

Macros and VBA tools can be found on the **Developer** tab, which is hidden by default, so the first step is to enable it. For more information, see **Show the Developer tab**.

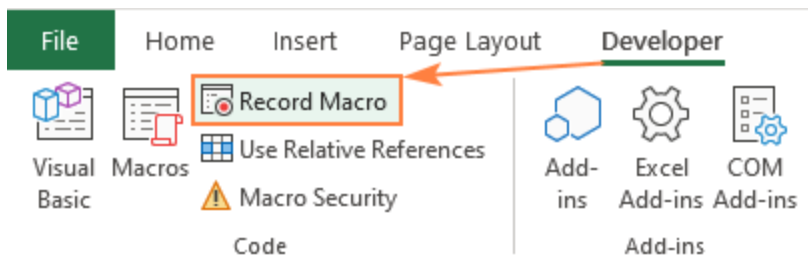
## Show the Developer tab

The Developer tab isn't displayed by default, but you can add it to the ribbon.

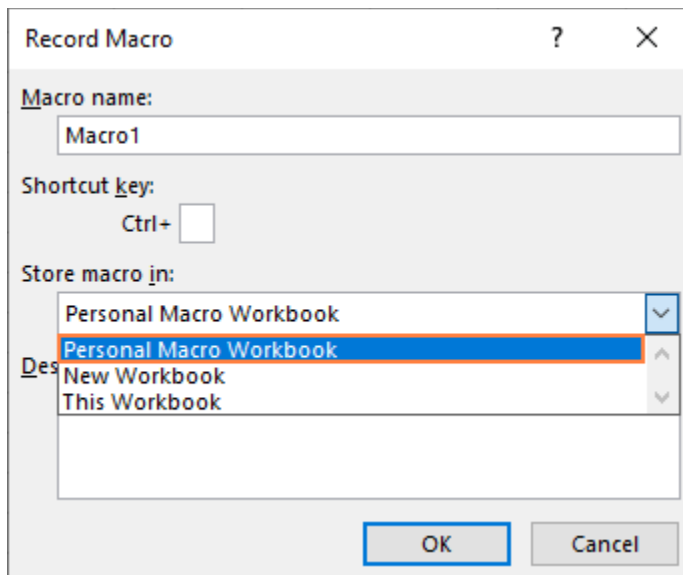
1. On the **File** tab, go to **Options > Customize Ribbon**.
2. Under **Customize the Ribbon** and under **Main Tabs**, select the **Developer** check box.

## Save a macro to your Personal Macro Workbook

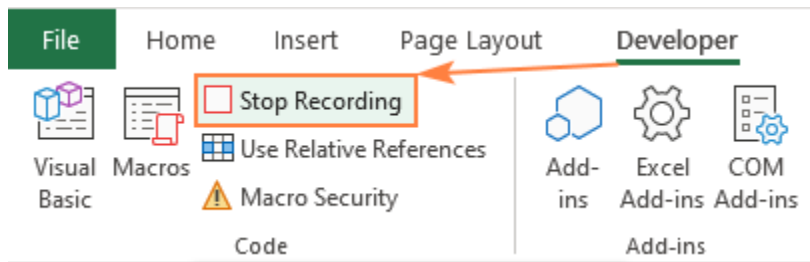
1. In any workbook, go to the **Developer** tab > **Code** group, and click **Record Macro**.



2. The *Record Macro* dialog box will show up. In the *Store Macro* in drop-down list, select **Personal Macro Workbook** and click **OK**. Optionally, you can change the default name such as Macro1 to a more meaningful one.



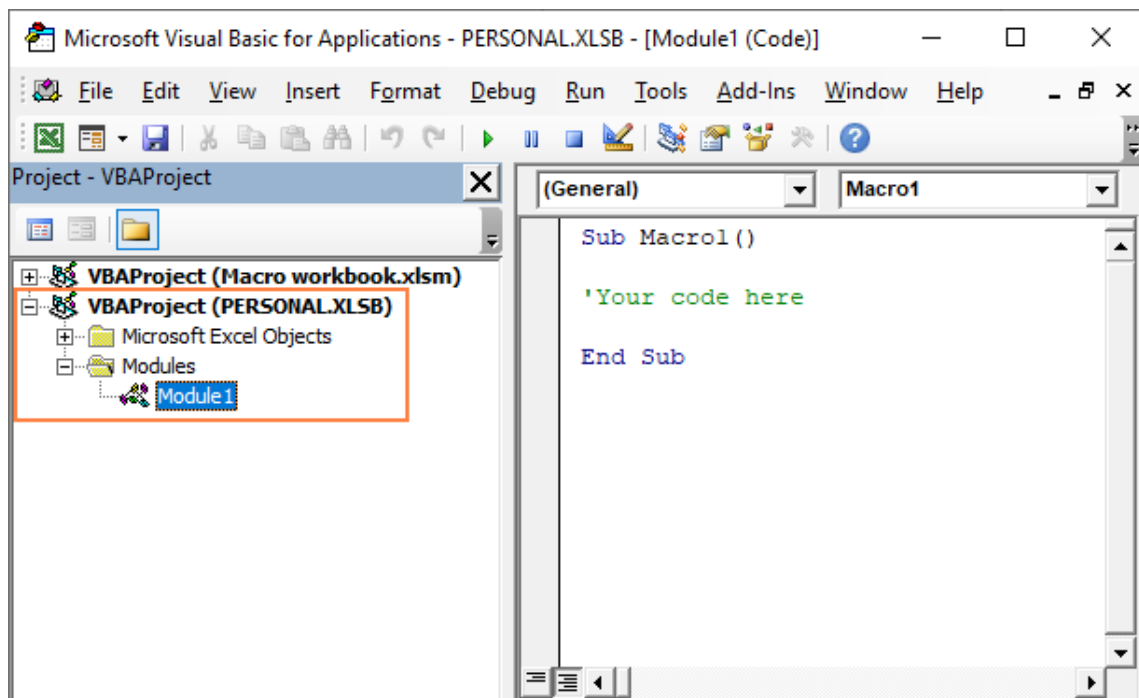
3. On the *Developer* tab or the *Status* bar, click **Stop Recording**.



4. When you close the workbook, you'll be prompted to save both the workbook, and the Personal Macro workbook.

The Personal.xlsm file is created and will be automatically opened in the background every time you start Excel.

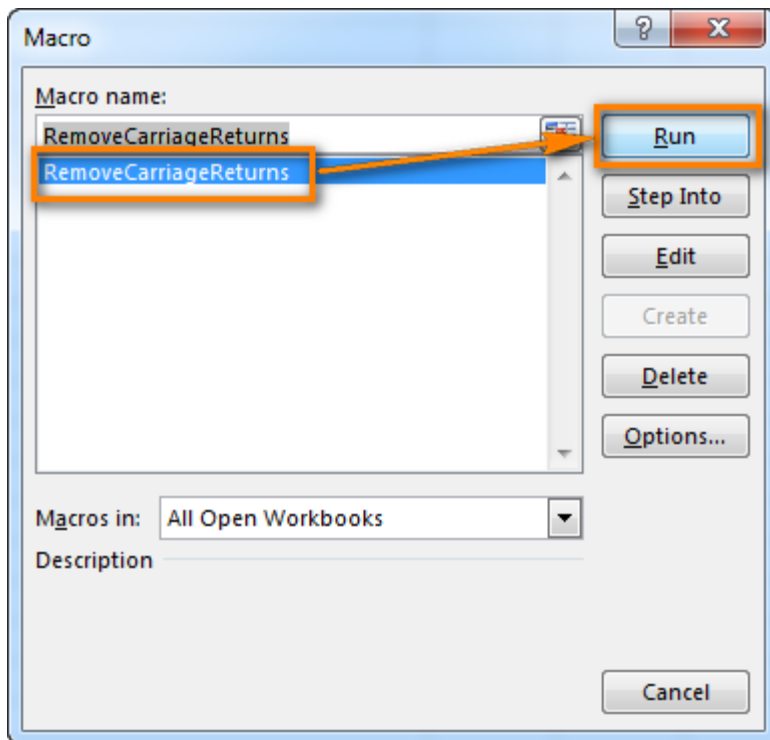
5. Open the VB Editor. For this, open your personal workbook and press **ALT + F11** or click the **Visual Basic** button on the *Developer* tab, in the *Code* group.
6. In the Project Explorer window, find the **PERSONAL.XLSB** object, expand it, and double-click on *Module1* to open its *Code* window.
7. In the Code window, delete the existing code creating a blank code window; copy/paste the VBA code (see the code at the end of the instruction).



8. Close the VB Editor.
9. Close Excel and save the changes you've made to the Personal Macro Workbook when prompted.

### How to run VBA macros in Excel

1. When you want to run the VBA code that you added as described in the section above: open your downloaded Online Director Report and press **Alt+F8** to open the "Macro" dialog.
2. Select the wanted macro from the "Macro Name" list and click the "Run" button.



## These VBA codes will help you to format your spreadsheet.

```
Function FindRowWithColumnName(ws As Worksheet, columnName As String) As Long
    Dim rng As Range
    ' Start searching from row 1 to row 20 (change if your headers might be further down)
    For i = 1 To 20
        Set rng = ws.Rows(i).Find(What:=columnName, _
            LookIn:=xlValues, _
            LookAt:=xlWhole, _
            SearchOrder:=xlByRows, _
            SearchDirection:=xlNext, _
            MatchCase:=False)
        If Not rng Is Nothing Then
            ' If column name is found, return the row number
            FindRowWithColumnName = i
            Exit Function
        End If
    Next i
    ' If column name is not found, return 0
    FindRowWithColumnName = 0
End Function

Sub NYOnlineDirectorFormatting()
    ' Define the worksheet variable for the main data
    Dim wsData As Worksheet

    ' Define the worksheet variable for the filters and metadata
    Dim wsFilters As Worksheet
    Dim filterRows As Long

    ' Set the worksheet to the one where data from Method 2 is expected to be
    Set wsData = Sheets("Online Directory Businesses")

    ' Find the row with the specified column name
    filterRows = FindRowWithColumnName(wsData, "Account Number")

    ' Create a new sheet for filters or clear it if it already exists
    On Error Resume Next ' Ignore errors if the sheet doesn't exist
    Set wsFilters = Sheets("Summary-Parameters")
    If wsFilters Is Nothing Then
        ' If the sheet doesn't exist, create it
        Set wsFilters = Sheets.Add(After:=wsData)
        wsFilters.Name = "Summary-Parameters"
    Else
        ' If it exists, clear it
        wsFilters.Cells.Clear
    End If
End Sub
```

```

End If
On Error GoTo 0 ' Stop ignoring errors

' Move filters to the Summary-Parameters sheet
wsData.Rows("1:" & filterRows - 1).Cut Destination:=wsFilters.Range("A1")

' Remove the filter rows and the empty separator row
wsData.Rows("1:" & filterRows - 1).Delete Shift:=xlUp

' Ensure the header is the first row in wsData
Dim headerRow As Long
headerRow = 1

' Determine the last row of the data
Dim LastRow As Long
LastRow = wsData.Cells(wsData.Rows.Count, "A").End(xlUp).Row

' Define header range
Dim headerRange As Range
Set headerRange = wsData.Range("A" & headerRow & ":AV" & headerRow)

' Format the header range
FormatRange headerRange

' Apply the AutoFilter to the header row only
headerRange.AutoFilter

' Apply blue background and white font color only to the header row
With headerRange
    .Interior.Color = RGB(0, 32, 96) ' Dark blue background color
    .Font.Color = RGB(255, 255, 255) ' White font color
    .Font.Bold = True           ' Bold text
End With

' Format the data range if there is any data below the header

If LastRow > headerRow Then
    Dim dataRange As Range
    Set dataRange = wsData.Range("A" & headerRow + 1 & ":AV" & LastRow)
    FormatRange dataRange
End If

' Adjust column widths based on the header text width
AdjustColumnWidthsAndRowHeights wsData

```

```

' Autofit columns for the "Summary-Parameters" sheet
If Not wsFilters Is Nothing Then
    Call AdjustFilterSheet(wsFilters)
End If

' Focus back to the "Online Directory Businesses" sheet
wsData.Activate

' Inform the user that the operation is complete
' MsgBox "Filters have been moved to 'Summary-Parameters', and headers and data have
been formatted.", vbInformation
End Sub

Sub FormatRange(rng As Range)
' Apply font settings to the range
With rng.Font
    .Name = "Calibri"
    .Size = 12
End With

' Apply general settings to the range
With rng
    .HorizontalAlignment = xlGeneral
    .VerticalAlignment = xlCenter
    .WrapText = True
End With

' Apply border formatting to the range
Dim border As Variant
For Each border In Array(xlEdgeLeft, xlEdgeTop, xlEdgeBottom, xlEdgeRight, xlInsideVertical,
xlInsideHorizontal)
    With rng.Borders(border)
        .LineStyle = xlContinuous
        .Color = RGB(0, 0, 0)
        .Weight = xlThin
    End With
Next border
End Sub

Sub AdjustColumnWidthsAndRowHeights(ws As Worksheet)
    Dim headerRange As Range
    Dim dataRange As Range
    Dim cell As Range
    Dim maxWidth As Double

```

```

Dim firstDataRow As Long
Dim headerRowIndex As Long

headerRowIndex = 1
Set headerRow = ws.Rows(headerRowIndex)
' Assuming the headers are in row 1
Set headerRange = ws.Range("A1").Resize(1, ws.Cells(1,
ws.Columns.Count).End(xlToLeft).Column)

' Loop through each cell in the header range and autofit the column
For Each cell In headerRow.Cells
    cell.EntireColumn.AutoFit
Next cell

' Autofit the header row height
headerRow.AutoFit

' Assuming data starts at row 2; adjust if your data starts at a different row
firstDataRow = 2

' Apply the width and height settings to the data range below the header
Set dataRange = ws.Range(ws.Cells(firstDataRow, 1), ws.Cells(ws.Rows.Count,
headerRange.Columns.Count).End(xlUp))

' Set the row height for the data range
dataRange.RowHeight = 90 ' Adjust the height as needed

' Set the column width for the data range
For Each cell In dataRange.Columns
    cell.ColumnWidth = 25 ' Adjust the width as needed
Next cell
End Sub
Sub AdjustFilterSheet(ws As Worksheet)
' Autofit the first two columns which contain the filters
ws.Columns("A:B").AutoFit

' Apply left alignment and vertical centering to these columns
With ws.Columns("A:B")
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlCenter
End With

' Format the third row in bold
With ws.Rows("3:3")

```

```
.Font.Bold = True  
End With  
End Sub
```