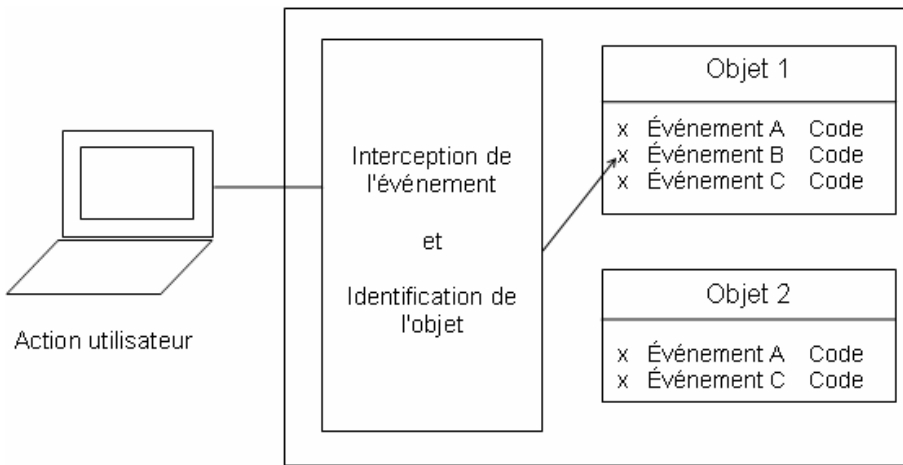


A. Présentation

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...).



B. Écriture des événements

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.

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

Exemple

Cet exemple montre comment obtenir l'historique de tous les liens hypertextes qui ont été visités à partir de la feuille de calcul active.

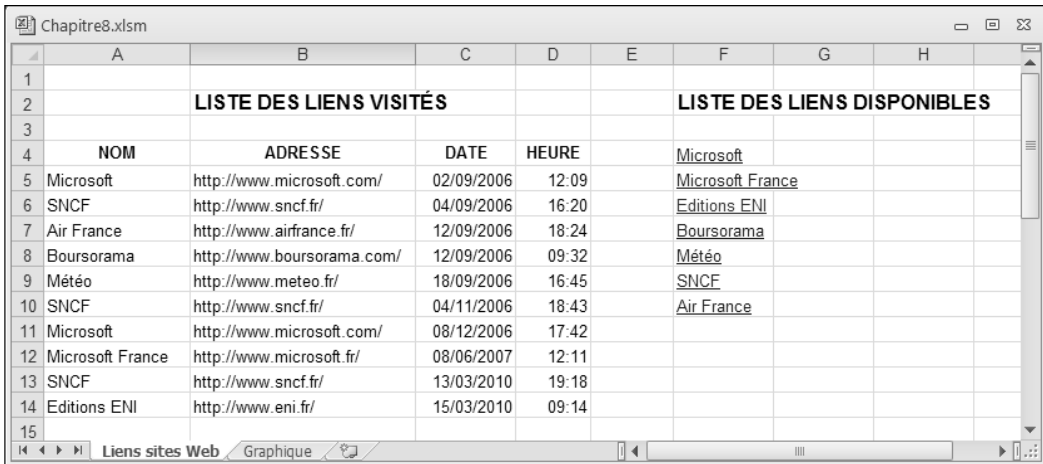
The screenshot shows the Microsoft Visual Basic for Applications editor window titled "Microsoft Visual Basic pour Applications - Chapitre8.xlsm - [Feuil1 (Code)]". The interface includes a menu bar (Fichier, Edition, Affichage, Insertion, Format, Débogage, Exécution, Outils, Compléments, Fenêtre), a toolbar, and a Project Explorer on the left. The Project Explorer shows a project named "EuroTool (EUROTOOL.XLAM)" containing a "VBAProject (Chapitre8.xlsm)" with several worksheets: "Feuil1 (Liens sites Web)", "Feuil2 (Graphique)", and "ThisWorkbook". The active worksheet is "FollowHyperlink". The main editor area displays the following VBA code:

```
Option Explicit

Private Sub Worksheet_FollowHyperlink _
    (ByVal Target As Hyperlink)
    Dim rng As Range
    Dim numLig As Integer

    ' Ajout d'une ligne à la fin de la liste
    ' des sites visités
    Set rng = Range("A4").CurrentRegion
    numLig = rng.Rows.Count + 4
    Cells(numLig, 1) = Target.Name
    Cells(numLig, 2) = Target.Address
    Cells(numLig, 3) = Date
    Cells(numLig, 4) = Format(Time, "hh:mm")
End Sub
```

Résultat dans Excel :



	A	B	C	D	E	F	G	H
1								
2		LISTE DES LIENS VISITÉS				LISTE DES LIENS DISPONIBLES		
3								
4		NOM	ADRESSE	DATE	HEURE	Microsoft		
5	Microsoft	http://www.microsoft.com/	02/09/2006	12:09		Microsoft France		
6	SNCF	http://www.sncf.fr/	04/09/2006	16:20		Editions ENI		
7	Air France	http://www.airfrance.fr/	12/09/2006	18:24		Boursorama		
8	Boursorama	http://www.boursorama.com/	12/09/2006	09:32		Météo		
9	Météo	http://www.meteo.fr/	18/09/2006	16:45		SNCF		
10	SNCF	http://www.sncf.fr/	04/11/2006	18:43		Air France		
11	Microsoft	http://www.microsoft.com/	08/12/2006	17:42				
12	Microsoft France	http://www.microsoft.fr/	08/06/2007	12:11				
13	SNCF	http://www.sncf.fr/	13/03/2010	19:18				
14	Editions ENI	http://www.eni.fr/	15/03/2010	09:14				
15								

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 **MonAppli** en tant qu'application.

```
Public WithEvents MonAppli 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 MonAppli As Excel.Application

Private Sub MonAppli_WorkbookNewSheet _
    (ByVal Wb As Workbook, ByVal Sh As Object)
    Dim NomFeuille 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
    NomFeuille = InputBox("Entrez le nom de la feuille")
    ActiveSheet.Name = NomFeuille
    ActiveSheet.Move After:=Sheets(Sheets.Count)
End Sub

Private Sub MonAppli_NewWorkbook(ByVal Wb As Workbook)
    Dim NbFeuilles As Integer
    Dim NbActuel As Integer
    Dim Différence 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
        NbFeuilles = Application.InputBox _
            ("Nombre de feuilles ?", Type:=1)
    Loop While NbFeuilles = False
    NbActuel = Sheets.Count
    Différence = NbActuel - NbFeuilles
    ' 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 Différence > 0
    Application.DisplayAlerts = False
    Sheets.Item(Différence).Select
    ActiveWindow.SelectedSheets.Delete
    Différence = Différence - 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 Différence < 0
    Application.EnableEvents = False
    Sheets.Add
    Différence = Différence + 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 NomVariable As New NomModuleDeClasse

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

Exemple

Ajoutez le code suivant dans le module Déclarations.

```
Option Explicit
Dim app As New ObjApplication

Sub InitializeMonAppli()
    Set app.MonAppli = 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.

3. Événement d'un graphique incorporé

La collection **Charts** (de l'objet **Workbook**) contient tous les graphiques du classeur spécifié.

Trois étapes sont nécessaires à l'écriture et à l'exécution des événements d'un graphique incorporé.

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 **ObjGraphiques**.

Étape 2

→ Dans le module de classe, créez un objet de type graphique par le code suivant :

```
Public WithEvents NomObjet As Chart
```