the commonly recognized word processor benefits of spell checking, backup copies, and collaboration features rests the productivity of the macro language.

In Microsoft Word, macros are created with a scripting language loosely based on Visual Basic called [Visual Basic for Applications](#) (VBA). Most often, Word macros are used to automate procedures, establish common forms and templates, and enforce uniformity in an enterprise setting. Because VBA is a scripting language, macro code is capable of structured logic, accepts user input, and can catch exceptions and errors. This means a Word macro can be as simple or as complicated as you need it to be.

My first macro

In Microsoft Word, you can create a very basic macro by simply capturing key strokes using the built-in macro recorder. For this example, we'll use Word 2002 to create a macro that will insert a corporate letterhead into any document created in the Normal.dot template.

Begin

While recording keystrokes, mouse movements, and commands seems to be a simple proposition, remember that the macro recorder will record everything—so if you mistype some text and then correct it while still recording, those events will be reproduced each time the macro is run. Those excess steps will decrease the efficiency of your macro, so it is a good idea to practice what you will be recording.

For this example, you will be typing this text and issuing these commands :

- In Word 2002, start a new blank document.
- Type [ALT], t, m, r, and change Macro1 to Spacely.
- Click OK.
- Navigate with the mouse to View | Header And Footer.
- In the Formatting toolbar, click Align Center.
- Type this text:
Spacely's Sprockets
1630 Lyndon Farm Court
Louisville, KY 40223
- Click the Close button on the Header And Footer toolbar.
- Click the Stop Recording button on the Recorder toolbar.

You have just created a Word Macro. To see it in action, start a new Word document and go to Tools | Macro | Macros, click on Spacely in the Macro Name box, then click the Run button.

The code

Here is what the above macro code looks like:

```
Sub Spacely()  
`  
` Spacely Macro  
` Macro recorded 5/21/2004 by  
`  
    If ActiveWindow.View.SplitSpecial <> wdPaneNone Then  
        ActiveWindow.Panes(2).Close  
    End If  
    If ActiveWindow.ActivePane.View.Type = wdNormalView Or ActiveWindow. _  
        ActivePane.View.Type = wdOutlineView Then
```

```
ActiveWindow.ActivePane.View.Type = wdPrintView
End If
ActiveWindow.ActivePane.View.SeekView = wdSeekCurrentPageHeader
Selection.ParagraphFormat.Alignment = wdAlignParagraphCenter
Selection.TypeText Text:="Spacely's Sprockets"
Selection.TypeParagraph
Selection.TypeText Text:="1630 Lyndon Farm Court"
Selection.TypeParagraph
Selection.TypeText Text:="Louisville, KY 40223"
ActiveWindow.ActivePane.View.SeekView = wdSeekMainDocument
End Sub
```

Note how the recorder transformed your various keystrokes and mouse movements into the proper VBA form. Actually coding this simple macro by hand would have taken a considerable amount of effort, even for someone well-versed in Visual Basic for Applications.

Common tasks

Task

Steps

Find

```
Sub Test ()
\
\ Test Macro
\ Macro recorded 5/24/2004 by
\
    Selection.Find.ClearFormatting
    With Selection.Find
        .Text = "QuickStart"
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute
End Sub
```

Search and replace

```
Sub Test ()
\
\ Test Macro
\ Macro recorded 5/24/2004 by
\
    Selection.Find.ClearFormatting
    Selection.Find.Replacement.ClearFormatting
    With Selection.Find
        .Text = "QuickStart"
        .Replacement.Text = "Quick Start"
        .Forward = True
        .Wrap = wdFindContinue
```

```
.Format = False
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute Replace:=wdReplaceAll
End Sub
```

If... then... else
conditional
statement

The code below will search for the text "QuickStart" and if the font is bold change the color to red; if not it will change the color to green and the font to all caps.

```
Sub Test()
`
` Test Macro
` Macro recorded 5/24/2004 by
`

Selection.Find.ClearFormatting
With Selection.Find
.Text = "QuickStart"
.Replacement.Text = "Quick Start"
.Forward = True
.Wrap = wdFindContinue
.Format = True
.MatchCase = False
.MatchWholeWord = False
.MatchWildcards = False
.MatchSoundsLike = False
.MatchAllWordForms = False
End With
Selection.Find.Execute
If Selection.Font.Bold = True Then
Selection.Font.Name = "Times New Roman"
Selection.Font.Size = 12
Selection.Font.Bold = True
Selection.Font.Italic = False
Selection.Font.Underline = wdUnderlineNone
Selection.Font.UnderlineColor = wdColorAutomatic
Selection.Font.StrikeThrough = False
Selection.Font.DoubleStrikeThrough = False
Selection.Font.Outline = False
Selection.Font.Emboss = False
Selection.Font.Shadow = False
Selection.Font.Hidden = False
Selection.Font.SmallCaps = False
Selection.Font.AllCaps = False
Selection.Font.Color = wdColorRed
Else
Selection.Font.Name = "Times New Roman"
Selection.Font.Name = "Times New Roman"
Selection.Font.Size = 12
Selection.Font.Bold = True
Selection.Font.Italic = False
```

```
Selection.Font.Underline = wdUnderlineNone
Selection.Font.UnderlineColor = wdColorAutomatic
Selection.Font.StrikeThrough = False
Selection.Font.DoubleStrikeThrough = False
Selection.Font.Outline = False
Selection.Font.Emboss = False
Selection.Font.Shadow = False
Selection.Font.Hidden = False
Selection.Font.SmallCaps = False
Selection.Font.AllCaps = True
Selection.Font.Color = wdColorGreen
End If
End Sub
```

Strengths

Easy to create and use	With the built-in recording feature, it is easy to create sophisticated Word macros just by recording the keystrokes and mouse movements necessary to complete the task at hand.
Availability of program conditional branching	While many macros can be created solely by the capture of keystrokes and mouse movements, the ability to execute more complex conditional branching and looping functions is also available.
Automation of repetitive tasks	In many situations, the required manipulation of document data is repetitive. Word macros can automate these repetitive tasks, increasing overall productivity.
Transferable	As a scripting language, VBA macros can be applied to any Word document running on any PC with Word installed. The macros can easily be applied to all applicable documents or workstations. A macro can be saved with a document, which means it is always available.
Third-party software not required	The Word VBA macro language is part of the standard Microsoft Word installation.

Weaknesses

Easy to create and transfer	When it comes to security, macro programs that are easy to create and transfer can be a problem. In recent years, some of the more destructive viruses have been written using the features of VBA.
Strictly a Microsoft Office language	The VBA macro language is only applicable to the Microsoft Office Suite.
Lacks a robust development environment	Writing VBA code from scratch to create a Word macro can sometimes be cumbersome.
Lack of control	Because macros can literally be recorded and saved by anyone, it may be difficult to completely control the makeup of workstations.

Best uses

Repetitive tasks in a Microsoft Office environment

For those enterprises using Microsoft Office in general and Word specifically, the VBA macro language can be used to automate repetitive tasks, such as filling out forms, manipulating and inserting text, or moving files across a network. With the help of the built-in recorder, users can often write their own macros to address individual or department-specific tasks that may not be applicable to other parts of the organization.

When consistency is important

The use of Word macros in the form of document templates can ensure consistency across the enterprise in situations where consistency in presentation, wording, and coding is paramount to the organization's mission, such as law firms, doctor's offices, or other professional services.

Online resources

[Microsoft Office Developer Center: Word](#)

Get information about creating and deploying Word-based solutions, including code samples and downloads.

[Word macros—Working with Word macros](#)

A four-part tutorial on using Word macros is presented and several links to advanced tutorials are included.

[Microsoft Visual Basic for Applications \(VBA\) homepage](#)

This site includes a comprehensive list of additional resources on VBA, including recent feature additions and available technical resources.

[Free Visual Basic for Applications—VBA Training & Tutorials](#)

This page presents an extensive list of free VBA training guides and tutorials.

[Microsoft Word public newsgroup forum](#)

The public Internet newsgroup forum where questions relating to Microsoft Word are answered.

Other resources

[VBA Developer's Handbook, 2nd Edition](#)

By Ken Getz and Mike Gilbert. Sybex, 2001, 2nd Book and CD-ROM edition, ISBN: 0782129781.

[Microsoft Office XP Inside Out](#)

By Michael J. Young and Michael Halvorson. Microsoft Press, 2001, Book and CD-ROM edition, ISBN: 0735612773.

[Mastering VBA 6](#)

By Guy Hart-Davis. Sybex, 2000, Book and CD-ROM edition, ISBN: 0782126367.

[VBA for Dummies](#)

By Steve Cummings. For Dummies, 2001, 3rd edition, ASIN: 0764508563.

Additional resources

[Use this Word macro to paste unformatted text quickly and easily](#)

About TechRepublic.com

TechRepublic is dedicated to serving the needs of IT professionals with its informative articles, books and CDs, downloads, discussion center, technical Q&A, and peer directory. Our mission is to focus our content offerings on the IT topics that keep you up at night. After identifying these areas of concern to IT professionals, we concentrate our editorial efforts on providing relevant, real-world solutions. This QuickStart Tool is just one of the many tools we have on hand to help you with your difficult IT projects.

Books & CDs: Browse our extensive catalog of highly sought after reference material.

Downloads: A TechRepublic membership provides you with free access to our expansive download library of helpful tools, white papers, and article compilations.

Discussion Center: Join other IT pros in discussions about technical, career, and management issues.

Technical Q & A: A free marketplace for questions and answers about technical, strategic, and career development issues.

Peer Directory: A home base to find other TechRepublic members with similar professional interests.

Sign up: Get informative e-newsletters sent directly to your inbox: Get up-to-date information sent directly to your inbox with TechRepublic's e-newsletters. Choose from Strategic Edge topics tailored to IT Decision makers, or get helpful technical solutions with our Technical Tips titles. We also have coverage of past articles appearing on our Web site in our NetNote e-newsletters. Give it a try; it's free with a TechRepublic membership, and you can unsubscribe at any time.