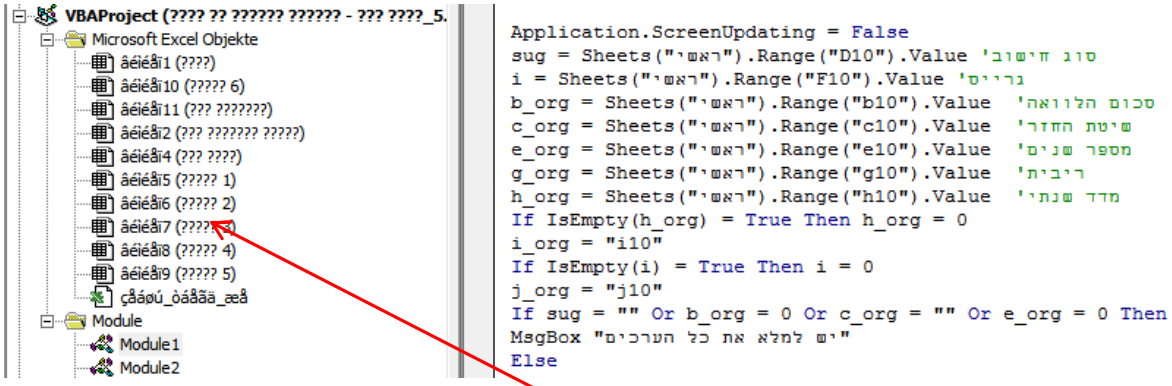
https://social.msdn.microsoft.com/Forums/de-DE/05073941-e83e-4606-bcb8-edecbd61f7f5/gibberish-instead-of-hebrew-in-vba?forum=isvvba#82e7223f-a319-4096-ad2c-c6dbfcaddc9a

Okay, you have a problem.

The text inside the VBA editor is not a problem. I live in Germany and my default font is "Courier New (Middle Europe)". After I change the font to "Courier New (Hebrew)" I see this:



As you see the code and comments itself looks okay, but it doesn't work either.

The basic issue is that the sheet names itself (inside the tree of the project explorer) are not changed.

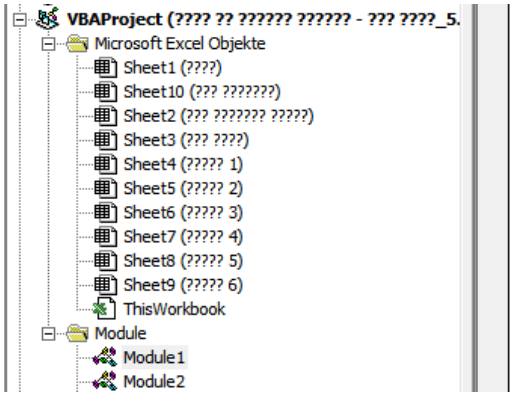
IMHO the only way out of this disaster is to use an international compatible sheet name. But the user should only see the Hebrew name of the sheet as is.

There is a way to accomplish that, because any sheet has a CodeName property and we can use it for our purpose.

Setup your macro option inside the Security Center to allow the code to access the VBA Project object model and then execute this sub:

```
Sub Setup()
  Dim vbComp As Object 'VBIDE.VBComponent
  Dim i As Long
  For Each vbComp In ThisWorkbook.VBProject.VBComponents
    Select Case vbComp.Type
      Case 100 'vbext_ct_Document
        If i = 0 Then
          vbComp.Name = "ThisWorkbook"
        Else
          vbComp.Name = "Sheet" & i
        End If
        i = i + 1
    End Select
  Next
End Sub
```

After that, your file looks like this:




```
Application.ScreenUpdating = False
sug = Sheets("ראשי").Range("D10").Value 'סוג חישוב
i = Sheets("ראשי").Range("F10").Value 'גרייס
b_org = Sheets("ראשי").Range("b10").Value  'סכום הלוואה
c_org = Sheets("ראשי").Range("c10").Value  'שיטת החזר
e_org = Sheets("ראשי").Range("e10").Value  'מספר שנים
g_org = Sheets("ראשי").Range("g10").Value  'ריבית
h_org = Sheets("ראשי").Range("h10").Value  'מדד שנתי
If IsEmpty(h_org) = True Then h_org = 0
i_org = "i10"
If IsEmpty(i) = True Then i = 0
j_org = "j10"
If sug = "" Or b_org = 0 Or c_org = "" Or e_org = 0 Then
MsgBox "יש למלא את כל הערכים"
Else
```

Add a new regular module and paste in this code:

```
Option Explicit


Public Property Get CodeSheets(ByVal CodeName As String) As Worksheet
  'Returns the worksheet which has this codename
  For Each CodeSheets In ThisWorkbook.Worksheets
    If StrComp(CodeName, CodeSheets.CodeName, vbTextCompare) = 0 Then Exit
Property
  Next
End Property
```

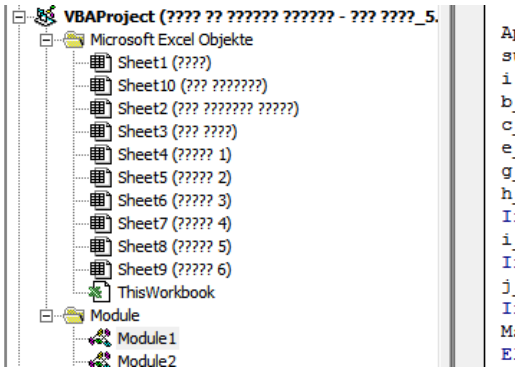
Okay, that was the easy part.

Now you can see the Hebrew sheet names and the new international sheet names inside the project explorer. Do as follows:

a) Replace all Hebrew sheet names in your code with the international sheet names

b) Replace any "Sheets" with "CodeSheets"

And your code looks like this:




```
Application.ScreenUpdating = False
sug = CodeSheets("Sheet1").Range("D10").Value 'סוג חישוב
i = CodeSheets("Sheet1").Range("F10").Value 'גרייס
b_org = CodeSheets("Sheet1").Range("b10").Value  'סכום הלוואה
c_org = CodeSheets("Sheet1").Range("c10").Value  'שיטת החזר
e_org = CodeSheets("Sheet1").Range("e10").Value  'מספר שנים
g_org = CodeSheets("Sheet1").Range("g10").Value  'ריבית
h_org = CodeSheets("Sheet1").Range("h10").Value  'מדד שנתי
If IsEmpty(h_org) = True Then h_org = 0
i_org = "i10"
If IsEmpty(i) = True Then i = 0
j_org = "j10"
If sug = "" Or b_org = 0 Or c_org = "" Or e_org = 0 Then
MsgBox "יש למלא את כל הערכים"
Else
```

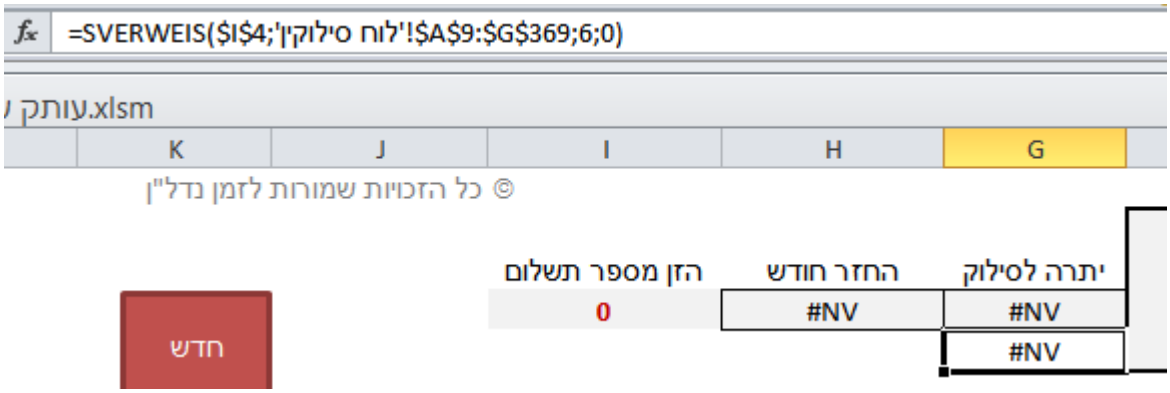
That should work. Okay there are 2 steps left, unfortunately they are much harder:

a) I've seen that you create formulas that also have Hebrew names, means you have to go new ways also here.

For an example I've created the same formula as in G4 in the 1st sheet in G5 with this code:

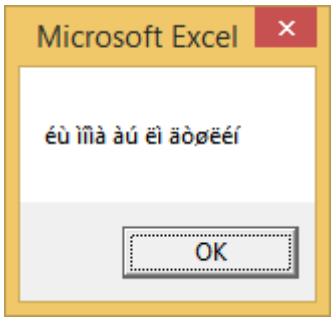
```
Sub Example_Formula()
  Dim MyFormula As String
  Dim R As Range
  'Define a formula, but don't use sheet names!
  'Use a "placeholder" instead.
  MyFormula = "=VLOOKUP($I$4,@Where,6,0)"
  'Refer to the cells
  Set R = CodeSheets("Sheet10").Range("A9:G369")
  'Replace the placeholder with the real cell address
  MyFormula = Replace(MyFormula, "@Where", R.Address(External:=True))
  'Write it into the cell
  ActiveCell.Formula = MyFormula
End Sub
```

Result is:



Looks perfect for me, G4 contains exact the same formula. :-)

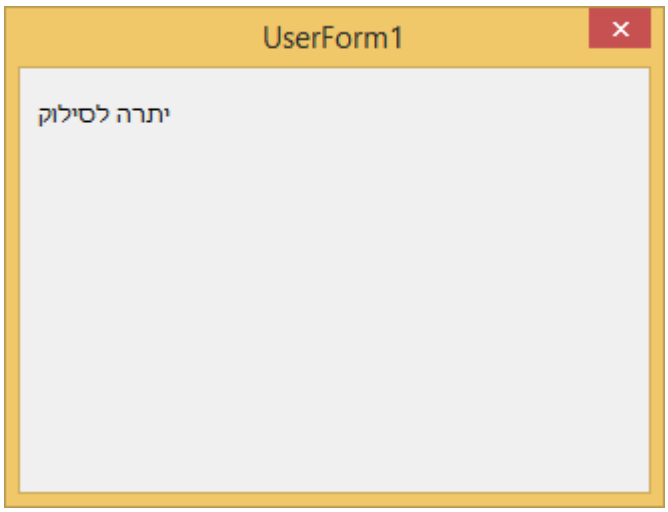
b) The last step is the MsgBox, because a MsgBox doesn't show Hebrew signs. The MsgBox from the code above look as this on my PC:



But a MsgBox is more or less a Userform with a label and button(s), so you can create your own. For testing purposes add a Userform and add a Label to that form, then execute this code from a regular module:

```
Sub Example_MsgBox()
  'Load the form into memory
  Load UserForm1
  With UserForm1
    'Setup the message to show
    .Label1 = CodeSheets("Sheet1").Range("G3").Value
    'Do it
    .Show
  End With
End Sub
```

And you see this:



Okay, that's it. IMHO that is the only way to make your file compatible with international PC's.

Andreas.