VBA for Technical Writers

a.k.a what to do when the macro recorder isn't enough

Adrian Morse
Documentation Manager at Picis

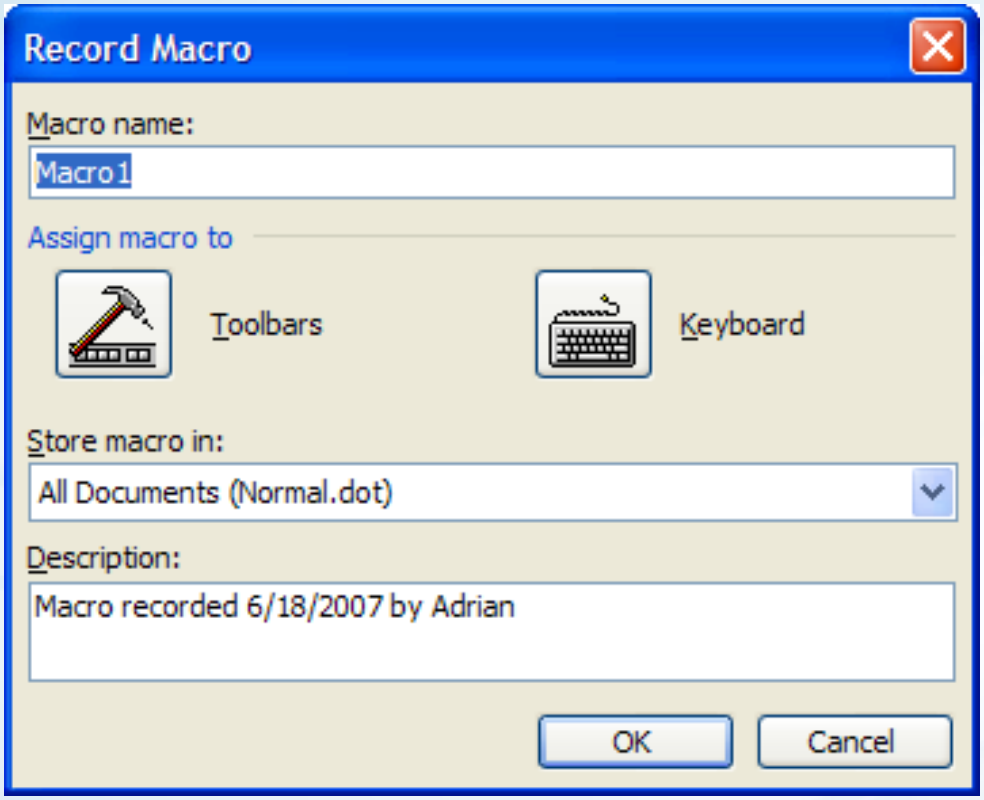
Topics

- The macro recorder
- The VBA Editor
- Basic VBA concepts
- Adding a user interface
- Adding “intelligence”

Topics

- **The macro recorder**
- The VBA Editor
- Basic VBA concepts
- Adding a user interface
- Adding “intelligence”

The Macro Recorder: Introduction

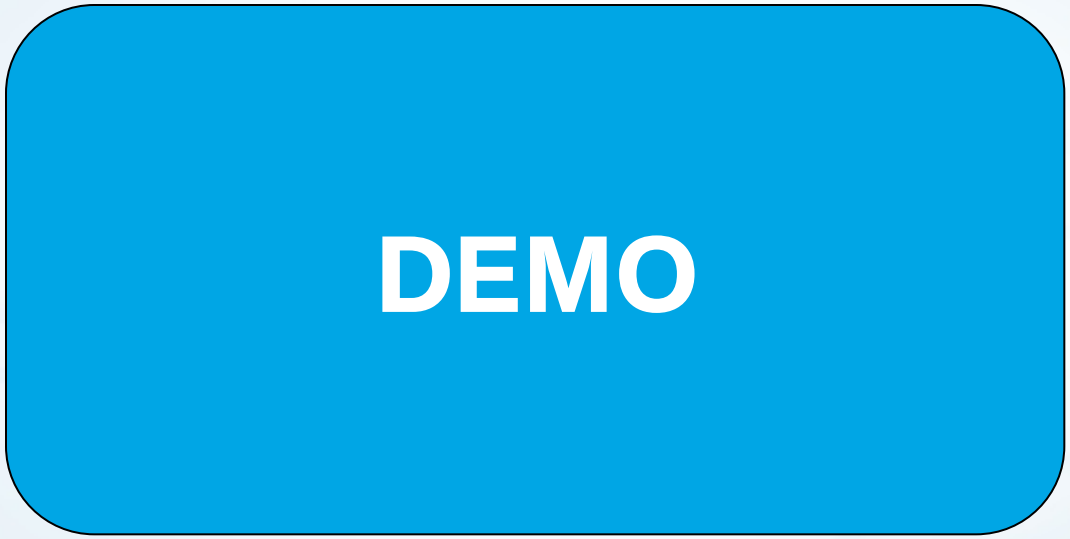


- Macros automate tasks that we expect to repeat
- Macros can be stored in the document or the “Normal” template
- Can create a toolbar button and keyboard shortcut for the macro.

The Macro Recorder: Limitations

- Guesses the user intentions — interprets actions in the most primitive way
- No user interface (for showing data or allowing user to enter it)
- No “intelligence” — for example, it cannot take action based on the outcome of previous events
- Cannot handle errors

The blue monkey

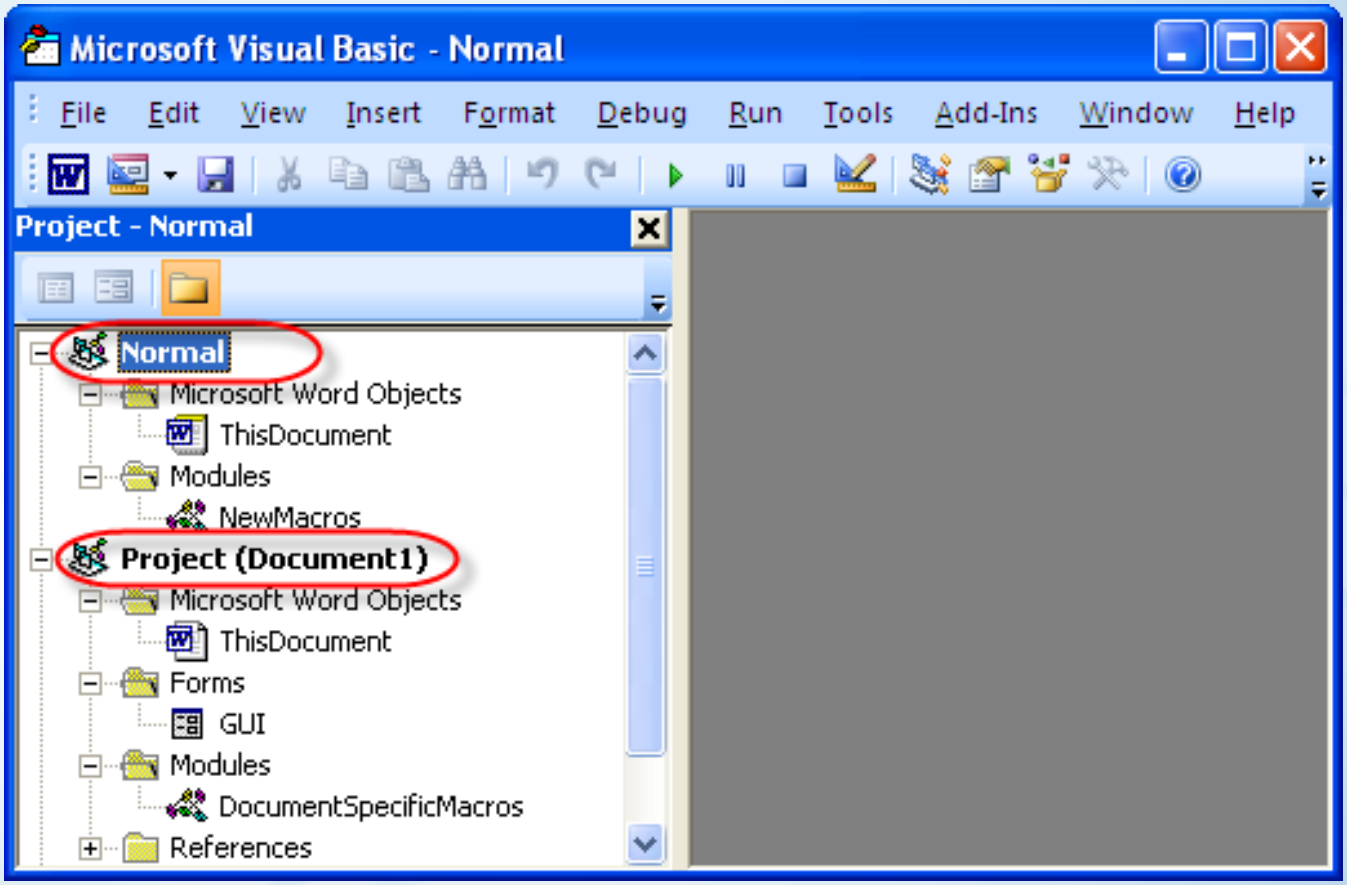


Topics

- The macro recorder
- **The VBA Editor**
- Basic VBA concepts
- Adding a user interface
- Adding “intelligence”

The VBA Editor: The project pane i

- **ALT+F11** opens the VBA Editor
- There is a “project” for the Normal template and each open document
- You may also see a project for any addins you have



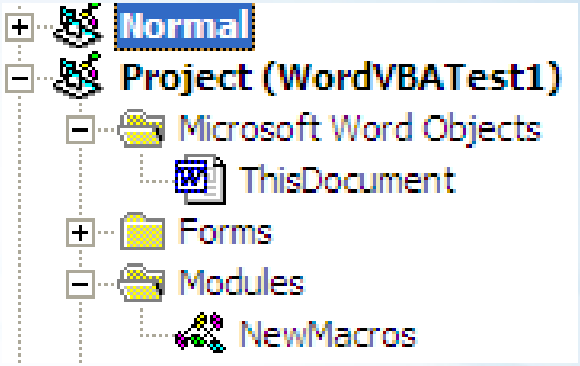
The VBA Editor: The project pane ii

- **ThisDocument** stores any code you want to run in response to an action (event).

  Example: When the doc is opened or closed

- **Forms** stores any GUI windows that you want to show to the reader (to provide info or so the user can enter data)

- **Modules** stores macros you create with the recorder or directly in the VBA Editor



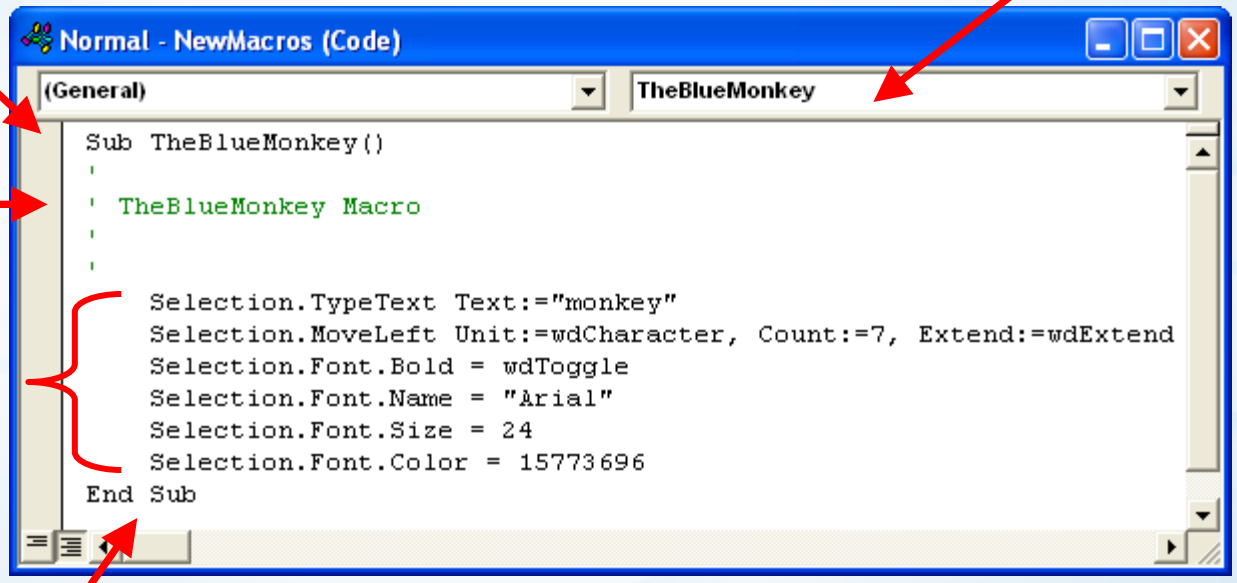
The VBA Editor: Inside the blue monkey

Macros start with “Sub”

Macro name

Comments added automatically

Actions performed by the user



Macros end with “End Sub”

The *bold* blue monkey

DEMO

Topics

- The macro recorder
- The VBA Editor
- **Basic VBA concepts**
- Adding a user interface
- Adding “intelligence”

Basic VBA Concepts: Objects i

- VBA is based on the idea of “objects”
- An object is anything that users can see and manipulate in some way e.g. a paragraph, a table, a text selection...
- Objects are arranged in a hierarchy with object classes including other objects.

  Example: The ‘Document’ object contains the ‘Paragraph’ object

Basic VBA Concepts: Objects ii

- Objects are based on classes, which are like templates for the object. You can create objects from existing classes or define your own classes.

  Example: You could create an object class to keep track of the time you spend working on documents during the day.

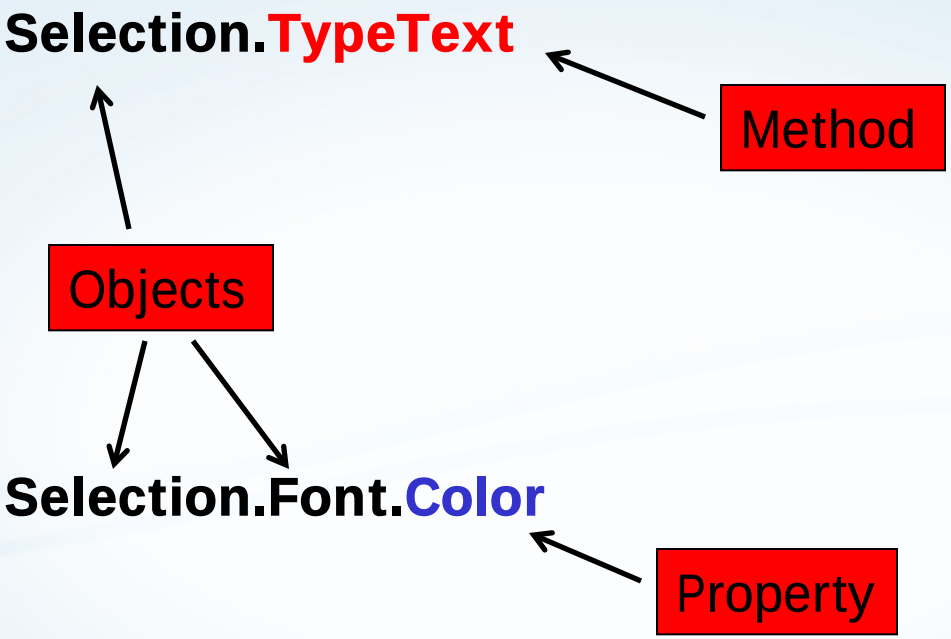
- You can interact with an object in three ways:
  - Change a **property** associated with it
  - Make it perform a task by activating an associated **method**
  - Define a procedure that runs whenever an **event** affects the object

Basic VBA Concepts: Object analogy

You can think of an object like a car...

- Its **properties** are its physical characteristics—model, colour, engine size...
- Its **methods** are actions the car can do—accelerate, brake, turn...
- Its **events** are actions that can be performed on it—turn the ignition key, press the horn, open the door...

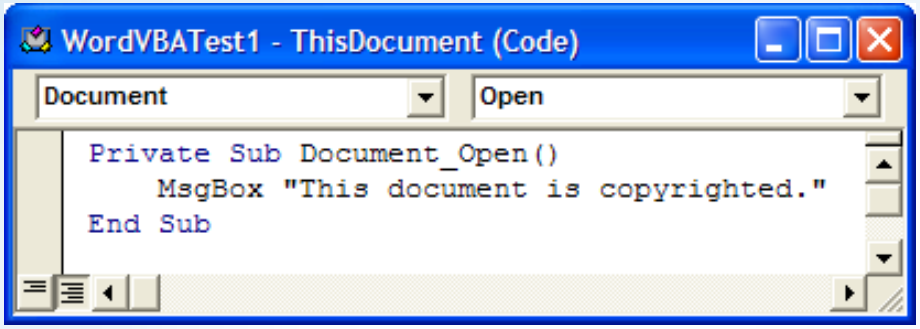
Basic VBA Concepts: Methods and properties



- The macro recorder often bases its code on the “Selection” object, one of the most important objects in Word VBA

Basic VBA Concepts: Events

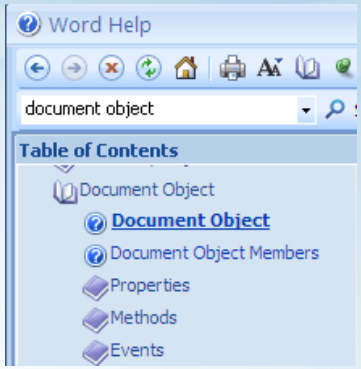
- This code in the *ThisDocument* file...



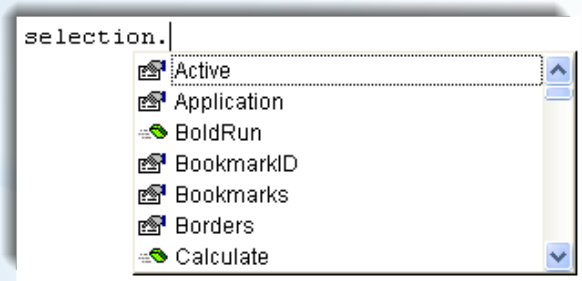
...causes this message box to be shown when the document is opened

- The object is the active document. The event is “open”
- Many objects do not have events associated with them

Basic VBA Concepts: Getting help



- Search for an object name in the help **TOC** to see methods, properties & events for that object



- Type a dot after an object name to see all possible methods and properties for that object

- Use the object browser

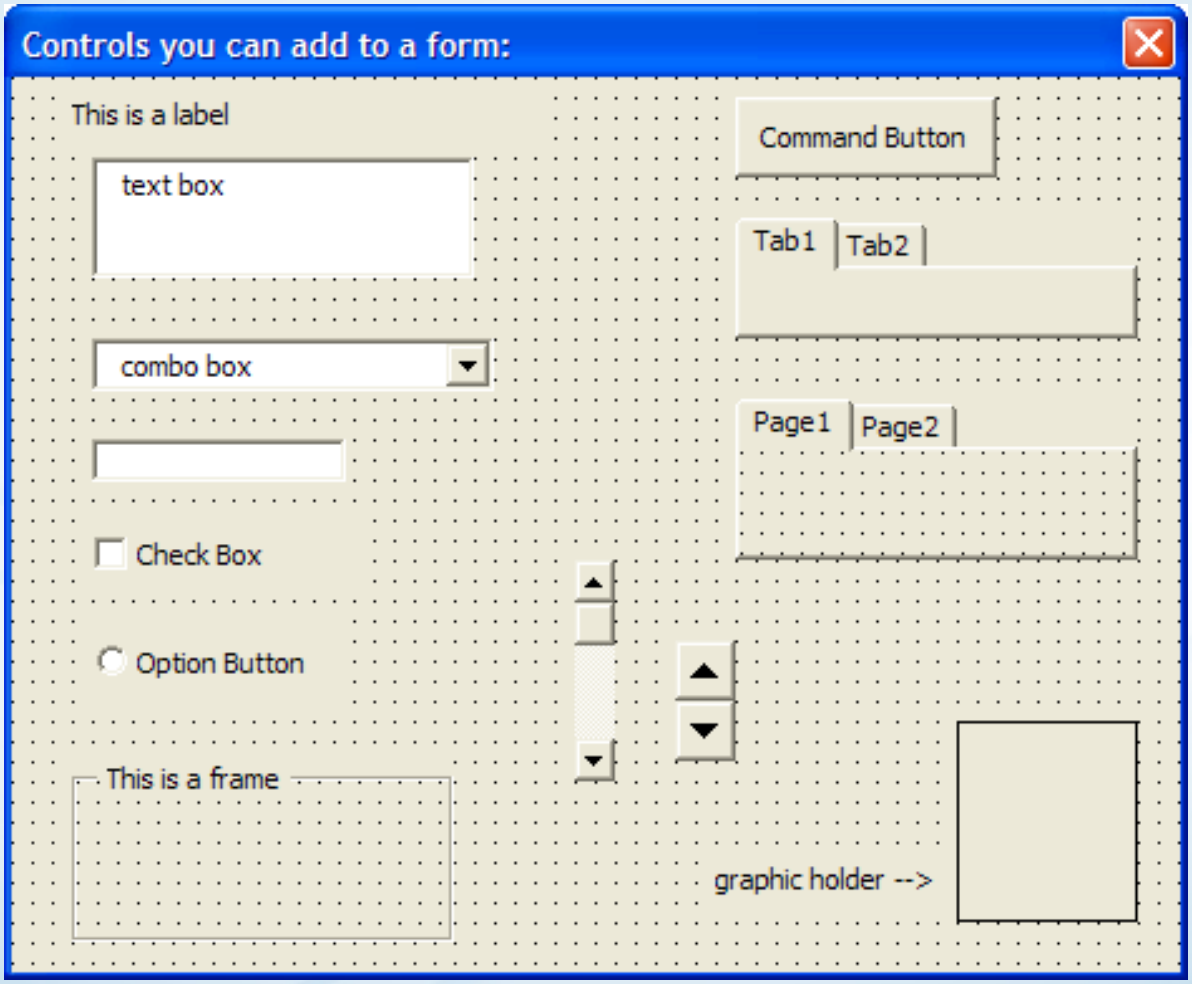
The *responsive* blue monkey

DEMO

Topics

- The macro recorder
- The VBA Editor
- Basic VBA concepts
- **Adding a user interface**
- Adding “intelligence”

Adding a User Interface: What you can do



The *inquisitive* blue monkey

DEMO

Topics

- The macro recorder
- The VBA Editor
- Basic VBA concepts
- Adding a user interface
- **Adding “intelligence”**

Adding Intelligence: Loops i

Means “repeat the code below for each value of x between 1 and10”.

```
For x = 1 To 10
  Selection.TypeText Text:="monkey"
  Selection.MoveLeft Unit:=wdCharac
  Selection.Font.Bold = True
  Selection.Font.Name = "Arial"
  Selection.Font.Size = 24
  Selection.Font.Color = 15773696
  Selection.MoveRight Unit:=wdCharacter, Count:=1
  Selection.TypeParagraph
Next x
```

Repeating the code is called *looping*. There are many ways to do it. This example shows a simple ‘For...Next’ loop.

The *loopy* blue monkey

DEMO

Adding Intelligence: Loops ii

```
For x = 1 To 10
  Selection.TypeText Text:="monkey"
  Selection.MoveLeft Unit:=wdCharac
  Selection.Font.Bold = True
  Selection.Font.Name = "Arial"
  Selection.Font.Size = 24 + 2*x
  Selection.Font.Color = 15773696
  Selection.MoveRight Unit:=wdCharacter, Count:=1
  Selection.TypeParagraph
Next x
```

In this loop we increase the font size by 2 points each loop

The *increasingly loopy* blue monkey

DEMO

Adding Intelligence: Conditions

There are different ways to apply conditions. This example shows a simple 'If...Then' loop

Means "run the code below if the document file name is *Blue.docm* "

```
If ActiveDocument.Name = "Blue.docm" Then
      Load UserForm1
      UserForm1.Show
Else: MsgBox ("Rename the document to Blue.docm")
End If
```

Means "otherwise (if the document is not named *Blue.docm*) show this message…

The *conditional* blue monkey

DEMO

Key Points to Remember

- The macro recorder — Great for getting started but many limitations
- The VBA Editor — **ALT+F11** opens the editor
- Basic VBA concepts — Macros are based on **objects** that have **properties**, **methods** & **events**.
- Adding a user interface — You can easily create a GUI and add code “behind” its controls
- Adding “intelligence” — You can enhance code with looping, conditions…

What Next?

- Play around with recording and editing macros
- Look at the Word MVPS site (http://word.mvps.org/FAQs/MacrosVBA/)
- Get a book!
- Join a forum...e.g. the VBA forum on the MSDN site

That's all folks!