Starting a macro with a button

About this document

This document is a “Cookbook” to create a button on a Excel worksheet which can be used to start a macro or a Visual Basic for Applications (VBA) application. This instruction consists of three items:

- Some hints in advance for the use of Excel in relation to VBA and macros.
- The step-by-step instructions to create the button, creating a VBA “UserForm” (form) and attaching the button to the “UserForm”.
- A some “decorations”.

This instruction is based on Excel 2013. However, it can also be used for the versions 2007, 2010 and 2015. Maybe you have to consult the “Help” for the retrieval of the various detail aspects. The instruction is based on reasonable skill in working with Excel.

This document is written in English and the English “windows nomenclature”. The images however were taken from my Dutch version of Excel, the only version of Excel 2013 I own.

Some hints in advance

When working with macros and VBA it is necessary to add or change some setting in Excel.

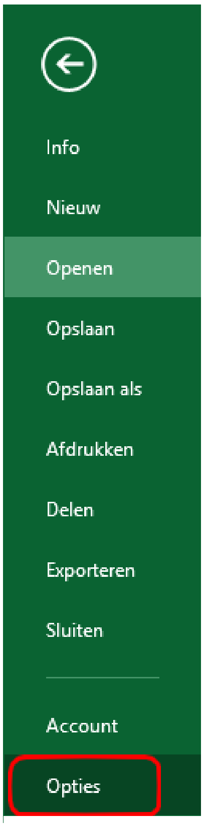


Image 1

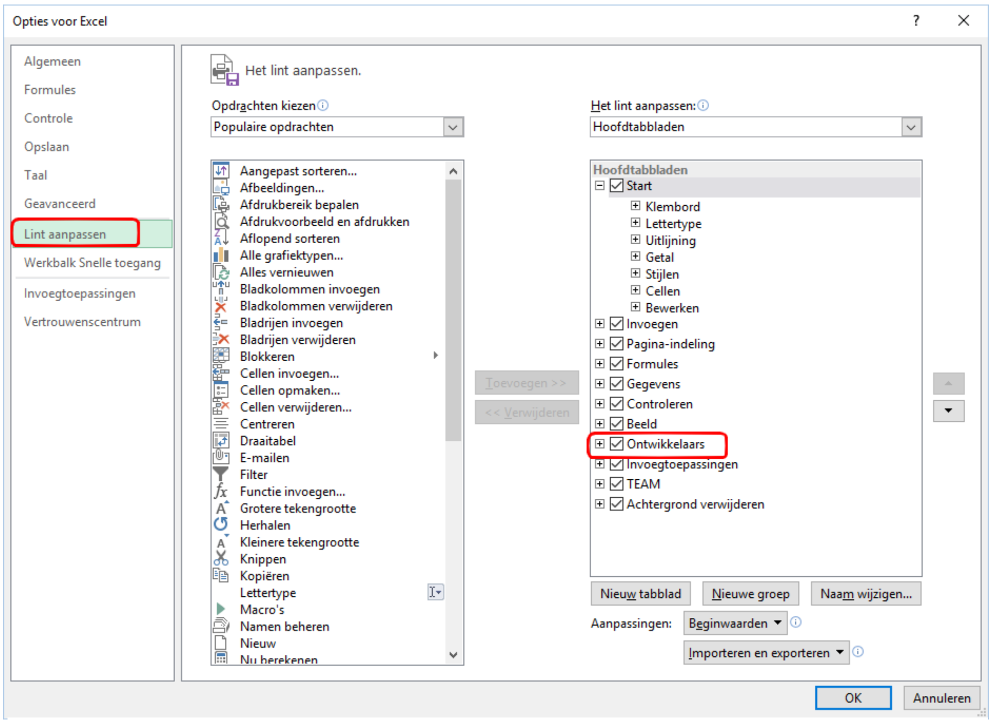


Image 2

First you must activate the ribbon “Developer”. To do so follow the next steps. Open in Excel the “FILE” tab en choose for “Options” (Image 1). Next step is to select in the categories pane for the item “Customize Ribbon”. Select in the list “Main tabs” for “Developer” and check (V) this option. (Image 2).

If you want to configure a “ Form Control” or an “ActiveX Control” you have to activate the “Design Mode”. To do this go to the “Developer Ribbon” and click on “Design Mode”. This mode is active when the background of this option is florescent (Image 3).

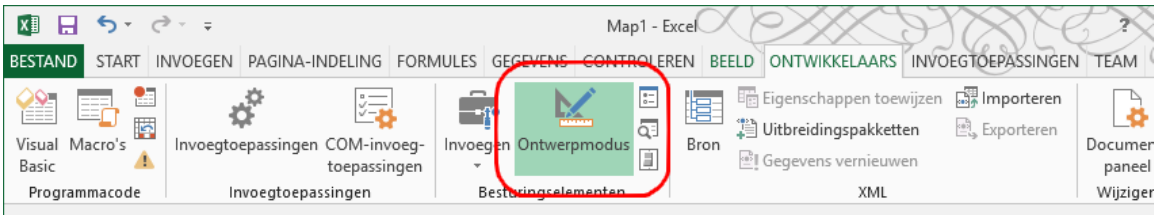


Image 3

The third point of attention are the settings of the "Trust Center" ("file" > "options" > "Trust Center" > "Trust Center settings"). The following settings are recommended:

- ActiveX Settings: “Prompt me before enabling all controls with minimal restrictions”
- Macro Settings: “Disable all macros with notifications”

When opening Workbooks with macros and/or “ActiveX controls” you will be confronted with security warnings. You will be asked to activate macros and/or “ActiveX controls”. Although activating is necessary but be always alert and cautious.

The bare functionality

Step 1: Creating a blank Workbook with one worksheet and saving it.

Creating a workbook: “File” > “New” > “Blank workbook”. If the workbook consists of more the one worksheet delete all but one (If necessary you can add worksheets later). The second stage is saving your workbook as a workbook with code (macros and/or VBA code). Saving your workbook: “File” > “Save as” > select your file location > choose for “Save as type:” “Excel workbook (code) .xlsm”.

Step 2: Creating a button on your Excel worksheet.

Before adding the button check if the “Design mode” is on. Select in the ribbon “Developer” the option “Insert” > “ActiveX control” > “Command Button” (Image 4). Now go to the place where the button should be in the worksheet and tear a rectangle with the mouse (keep left mouse button pressed). The button appears on your worksheet.

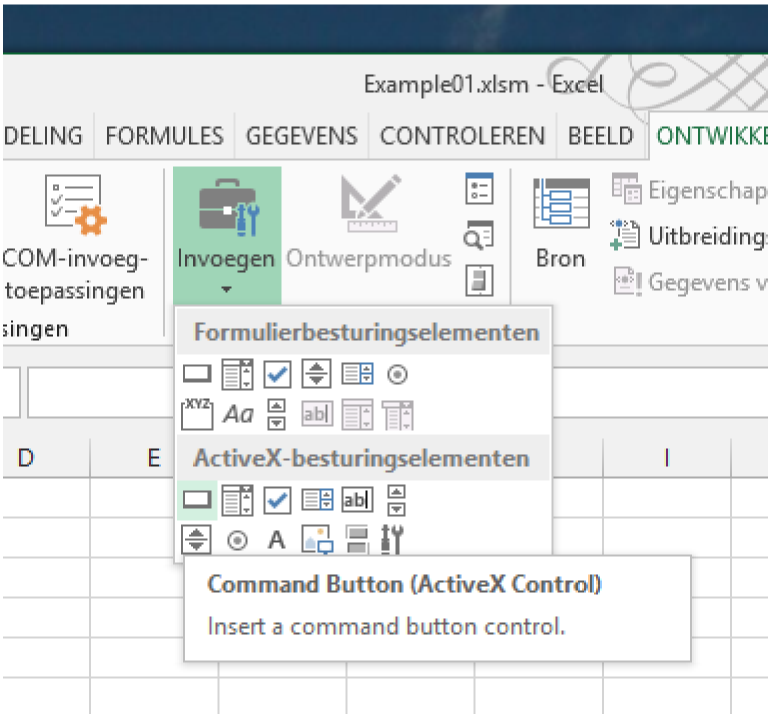
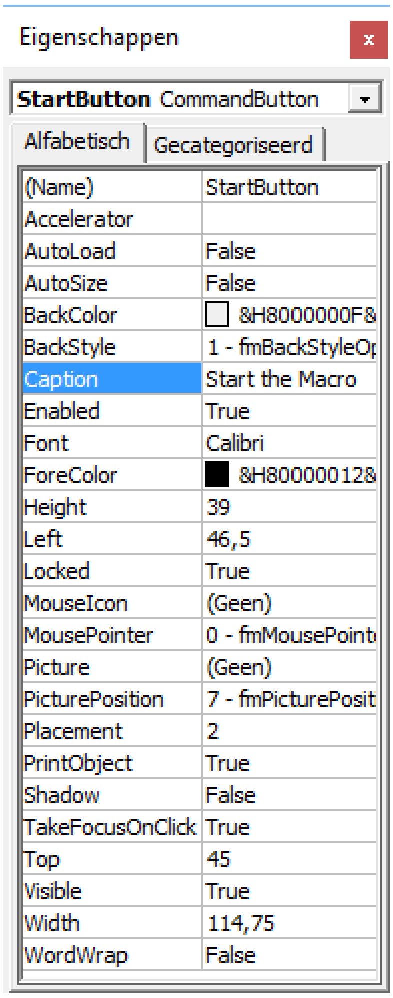


Image 4

Before we save the result me must change some of the buttons properties. Click with the right mouse button on “Commandbutton1”. Select in the menu the option “Properties”. Chance “(name)” in “StartButton” and “Caption” in “Start the Macro” (Image 5). “(Name)” is the name of the object and ”Caption” the text on the button. End this step by saving the result.



Eigenschappen

StartButton CommandButton

Alfabetisch | Gecategoriseerd

(Name)	StartButton
Accelerator	
AutoLoad	False
AutoSize	False
BackColor	□ &H8000000F&
BackStyle	1 - fmBackStyleO
Caption	Start the Macro
Enabled	True
Font	Calibri
ForeColor	■ &H80000012&
Height	39
Left	46,5
Locked	True
MouseIcon	(Geen)
MousePointer	0 - fmMousePoint
Picture	(Geen)
PicturePosition	7 - fmPicturePosit
Placement	2
PrintObject	True
Shadow	False
TakeFocusOnClick	True
Top	45
Visible	True
Width	114,75
WordWrap	False

Image 5

Step 3: creating a “UserForm”

Before you can create a “UserForm” you must open VBA. Select in the ribbon “Developer” the option “Visual Basic”. To add a “UserForm” to our workbook choose “Insert” > “UserForm” (Image 6).

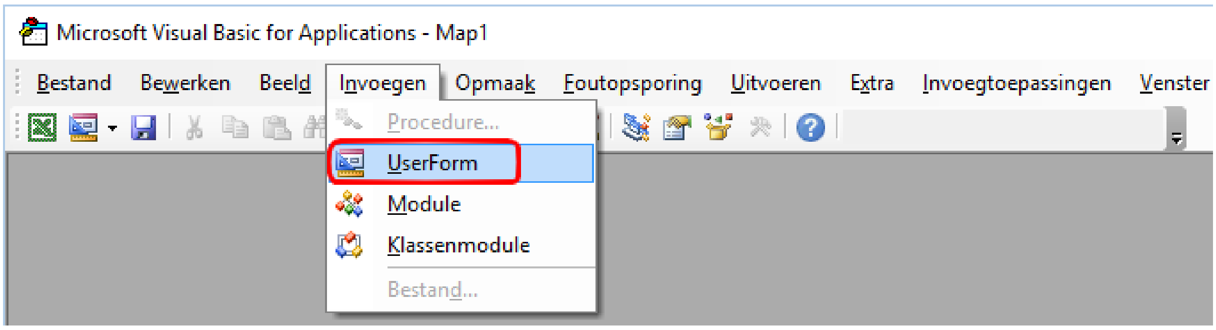
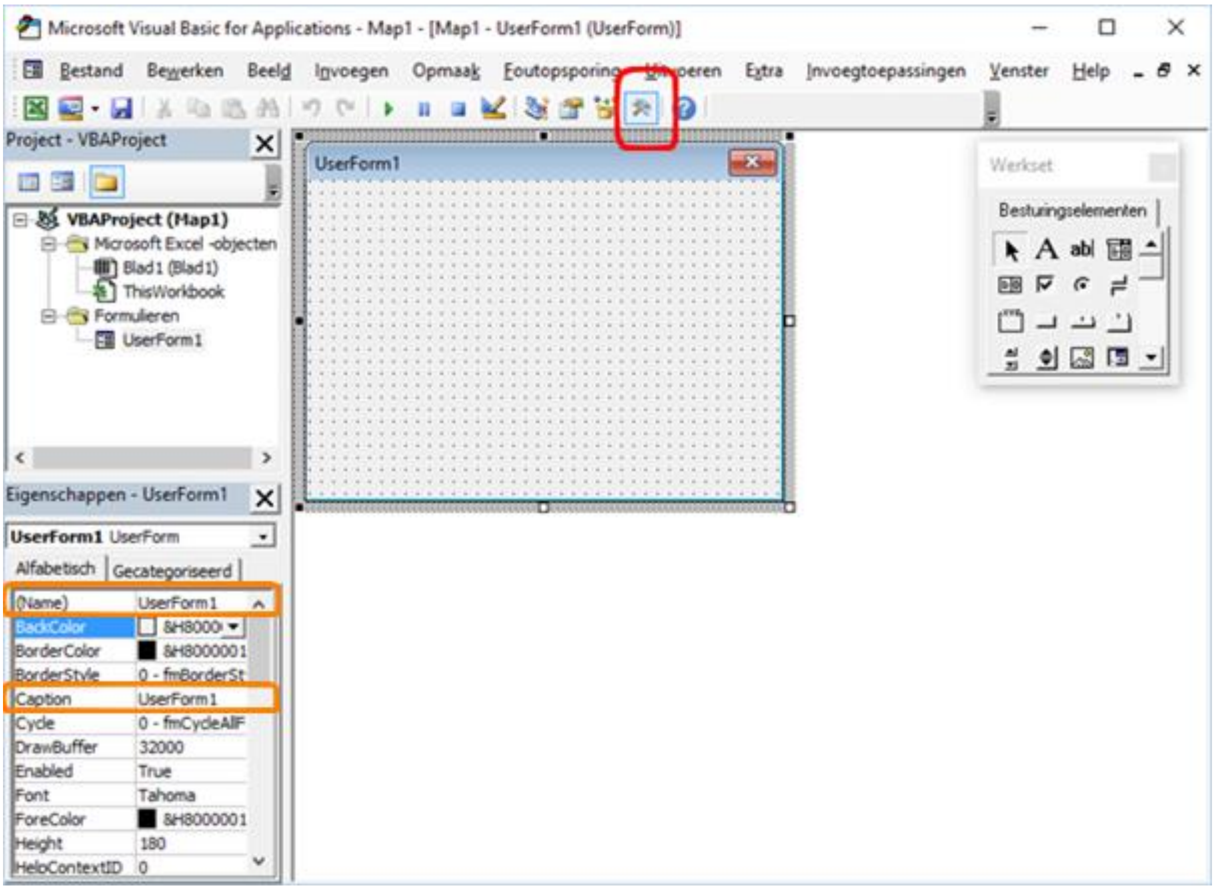


Image 6



(Name)	UserForm1
BackColor	&H8000(
BorderColor	&H8000001
BorderStyle	0 - fmBorderSt
Caption	UserForm1
Cycle	0 - fmCycleAllF
DrawBuffer	32000
Enabled	True
Font	Tahoma
ForeColor	&H8000001
Height	180
HelpContextID	0

Image 7

A “UserForm” appears (Image 7). If the Project Explorer hidden, use the key combination CTRL-R to popup the explorer. For the Properties window use the F4 key and to display the toolbox click on the Toolbox icon in the toolbar (Image 7, red box).

Before you continue, modify the "(Name)" from the "UserForm1" in "StartForm" and the "Caption" in "Example" (title above it in the form) (Image 7, orange box). Finalize this step by saving.

Step 4: Linking the “StartButton” (step 2) to the “StartForm” (step 3).

Switch to the Excel worksheet. Check the “Design mode” in the ribbon “Developer”. It must be on (florescence). Click with the right mouse button on the ”StartButton” and select the option “View code”. A Code window will open with the following code:

```
Private Sub StartButton_Click()

End Sub
```

By adding one line of code (`StartForm.Show`) the link is created

```
Private Sub StartButton_Click()
    StartForm.Show
End Sub
```

In fact, the project is now completed. The last action is of course saving the project.

Step five: test the operation.

To test the project return to the Excel worksheet. Close the “Developer mode” and click on the button “Start the macro”. A window with the title "example" opens. You close the window by clicking the X at the top right of the window.

Decorating our project

Option 1. Including a “Text Box” in our worksheet.

Inse rting a “Text Box” behind the button on the worksheet gives us the opportunity to add some text with some instructions, explanations or comments about the functionality of our button. To do this select in the “Insert” ribbon the option “Illustrations” > “Shapes” > “Basic Shapes” > “Text Box” (Image 8). Now go to the place where the “Text Box” should be in the worksheet and tear a rectangle with the mouse (keep left mouse button pressed). The “Text Box” is created and you can add text in to this box. The best moment to create the “Text Box” is between step 1 and step 2.

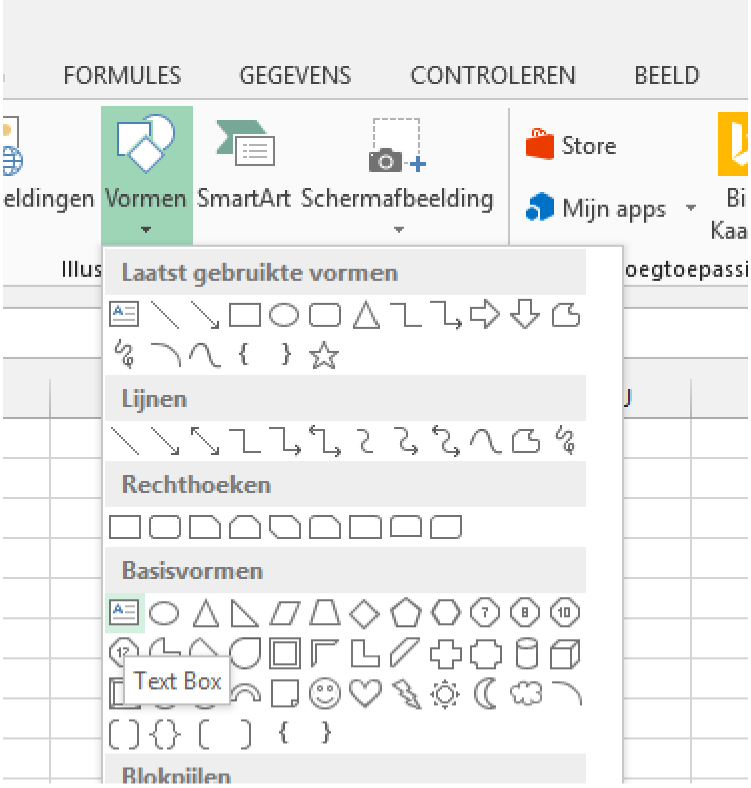
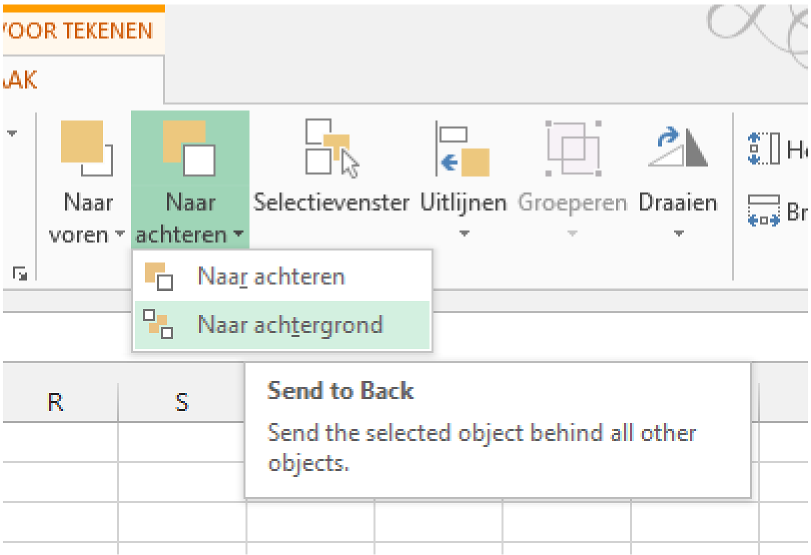


Image 8

If you insert the “Text Box” on a later moment, the “Text Box” will hid the button. You can “repair” it by moving the to the background. To do so: Select in the ribbon “Format” > Option “Arrange” > “Send Backward” > “Send to back”. The button will be shown again (Image 9).



Send to Back

Send the selected object behind all other objects.

Image 9

Option 2: Changing the worksheets background color

By changing the “fill color” and the “Border color” in the same color you can “hide” the cell structure. You create a quiet environment, which doesn’t invite to use them. Of course you can (also) protect the cells.

Option 3: creating an “Close” button on the form

First go to the Visual Basic Editor. Choose under “View” the option “Object”. You will see your “Form”. Open the “Toolbox”: under “View” the option “Toolbox”. Select the “Control” “Command Button” (Image 10). Draw with the mouse (left button down) a square on your form. When ready loosen the left mouse button. The “Command Button” is created. Next is changing some properties of our button. Select in the top of the “Property window” “CommandbButton1”. Change “(Name)” in “CloseButton” and “Caption” in “Close form!”. We finish with adding some code to our button. By double clicking with the left mouse button on our button the “Code” will open with the text:

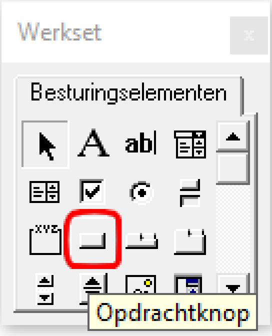


Image 10

```
Private Sub CloseButton_Click()

End Sub
```

This is the subroutine for clicking once with the left mouse button on our button. By adding the code line `Unload Me` the button gets the functionality “Closing the form”. The code will look like:

```
Private Sub CloseButton_Click()
     Unload Me
End Sub
```

Last but not least: safe the project.

Some last words

- You can use VBA not only with Excel but with other MS-Office applications too. For example you can create the same button in a Word-document (save as *.docm).
- You can download two spreadsheets: the first (“example01.xlsm”) was created by using step one to five. The second (“example02.xlsm”) incorporated the decorations.