



# Automating Excel® Macros and VBA Fundamentals

Prepared by:  
**Mike Thomas**  
*[theexceltrainer.co.uk](http://theexceltrainer.co.uk)*

**LORMAN**®

Published on [www.lorman.com](http://www.lorman.com) - December 2018

Automating Excel® - Macros and VBA Fundamentals, ©2018 Lorman Education Services. All Rights Reserved.

EXCEL is a registered trademark of Microsoft Corporation and this material is not sponsored by or affiliated with Microsoft Corporation.

## INTRODUCING

Lorman's New Approach to Continuing Education

# ALL-ACCESS PASS

The All-Access Pass grants you **UNLIMITED** access to Lorman's ever-growing library of training resources:

- ☑ Unlimited Live Webinars - 120 live webinars added every month
- ☑ Unlimited OnDemand and MP3 Downloads - Over 1,500 courses available
- ☑ Videos - More than 1300 available
- ☑ Slide Decks - More than 2300 available
- ☑ White Papers
- ☑ Reports
- ☑ Articles
- ☑ ... and much more!

Join the thousands of other pass-holders that have already trusted us for their professional development by choosing the All-Access Pass.



**Get Your All-Access Pass Today!**

# SAVE 20%

Learn more: [www.lorman.com/pass/?s=special20](http://www.lorman.com/pass/?s=special20)

Use Discount Code Q7014393 and Priority Code 18536 to receive the 20% AAP discount.

\*Discount cannot be combined with any other discounts.

# **Automating Excel: Macros and VBA Fundamentals**

**Trainer: Mike Thomas**

## What is VBA?

- It stands for Visual Basic for Applications
- It's a programming language
- It's built into Excel, Word, Excel, PowerPoint and several other MS applications. The training focusses on VBA in Excel

## What is a Macro?

- In Excel, a macro is a set of keystrokes or commands stored under a name, which can be "played back" by issuing a single instruction, thereby saving you time.

## The Macro Recorder and VBA

- Excel has a built-in macro recorder that lets you create a macro by actually recording the actions, however, the macro recorder does have limitations so to be able to do anything more than just record a series of commands or keypresses, you need to learn VBA.
- Another reason for learning VBA is to be able to edit existing macros. Even if the macro was created with the macro recorder, the only way to amend it is to either record it again from scratch and replace the existing one or use VBA to edit it.

## XLSX v XLSM

- XLSX files (i.e. normal Excel workbook files) cannot store VBA code.
- You must save the file as a Macro-Enabled file (with an XLSM file extension)

## How to Display the Developer Tab: Office 2010/2013/2016

- Click File > Options
- Click Customize Ribbon
- Put a tick in the Developer checkbox in the right-hand column

## How to Display the Developer Tab: Office 2007

- Click the Office button and select Options
- In the first set of options tick the box "show developer tab on the Ribbon"

## 4 Ways to Run a Macro

1. Developer Tab > Macros and select the macro to run
2. Press ALT+F8 (shortcut to the above)
3. Assign a keyboard shortcut to the macro
4. Add a button to the spreadsheet and attach the macro to it

## Event Procedures

- An event procedure is code that runs automatically when an event occurs
- Examples of events
  - Workbook opening
  - Workbook being closed
  - Workbook being saved
  - Switching from one worksheet to another
  - Switching from one cell to another

## The VBA Editor

- ALT+F11 to switch between Excel and VBA Editor or Developer > Visual Basic
- Project Explorer and Properties Window are displayed on the left-hand side
- If they are not displayed, use the **View** menu to display them
- Use the Run Button (green triangle) to run the current procedure (current = where the cursor is)

## Procedures

- In VBA, macros are called Procedures
- Procedure is a programming term, macro is a friendlier end-user term
- All procedures start with the word SUB followed by a pair of brackets
- All procedures end with END SUB

## Modules

- A module is a container for procedures
- There is no limit to the number of procedures that can be stored in a single module
- You can create more than one module in a file to aid organising procedures

## Event Procedures

- Can be workbook-specific or worksheet-specific
- Must be stored in the appropriate "sheet" module or in the "ThisWorkbook" module

## Editing a Macro

- Press ALT+F8 (or Developer > Macros)
- Select the name of the macro
- Click Edit

## Step Through a Procedure

- Useful when troubleshooting
- Debug > Step into from the toolbar or F8

## Create a New Procedure

- Place cursor where new procedure is to be stored
- Type SUB followed by a space followed by the name and press Enter

## Modules

- New Module: Select Insert > Module
- Rename a Module: Click the module and rename via the Properties window
- Delete a Module: Right click the module and select Remove

## Indenting

- Use the TAB key to indent
- Use Shift and TAB to remove an indent

## Comments

- Type an apostrophe at the start of the line to add a comment / note

## To Add a Button to the Spreadsheet

- Developer tab > Insert
- Click the first icon in the Forms section
- Draw the button
- Select the macro to attach to the button
- Right click the button to change text / location / macro

## Protecting the Code

- Tools > VBA Project Properties > Protection

## Sample Code: Selecting Cells

**Select a single cell**

```
Range("A1").Select
```

**Select a range of cells**

```
Range("A1:B10").Select
```

**Select a cell one below A1**

```
Range("A1").Offset(1,0).Select
```

**Select a cell one to right of current cell**

```
Activecell.Offset(0,1).Select
```

**Select from current cell to last non-blank cell in column (inclusive)**

```
Range(Selection,Selection.End(xlDown)).Select
```

**Select the next blank cell in current column**

```
ActiveCell.End(xlDown).Select
```

```
ActiveCell.Offset(1, 0).Select
```

## Sample Code: Setting Cell Value

**Set the value of the current cell to "The Excel Trainer"**

```
Activecell.value = "The Excel Trainer"
```

**Set the value of the A1 to "The Excel Trainer"**

```
Range("A1").value = "The Excel Trainer"
```

**Set the value of the A1 to 100**

```
Range("A1").value = 100
```

**Enter a formula in A1**

```
Range("A1").Value = "=A3+A4"
```

**Enter Current time into current cell**

```
ActiveCell.Value = "=Now()"
```

**Enter name of current user into A1 (Windows username)**

```
Range("A1").Value = Environ("USERNAME")
```

## Logic (IF)

**If A1 contains "Mike Thomas" then put "Excel VBA" into B1 and go to C1. If A1 does not contain "Mike Thomas" then put "Excel" in B1 and go to D1**

```
If Range("A1").Value = "Mike Thomas" Then
    Range("B1").Value = "Excel VBA"
    Range("C1").Select
Else
    Range("B1").Value = "Excel"
    Range("D1").Select
End If
```

**If A1 contains "Mike Thomas" then put "Excel VBA" into B1. If A1 does not contain "Mike Thomas" then continue processing the code following the "End If"**

```
If Range("A1").Value = "Mike Thomas" Then
    Range("B1").Value = "Excel VBA"
End If
```

**If A1 contains "Mike Thomas" then put "Excel VBA" into B1. If A1 does not contain "Mike Thomas" then ignore this line of code**

```
If Range("A1").Value = "Mike Thomas" Then Range("B1").Value = "Excel VBA"
```

**Select Case – an alternative to a lengthy IF function**

**If the value (content) of A1 is Mike Thomas, set B1 to MT. If the value (content) of A1 is John Doe, set B1 to JD. If the value (content) of A1 is Robert Doe, set B1 to RD. If the value (content) of A1 is anything else, set B1 to Unknown**

```
Select Case Range("A1").Value
    Case "Mike Thomas"
        Range("B1").Value = "MT"

    Case "John Doe"
        Range("B1").Value = "JD"

    Case "Robert Doe"
        Range("B1").Value = "RD"

    Case Else
        Range("B1").Value = "Unknown"
End Select
```



## Looping: Repeating the Process

**Go to B2. For each cell in column B, set the format to dd-mmmm-yyyy. Stop when it gets to a blank cell**

```
Range("B2").Select
Do Until ActiveCell.Value = ""
    ActiveCell.NumberFormat = "dd-mmmm-yyyy"
    ActiveCell.Offset(1, 0).Select
Loop
```

**Go to B2. For each cell in column B, set the format to dd-mmmm-yyyy. As it moves down the column, check column D. Stop when there is a blank cell in column D on that row**

```
Range("B2").Select
Do Until ActiveCell.Offset(0,2).Value = ""
    ActiveCell.NumberFormat = "dd-mmmm-yyyy"
    ActiveCell.Offset(1, 0).Select
Loop
```

## Variables

**Create 2 variables to store "string" (i.e. text) values. The value of the variables comes from 2 cells**

```
Dim vOrientation As String
Dim vPaperSize As String

vOrientation = Range("M1").Value
vPaperSize = Range("M2").Value
```

**Using a variable as a counter to track the number of times a process is performed**

```
Dim Cntr as byte
Dim Cntr1 as byte

Cntr1 = Range("A1").Value

For Cntr = 1 To Cntr1
    'Code goes here
    'Code goes here
Next Cntr
```

## Functions

### **Function to convert a Celsius value to Fahrenheit**

Function CelFar(vCelsius As Integer)

    CelFar = vCelsius \* (9 / 5) + 32

End Function

**Scenario: A1 = 20. This is a Celsius temperature. In another cell, type =CelFar(A1)**

**The result of the function is the Fahrenheit equivalent of 20 degrees Celsius**

The material appearing in this website is for informational purposes only and is not legal advice. Transmission of this information is not intended to create, and receipt does not constitute, an attorney-client relationship. The information provided herein is intended only as general information which may or may not reflect the most current developments. Although these materials may be prepared by professionals, they should not be used as a substitute for professional services. If legal or other professional advice is required, the services of a professional should be sought.

The opinions or viewpoints expressed herein do not necessarily reflect those of Lorman Education Services. All materials and content were prepared by persons and/or entities other than Lorman Education Services, and said other persons and/or entities are solely responsible for their content.

Any links to other websites are not intended to be referrals or endorsements of these sites. The links provided are maintained by the respective organizations, and they are solely responsible for the content of their own sites.