

Utiliser les évènements dans les graphiques Excel

par [SilkyRoad \(silkyroad.developpez.com\)](http://silkyroad.developpez.com)

Date de publication : 06.11.2006

Dernière mise à jour :

Les graphiques sont des supports incontournables pour visualiser et analyser des résultats de tous types.

Les outils disponibles dans le tableur Excel sont très complets et offrent de nombreuses possibilités dans ce domaine.

Notamment les évènements qui permettent l'interaction entre votre programme et l'utilisateur. Les évènements permettent d'intercepter les déplacements de la souris, la sélection d'un objet, la modification d'une donnée...

Ce document décrit les évènements disponibles pour manipuler les graphiques et propose quelques exemples d'utilisation.

Toutes les macros de ce document ont été testées à partir d'Excel2002.

I - Préambule

II - Les évènements dans un graphique

II-A - La description des évènements

II-A-1 - L'activation du graphique

II-A-2 - Le double clic dans un graphique

II-A-3 - Le clic droit dans un graphique

II-A-4 - La mise à jour du graphique

II-A-5 - La désactivation du graphique

II-A-6 - Les mouvements de la souris sur le graphique

II-A-7 - La sélection d'un objet dans le graphique

II-A-8 - La modification des séries

II-A-9 - Le redimensionnement du graphique

II-B - Les évènements dans les graphiques incorporés

III - Conclusion

IV - Téléchargements

I - Préambule

Si vous débutez dans le monde des graphiques, je vous conseille la lecture préalable de l'excellent tutoriel de J-M Rabilloud. Les graphiques y sont traités à partir de la page 19.

Programmer efficacement avec Excel en VBA (pdf, 46 pages, 711 Ko) par J-M Rabilloud

et

La FAQ VBA

Vous y trouverez des exemples complets et détaillés pour créer des graphiques dans le classeur.

II - Les évènements dans un graphique

Il existe 2 types de graphiques dans Excel:

Les onglets graphiques (Charts)

Les graphiques incorporés dans la feuille de calcul (ChartObjects)

La syntaxe est identique pour utiliser les évènements. Par contre la méthode diffère pour y accéder.

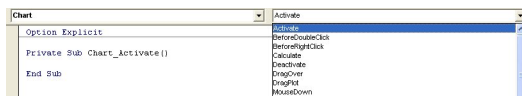
Les feuilles graphiques:

Les évènements sont disponibles depuis l'éditeur de macros (Alt+F11)

Après avoir créé votre graphique, double cliquez sur l'onglet graphique dans l'explorateur de projets afin d'accéder au module objet.



Ensuite, sélectionnez "Chart" dans le menu déroulant de gauche. La liste des évènements est alors visible dans le menu déroulant de droite.



Les graphiques incorporés:

Vous devez utiliser des modules de classe pour gérer les évènements.

Les classes définissent le fonctionnement des objets. Un objet peut être défini par une propriété, une méthode ou un évènement.

(Consultez le chapitre II-B pour plus de détails).



II-A - La description des évènements

II-A-1 - L'activation du graphique

`Private Sub Chart_Activate()`

La procédure est déclenchée dès que le graphique est activé.

```

Vba

Private Sub Chart_Activate()
    MsgBox "Bonjour " & Environ("username") & vbCrLf & _
        "Vous avez activé le graphique " & ActiveChart.Name
End Sub
    
```

II-A-2 - Le double clic dans un graphique

`Private Sub Chart_BeforeDoubleClick(ByVal ElementID As Long, ByVal Arg1 As Long, ByVal Arg2 As Long, Cancel As Boolean)`

L'argument ElementID correspond à l'objet cliqué.

Les Arguments Arg1 et Arg2 dépendent de l'objet cliqué.

Le tableau ci dessous présente la liste des arguments ElementID ainsi que les constantes Arg1 et Arg2 associées:

ElementID	Arg1	Arg2
xlAxis	AxisIndex	AxisType
xlAxisTitle	AxisIndex	AxisType
xlDisplayUnitLabel	AxisIndex	AxisType
xlMajorGridlines	AxisIndex	AxisType
xlMinorGridlines	AxisIndex	AxisType
xlPivotChartDropZone	DropZoneType	Aucune

xlPivotChartFieldButton	DropZoneType	PivotFieldIndex
xIDownBars	GroupIndex	Aucune
xIDropLines	GroupIndex	Aucune
xIHiLoLines	GroupIndex	Aucune
xIRadarAxisLabels	GroupIndex	Aucune
xISeriesLines	GroupIndex	Aucune
xIUpBars	GroupIndex	Aucune
xIChartArea	Aucune	Aucune
xIChartTitle	Aucune	Aucune
xICorners	Aucune	Aucune
xIDataTable	Aucune	Aucune
xIFloor	Aucune	Aucune
xILegend	Aucune	Aucune
xINothing	Aucune	Aucune
xIPlotArea	Aucune	Aucune
xIWalls	Aucune	Aucune
xIDataLabel	SeriesIndex	PointIndex
xIErrorBars	SeriesIndex	Aucune
xILegendEntry	SeriesIndex	Aucune
xILegendKey	SeriesIndex	Aucune
xISeries	SeriesIndex	PointIndex
xITrendline	SeriesIndex	TrendLineIndex
xIXErrorBars	SeriesIndex	Aucune
xIYErrorBars	SeriesIndex	Aucune
xIShape	ShapeIndex	Aucune

La description des arguments Arg1 et Arg2:

Argument	Description
AxisIndex	Indique si l'axe est principal ou secondaire. Il peut s'agir de l'une des constantes XIAxisGroup suivantes : xIPrimary ou xISecondary
AxisType	Indique le type d'axe. Il peut s'agir de l'une des constantes XIAxisType suivantes : xICategory, xISeriesAxis ou xIValue.
DropZoneType	Spécifie le type de zone de projection : champ de colonne, de données, de page ou de ligne. Il peut s'agir de l'une des constantes XIPivotFieldOrientation suivantes: xIColumnField, xIDataField, xIPageField ou xIRowField. Les constantes de champ de colonne et de ligne spécifient

	respectivement les champs de série et de catégorie.
GroupIndex	Indique l'index d'un ChartGroup (plusieurs séries tracées dans un graphique avec le même format) dans une collection ChartGroups.
PivotFieldIndex	Indique l'index d'un champ de colonne (série), de données, de page ou de ligne (catégorie) spécifique de la collection PivotFields. Renvoie -1 si le type de zone de projection est xlDataField.
PointIndex	Renvoie l'index du point dans la collection de points de la série. la valeur #1 indique que tous les points sont sélectionnés.
SeriesIndex	Renvoie l'index de la série dans la collection de séries dans le graphique.
ShapeIndex	Renvoie l'index d'une forme automatique (Shapes) dans une collection de formes.
TrendlineIndex	Renvoie l'index d'une courbe de tendance dans une collection de courbes de tendances.

Voici un exemple pour empêcher l'affichage de la boîte de dialogue "Format de zone de traçage" lors du double clic dans le graphique.

Vba

```
Private Sub Chart_BeforeDoubleClick(ByVal ElementID As Long, _
    ByVal Arg1 As Long, ByVal Arg2 As Long, Cancel As Boolean)
    Cancel = True
End Sub
```

II-A-3 - Le clic droit dans un graphique

```
Private Sub Chart_BeforeRightClick(Cancel As Boolean)
```

L'évènement survient lors de l'utilisation du clic droit dans un graphique.

Cet exemple empêche l'affichage du menu contextuel.

Vba

```
Private Sub Chart_BeforeRightClick(Cancel As Boolean)
    MsgBox "Vous venez d'effectuer un clic droit."
    Cancel = True
End Sub
```

II-A-4 - La mise à jour du graphique

```
Private Sub Chart_Calculate()
```

Cet évènement permet d'intercepter les mises à jour du graphique, notamment la modification des valeurs dans la source de données.

Vba

```
Private Sub Chart_Calculate()
    MsgBox "Graphique mis à jour"
End Sub
```

Nota:

Les mises en forme personnalisées des graphiques croisés dynamiques ne sont pas conservées lors des réactualisations de données. Cet évènement peut donc être utile afin de forcer une mise en forme.

Un exemple pour que la 1ere série du graphique soit toujours de couleur rouge

Vba

```
Private Sub Chart_Calculate()
    Graph1.SeriesCollection(1).Border.ColorIndex = 3
End Sub
```

II-A-5 - La désactivation du graphique

```
Private Sub Chart_Deactivate()
```

Permet d'intercepter la désactivation du graphique.

Exemple: Empêcher le Copier/Coller d'un graphique.

Cette procédure annule le mode Couper ou Copier lorsque l'on désactive le graphique.

Vba

```
Private Sub Chart_Deactivate()  
    Application.CutCopyMode = False  
End Sub
```

II-A-6 - Les mouvements de la souris sur le graphique

`Private Sub Chart_MouseDown(ByVal Button As Long, ByVal Shift As Long, ByVal x As Long, ByVal y As Long)`

`Private Sub Chart_MouseMove(ByVal Button As Long, ByVal Shift As Long, ByVal x As Long, ByVal y As Long)`

`Private Sub Chart_MouseUp(ByVal Button As Long, ByVal Shift As Long, ByVal x As Long, ByVal y As Long)`

Chart_MouseDown survient lorsque l'utilisateur clique sur la souris alors qu'elle est placée au dessus du graphique.

Chart_MouseUp survient lorsque l'utilisateur relâche le bouton de la souris.

Chart_MouseMove survient lorsque l'utilisateur déplace la souris sur le graphique.

La description des arguments:

Button: Indique quel bouton est activé lors de l'évènement Chart_Mouse:

xlNoButton

xlPrimaryButton

xlSecondaryButton

xlMiddleButton

Shift: Indique quelle touche du clavier est enfoncée lors de l'évènement Chart_Mouse:

0 = Pas de touche

1 = Touche Shift

2 = Touche Ctrl

4 = Touche Alt

Vous pouvez aussi obtenir des associations de touches:

6 = Touches Ctrl & Alt

...etc...

X: Renvoie la position horizontale dans le graphique.

Y: Renvoie la position verticale dans le graphique.

II-A-7 - La sélection d'un objet dans le graphique

```
Private Sub Chart_Select(ByVal ElementID As Long, ByVal Arg1 As Long, ByVal Arg2 As Long)
```

L'argument ElementID correspond à l'objet sélectionné. Les Arguments Arg1 et Arg2 dépendent de l'objet cliqué.

Les arguments sont identiques à ceux de l'évènement [Chart_BeforeDoubleClick](#) .

Cet exemple identifie la sélection de la 3eme série du graphique.

Vba

```
Private Sub Chart_Select(ByVal ElementID As Long, ByVal Arg1 As Long, ByVal Arg2 As Long)
    If ElementID = xlSeries And Arg1 = 3 Then _
        MsgBox "Vous avez sélectionné la 3eme série."
End Sub
```

II-A-8 - La modification des séries

```
Private Sub Chart_SeriesChange(ByVal SeriesIndex As Long, ByVal PointIndex As Long)
```

L'évènement est déclenché lorsque vous modifiez la position d'un point dans le graphique.

Remarque:

Pour gérer les modifications issues de changements dans la source de données , utilisez l'évènement [Chart_Calculate\(\)](#).

Vba

```
Private Sub Chart_SeriesChange(ByVal SeriesIndex As Long, ByVal PointIndex As Long)
    MsgBox "Le point " & PointIndex & " de la série " _
        & SeriesIndex & " vient d'être modifié."

    'Modifie la couleur du point
    ActiveChart.SeriesCollection(SeriesIndex).Points(PointIndex). _
        MarkerBackgroundColorIndex = 12
End Sub
```

II-A-9 - Le redimensionnement du graphique

[Private Sub Chart_Resize\(\)](#)

L'évènement est déclenché lorsque vous modifiez les dimensions du graphique.

II-B - Les évènements dans les graphiques incorporés

Ce chapitre décrit comment gérer les évènements dans les graphiques incorporés.

Il s'agit de la description du classeur proposé en téléchargement, à la fin de ce document. Il existe bien sur d'autres solutions et des adaptations devront être apportées en fonction de vos projets.

Vous devez tout d'abord placer cette procédure dans le module objet ThisWorkbook.

Ces lignes permettront d'intégrer l'application Excel dans la classe "ClasseAppli".

Vba

```
Option Explicit

Dim XlAppli As New ClasseAppli

Private Sub Workbook_Open()
```

Vba

```
Set XlAppli.XL = Excel.Application
End Sub
```

Ensuite, créez un module de classe nommé "ClasseAppli".

Placez y ces procédures qui permettront de gérer les graphiques contenus dans la feuille active.

Les graphiques seront pris en compte dans la classe, dès l'activation de l'onglet.

Vba

```
Option Explicit

Public WithEvents XL As Excel.Application
Dim ClTabChart() As ClasseChart

'Permet de prendre en compte les graphiques incorporés de la feuille active
Public Sub XL_SheetActivate(ByVal Feuille As Object)
    Dim i As Integer

    'Vérifie qu'il s'agit d'une feuille de calcul
    If TypeOf Feuille Is Worksheet Then
        'Vérifie s'il y a des graphiques dans la feuille
        If Feuille.ChartObjects.Count = 0 Then Exit Sub

        'S'il y a des graphiques,
        'boucle pour les intégrer dans le module de classe
        For i = 1 To Feuille.ChartObjects.Count
            ReDim Preserve ClTabChart(i)
            Set ClTabChart(i) = New ClasseChart
            Set ClTabChart(i).Graph = Feuille.ChartObjects(i).Chart
        Next i

    End If
End Sub

'Permet de vider la classe lors de la désactivation de la feuille
Private Sub XL_SheetDeactivate(ByVal Feuille As Object)
    Dim j As Integer

    On Error Resume Next

    For j = 1 To UBound(ClTabChart)
        Set ClTabChart(j).Graph = Nothing
    Next

End Sub
```

Ensuite, créez un module de classe nommé "ClasseChart".

Placez y les procédures suivantes.

Il s'agit des macros événementielles qui permettront de gérer les graphiques. Vous pouvez constater que les syntaxes sont similaires au chapitre II-A.

La seule différence concerne le nom des procédures:

Dans un onglet graphique le nom s'affiche automatiquement, dans le style:

`Private Sub Chart_Activate()`

Ici, vous allez déclarer: `Public WithEvents Graph As Chart`

Il faudra donc écrire:

`Private Sub Graph_Activate()`

```
Vba

Option Explicit

Public WithEvents Graph As Chart
Dim NomGraph As String

'*** Utilisation des évènements ****

Private Sub Graph_Activate()
    MsgBox "Bonjour " & Environ("username") & vbCrLf & _
        "Vous avez activé le graphique " & ActiveChart.Name

    NomGraph = ActiveChart.Name
End Sub

Private Sub Graph_BeforeDoubleClick(ByVal ElementID As Long, _
    ByVal Arg1 As Long, ByVal Arg2 As Long, Cancel As Boolean)

    Range("F4") = RetourneDescriptionID(ElementID, Arg1, Arg2)

    Cancel = True
End Sub

Private Sub Graph_BeforeRightClick(Cancel As Boolean)
    Range("F4") = "Vous avez fait un clic droit."
    Cancel = True
End Sub

Private Sub Graph_Calculate()
    Range("F4") = "Graphique mis à jour"
```

Vba

```
End Sub

Private Sub Graph_Deactivate()
    On Error Resume Next
    Range("F4") = NomGraph & " désactivé"
End Sub

Private Sub Graph_Select(ByVal ElementID As Long, _
    ByVal Arg1 As Long, ByVal Arg2 As Long)

    Range("F4") = "L'objet sélectionné: " & _
        RetourneDescriptionID(ElementID, Arg1, Arg2)
End Sub

Private Sub Graph_MouseMove(ByVal Button As Long, ByVal Shift As Long, _
    ByVal x As Long, ByVal y As Long)
    Dim ElementID As Long
    Dim Arg1 As Long, Arg2 As Long

    'Filtre pour que cette procédure soit appliquée sur un graphique spécifique
    If ActiveChart.Name = "Feuill Graphique 2" Then

        If Feuill.CheckBox1 = False Then
            'Button: Indique quel bouton est activé lors de l'évènement Chart_Mouse
            'xlNoButton
            'xlPrimaryButton
            'xlSecondaryButton
            'xlMiddleButton
            'Shift: indique quelle touche du clavier est enfoncée lors des
            'mouvements de la souris:
            '0 = pas de touche
            '1 = Touche Shift
            '2 = Touche Ctrl
            '4 = Touche Alt
            'Vous pouvez aussi obtenir des associations de touches:
            '6 = Touches Ctrl & Alt
            '...etc...
            'x: position horizontale dans le graphique
            'y: position verticale dans le graphique
            Range("F4") = Button & " / " & Shift & " / " & x & " / " & y
        Else
            ActiveChart.GetChartElement x, y, ElementID, Arg1, Arg2

            Range("F4") = "Le curseur se déplace sur : " & _
                RetourneDescriptionID(ElementID, Arg1, Arg2)
        End If
    End If
End Sub

Private Sub Graph_Resize()
    MsgBox "Pourquoi avez vous modifié les dimensions du graphique?"
End Sub

Private Sub Graph_SeriesChange(ByVal SeriesIndex As Long, ByVal PointIndex As Long)

    Range("F4") = "Le point " & PointIndex & " de la série " & _
        & SeriesIndex & " vient d'être modifié."
```

Vba

```
'Modifie la couleur du point
ActiveChart.SeriesCollection(SeriesIndex).Points(PointIndex). _
    MarkerBackgroundColorIndex = 12
End Sub
```

Enfin, placez la fonction [RetourneDescriptionID](#) dans un module standard.

Cette fonction permet de traduire les arguments ElementID, Arg1 et Arg2 (constantes de type Long) en texte. Consultez les tableaux descriptifs dans le chapitre II-A-2 pour plus de détails.

Cette fonction est traduite et adaptée d'une procédure de Jon Peltier:



<http://www.comptorcompanion.com/LPMArticle.asp?ID=221>

Vba

```
Option Explicit

'-----
'La fonction "RetourneDescriptionID" permet de traduire les arguments
'ElementID , Arg1 et Arg2 (constantes de type Long) en texte.
'-----

Public Function RetourneDescriptionID _
    (ElementID As Long, Arg1 As Long, Arg2 As Long)

'Adapté et traduit d'une procédure de Jon Peltier
'http://www.comptorcompanion.com/LPMArticle.asp?ID=221

    Dim Arg As String, CstElement As String

    Select Case ElementID
        Case 0
            CstElement = "Etiquette de donnée" '0
            Arg = "Series " & Arg1
            If Arg2 > 0 Then Arg = Arg & ", Point " & Arg2

        Case 2: CstElement = "Zone de graphique" '2

        Case 3:
            CstElement = "Série " & Arg1 '3
            If Arg2 > 0 Then Arg = "Point " & Arg2
            If Arg2 = -1 Then Arg = "Toute la série est sélectionnée : " & _
                ActiveChart.SeriesCollection(Arg1).Formula

        Case 4: CstElement = "Titre" '4

        Case 5: CstElement = "Panneau" '5

        Case 6: CstElement = "Coins" '6

        Case 7: CstElement = "Table de données" '7
```

Vba

```
Case 8
    CstElement = "Courbe de tendance: " & Arg1
    Arg = "Série " & Arg1 & ", Courbe de tendance " & Arg2

Case 9
    CstElement = "Barre d'erreur" & Arg1
    Arg = "Série " & Arg1

Case 10
    CstElement = "Barre d'erreur X" & Arg1
    Arg = "Série " & Arg1

Case 11
    CstElement = "Barre d'erreur Y" & Arg1
    Arg = "Série " & Arg1

Case 12
    CstElement = "Élément de légende" & Arg1
    Arg = "Série " & Arg1

Case 13
    CstElement = "Symbole de légende" & Arg1
    Arg = "Série " & Arg1

Case 14
    CstElement = "Forme automatique" & Arg1
    Arg = "Forme automatique n° " & Arg1

Case 15
    CstElement = "Quadrillage Principal " & Arg1
    CstElement = CstElement & IIf(Arg1 = 1, "Principal ", "Secondaire ")
    CstElement = CstElement & IIf(Arg2 = 1, "Abscisses ", "Ordonnées ")

Case 16
    CstElement = CstElement & "Quadrillage Secondaire "
    CstElement = IIf(Arg1 = 1, "Principal ", "Secondaire ") & Arg1
    CstElement = CstElement & IIf(Arg2 = 1, "Abscisses ", "Ordonnées ")

Case 17
    CstElement = "Titre de l'axe "
    CstElement = CstElement & IIf(Arg1 = 1, "Principal ", "Secondaire ") & Arg1
    CstElement = CstElement & IIf(Arg2 = 1, "Abscisses ", "Ordonnées ")

Case 18
    CstElement = "Barre de hausse" & Arg1
    Arg = "Groupe Index " & Arg1

Case 19: CstElement = "Zone de traçage" & Arg1

Case 20
    CstElement = "Barre de baisse" & Arg1
    Arg = "Groupe Index " & Arg1

Case 21
    CstElement = "Axe " & Arg1
    CstElement = CstElement & IIf(Arg1 = 1, "Principal ", "Secondaire ")
    CstElement = CstElement & IIf(Arg2 = 1, "Abscisses ", "Ordonnées ")

Case 22
    CstElement = "Ligne de Séries" & Arg1
    Arg = "Groupe Index " & Arg1

Case 23: CstElement = "Plancher" & Arg1

Case 24: CstElement = "Légende" & Arg1

Case 25
    CstElement = "Lignes Haut-Bas" & Arg1
```


Vba

```
Arg = "Groupe Index " & Arg1

Case 26
CstElement = "Lignes de projection" '26
Arg = "Groupe Index " & Arg1

Case 27
CstElement = "Etiquette de catégorie axe Radar" '27
Arg = "Groupe Index " & Arg1

Case 28: CstElement = "Aucune" '28

Case 30
CstElement = "Etiquette d'unités d'affichage "
CstElement = CstElement & IIf(Arg1 = 1, " Principal ", " Secondaire ") '30
CstElement = CstElement & IIf(Arg2 = 1, "Abscisses ", "Ordonnées ")
End Select

RetourneDescriptionID = CstElement & IIf(Len(Arg) > 0, " " & Arg, "")
End Function

'-----
```

III - Conclusion

J'espère que toutes ces informations pourront vous être utiles. Il s'agit d'exemples généraux qu'il conviendra d'adapter en fonction de vos projets.

Les événements offrent de nombreuses possibilités pour personnaliser votre application et interagir avec les actions de l'utilisateur final.

Si vous souhaitez contrôler et limiter les manipulations possibles dans le classeur, il sera important:

De définir ce qui est autorisé et ce qui ne l'est pas,

D'anticiper les actions involontaires des utilisateurs.

Et ensuite d'intégrer ces paramètres dans les macros événementielles.

IV - Téléchargements

Téléchargez un exemple de classeur pour gérer les évènements graphiques.

