

Les évènements du module objet ThisWorkbook, dans Excel

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

Date de publication : 21/10/2006

Dernière mise à jour : 22/10/2007

Ce tutoriel décrit les évènements associés au niveau du classeur (module objet ThisWorkbook), disponibles dans Excel.

Mise à jour pour présenter les évènements jusqu'à la version Excel2007.

I - Introduction

II - Les événements

- II-1 - Workbook_Activate
- II-2 - Workbook_AddinInstall et Workbook_AddinUninstall
- II-3 - Workbook_AfterXmlExport
- II-4 - Workbook_AfterXmlImport
- II-5 - Workbook_BeforeClose
- II-6 - Workbook_BeforePrint
- II-7 - Workbook_BeforeSave
- II-8 - Workbook_BeforeXmlExport
- II-9 - Workbook_BeforeXmlImport
- II-10 - Workbook_Deactivate
- II-11 - Workbook_NewSheet
- II-12 - Workbook_Open
- II-13 - Workbook_PivotTableCloseConnection
- II-14 - Workbook_PivotTableOpenConnection
- II-15 - Workbook_RowsetComplete
- II-16 - Workbook_SheetActivate
- II-17 - Workbook_SheetBeforeDoubleClick
- II-18 - Workbook_SheetBeforeRightClick
- II-19 - Workbook_SheetCalculate
- II-20 - Workbook_SheetChange
- II-21 - Workbook_SheetDeactivate
- II-22 - Workbook_SheetFollowHyperlink
- II-23 - Workbook_SheetPivotTableUpdate
- II-24 - Workbook_SheetSelectionChange
- II-25 - Workbook_Sync
- II-26 - Workbook_WindowActivate
- II-27 - Workbook_WindowDeactivate
- II-28 - Workbook_WindowResize

III - Téléchargement

I - Introduction

Les événements permettent l'interaction entre votre programme et l'utilisateur. Il s'agit de procédures qui se déclenchent automatiquement lorsqu'une action prédéfinie survient dans:

- * L'application Excel
- * Le classeur
- * **La feuille de calcul**
- * **Les graphiques**
- * **Les UserForm**
- * Les contrôles

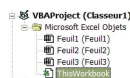
Vous pouvez ainsi intercepter l'activation ou la désactivation du classeur ou un de ses onglets, la modification d'une donnée...etc...

Les procédures événementielles du modules objet [ThisWorkbook](#) prennent en compte:

- * Le classeur (Workbook_...)
- * La collection de feuilles dans le classeur(Workbook_Sheet..)
- * La fenêtre contenant le classeur(Workbook_Window...)

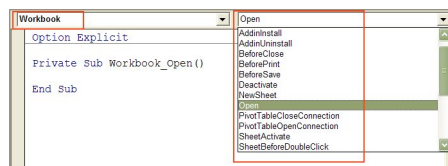
La liste des événements est accessible dans l'éditeur de macros (Alt+F11).

Double cliquez sur le module objet ThisWorkbook.



Puis sélectionnez **Workbook** dans la liste déroulante de gauche.

Vous pouvez maintenant visualiser et choisir un événement dans la liste déroulante de droite.



La liste des événements disponibles est différente en fonction des versions d'Excel.

Utilisez la propriété `EnableEvents` pour désactiver provisoirement les événements. Si vous devez par exemple écrire dans une cellule à partir d'une procédure événementielle qui est déclenchée par la modification des cellules, vous allez provoquer une boucle récursive infinie. La propriété `Application.EnableEvents = False` permet de régler ce problème.

Cet exemple incrémente d'une unité la cellule A1, à chaque modification dans le classeur.

Vba

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    'Désactive l'évènement
    Application.EnableEvents = False
    'modification dans la cellule A1
    Sheets(1).Range("A1") = Sheets(1).Range("A1") + 1
    'Réactive l'évènement
    Application.EnableEvents = True
End Sub
```

II - Les événements

II-1 - Workbook_Activate

```
Private Sub Workbook_Activate()
```

Cet événement est déclenché lorsque le classeur est activé (basculement entre deux classeurs).

La procédure n'est pas lancée si vous revenez d'une autre application alors que le classeur était précédemment actif.

II-2 - Workbook_AddinInstall et Workbook_AddinUninstall

```
Private Sub Workbook_AddinInstall()
```

Cet événement est utilisé dans les macros complémentaires (xla).

Il est déclenché lors de l'installation de la macro complémentaire qui contient cette procédure événementielle.

Nota:

Une macro complémentaire est un fichier Excel, dont les feuilles ne sont pas visibles. Ce type de fichier est principalement conçu pour stocker des macros. Il est possible d'installer une macro complémentaire afin qu'elle se charge automatiquement dès le démarrage d'Excel. Vous disposez ainsi en permanence des macros contenues dans le fichier.

```
Private Sub Workbook_AddinUninstall()
```

Cet événement est utilisé dans les macros complémentaires.

Il est déclenché lorsque la macro complémentaire est désinstallée.

II-3 - Workbook_AfterXmlExport

```
Private Sub Workbook_AfterXmlExport(ByVal Map As XmlMap, ByVal Url As String, ByVal Result As XIXmlExportResult)
```

Est déclenché après l'export des données vers un fichier xml (à partir d'Excel2003).

Le paramètre **Map** représente le mappage xml de la table utilisée pour l'export.

Le paramètre **Url** renvoie le chemin complet du fichier xml enregistré lors de l'export.

Le paramètre **Result** renvoie le résultat de l'export:

- * 0 (xlXmlExportSuccess) L'export a réussi.
- * 1 (xlXmlExportValidationFailed) L'export a échoué.

Vba

```
Private Sub Workbook_AfterXmlExport(ByVal Map As XmlMap, ByVal Url As String, _  
    ByVal Result As xlXmlExportResult)  
  
    MsgBox Map.Name & vbCrLf & _  
        Url & vbCrLf & _  
        Result  
  
End Sub
```

II-4 - Workbook_AfterXmlImport

```
Private Sub Workbook_AfterXmlImport(ByVal Map As XmlMap, ByVal IsRefresh As Boolean, ByVal Result As  
xlXmlImportResult)
```

Est déclenché après l'insertion ou l'actualisation des données xml dans la feuille de calcul (à partir d'Excel2003).

Le paramètre **Map** représente le mappage xml de la table résultant de l'import.

Le paramètre **IsRefresh** permet d'identifier si l'import provient d'une nouvelle source de données ou s'il s'agit de l'actualisation d'un mappage existant dans la feuille. La valeur True est renvoyée s'il s'agit d'une actualisation.

Le paramètre **Result** renvoie le résultat de l'import:

- * 0 (xlXmlImportSuccess) L'import a réussi.
- * 1 (xlXmlImportElementsTruncated) Le contenu du fichier xml a été tronqué lors de l'import car il est trop grand pour la feuille de calcul.
- * 2 (xlXmlImportValidationFailed) L'import a échoué.

Vba

```
Private Sub Workbook_AfterXmlImport(ByVal Map As XmlMap, ByVal IsRefresh As Boolean, _  
    ByVal Result As XmlImportResult)  
  
    MsgBox IsRefresh & vbCrLf & _  
        Map.Name & vbCrLf & _  
        Result  
  
End Sub
```

II-5 - Workbook_BeforeClose

`Private Sub Workbook_BeforeClose(Cancel As Boolean)`

Cet évènement est déclenché avant la fermeture du classeur.

Le classeur se ferme uniquement quand la procédure événementielle est arrivée à terme.

Le paramètre `Cancel = True` empêche la fermeture du fichier.

Cet exemple bloque la fermeture du classeur tant que la cellule A1 est vide.

Vba

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    'Vérifie le contenu de la cellule A1 dans la Feuille  
    If Sheets("Feuille").Range("A1") = "" Then  
        MsgBox "Complétez la cellule A1"  
        'Empêche la fermeture si la cellule A1 est vide  
        Cancel = True  
    Else  
        'Enregistre les modifications avant la fermeture du classeur  
        ThisWorkbook.Save  
    End If  
End Sub
```

Description générale du paramètre CANCEL:

Certains événements (BeforeClose, BeforePrint, BeforeSave, SheetBeforeDoubleClick, SheetBeforeRightClick) disposent de cette option. La valeur par défaut de ce paramètre est égale à False. Si vous souhaitez désactiver l'action associée à une macro événementielle, spécifiez `Cancel = True` dans la procédure.

II-6 - Workbook_BeforePrint

`Private Sub Workbook_BeforePrint(Cancel As Boolean)`

Cet évènement survient avant l'impression. L'impression commence uniquement à l'issue de cette procédure.

Le paramètre `Cancel = True` bloque toute impression.

II-7 - Workbook_BeforeSave

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

Cet évènement est déclenché avant de débiter la sauvegarde du classeur.

Le paramètre `SaveAsUI` renvoie VRAI si la boîte de dialogue "Enregistrer Sous" va être affichée.

La sauvegarde du classeur n'est pas possible si vous spécifiez Le paramètre `Cancel = True`.

Vba

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

Mais vous allez dire, comment sauvegarder un classeur dans lequel on vient de placer une procédure qui empêche l'enregistrement...?

Suivez la description ci dessous:

Placez un point d'arrêt sur la ligne `Private Sub Workbook_BeforeSave`.

Lancez la sauvegarde.

Lorsque l'enregistrement atteint le point d'arrêt, Faites glisser la flèche (qui apparait sur la gauche de la ligne surlignée en jaune) vers le bas jusqu'à dépasser la ligne `Cancel = True`.

Appuyez sur le touche raccourci F8 pour continuer l'enregistrement.

II-8 - Workbook_BeforeXmlExport

```
Private Sub Workbook_BeforeXmlExport(ByVal Map As XmlMap, ByVal Url As String, Cancel As Boolean)
```

Est déclenché avant l'export des données xml (à partir d'Excel2003).

Le paramètre `Map` représente le mappage xml de la table utilisée pour l'export.

Le paramètre **Url** renvoie le chemin complet du fichier xml qui va être utilisé pour l'export.

Spécifiez **Cancel = True** pour empêcher l'export.

Vba

```
Private Sub Workbook_BeforeXmlExport(ByVal Map As XmlMap, ByVal Url As String, _  
    Cancel As Boolean)  
  
    MsgBox Map.Name & vbCrLf & Url  
End Sub
```

II-9 - Workbook_BeforeXmlImport

Private Sub Workbook_BeforeXmlImport(ByVal Map As XmlMap, ByVal Url As String, ByVal IsRefresh As Boolean, Cancel As Boolean)

Est déclenché avant l'insertion ou l'actualisation des données xml dans la feuille de calcul (à partir d'Excel2003).

Le paramètre **Map** représente le mappage xml des données importées.

Le paramètre **Url** renvoie le chemin complet du fichier xml importé.

Le paramètre **IsRefresh** permet d'identifier si l'import provient d'une nouvelle source de données ou s'il s'agit de l'actualisation d'un mappage existant dans la feuille. La valeur True est renvoyée s'il s'agit d'une actualisation.

Spécifiez **Cancel = True** pour empêcher l'import.

Vba

```
Private Sub Workbook_BeforeXmlImport(ByVal Map As XmlMap, ByVal Url As String, _  
    ByVal IsRefresh As Boolean, Cancel As Boolean)  
  
    MsgBox IsRefresh & vbCrLf & _  
        Map.Name & vbCrLf & _  
        Url  
End Sub
```

II-10 - Workbook_Deactivate

`Private Sub Workbook_Deactivate()`

Cet événement est déclenché lorsque vous sélectionnez un autre classeur.

La procédure n'est pas lancée si vous sélectionnez une autre application.

II-11 - Workbook_NewSheet

`Private Sub Workbook_NewSheet(ByVal Sh As Object)`

Cet événement est déclenché lorsqu'une nouvelle feuille est insérée dans le classeur.

Le paramètre `Sh` correspond à l'objet feuille créé (feuille de calcul ou feuille graphique).

Cet exemple affiche le nom et l'index de la nouvelle feuille.

Vba

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    MsgBox Sh.Name & " : " & Sh.Index

    'Exemple pour affecter un nouveau nom à la feuille créée
    'Sh.Name = "Nouveau nom"
End Sub
```

II-12 - Workbook_Open

`Private Sub Workbook_Open()`

Cet événement permet de déclencher une procédure lors de l'ouverture du classeur.

Voici un exemple de message d'accueil qui s'affiche à l'ouverture d'un classeur:

Vba

```
Private Sub Workbook_Open()
    MsgBox "Bonjour " & Environ("UserName")
End Sub
```

Nota important:

Il est conseillé de paramétrer la sécurité des macros au niveau moyen, pour que les macros ne se déclenchent pas sans autorisation de l'utilisateur. Quand vous ouvrez un classeur émanant d'une personne que vous ne connaissez pas, il est toujours préférable d'ouvrir une première fois le fichier en désactivant les macros afin de vérifier que celui-ci ne contient pas des procédures effectuant des actions indésirables.

Pour paramétrer le niveau de sécurité des macros:

Menu Outils

Macro

Sécurité

Onglet niveau de sécurité

Sélectionnez l'option "Niveau de sécurité moyen."

Pour ne pas déclencher la procédure événementielle lors de l'ouverture du classeur:

Si vous ouvrez le classeur manuellement, cliquez sur le bouton "Désactivez les macros" dans la boîte d'alerte des macros qui s'affiche avant l'ouverture.

Si vous ouvrez le classeur par macro, utilisez la propriété `EnableEvents` pour désactiver la procédure `Workbook_Open`.

Vba

```
Application.EnableEvents = False
Workbooks.Open Filename:="C:\LeClasseur.xls"
Application.EnableEvents = True
```

II-13 - Workbook_PivotTableCloseConnection

`Private Sub Workbook_PivotTableCloseConnection(ByVal Target As PivotTable)`

L'évènement est déclenché lorsqu'un tableau croisé dynamique se déconnecte de sa source de données.

Le paramètre `Target` correspond au TCD déconnecté.

II-14 - Workbook_PivotTableOpenConnection

`Private Sub Workbook_PivotTableOpenConnection(ByVal Target As PivotTable)`

L'évènement est déclenché lorsqu'un tableau croisé dynamique se connecte à une source de données.

Le paramètre **Target** correspond au TCD connecté.

II-15 - Workbook_RowsetComplete

`Private Sub Workbook_RowsetComplete(ByVal Description As String, ByVal Sheet As String, ByVal Success As Boolean)`

Survient lorsque l'utilisateur étudie le jeu d'enregistrements ou appelle l'action **rowset** pour un objet **PivotTable OLAP** (à partir d'Excel2007).


Le paramètre **Description** renvoie une brève description de l'évènement.

Le paramètre **Sheet** renvoie le nom de la feuille qui contient le Recordset créé.

II-16 - Workbook_SheetActivate

`Private Sub Workbook_SheetActivate(ByVal Sh As Object)`

Identifie l'activation d'un onglet dans le classeur. Le paramètre **Sh** correspond à l'onglet sélectionné.

 *Tous les évènements dont le nom commence par **Workbook_Sheet** ont leur équivalent dans chaque module feuille. Le principe de fonctionnement est identique.*

Utiliser la procédure au niveau du module objet ThisWorkbook a pour avantage d'écrire une seule procédure événementielle pour gérer l'ensemble des onglets dans le classeur.

Cet exemple renvoie le nom de la feuille sélectionnée.

Vba

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    MsgBox Sh.Name
End Sub
```

II-17 - Workbook_SheetBeforeDoubleClick

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

Identifie le double clic dans une cellule.

Cet évènement est déclenché à la suite de `WorkSheet_BeforeDoubleClick`.

Le paramètre `Sh` correspond à l'onglet actif.

Le paramètre `Target` correspond à la cellule qui reçoit le double clic.

Le paramètre `Cancel` désactive l'action associée à une macro événementielle.

II-18 - `Workbook_SheetBeforeRightClick`

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

Cet évènement survient lors de l'utilisation du clic droit dans une des feuilles du classeur.

Cet évènement est déclenché à la suite de `WorkSheet_BeforeRightClick`.

Le paramètre `Sh` correspond à l'onglet actif.

Le paramètre `Target` correspond à la cellule qui reçoit le clic droit.

Le paramètre `Cancel` désactive l'action associée à une macro événementielle.

Un exemple qui désactive l'affichage du menu contextuel dans toutes les feuilles du classeur, et le remplace par la palette de couleurs afin de colorier la ou les cellules sélectionnées.

Vba

```
Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
    Dim x As Long
    Dim lngColorIndex As Long

    'Désactive le menu contextuel
    Cancel = True

    'Affiche la palette couleur
    lngColorIndex = Application.Dialogs(xlDialogPatterns).Show

    'Récupère la couleur de cellule actuelle
    x = ActiveCell.Interior.ColorIndex

    If lngColorIndex = xlColorIndexAutomatic Then x = xlColorIndexNone
    ActiveCell.Interior.ColorIndex = x

End Sub
```

II-19 - Workbook_SheetCalculate

`Private Sub Workbook_SheetCalculate(ByVal Sh As Object)`

Cet événement est déclenché lors du recalcul (validation ou mise à jour des formules) dans une des feuilles du classeur.

L'évènement survient à la suite de `Worksheet_Calculate`.

Le paramètre `Sh` correspond à l'onglet contenant la formule.

II-20 - Workbook_SheetChange

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

Cet événement est déclenché lorsqu'une des cellules du classeur est modifiée.

Le paramètre `Sh` correspond à l'onglet contenant la cellule modifiée.

Le paramètre `Target` correspond à la cellule modifiée.

Cet exemple permet d'identifier la cellule que vous venez de modifier.

Vba

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

    'permet de sortir de la procédure si plus d'une cellule est sélectionnée
    '(sinon la suite de la macro renvoie un message d'erreur)
    If Target.Count > 1 Then Exit Sub

    MsgBox "Vous venez de modifier la cellule " & Target.Address & _
        " (" & Target.Value & ")" & _
        " dans la feuille nommée " & Sh.Name

End Sub
```

II-21 - Workbook_SheetDeactivate

`Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)`

Survient lorsqu'une feuille du classeur contenant la procédure événementielle est désactivée.

Le paramètre **Sh** correspond à l'onglet désactivé.

II-22 - Workbook_SheetFollowHyperlink

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

L'évènement survient lorsqu'un lien hypertexte est déclenché.

Le paramètre **Sh** correspond à l'onglet contenant le lien.

Le paramètre **Target** correspond à l'objet lien hypertexte que vous déclenchez.

Un exemple qui affiche l'adresse du lien déclenché.

Vba

```
Private Sub Workbook_SheetFollowHyperlink(ByVal Sh As Object, ByVal Target As Hyperlink)
    MsgBox Target.Address & vbCrLf & Target.SubAddress
End Sub
```

Nota:

Il semblerait que l'évènement n'est pas déclenché lorsque le lien est issu de la formule **LIEN_HYPERTEXTE**.

```
=LIEN_HYPERTEXTE("[nomClasseur.xls]Feuil2!D13";"La description")
```

II-23 - Workbook_SheetPivotTableUpdate

```
Private Sub Workbook_SheetPivotTableUpdate(ByVal Sh As Object, ByVal Target As PivotTable)
```

Cet évènement survient lors de la mise à jour d'un tableau croisé dynamique.

Le paramètre **Sh** correspond à l'onglet contenant le TCD.

Le paramètre **Target** correspond à l'objet Tableau croisé dynamique mis à jour.

Cet exemple permet d'identifier quel TCD est mis à jour.

Vba

```
Private Sub Workbook_SheetPivotTableUpdate(ByVal Sh As Object, ByVal Target As PivotTable)
    MsgBox "Mise à jour " & Sh.Name & " / " & Target.Name
End Sub
```

II-24 - Workbook_SheetSelectionChange

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

Cet événement survient lors de la sélection d'une cellule dans une des feuilles du classeur.

Il est déclenché à la suite de `WorkSheet_SelectionChange`.

Le paramètre `Sh` correspond à l'onglet actif.

Le paramètre `Target` correspond à la cellule sélectionnée.

Cet exemple permet de récupérer le nom de la feuille et l'adresse de la cellule (ou de la plage de cellules) que vous venez de sélectionner.

Vba

```
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)
    MsgBox "Vous avez sélectionné la cellule " & Target.Address & _
        " dans la feuille nommée " & Sh.Name
End Sub
```

II-25 - Workbook_Sync

`Private Sub Workbook_Sync(ByVal SyncEventType As Office.MsoSyncEventType)`

Est déclenché lorsque la copie locale d'une feuille de calcul qui fait partie de l'espace de travail d'un document est synchronisée avec la copie sur le serveur (à partir d'Excel2003).

Le paramètre `SyncEventType` peut prendre une des valeurs suivantes:

- * `msoSyncEventDownloadFailed`
- * `msoSyncEventDownloadInitiated`
- * `msoSyncEventDownloadNoChange`
- * `msoSyncEventDownloadSucceeded`
- * `msoSyncEventOffline`
- * `msoSyncEventUploadFailed`

- * msoSyncEventUploadInitiated
- * msoSyncEventUploadSucceeded

Vba

```
'Renvoie un message d'alerte si la synchronisation échoue.  
Private Sub Workbook_Sync(ByVal SyncEventType As Office.MsoSyncEventType)  
  
    If SyncEventType = msoSyncEventDownloadFailed Or _  
        SyncEventType = msoSyncEventUploadFailed Then _  
        MsgBox "La synchronisation a échoué."  
  
End Sub
```

II-26 - Workbook_WindowActivate

```
Private Sub Workbook_WindowActivate(ByVal Wn As Window)
```

Cet évènement survient lorsque vous activez la fenêtre qui contient le classeur.

Le paramètre **Wn** correspond à la fenêtre activée.

II-27 - Workbook_WindowDeactivate

```
Private Sub Workbook_WindowDeactivate(ByVal Wn As Window)
```

Cet évènement survient lorsque vous désactivez la fenêtre qui contient le classeur pour activer une autre fenêtre Excel.

Le paramètre **Wn** correspond à la fenêtre désactivée.

II-28 - Workbook_WindowResize

```
Private Sub Workbook_WindowResize(ByVal Wn As Window)
```

Cet évènement survient lors du redimensionnement de la fenêtre qui contient le classeur.

Le paramètre **Wn** correspond à cette fenêtre.

III - Téléchargement

