

Les événements avec Microsoft Access

Les événements avec Microsoft Access : Les Formulaires

par Fabrice CONSTANS ([ACCESSite](#))

Date de publication : 28/09/2009

Dernière mise à jour : 1/10/2009

Véritable centre nerveux de l'IHM (Interface Homme Machine), les événements créés l'interactivité d'une application en réagissant aux actions de l'utilisateur. Plus un logiciel est riche plus ces possibilités événementielles sont nombreuses. C'est pour cette raison que les dernières versions de Microsoft ACCESS ont vu une augmentation sensible du nombre d'événements. Dans ce tutoriel nous allons passer en revue les principaux événements pour se familiariser à leur utilisation.

I - Glossaire.....	4
I-A - Focus.....	4
I-B - Touche morte.....	4
I-C - Contrôle éligible.....	4
II - Avertissement.....	5
III - Qu'est-ce qu'un événement.....	6
IV - Règles.....	7
V - Groupes d'événements.....	8
VI - Comment utiliser un événement.....	9
VII - Les événements formulaires.....	10
VII-A - A l'ouverture.....	10
VII-A-1 - Sur ouverture - Form_Open.....	10
VII-A-2 - Sur chargement - Form_Load.....	10
VII-A-3 - Sur redimensionnement - Form_Resize.....	10
VII-A-4 - Sur activé - Form_Activate.....	11
VII-A-5 - Sur activation - Form_Current.....	11
VII-A-6 - Enchaînement des événements lors de l'ouverture d'un formulaire.....	11
VII-A-7 - Enchaînement lors de l'activation d'un formulaire déjà ouvert.....	11
VII-B - A la fermeture.....	11
VII-B-1 - Sur libération - Form_Unload.....	11
VII-B-2 - Sur désactivé - Form_Deactivate.....	12
VII-B-3 - Sur fermeture- Form_Close.....	12
VII-B-4 - Enchaînement des événements de fermeture.....	12
VII-C - Ajout, Suppression, modification d'enregistrement.....	12
VII-C-1 - Avant insertion - Form_BeforeInsert.....	12
VII-C-2 - Après insertion - Form_AfterInsert.....	13
VII-C-3 - Avant MAJ - Form_BeforeUpdate.....	13
VII-C-4 - Après MAJ - Form_AfterUpdate.....	13
VII-C-5 - Avant suppression - Form_BeforeDelConfirm.....	13
VII-C-6 - Sur suppression - Form_Delete.....	13
VII-C-7 - Après Suppression - Form_AfterDelConfirm.....	13
VII-D - Détection d'erreur, de modification et d'annulation.....	14
VII-D-1 - Sur annulation - Form_Undo.....	14
VII-D-2 - Si modification - Form_Dirty.....	14
VII-E - Sur erreur - Form_Error.....	14
VII-F - Perte et réception de focus.....	15
VII-F-1 - Sur réception focus - Form_GotFocus.....	15
VII-F-2 - Sur perte focus - Form_LostFocus.....	15
VII-G - Périphériques d'entrées/sorties.....	15
VII-G-1 - Sur clic - Form_Click.....	15
VII-G-2 - Sur double-clic - Form_DbClick.....	15
VII-G-3 - Enchaînement des événements clic de souris.....	15
VII-G-4 - Sur souris appuyée - Form_MouseDown.....	15
VII-G-5 - Sur souris relâchée - Form_MouseUp.....	16
VII-G-6 - Sur souris déplacée - Form_MouseMove.....	16
VII-G-7 - Sur touche appuyée - Form_KeyDown.....	16
VII-G-8 - Sur touche relâchée - Form_KeyUp.....	16
VII-G-9 - Sur touche activée - Form_KeyPress.....	17
VII-G-10 - Enchaînement des événements Key.....	17
VII-G-11 - Sur roulement de la souris - Form_MouseWheel.....	17
VII-H - Filtrer des données.....	17
VII-H-1 - Sur filtre - Form_Filter.....	17
VII-H-2 - Sur filtre appliqué - Form_Apply_filter.....	17
VII-I - Événement autonome.....	18
VII-I-1 - Sur minuterie (intervalle minuterie) - Form_Timer.....	18
VII-J - Les événements pour formulaire d'analyse croisée dynamique.....	18
VIII - Les événements de sections.....	19
VIII-A - Périphériques d'entrées/sorties.....	19

VIII-A-1 - Sur clic - NomdelaSection_Click.....	19
VIII-A-2 - Sur double clic - NomdelaSection_DblClick.....	19
VIII-A-3 - Sur souris appuyée - NomdelaSection_MouseDown.....	19
VIII-A-4 - Sur souris relâchée - NomdelaSection_MouseMove.....	19
VIII-A-5 - Sur souris déplacée - NomdelaSection_MouseUp.....	19
VIII-B - Affichage.....	19
VIII-B-1 - Sur le dessin - NomdelaSection_Paint.....	19
IX - Événement Contrôles.....	21
IX-A - Etiquette.....	21
IX-A-1 - Sur clic - label_Click.....	21
IX-A-2 - Sur double clic - label_DblClick.....	21
IX-B - Ligne.....	21
IX-C - Cadre.....	21
IX-D - Zone de liste déroulante.....	22
IX-D-1 - Sur clic - cbo_Click.....	22
IX-D-2 - Avant MAJ- cbo_BeforeUpdate.....	22
IX-D-3 - Après MAJ - cbo_AfterUpdate.....	22
IX-D-4 - Si modification- cbo_Dirty.....	22
IX-D-5 - Sur changement - cbo_Change.....	22
IX-D-6 - Sur absence dans liste - cbo_NotInList.....	23
IX-D-7 - Sur entrée - cbo_GotFocus.....	23
IX-D-8 - Sur entrée - cbo_Enter.....	23
IX-D-9 - Sur sortie - cbo_Exit.....	23
IX-E - Onglet.....	23
IX-E-1 - Sur changement - tab_Change().....	24
IX-F - Enchaînement des événements d'un contrôle vers un autre.....	24
X - Déclencher manuellement un événement.....	25
XI - Astuce pour connaître le nom VBA d'un événement.....	26
XII - Conclusion.....	27
XIII - Liens.....	28

I - Glossaire

I-A - Focus

Le focus est l'activation d'un objet (formulaire, contrôle, état) de l'IHM ACCESS. Cette activation peut être faite au moyen de la souris, du clavier ou de tout autre périphérique de conversation avec la machine. Il peut également être activé par programme via la méthode *setfocus*.

I-B - Touche morte

Les touches mortes sont les touches du clavier ne générant aucun effet seule. C'est le cas de Shift (Maj. gauche ou droit), Contrôle (Ctrl droit et gauche), Alt. Il y en d'autre sur le clavier mais seules celles-ci seront abordées dans ce cours.

I-C - Contrôle éligible

Un contrôle est éligible lorsqu'il a la capacité d'accepter la modification pour laquelle il a été cité. Par exemple la section Entête de formulaire n'est pas éligible pour l'événement *Après MAJ*. Il ne possède pas cet événement.

II - Avertissement

Le code contenu dans ce cours n'est là que pour l'exemple, il ne peut être utilisé tel quel dans une application du fait son caractère incomplet. Ce cours commente l'application **Chasseur d'événements** disponible gratuitement à l'adresse suivante :

<http://loufab.developpez.com/>

III - Qu'est-ce qu'un événement

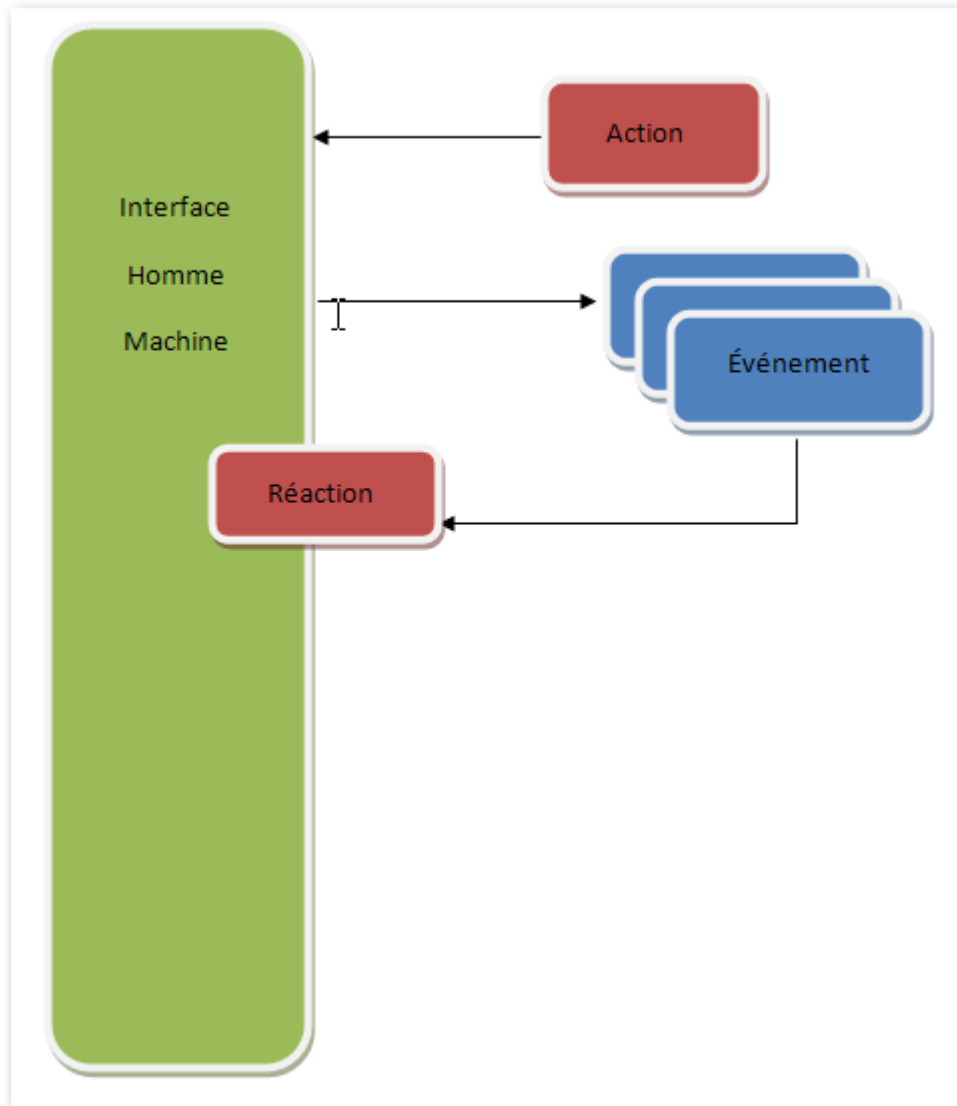
L'événement permet d'exécuter une ou plusieurs instructions suite à l'action d'un utilisateur. Lorsque vous cliquez sur un bouton, vous produisez une action qui engendre un ou plusieurs événements.

Chaque événement possède sa procédure (ensemble de ligne de code) appelée **Procédure événementielle**. Lorsque vous utilisez les événements dans une application Ms ACCESS vous faite de la programmation événementielle.

IV - Règles

Les événements sont encadrés par quelques règles simples.

- Chaque action entraîne un ou plusieurs événements.
- Sauf exception, une action est produite par l'utilisateur.
- Dans le cas où la précédente loi ne pourrait pas engendrer la loi numéro 1, le développeur devra s'en charger.
- Action Événement Interface Homme Machine Le déclenchement d'événements est séquentiel, jamais simultané.



V - Groupes d'événements

Les événements sont classés en 2 groupes qui ne fonctionnent pas de la même manière. Ceux des formulaires et ceux des états.

VI - Comment utiliser un événement

Seuls les événements qui sont activés par l'utilisateur sont utilisés. Pour activer un événement il suffit d'y assigner du code. Le plus simple est de se positionner en mode Création ou Page sur l'objet dont on souhaite utiliser un événement, pour en afficher les propriétés.

Cliquez sur l'onglet **Événements**.

Cliquez sur l'événement et choisir dans la liste **[Procédure événementielle]**.

Cliquez sur l'icône



Si vous préférez utiliser des macros écrivez directement le nom de la macro dedans.

Vous pouvez également mettre le nom d'une fonction précédée du signe égale. Combiné à la multi sélection de contrôles ayant des événements communs vous attribuerez rapidement une fonction à un groupe de contrôle.

Vers une fonction VBA

```
=mafonction()
```

Ou encore

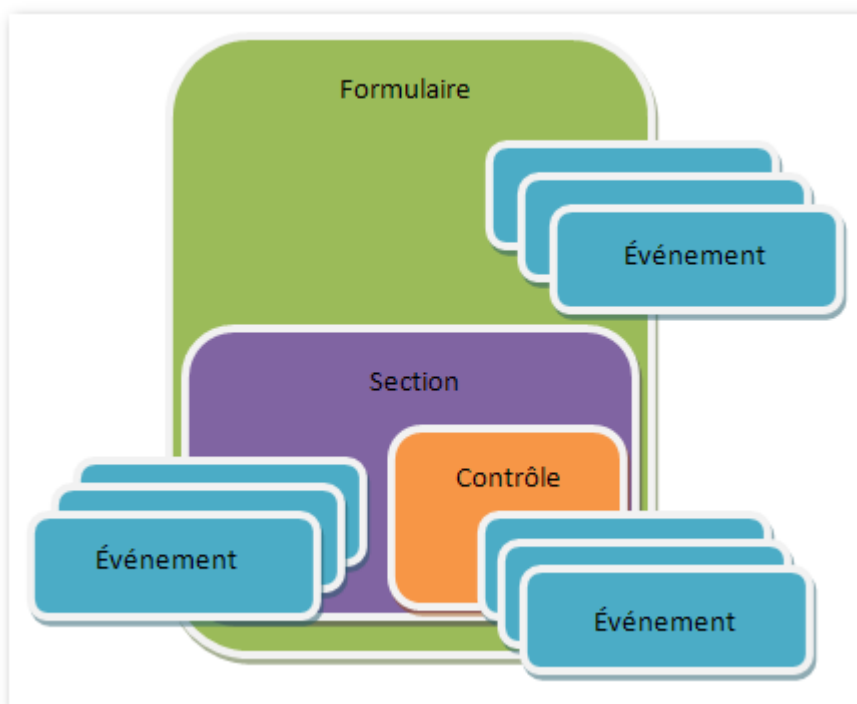
La même avec une référence à un formulaire

```
=mafonction(forms.monform)
```

VII - Les événements formulaires

Le formulaire est en majorité doté d'événements dit « utilisateur ». C'est donc l'utilisateur qui les déclenche par ses actions. L'exception concerne l'événement **Sur minuterie** du formulaire car celui-ci est déclenché par l'horloge de la machine suivant un intervalle définie par le développeur. Nous verrons ce cas particulier en même temps que les autres.

Dans le groupe des événements formulaires on trouve 3 groupes. Les événements de Formulaires, de Sections et de Contrôles.



Le plan des événements Formulaire

Vous constaterez que certains événements existent dans les 3 groupes. Ils sont cependant totalement indépendants.

VII-A - A l'ouverture

VII-A-1 - Sur ouverture - Form_Open

Se produit lorsqu'un formulaire est ouvert, mais avant l'affichage du premier enregistrement.

Au cours de cet événement vous pouvez agir sur les contrôles du formulaire : placer le focus sur un contrôle particulier, définir ou modifier une source de données.

VII-A-2 - Sur chargement - Form_Load

Se produit lorsqu'un formulaire est ouvert et que ses enregistrements sont affichés.

Identique au précédent mais ne se produit qu'une seule fois.

VII-A-3 - Sur redimensionnement - Form_Resize

Se produit lorsque le formulaire change de dimension, soit par l'action de l'utilisateur sur le cadre du formulaire, soit lorsqu'il s'affiche à l'ouverture ou soit qu'il est redimensionné par le code.

L'événement se produit une fois à chaque pas de la souris.


VII-A-4 - Sur activé - Form_Activate

Se produit lorsque le formulaire devient la fenêtre active.

VII-A-5 - Sur activation - Form_Current

Se produit lorsque le focus passe à un enregistrement donné pour en faire l'enregistrement activé, ou lorsque le formulaire est actualisé ou qu'il fait l'objet d'une nouvelle requête. On l'utilise largement pour modifier la présentation ou activer/désactiver des contrôles.

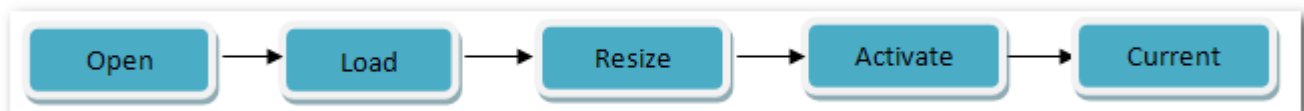
Ne pas confondre avec l'événement Sur activé - Form_Activate.

 *Source de données obligatoire.*

Dans l'exemple suivant on utilise l'événement pour changer l'aspect d'un contrôle suivant sa valeur.

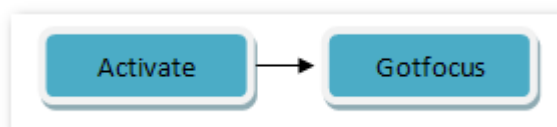
```
Private Sub Form_Current()
    If Me.Type = "Formateur" Then
        Me.Type.BackColor = RGB(255, 255, 0) 'jaune
    Else
        Me.Type.BackColor = RGB(255, 255, 255) 'blanc
    End If
End Sub
```

VII-A-6 - Enchaînement des événements lors de l'ouverture d'un formulaire



Load / Resize / Open / Activate / Current

VII-A-7 - Enchaînement lors de l'activation d'un formulaire déjà ouvert.



Activate / Gotfocus

VII-B - A la fermeture

VII-B-1 - Sur libération - Form_Unload

Se produit lorsque le formulaire est fermé mais toujours à l'écran. Il est peut être provoqué par toutes actions de fermeture du formulaire, de sa mise en mode création (design), de la fermeture de l'application, d'ACCESS ou de Windows. On peut demander une confirmation de fermeture sur cet événement mais pas sur les suivants.

i *Prise en charge de l'annulation.*

Dans l'exemple suivant on demande la confirmation à l'utilisateur avant de fermer le formulaire. Si l'utilisateur répond Non l'événement est annulé et le processus de fermeture est stoppé.

```
Private Sub Form_Unload(Cancel As Integer)
    If MsgBox("Souhaitez-vous fermer le formulaire ?", vbYesNo, "titre") = vbNo Then Cancel = True
End Sub
```

VII-B-2 - Sur désactivé - Form_Deactivate

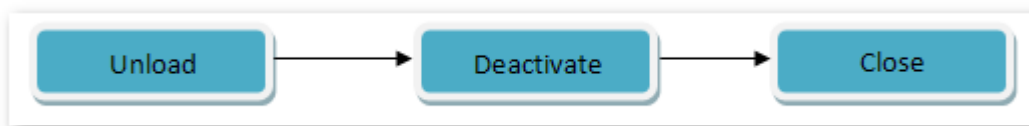
Se produit chaque fois que le formulaire perd le focus. Elle se trouve alors dans un état désactivé.

VII-B-3 - Sur fermeture- Form_Close

Se produit lors de la fermeture du formulaire et qu'il est supprimé de l'écran. Cet événement ne peut être annulé. C'est le dernier événement de la chaîne de fermeture.

VII-B-4 - Enchaînement des événements de fermeture

Lorsqu'un ordre de fermeture est donné au formulaire, trois événements s'enchaînent.



Unload / Deactivate / Close

VII-C - Ajout, Suppression, modification d'enregistrement

Les événements suivants s'appliquent à l'enregistrement courant.

VII-C-1 - Avant insertion - Form_BeforeInsert

Se produit avant que les données d'un nouvel enregistrement ne soient enregistrées. Cet événement est souvent utilisé pour demander à l'utilisateur de confirmer l'ajout de données.

! *Source de données obligatoire.*


i *Prise en charge de l'annulation.*

Dans l'exemple suivant on pose la demande confirmation à l'utilisateur pour la création de l'enregistrement. Si celui-ci clique sur Non la création est annulée.

```
Private Sub Form_BeforeInsert(Cancel As Integer)
    If MsgBox("Souhaitez-vous réellement créer cet enregistrement ?", vbYesNo, "Titre") = vbNo Then
        Cancel = True
    End If
End Sub
```

VII-C-2 - Après insertion - Form_AfterInsert


Se produit après l'ajout d'un nouvel enregistrement.

 *Source de données obligatoire.*

VII-C-3 - Avant MAJ - Form_BeforeUpdate

Se produit avant que les données modifiées d'un enregistrement soient mise à jour.

Cet événement peut être utilisé pour contrôler la validité des données avant l'enregistrement.

 *Source de données obligatoire.*

 *Prise en charge de l'annulation.*


VII-C-4 - Après MAJ - Form_AfterUpdate

Se produit après que les données modifiées d'un enregistrement ont été mises à jour.

 *Source de données obligatoire.*

VII-C-5 - Avant suppression - Form_BeforeDelConfirm

Se produit avant le message de confirmation de la suppression d'un enregistrement.


 *Source de données obligatoire.*

 *Prise en charge de l'annulation.*

 *Contrôle du message ACCESS possible.*

VII-C-6 - Sur suppression - Form_Delete

Se produit avant la suppression d'un enregistrement.

 *Source de données obligatoire.*


 *Prise en charge de l'annulation.*

VII-C-7 - Après Suppression - Form_AfterDelConfirm

Se produit après chaque suppression d'un enregistrement

Paramètre Status :

- **acDeleteOk** (effacement effectué),
- **acDeleteCancel** (effacement annulé par programme),
- **asDeleteUserCancel** (effacement annulé par l'utilisateur)

 *Source de données obligatoire.*

VII-D - Détection d'erreur, de modification et d'annulation.

VII-D-1 - Sur annulation - Form_Undo

Se produit lorsque l'utilisateur annule une modification. L'événement est déclenché par l'appuie sur la touche ESC (Echap), par un bouton d'annulation déclaré comme tel ou encore par la méthode d'annulation.


```
Me.Undo
```

VII-D-2 - Si modification - Form_Dirty

Se produit lors de la modification de la valeur d'un enregistrement. On peut connaître cet état en consultant la propriété **Dirty**

```
Me.dirty
```

Notez que si vous modifiez par code l'enregistrement, que vous initialisez les champs d'un nouvel enregistrement ou que vous déclarez une source de données par programme, la propriété Dirty sera vrai.

 *Source de données obligatoire.*

VII-E - Sur erreur - Form_Error

Se produit lorsqu'une erreur est levée dans le formulaire. Elles peuvent être traitées globalement dans cet événement.

Les paramètres permettent de connaître l'erreur et de modifier le comportement d'ACCESS.

- **DataErr** contient le numéro de l'erreur levée.
- **Response** peut prendre plusieurs valeurs suivant le comportement souhaité.
- **acDataErrContinue** ignore l'erreur, n'affiche pas de message. Permet de traiter l'erreur et/ou de placer un message.
- **acDataErrDisplay** affiche le message d'ACCESS.

L'exemple suivant montre l'utilisation d'une gestion d'erreur commune à l'ensemble des contrôles d'un formulaire.

```
Private Sub Form_Error(DataErr As Integer, Response As Integer)
    Select Case DataErr
        Case 2237
            Response = acDataErrContinue
            MsgBox "Vous devez sélectionner une valeur de la liste.", vbOKOnly, "titre"
            Me.Undo
        Case 2113
```

```

        Response = acDataErrContinue
        MsgBox "Le valeur pour ce champ est incorrecte.", vbOKOnly, "titre"
        Me.Undo
    Case Else
        Response = acDataErrDisplay
    End Select
End Sub
    
```

VII-F - Perte et réception de focus

VII-F-1 - Sur réception focus - Form_GotFocus

Se produit lorsque le formulaire reçoit le focus. Lorsque vous êtes sur un autre objet ou une autre application et que vous cliquez sur le formulaire.

VII-F-2 - Sur perte focus - Form_LostFocus

Se produit lorsque le formulaire perd le focus. Le focus est reçu par un autre objet généralement un autre formulaire ou état.

VII-G - Périphériques d'entrées/sorties

VII-G-1 - Sur clic - Form_Click

Se produit lorsque l'utilisateur appuie puis relâche le bouton gauche de la souris sur le sélecteur du formulaire. Se produit également au premier clic d'un double-clic. Ne se produit ni si l'utilisateur clique sur un contrôle ou une Section du formulaire, ni sur les éléments comme la barre de titre ou la barre d'état et autre ascenseur.

VII-G-2 - Sur double-clic - Form_DblClick

Se produit lors du deuxième clic dans la période fixé par le réglage de la souris dans le panneau de configuration de Windows. Il succède obligatoire à l'événement **Sur clic** (form_clic).

 *Prise en charge de l'annulation.*

VII-G-3 - Enchaînement des événements clic de souris



Click / Dblick

VII-G-4 - Sur souris appuyée - Form_MouseDown

Se produit lors de l'appuie sur un bouton de la souris. On peut connaître le bouton ou la séquence de bouton+touche que l'utilisateur a utilisée ainsi que l'endroit où est positionné le pointeur de la souris grâce aux paramètres suivant.

Button représente le bouton pressé.

- **acLeftButton** bouton gauche de la souris.
- **acRightButton** bouton droit de la souris.
- **acMiddleButton** bouton du milieu de la souris.

Shift représente la touche morte enfoncée au moment du clic.

- **acShiftMask** Masque de bits pour la touche MAJ.
- **acCtrlMask** Masque de bits pour la touche CTRL.
- **acAltMask** Masque de bits pour la touche ALT.

X et Y représente la position X et Y du pointeur. Valeur en twips par rapport au bord haut-gauche du formulaire. Dans l'exemple suivant un message très secret s'affiche lorsque l'utilisateur clique dans une zone du Détail avec le bouton gauche tout en maintenant les 3 touches mortes. Le clic doit être fait dans un rectangle déterminé par X et Y.

```
Private Sub Détail_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    'GetInput "Détail", "Détail_MouseDown", "Sur souris appuyée", Button, Shift, X, Y
    If Button = acLeftButton And Shift = acShiftMask + acCtrlMask + acAltMask And X > 200 And
    Y < 1000 Then
        MsgBox "Ce message est top secret !"
    End If
End Sub
```

VII-G-5 - Sur souris relâchée - Form_MouseUp

Se produit lorsqu'un bouton de la souris est relâché. Cet événement survient immédiatement après l'événement *Sur souris appuyée*. Il est doté des mêmes paramètres que *MouseDown*

VII-G-6 - Sur souris déplacée - Form_MouseMove

Se produit lors du déplacement du pointeur de la souris. Il est doté des mêmes paramètres que *MouseDown*.

VII-G-7 - Sur touche appuyée - Form_KeyDown

Se produit lors de l'appuie sur une touche ou d'une séquence de touches ou lorsque est envoyée par code (Sendkeys ou EnvoiTouche). Cet événement possède des paramètres permettant de connaître la combinaison de touches pressée et d'intervenir dessus.

- 1 **KeyCode** Indique la touche frappée. Vous pouvez utiliser les constantes de touches VbKeyXXX ou leur code ASCII/ANSI.
- 2 **Shift** Indique si une touche morte est activé lors de l'appuie. Syntaxe identique que pour les événements Mouse.

L'utilisation des événements de type Key ne fonctionnent que si la propriété Aperçu Touches (KeyPreview) du formulaire est à Oui. Si vous souhaitez annuler la frappe utilisez la syntaxe suivante.

```
KeyCode = 0
```

Vous pouvez également la substitué en affectant le code de substitution au paramètre KeyCode.

VII-G-8 - Sur touche relâchée - Form_KeyUp

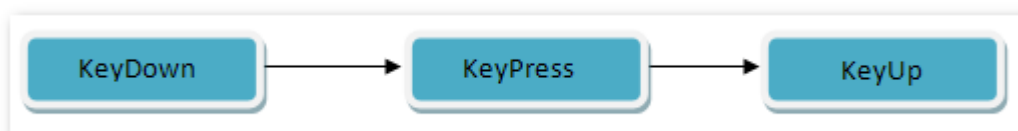
Se produit lorsque la touche est relâchée. Identique au fonctionnement précédent.

VII-G-9 - Sur touche activée - Form_KeyPress

Se produit lorsque la touche est enfoncée. Identique au fonctionnement précédent.

VII-G-10 - Enchaînement des événements Key

Les événements se produisent suivant ce diagramme.



KeyUp / KeyPress / KeyDown

Si l'utilisateur maintient la touche appuyée cette séquence se répète jusqu'à relâchement de la touche.

VII-G-11 - Sur roulement de la souris - Form_MouseWheel

Se produit lors de l'action sur la roulette de la souris. On peut contrôler l'effet de la roulette en agissant sur les paramètres de l'événement.

- **Page** indique si la page est modifiée True / False.
- **Count** Indique le nombre de lignes qui défile lors de l'action.

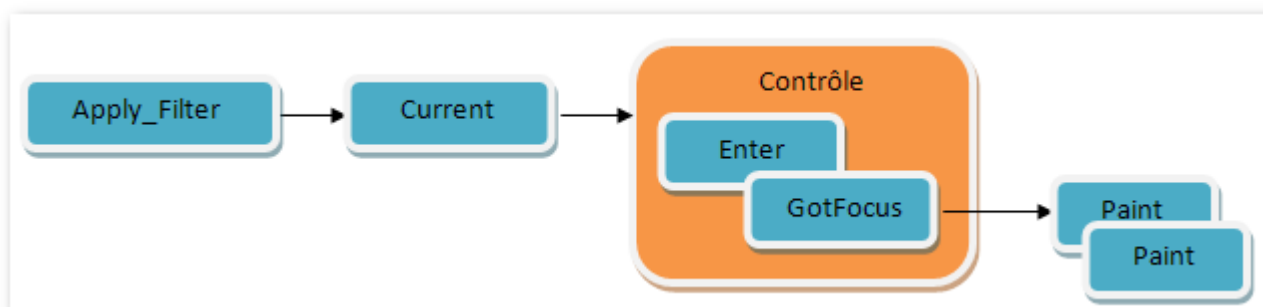
VII-H - Filtrer des données

VII-H-1 - Sur filtre - Form_Filter

Se produit lorsque l'utilisateur sollicite le Filtre par formulaire.

VII-H-2 - Sur filtre appliqué - Form_Apply_filter

Se produit au moment de l'application du filtre. Il entraîne d'autres événements dont ceux du contrôle qui reçoit le focus. Le contrôle actif perd le focus lors du clic sur le ruban, puis la pose du filtre se déclenche et le focus revient sur le contrôle. Notez que si le filtre est posé par le code il n'y a pas de perte/reprise de focus sur le contrôle.



Apply_Filter / Enter / GotFocus / Current/ Paint / Paint

Dans l'exemple suivant on empêche l'utilisateur d'enlever le filtre.

```

Private Sub Form_ApplyFilter(Cancel As Integer, ApplyType As Integer)
If ApplyType = acShowAllRecords Then
    Cancel = true
Endif
End Sub
    
```


VII-I - Evénement autonome

VII-I-1 - Sur minuterie (intervalle minuterie) - Form_Timer

Seul événement à ne pas être déclenché par une action de l'utilisateur, le timer permet de lancer la procédure toutes les n millisecondes. L'intervalle est défini par la propriété **TimerInterval** du formulaire. Cet intervalle est compris entre 0 ; pas de déclenchement, et 2.147.483.647 millisecondes soit presque 25 jours ! On peut attribuer cet intervalle par programme. Cela permet de bloquer le processus (0) ou de le réactiver (>0). Dans cet exemple on affiche l'heure dans une zone de texte nommée Heure.

```
Me.Heure = Format(Now(), "hh:mm:ss")
```

La propriété **Intervalle** est réglée sur 1000 millisecondes.

 *Si vous souhaitez intégrer une horloge il en existe de très belle en Flash. Vous pouvez intégrer très facilement un composant Flash en insérant un contrôle ActiveX dans le formulaire.*

VII-J - Les événements pour formulaire d'analyse croisée dynamique

Les événements suivants ne sont pas commentés dans cette version du tutoriel.

- Lors de la modification de la sélection
- Avant rendu
- Après mise en page
- Lors de la connexion
- Lors de la déconnexion
- Avant requête
- Sur requête
- Lors de modifications des données
- Lors de modifications du jeu de données
- Lors de la commande Cmd Execute
- Lors de la commande Cmd Before Execute
- Lors de la commande Cmd Enabled
- Lors de la commande Cmd Checked
- Lors de la modification de l'affichage
- Lors de la commande du tableau croisé dynamique
- Avant info-bulle

VIII - Les événements de sections


Pour le formulaire les sections sont aux nombres de 5 maximums.

- Entête de Formulaire
- Entête de Page
- Détail
- Pied de Page
- Pied de Formulaire

Généralement la section Détail est la plus utilisée. Les événements de la section fonctionnent de la même manière que pour le formulaire mais leur périmètre d'action est limité à leur section. Par exemple si vous cliquez sur une entête de formulaire, ne vous attendez pas voir l'événement éponyme de la section Détail fonctionner.

VIII-A - Périphériques d'entrées/sorties

Les périphériques d'entrées les plus fréquents sont le clavier, la souris. L'écran tactile plus rarement installé utilise les événements de la souris, de même que les tablettes. Actuellement seul l'écran représente un périphérique de sortie géré nativement par les événements MS ACCESS.

 *Nomdelasection est à remplacer par le nom de la section en anglais (detail, header, footer).*

VIII-A-1 - Sur clic - NomdelaSection_Click

Se produit lorsqu'on clique sur une zone de la section qui ne contient aucun contrôle. Son fonctionnement est identique à l'événement éponyme du formulaire.

VIII-A-2 - Sur double clic - NomdelaSection_DbClick

Identique à l'événement éponyme du formulaire.

VIII-A-3 - Sur souris appuyée - NomdelaSection_MouseDown

Se produit lorsqu'un bouton de la souris est enfoncé.

VIII-A-4 - Sur souris relâchée - NomdelaSection_MouseMove

Se produit lorsque le pointeur de la souris est déplacé.

VIII-A-5 - Sur souris déplacée - NomdelaSection_MouseUp

Se produit lorsqu'un bouton de la souris est relâché.

VIII-B - Affichage

VIII-B-1 - Sur le dessin - NomdelaSection_Paint

Se produit lors d'une modification de l'affichage même partiel de la section.

Quelques cas :

- Fermeture ou déplacement d'une fenêtre masquant même partiellement la section. Il est à noter que même si l'application n'est pas active et que la fenêtre masquant ne fait pas partie de celle-ci l'événement se produit.
- Lorsque le curseur passe d'un contrôle à un autre.
- Ne se produit pas lorsque qu'un menu contextuel est affiché ou effacé.
- Ne se produit pas lorsque le focus passe d'une fenêtre contigüe mais pas chevauchante à une autre.

IX - Événement Contrôles

Les contrôles possèdent également leurs événements. Certains sont communs alors que d'autres sont spécifiques à un type de contrôle. Par exemple une zone de liste modifiable possède en plus d'une zone de texte l'événement **Sur absence dans liste**.

IX-A - Etiquette

Les étiquettes ont un fonctionnement événementiel particulier. Si elle est indépendante, qu'elle n'est pas liée à un contrôle, elle possède ses propres événements :

- Sur clic
- Sur double clic
- Sur souris appuyée
- Sur souris relâchée
- Sur souris déplacée

Lorsqu'elle est liée, elle déclenche les événements du contrôle auquel elle est liée.

IX-A-1 - Sur clic - label_Click


L'événement clic sur une étiquette semble inutile à première vue mais un bon exemple est celui de la mise en place d'un tri sur le champ.


```
Private Sub EtNumero_Click()  
    Me.OrderBy = "Numero ASC"  
    Me.OrderByOn = True  
End Sub
```

Il s'agit là d'un code non finalisé, il faudra imaginer un système d'inversion (ASC, DESC et non trié) et pourquoi pas l'affichage de symboles pour identifier le tri en cours.

IX-A-2 - Sur double clic - label_DbIcClick

Le double clic se déclenche au deuxième clic, le premier étant pris en charge par l'événement précédent.

 *Prise en charge de l'annulation.*

 *Si une action dans l'événement Clic interrompt l'enchaînement comme une attente d'une réponse de la part de l'utilisateur, le message box (msgbox) étant un bon exemple, cet événement ne se produira pas.*

IX-B - Ligne

Aucun événement sur les lignes.

IX-C - Cadre

Le cadre possède les mêmes événements que l'étiquette.

IX-D - Zone de liste déroulante

Pour la commodité nous nommerons ce contrôle cbo (combobox). Comme pour les autres objets vus précédemment celui-ci comporte des événements communs.

IX-D-1 - Sur clic - cbo_Click

Se produit lors du clic dans le contrôle ou sur l'étiquette liée. Pour lier une étiquette à un contrôle éligible vous devez :

- Sélectionner l'étiquette et faire un ctrl+X
- Sélectionner le contrôle et faire un ctrl+V

IX-D-2 - Avant MAJ- cbo_BeforeUpdate

Se produit avant la mise à jour de la valeur contenue dans le contrôle.

IX-D-3 - Après MAJ - cbo_AfterUpdate

Se produit après avoir choisi une nouvelle valeur de la liste ou après la saisie d'une nouvelle valeur. On utilise souvent cet événement pour effectuer des rafraichissements de contrôle dépendant de la valeur de la liste. Dans l'exemple suivant la liste permet de filtrer une autre liste.

```

Private Sub cbo_AfterUpdate ()
    Me.ListeFiltree.Requery
End Sub
```

IX-D-4 - Si modification- cbo_Dirty

Se produit quand le contenu de l'enregistrement est modifié. Soit parce que l'utilisateur a choisi une valeur de la liste soit qu'il a saisi un caractère.

IX-D-5 - Sur changement - cbo_Change

Se produit lorsque l'utilisateur effectue un changement dans la zone de texte de la liste. L'événement se produit à la première frappe d'une touche du clavier (suppression, saisie, remplacement par un caractère différent) modifiant la valeur de la zone. Dans l'exemple suivant on calcule le nombre de caractères saisis. Si il y a dépassement un message d'affiche.

```

Private Sub cbo_Change ()
    Dim lenmax As Byte
    lenmax = Len(Me.Type.Text)
    If lenmax > 15 Then
        MsgBox "Attention vous dépassez le nombre de caractères autorisé pour ce champ !"
    End If
    Me.lenType.Caption = lenmax & " /15"
End Sub
```



Vous pouvez remarquer que l'on ne teste pas la propriété Value ou le contrôle pour connaître le contenu mais la propriété Text. En effet seule cette propriété est à même de retourner une saisie en cours mais pas validée.

IX-D-6 - Sur absence dans liste - cbo_NotInList

Se produit lorsque l'utilisateur saisie une nouvelle valeur dans la zone de liste déroulante. La propriété Limité à liste (LimitToList) doit être réglée sur Oui.

NewData contient le texte saisie par l'utilisateur.

Response Gère le message et le comportement d'ACCESS.

- **acDataErrDisplay** affiche le message d'ACCESS
- **acDataErrContinue** n'affiche pas le message d'ACCESS
- **acDataErrAdded** n'affiche pas le message d'ACCESS et permet le rajout de la valeur à la liste. ACCESS réactualise seul la liste.

Dans l'exemple suivant nous utilisons le paramètre **acDataErrAdded** pour rajouter la nouvelle entrée à la table source de la zone de liste.

```
Dim rst As DAO.Recordset
Response = acDataErrAdded
Set rst = CurrentDb.OpenRecordset("tblFonction", dbopendynaset)
rst.AddNew
rst.Fields("Metier") = NewData
rst.Update
rst.Close
Set rst = Nothing
```

IX-D-7 - Sur entrée - cbo_GotFocus

Se produit à la réception du focus par le contrôle. On utilise souvent cet événement pour dérouler automatiquement la liste comme dans l'exemple ci-dessous.

```
Private Sub cbo_GotFocus()
Me.cbo.DropDown
End Sub<
```

IX-D-8 - Sur entrée - cbo_Enter

Se produit à l'entrée dans la liste après que le focus ait été reçu.

IX-D-9 - Sur sortie - cbo_Exit

Se produit après la perte du focus.

 *Prise en charge de l'annulation.*

IX-E - Onglet

Nous ne détaillerons pas l'intégralité des événements des onglets mais nous nous attacherons à l'un d'eux qui est très important.

IX-E-1 - Sur changement - tab_Change()

Parmi les quelques événements du contrôle d'onglet le plus important est bien celui-ci. En effet lorsque vous cliquez sur un des onglets c'est celui-ci qui se produit. Paradoxalement clic et double clic ne se produisent que lorsque vous cliquez hors des onglets. Pour illustrer cet événement voici une petite astuce pour déclencher une action lors du clic sur un onglet particulier. Dans l'exemple il s'agit de quitter l'application.

```

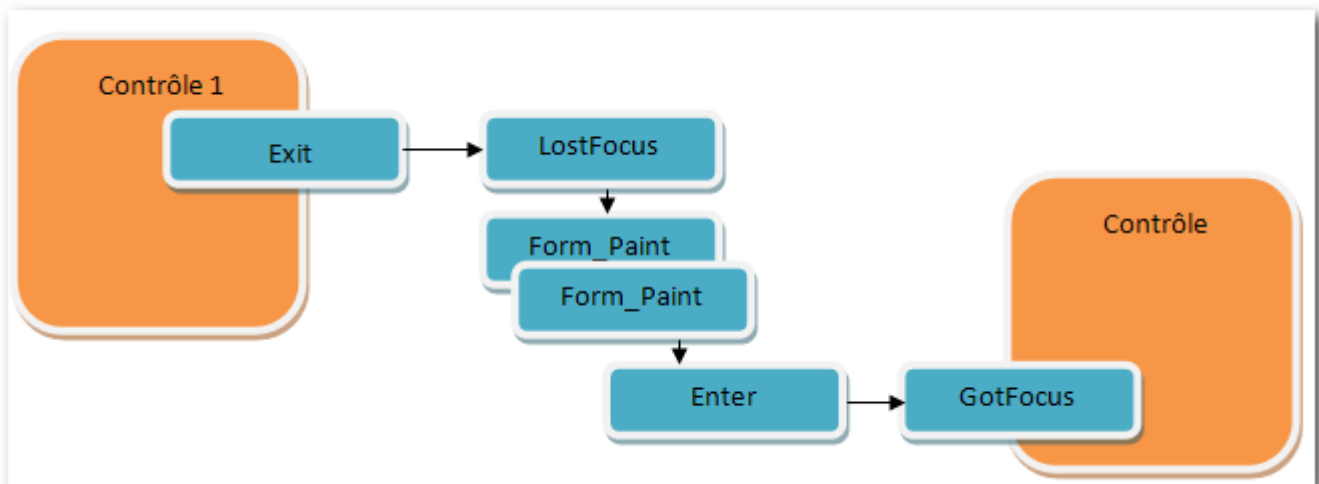
If Me.CtlTab0.Pages.Item(Me.CtlTab0).Caption = "Quitter" Then
    If MsgBox("Etes-
vous sur(e) de vouloir quitter DVDtek ?", vbYesNo + vbDefaultButton1, "info") = vbYes Then
        DoCmd.Quit
    Else
        Me.PageDVD.SetFocus
    End If
End If

```

Ce code est tiré de l'application DVDtek.

IX-F - Enchaînement des événements d'un contrôle vers un autre

Le schéma suivant montre le mécanisme des événements lors du déplacement du curseur d'un contrôle à un autre.



Control / Form_Paint / LostFocus / Exit / Enter / GotFocus / Form_Paint / Control

L'événement Form_Paint appartient au formulaire.

X - Déclencher manuellement un événement

Dans les règles de fonctionnement des événements nous avons pu voir que le déclenchement d'un événement pouvait être réalisé par le programmeur. Pour déclencher un événement à partir du code VBA il suffit d'appeler la procédure comme n'importe quelle procédure.

```
btnFermer_Click
```

Ici on déclenche l'événement **Sur clic** d'un contrôle nommé **btnFermer**. Pour les événements ayant des paramètres vous devez les spécifier car ceux-ci ne sont jamais optionnels. Pour déclencher le même événement mais de l'extérieur du formulaire vous devez dans un premier temps rendre la procédure publique.

Transformez cette instruction

```
Private Sub btnFermer_Click()
```

En cette instruction

```
Public Sub btnFermer_Click()
```

Maintenant l'événement peut être déclenché de l'extérieur du formulaire. Comme ceci :

```
Form_frmClient.btnFermer_Click
```

XI - Astuce pour connaître le nom VBA d'un événement

Il est souvent utile de connaître le nom d'un événement en VBA, pour cela rien de plus simple :

- Ouvrez le formulaire en mode création.
- Affichez les propriétés
- Cliquez sur l'événement et appuyez sur F1.

XII - Conclusion

Au terme de ce tutoriel vous avez pu remarquer la nécessité d'utiliser les événements. En faire un inventaire exhaustif est inutile, l'important sera d'acquérir la logique de fonctionnement et surtout l'ordre dans lequel ils s'enchainent. Pour en savoir plus je vous invite à consulter l'aide de Microsoft ACCESS, le centre de développement disponible sur le site MSDN de Microsoft ainsi que l'application Chasseur d'événements disponible sur ma page personnelle.

XIII - Liens

Portail des développeurs MS ACCESS : <http://msdn.microsoft.com/fr-fr/office/aa905400.aspx>

Le chasseur d'événements : <http://loufab.developpez.com/>

Vous souhaitez en savoir plus sur  <http://loufab.developpez.com/tutoriels/access/appelformulaire/>