Lesson 8

Sébastien Mathier

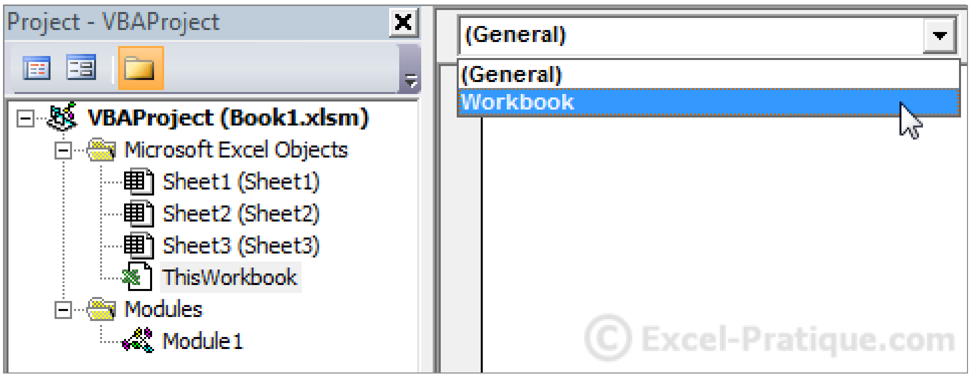
www.excel-pratique.com/en

Workbook events :

We can have workbook events (for example, opening, closing, etc) trigger VBA code.

Workbook_Open :

To execute instructions when the workbook is opened, go to **ThisWorkbook** and select **Workbook** :



The event **Workbook_Open** will be added by default, and takes effect when the workbook is opened :

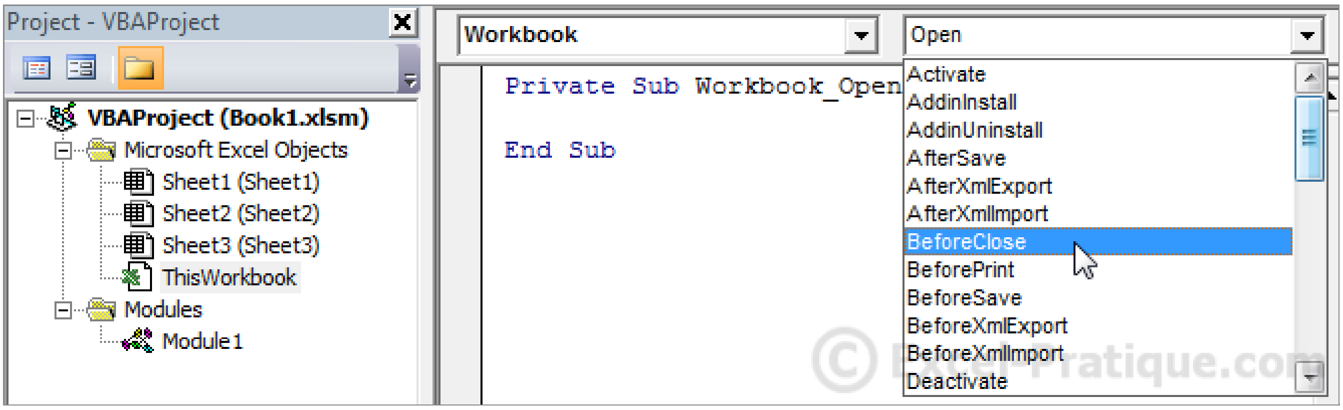
```
Private Sub Workbook_Open()

End Sub
```

For example, if we add the following instruction, a dialog box will be displayed when the workbook is opened :

```
Private Sub Workbook_Open()
    MsgBox "Welcome"
End Sub
```

Workbook_BeforeClose :

To execute instructions immediately before the workbook is closed, choose **BeforeClose** :



```
Private Sub Workbook_Open

End Sub
```

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)

End Sub
```

The closing of the workbook can be cancelled by assigning the value **True** to the variable "Cancel".

Here is an example in which the user is asked to confirm the closing of the workbook :

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    'If the user responds NO, the Cancel variable will have the value TRUE (which will cancel the closing of the workbook)
    If MsgBox("Are you sure that you want to close this workbook ?", 36, "Confirm") = vbNo Then
        Cancel = True
    End If
End Sub
```

Workbook_BeforeSave :

This event will fire immediately before the workbook is saved :

```
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)

End Sub
```

File save can be cancelled by assigning the value **True** to the variable "Cancel".

Workbook_BeforePrint :

This event will fire immediately before printing :

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)

End Sub
```

Printing can be cancelled by assigning the value **True** to the variable "Cancel".

Workbook_AfterSave :

This event will fire immediately after a save command is completed :

```
Private Sub Workbook_AfterSave(ByVal Success As Boolean)

End Sub
```

Workbook_SheetActivate :

This event will fire each time the user switches from one worksheet to another :

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)

End Sub
```

In this example, the name of the worksheet is displayed in a dialog box :

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    MsgBox "Name of Sheet : " & Sh.Name
End Sub
```

Workbook_SheetBeforeDoubleClick :

This event will fire immediately before a double click on a cell :

```vba
Private Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)

End Sub
```

For example, we can use this event to add a background color to a cell depending on which worksheet it is on :

```vba
Private Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
    If Sh.Name = "Sheet1" Then
        Target.Interior.Color = RGB(255, 108, 0) 'Orange color
    Else
        Target.Interior.Color = RGB(136, 255, 0) 'Green color
    End If
End Sub
```

Workbook_SheetBeforeRightClick :

This event will fire immediately before a right click on a cell :

```vba
Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)

End Sub
```

Workbook_SheetChange :

This event will fire each time the contents of a cell are modified :

```vba
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)

End Sub
```

Workbook_SheetCalculate :

This event will fire each time a worksheet's data is calculated or recalculated :

```vba
Private Sub Workbook_SheetCalculate(ByVal Sh As Object)

End Sub
```

Workbook_SheetSelectionChange :

This event will fire each time the selection changes on a calculation worksheet :

```
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)

End Sub
```

In this example, a background color is added if cell A1 is empty :

```
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)
    If Range("A1") = "" Then
        Target.Interior.Color = RGB(124, 255, 255) 'Blue color
   End If
End Sub
```

Workbook_NewSheet :

This event will fire each time a new worksheet is added to the workbook :

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)

End Sub
```

Workbook_SheetFollowHyperlink :

This event will fire whenever a hypertext link is clicked :

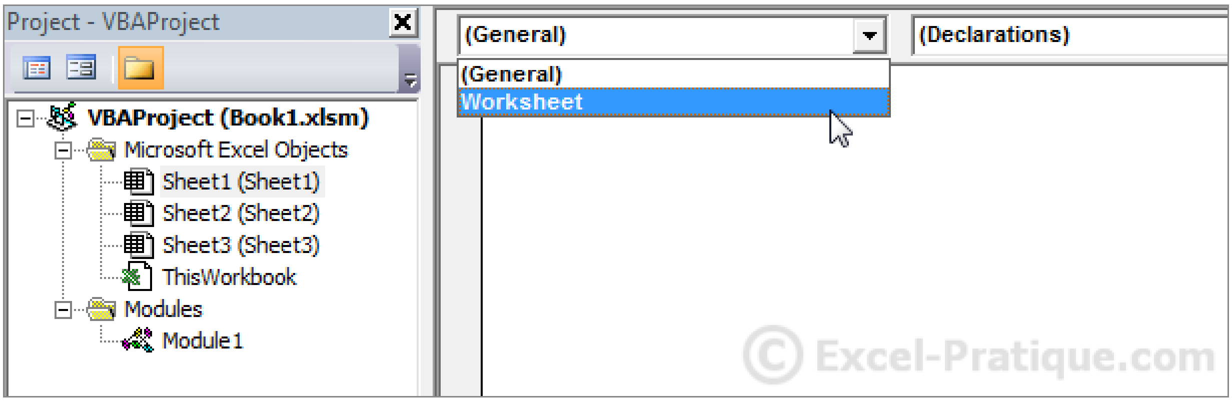
```
Private Sub Workbook_SheetFollowHyperlink(ByVal Sh As Object, ByVal Target As Hyperlink)

End Sub
```

Worksheet events :

On the last page, we looked at events related to the entire workbook. On this page, we'll move on to focus on events linked to a single worksheet.

Worksheet_SelectionChange :

To execute instructions based on events for a particular worksheet, select the sheet in the editor, and then **Worksheet** :



The **SelectionChange** event will be added by default. This event is executed whenever the selection is changed :

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)

End Sub
```

For example, the code below adds background colors to one or more selected cells and automatically removes the background color from the previous selection when the selection is changed :

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Static previous_selection As String

    If previous_selection <> "" Then
        'Removing background color from previous selection :
        Range(previous_selection).Interior.ColorIndex = xlColorIndexNone
    End If

    'Adding background color to current selection :
    Target.Interior.Color = RGB(181, 244, 0)

    'Saving the address of the current selection :
    previous_selection = Target.Address
End Sub
```

Worksheet_Activate :

This event will fire when the worksheet is activated :

```
Private Sub Worksheet_Activate()

End Sub
```

Worksheet_Deactivate :

This event will fire when another worksheet is activated :

```
Private Sub Worksheet_Deactivate()

End Sub
```

Worksheet_BeforeDoubleClick :

This event will fire when a cell in the worksheet is double clicked :

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)

End Sub
```

Worksheet_BeforeRightClick :

This event will fire when the user right clicks on the worksheet :

```
Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean)

End Sub
```

Worksheet_Calculate :

This event will fire when the data on the worksheet are calculated or recalculated :

```
Private Sub Worksheet_Calculate()

End Sub
```

Worksheet_Change :

This event will fire when the contents of cells in the worksheet are changed :

```
Private Sub Worksheet_Change(ByVal Target As Range)

End Sub
```

Worksheet_FollowHyperlink :

This event will fire when the user clicks on a hypertext link :

```
Private Sub Worksheet_FollowHyperlink(ByVal Target As Hyperlink)

End Sub
```

Temporarily deactivate all events :

To execute code without firing any events, put it between the following two lines of code :

```
Application.EnableEvents = False ' => deactivate events
'Instructions
Application.EnableEvents = True ' => reactivate events
```