



Introduction

Once a macro has been recorded, the resultant instructions are housed in a program called Visual Basic. These instructions (or code) is what Excel calls on when executing (running) the macro. Editing a macro involves opening the macro in Visual Basic to view the instructions and modifying the instructions accordingly. In this self help document, you will learn how to perform basic modifications to an Excel macro using Visual Basic.

Process Summary

Introduction

Editing a Macro

The Visual Basic environment

1. Editing Text
2. Adding a Message Box



This self help document is very technical in nature. A solid understanding of Excel is necessary to perform these steps.

Editing a Macro

1. Click the down arrow under **Macros** from the **Macros** group on the **View** tab (Fig 1, 1).
2. Choose **View Macros** (Fig 1, 2). A dialog box will appear (Fig 2).
3. Choose the macro that needs editing and click **Edit** (Fig 2, 1). Visual Basic will load and show you the macro code (Fig 3).

Figure 1

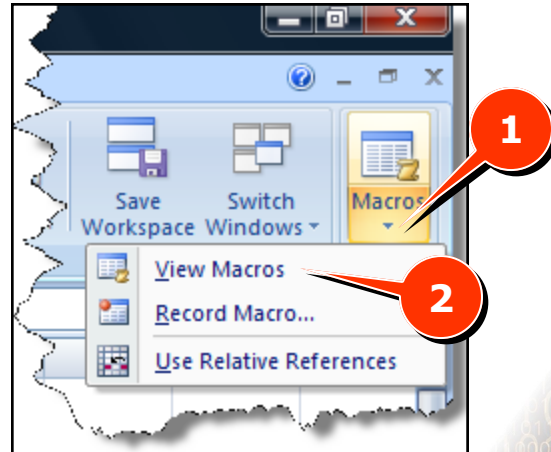
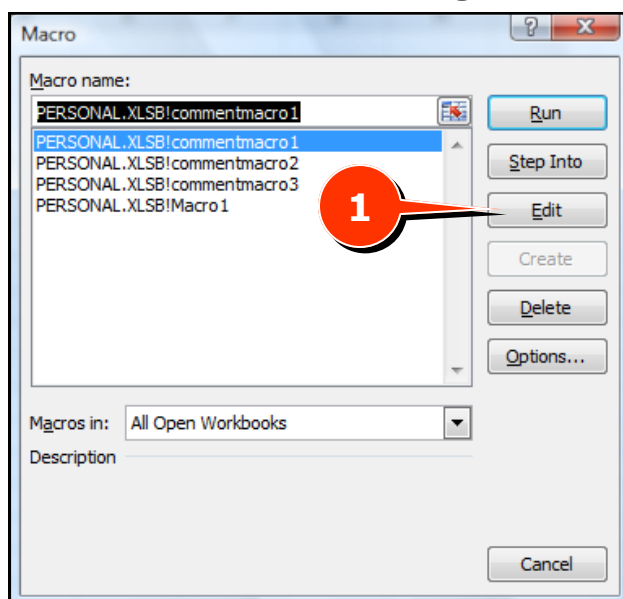


Figure 2



The Visual Basic Environment

Macros in edit mode are viewed in a program called Visual Basic. When Visual Basic launches, the Excel program is minimized and a Module is launched (Fig 3).

Figure 3

```

Sample Worksheets.xlsx - Module1 (Code)
(General) | timein

Sub timein()
    '
    ' timein Macro
    '
    Range("C8").Select
    ActiveCell.FormulaR1C1 = "=NOW()"
    Range("C8").Select
    Selection.Copy
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Application.CutCopyMode = False
    Range("C11").Select
End Sub
  
```

Changing Font Color

1. Locate the line of code that lists the font color (Fig 4, 1).
2. Highlight the text.
3. Change the font color name and number to a theme color of your choice. There are 8 to choose from ranging from Light, Dark, and Accent 1-6.

Figure 4

```

ActiveCell.Offset(-2, 0).Range("A1").Select
With Selection.Font
    .ThemeColor = xlThemeColorLight2
    .TintAndShade = 0.599993896298105
End With
  
```

Changing a Font

1. Locate the line of code that has the words **With Selection.Font** (Fig 5, 1)
2. Insert a line below.
3. **Type** `.Name = "FontName"` (where Fontname is the name of the desired font) (Fig 6)

Figure 5

```

With Selection.Font
    .ThemeColor = xlThemeColorLight2
    .TintAndShade = 0.499984740745262
End With
  
```

Figure 6

```

With Selection.Font
    .Name = "Algerian"
    .Size = 20
  
```

Changing a Cell Location

1. Located the line of code that has the cell reference. Note: there may be several instances of this.
2. Highlight the text (Fig 7, 1) and replace as necessary.

Figure 7

```

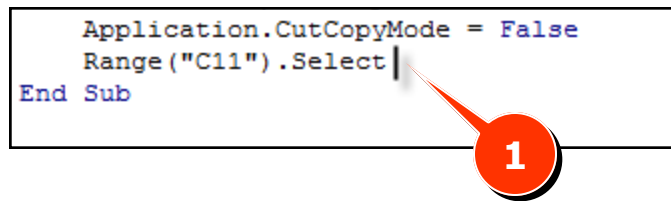
Range("C8").Select
ActiveCell.FormulaR1C1 = "=NOW()"
  
```

Inserting a Message Box

One of the most popular options with a macro is a message box. A message box uses on screen prompts to direct users. Perhaps the simplest message box is one that alerts the user that the macro is about to commence or one that alerts the user that the macro has completed. A message box can be added anywhere in the code. In this example, a Message box is instructing users that they have successfully clocked in.

1. Choose **Macros** from the **Macros** group on the **View** tab (Fig 1, 1).
2. Choose **View Macros** (Fig 1, 2). A dialog box will appear (Fig 2).
3. Select the macro that needs editing and click **Edit** (Fig 2, 1). Visual Basic will launch and display the macro code (Fig 3).
4. Locate the line of code above where you want to place the message box.
5. Press **Enter** to create a new line (Fig 5, 1).

Figure 5



The screenshot shows a Visual Basic code editor window with the following code:
`Application.CutCopyMode = False`
`Range("C11").Select`
`End Sub`
A red circle with the number '1' is positioned over the end of the second line of code, with a red arrow pointing to the cursor position at the end of the line, indicating where a new line should be inserted.

6. Type the following: **Msgbox "Text"** (where "Text" is the message that will display)
7. Close Visual Basic for Applications.
8. Run the macro. The message box will now display on the screen (Fig 6).

Figure 6

