

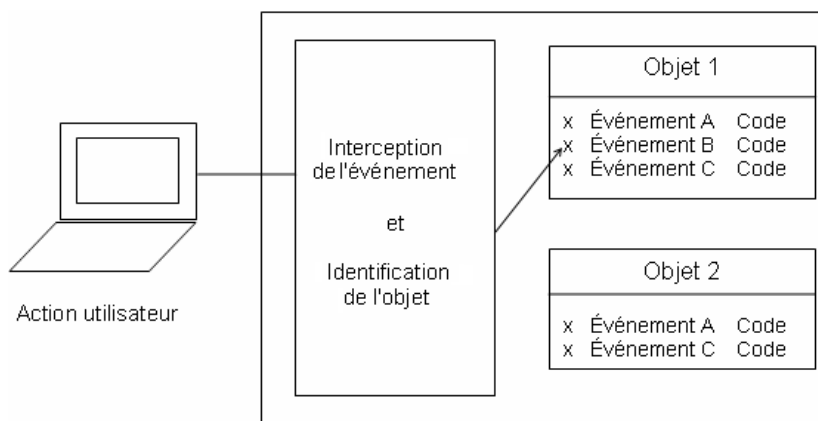
## Chapitre 9

# Gestion des événements

### 1. Présentation

Un événement est une **action** utilisateur ou système reconnue par un objet Microsoft Excel. Il déclenche la procédure événementielle associée à l'événement de l'objet activé.

Les procédures événementielles vous permettent d'associer un code personnalisé en réponse à un événement qui se produit sur un objet Excel (classeur, feuille, formulaire, graphique...).



## 2. Écriture des événements

### 2.1 Événements de classeur, de feuille ou de formulaire

Vous pouvez accéder aux procédures événementielles associées à un objet de la façon suivante :

- ▣ Dans la fenêtre **Explorateur de projet**, double cliquez sur l'objet souhaité (classeur, feuille, ou formulaire) afin de faire apparaître la fenêtre de code correspondante.
- ▣ Ouvrez la liste déroulante de gauche de la fenêtre de code et sélectionnez **Workbook**, **Worksheet** ou **UserForm** en fonction de l'objet sélectionné.
- ▣ Vous pouvez alors sélectionner l'un des événements liés à l'objet sélectionné dans la liste déroulante de droite afin de lui associer un code personnalisé.

#### ■ Remarque

*Vous pouvez à tout moment désactiver l'exécution des procédures événementielles en affectant **False** à la propriété **EnableEvents** de l'objet **Application**.*



## 2.2 Événements de l'objet Application

Trois étapes sont nécessaires à l'écriture et à l'exécution des événements de l'objet **Application**.

### Étape 1

► Insérez un module de classe :

**Insertion - Module de classe**

ou ouvrez la liste  et cliquez sur **Module de classe**.

► Une fois le module inséré, nommez-le.

### Exemple

Nommez le module de classe ObjApplication.

### Étape 2

► Dans le module de classe, créez un objet **Application** par le code suivant :

```
Public WithEvents NomObjet As Application
```

### Exemple

Création de l'objet appelé **oMonAppli** en tant qu'application.

```
Public WithEvents oMonAppli As Application
```

L'objet ainsi créé devient disponible dans la liste de gauche du module.

► Sélectionnez l'objet ainsi créé dans la liste de gauche du module puis sélectionnez l'événement attendu dans la liste de droite. Écrivez le code des procédures à générer.

Exemple

Création de deux procédures événementielles : la première concerne l'insertion d'une nouvelle feuille, la seconde la création d'un nouveau classeur.

```
Public WithEvents oMonAppli As Excel.Application

Private Sub oMonAppli_WorkbookNewSheet _
    (ByVal Wb As Workbook, ByVal Sh As Object)
    Dim oNomFeuille As String
    ' A chaque ajout de feuille, on demande à l'utilisateur
    ' de saisir un nom qui sera ensuite affecté à la feuille insérée
    ' après les feuilles existantes
    oNomFeuille = InputBox("Entrez le nom de la feuille")
    ActiveSheet.Name = oNomFeuille
    ActiveSheet.Move After:=Sheets(Sheets.Count)
End Sub

Private Sub oMonAppli_NewWorkbook(ByVal Wb As Workbook)
    Dim iNbFeuilles As Integer
    Dim iNbActuel As Integer
    Dim iDifférence As Integer
    ' Pour chaque nouveau classeur,
    ' on demande à l'utilisateur le nombre de feuilles
    ' Suivant les cas, on ajoute ou on supprime des feuilles
    Do
        iNbFeuilles = Application.InputBox(Title:="", _
            Prompt:="Nombre de feuilles ?", Default:="3", Type:=1)
    Loop While iNbFeuilles = False
    iNbActuel = Sheets.Count
    iDifférence = iNbActuel - iNbFeuilles
    ' Suppression des feuilles en trop
    ' Suppression des messages d'alerte afin de ne pas
    ' avoir de message lors de la suppression de feuilles
    Do While iDifférence > 0
        Application.DisplayAlerts = False
        Sheets.Item(iDifférence).Select
        ActiveWindow.SelectedSheets.Delete
        iDifférence = iDifférence - 1
    Loop

    ' Ajout de feuilles si nécessaire
    ' Les événements sont désactivés afin de ne pas avoir
    ' à saisir le nom des nouvelles feuilles
    Do While iDifférence < 0
        Application.EnableEvents = False
        Sheets.Add After:=Sheets(Sheets.Count)
```

# 312 \_\_\_\_\_ VBA Excel (2019 et Office 365)

Programmer sous Excel : macros et langage VBA

```
iDifférence = iDifférence + 1
Loop
' Réactivation des événements et alertes
Application.EnableEvents = True
Application.DisplayAlerts = True
End Sub
```

## Étape 3

▣ Activez un module quelconque et connectez l'objet déclaré dans le module de classe avec l'objet **Application** par les instructions suivantes :

```
Dim oNomVariable As New NomModuleDeClasse
```

```
Sub NomProcédure ()
Set oNomVariable.NomObjet = Application
End Sub
```

## Exemple

Ajoutez le code suivant dans le module Déclarations.

```
Option Explicit
Dim oApp As New ObjApplication

Sub InitializeMonAppli()
Set oApp.oMonAppli = Application
End Sub
```

Enfin appelez la procédure InitializeMonAppli lors de l'ouverture du classeur (module de classe ThisWorkbook).

```
Private Sub Workbook_Open()
InitializeMonAppli
End Sub
```

Lorsque ce classeur sera ouvert, les procédures événementielles créées au cours de l'étape 2 s'exécuteront automatiquement lors de l'ajout de classeurs ou de feuilles. Ces procédures seront désactivées à la fermeture du classeur.