



## Gli eventi della cartella di lavoro o ThisWorkbook

Per funzionare appropriatamente, la maggior parte dei programmi di Excel, VBA modifica in qualche modo l'ambiente dell'applicazione stessa, queste modifiche possono comportare l'aggiunta di comandi di menu, la visualizzazione di barre degli strumenti o l'impostazione di opzioni. Nella maggior parte dei casi, tutti questi cambiamenti devono essere apportati in Excel prima che l'utente avvii il programma di VBA, in modo che i comandi di menu e le barre degli strumenti siano disponibili fin dall'inizio.

Un programma di VBA non dovrebbe obbligare l'utente ad aggiungere menu e barre degli strumenti e a modificare le impostazioni delle opzioni, pertanto dobbiamo fare in modo che questi cambiamenti si verifichino automaticamente all'avvio usando una routine di gestione dell'evento **Open** della cartella di lavoro. L'evento Open viene generato quando viene aperta una cartella di lavoro e il codice di una routine Open, viene eseguito ogni volta che la cartella di lavoro viene aperta e rappresenta uno strumento ideale per impostare le eventuali condizioni speciali necessarie per la cartella stessa. Per poter accedere agli eventi della cartella si deve agire in questo modo:

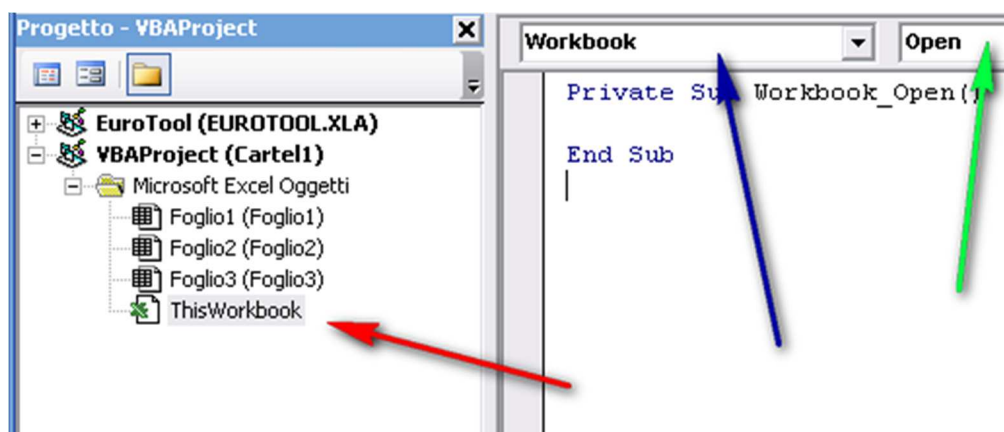


Fig. 1

Selezionare la voce *ThisWorkbook* nella finestra del progetto, indicato dalla *freccia rossa* in *figura 2*, per accedere al modulo di classe della cartella di lavoro (Workbook) di seguito selezionare Workbook nella casella a discesa sopra la finestra del codice, indicata dalla *freccia blu* e nella casella a fianco, indicata dalla *freccia verde* compariranno i vari eventi per l'oggetto. Di seguito riportiamo gli eventi principali associati all'oggetto Workbook:

### ■ Private Sub Workbook\_Open ()

Questo evento può innescare una procedura quando si apre la cartella di lavoro

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

```
Private Sub Workbook_Open()  
    MsgBox "Benvenuto"  
End Sub
```



*Si consiglia di impostare il livello di protezione macro sul livello **medio**, in modo che le macro non vengano attivate senza l'autorizzazione dell'utente, inoltre quando si apre una cartella di lavoro proveniente da una persona che non si conosce, è meglio aprire il file disattivando le macro per verificare che non contengano procedure che svolgono azioni indesiderate.*



### ■ Private Sub Workbook\_Activate()

Questo evento viene innescato quando viene attivata la cartella di lavoro. La procedura non viene avviata se si proviene da un'altra applicazione, mentre la cartella di lavoro era attiva

```
Private Sub Workbook_Activate()  
    'nascondere la barra della formula  
    Application.DisplayFormulaBar = False  
End Sub
```

```
Private Sub Workbook_Activate()  
    'mostra la userform1  
    Userform1.Show  
End Sub
```

### ■ Private Sub Workbook\_Deactivate ()

Questo evento viene attivato quando si seleziona un'altra cartella di lavoro.

```
Private Sub Workbook_Deactivate()  
    'attiva la barra della formula  
    Application.DisplayFormulaBar = True
```

### ■ Private Sub Workbook\_BeforeClose (Cancel As Boolean)

Questo evento viene generato prima di chiudere la cartella di lavoro, che si chiude solo quando l'evento è terminato. Il parametro Cancel impedisce la chiusura del file.

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    'blocca la chiusura della cartella di lavoro finchè la cella A1 è vuota.  
    If Sheets("Foglio1").Range("A1") = "" Then  
        MsgBox "Completare l'inserimento in A1"  
        Cancel = True  
    Else  
        ThisWorkbook.Save  
    End If  
End Sub  
  
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    If MsgBox("Vuoi veramente chiudere la cartella di lavoro?", 36, "Conferma") =  
vbNo Then  
        Cancel = True  
    End If  
End Sub  
  
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    'salvare data e ora nel foglio1  
    Worksheets("Foglio1").Range("A1").Value = _  
        Format(Date + Time, "dd/mm/yy hh:mm")  
    ThisWorkbook.Save  
End Sub
```



#### ■ **Private Sub Workbook\_BeforePrint** (Cancel As Boolean)

Questo evento si verifica prima della stampa che inizia solo dopo questa procedura, il parametro Cancel blocca l'esecuzione della macro

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)
    `ricalcola tutti i fogli della cartella di lavoro attiva prima di stampare
    For Each wk in Worksheets
        wk.Calculate
    Next
End Sub

Private Sub Workbook_BeforePrint(Cancel As Boolean)
    ActiveSheet.PageSetup.PrintArea = Selection
End Sub

Private Sub Workbook_BeforePrint(Cancel As Boolean)
    Cancel = false
    Worksheets("Foglio1").PageSetup.RightFooter = "Produced by " & myname
End Sub
```

#### ■ **Private Sub Workbook\_BeforeSave** (ByVal SaveAsUI As Boolean, Cancel As Boolean)

Questo evento viene generato prima di avviare il lavoro di backup, il parametro SaveAsUI restituisce TRUE se la casella "Salva con nome" verrà visualizzata, se si specifica il parametro Cancel = True verrà bloccata l'operazione di salvataggio

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

Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel as Boolean)
    a = MsgBox("Vuoi salvare la cartella corrente?", vbYesNo)
    If a = vbNo Then Cancel = True
End Sub

Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
    MsgBox "Arrivederci cartella salvata con successo"
End Sub
```



### ■ **Private Sub Workbook\_NewSheet** (ByVal Sh As Object)

Questo evento viene attivato quando un nuovo foglio viene inserito nella cartella di lavoro, il parametro **Sh** è il foglio di lavoro creato

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    'visualizza il nome e l'indice del nuovo foglio.
    MsgBox Sh.Name & " : " & Sh.Index
End Sub
```

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    'sposta nuovi fogli alla fine del lavoro
    Sh.Move After:= Sheets(Sheets.Count)
End Sub
```

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    Sheet1.UsedRange.Copy Sh.UsedRange
End Sub
```

### ■ **Private Sub Workbook\_WindowActivate** (ByVal Wn As Window)

Questo evento si verifica quando si attiva la finestra che contiene la cartella di lavoro e il parametro Wn corrisponde alla finestra attiva

```
Private Sub Workbook_WindowActivate(ByVal Wn As Excel.Window)
    Wn.WindowState = xlMaximized
End Sub
```

```
Private Sub Workbook_WindowActivate(ByVal Wn As Window)
    MsgBox "Attivato"
End Sub
```

```
Private Sub Workbook_WindowActivate(ByVal Wn As Window)
    MsgBox "Questa cartella è attiva"
End Sub
```

### ■ **Private Sub Workbook\_WindowDeactivate** (ByVal Wn As Window)

Questo evento si verifica quando si disattiva la finestra che contiene la cartella di lavoro per attivare un'altra finestra di Excel e il parametro Wn corrisponde alla finestra disabilitata.

```
Private Sub Workbook_WindowDeactivate(ByVal Wn As Window)
    MsgBox "Disattivato"
End Sub
```

```
Private Sub Workbook_WindowDeactivate(ByVal Wn As Window)
    'nasconde la barra multifunzione se l'altezza è inferiore a 100
    altezz = Application.CommandBars("Ribbon").Height
    If altezz < 100 Then Application.SendKeys ("^{F1}")
End Sub
```

```
Private Sub Workbook_WindowDeactivate(ByVal Wn As Window)
    MsgBox "Hai disattivato " & Wn.Caption
End Sub
```



### ■ **Private Sub Workbook\_WindowResize** (ByVal Wn As Window)

Questo evento si verifica quando si ridimensiona la finestra che contiene la cartella di lavoro e il parametro **Wn** corrisponde alla finestra.

```
Private Sub Workbook_WindowResize(ByVal Wn As Excel.Window)
    'Inserisce nella barra di stato il nome del file e la scritta "ridimensionata"
    Application.StatusBar = Wn.Caption & " Ridimensionata"
End Sub
```

```
Private Sub Workbook_WindowResize(ByVal Wn As Window)
    'ridimensionando la finestra inserisce il valore in B1
    With ThisWorkbook.ActiveSheet.Range("B1")
        .Value = .Value + 1
    End With
End Sub
```

```
Private Sub Workbook_WindowResize(ByVal Wn As Window)
    'parcheggia la finestra se viene ridimensionata
    Wn.WindowState = xlMinimized
End Sub
```

### ■ **Private Sub Workbook\_AddinInstall** ()

Questo evento viene utilizzato nei componenti aggiuntivi (XLA) e viene attivato durante l'installazione del componente aggiuntivo contenuto nella procedura evento.

```
Private Sub Workbook_AddinInstall()
    'aggiunge un controllo alla barra degli strumenti standard
    With Application.CommandBars("Standard").Controls.Add
        .Caption = "Aggiungi Voci al menu"
        .OnAction = "'pippo11.xls'!Amacro"
    End With
End Sub
```

```
Private Sub Workbook_AddinInstall()
    Dim MyForm As Object
    Set MyForm =
    ActiveWorkbook.VBProject.VBComponents.Add(vbext_ct_MSForm)
    'nuova userform
    With MyForm
        .Properties("Caption") = "My Form"
        .Properties("Width") = 450
        .Properties("Height") = 300
    End With
End Sub
```

```
Private Sub Workbook_AddinInstall()
    Run "AddMenus"
End Sub
```



### ■ **Private Sub Workbook\_AddinUninstall ()**

Questo evento viene utilizzato nei Componenti aggiuntivi e viene attivata quando il componente aggiuntivo viene disinstallato.

```
Private Sub Workbook_AddinUninstall()  
    Run "DeleteMenu"  
End Sub  
  
Private Sub Workbook_AddinUninstall()  
    Application.WindowState = xlMinimized  
End Sub  
  
Private Sub Workbook_AddinUninstall()  
    For Each Component In ActiveWorkbook.VBProject.VBComponents  
        If Component.Type = 3 Then  
            ActiveWorkbook.VBProject.VBComponents.Remove Component  
        End If  
    Next Component  
End Sub
```

### ■ **Private Sub Workbook\_BeforeXmlExport** (ByVal Map As XmlMap, ByVal Url As String, Cancel As Boolean)

Questo evento si attiva quando Excel salva o esporta i dati XML da questa cartella di lavoro specifica, il parametro **Map** rappresenta la tabella xml di mappatura utilizzata per l'esportazione e il parametro **Url** restituisce il percorso completo del file xml che verrà utilizzato per l'esportazione. Si specifica **Cancel = True** per impedire l'esportazione.

```
Private Sub Workbook_BeforeXmlExport(ByVal Map As XmlMap, ByVal Url As  
String, _  
    Cancel As Boolean)  
    MsgBox Map.Name & vbCrLf & Url  
End Sub  
  
Private Sub Workbook_BeforeXmlExport(ByVal Map As XmlMap, ByVal Url As  
String, _  
    Cancel As Boolean)  
    If (Map.IsExportable) Then  
        If MsgBox("Microsoft Excel è pronto" & " per esportare in XML da " &  
"Map.Name" & "." & vbCrLf & _  
" Vuoi continuare?", vbYesNo + vbQuestion, "Processo di esportazione in XML")  
= 7 Then Cancel = True  
    End If  
End Sub  
  
Private Sub Workbook_BeforeXmlExport(ByVal Map As XmlMap, _ ByVal Url As  
String, Cancel As Boolean)  
    Debug.Print "Verifica prima di esportare", Map, Url, IsRefresh, Cancel  
End Sub
```





■ **Private Sub Workbook\_AfterXmlExport** (ByVal Map As XmlMap, ByVal Url As String, ByVal Result As XIXmlExportResult)

Questo evento viene attivato dopo l'esportazione dei dati in un file xml (da Excel2003), il parametro Map rappresenta la tabella xml di mappatura utilizzata per l'esportazione il parametro URL restituisce il percorso completo del file xml salvato durante l'esportazione, mentre invece il parametro Result restituisce il risultato dell'esportazione che può essere:

- 0 (*xlXmlExportSuccess*): L'esportazione è stata eseguita correttamente oppure
- 1 (*xlXmlExportValidationFailed*): L'esportazione non riuscita

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

■ **Private Sub Workbook\_BeforeXmlImport** (ByVal Map As XmlMap, ByVal Url As String, ByVal IsRefresh As Boolean, Cancel As Boolean)

Questo evento si attiva quando si importano dei dati da un file XML nel foglio di lavoro. Il parametro Map rappresenta i dati di mapping che importano il file XML e il parametro URL restituisce il percorso completo del file XML importato. Il parametro *IsRefresh* identifica se l'importazione è di una nuova origine dati o se è un aggiornamento esistente nel foglio. Se viene restituito True è un aggiornamento, si specifica Cancel = True per impedire l'importazione.

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

```
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
```

```
Private Sub Workbook_BeforeXmlImport(ByVal Map As XmlMap, ByVal Url As String, ByVal IsRefresh As Boolean, Cancel As Boolean)
    Debug.Print "Verifica prima di importare", Map, Url, IsRefresh, Cancel
End Sub
```

```
Private Sub Workbook_BeforeXmlImport(ByVal Map As XmlMap, ByVal Url As String, ByVal IsRefresh As Boolean, Cancel As Boolean)
    If Map.RootElementName = "Modello_pippo" Then
        MsgBox "Il file XML che hai selezionato non può essere importato"
        Cancel = True
    End If
End Sub
```



■ **Private Sub Workbook\_AfterXmlImport** (ByVal Map As XmlMap, ByVal IsRefresh As boolean, ByVal result As XIXmlImportResult)

Questo evento viene attivato dopo l'inserimento o l'aggiornamento dei dati XML nel foglio di lavoro e il parametro Map rappresenta la tabella xml di mappatura derivante dall'importazione e il parametro IsRefresh identifica se l'importazione è di una nuova origine dati o se si sta aggiornando un foglio di lavoro esistente, e viene restituito True se si tratta di un aggiornamento. Il parametro result restituisce il risultato dell'importazione, che può essere:

- 0 (*xlXmlImportSuccess*): L'importazione è riuscita
- 1 (*xlXmlImportElementsTruncated*): Il contenuto del file xml è stato troncato durante l'importazione perché è troppo grande per il foglio di lavoro.
- 2 (*xlXmlImportValidationFailed*): L'importazione non è riuscita

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

■ **Private Sub Workbook\_PivotTableOpenConnection** (ByVal Target As pivot)

L'evento è attivato quando una tabella pivot si connette a un'origine dati e il parametro Target è corrisponde alla tabella

```
Private Sub Workbook_PivotTableOpenConnection(ByVal Target As PivotTable)  
    MsgBox "La connessione alla tabella Pivot è stata aperta"  
End Sub
```

```
Private Sub Workbook_PivotTableOpenConnection(ByVal Target As PivotTable)  
    Application.DisplayAlerts = False  
End Sub
```

■ **Private Sub Workbook\_PivotTableCloseConnection** (ByVal Target As pivot)

L'evento è attivato quando una tabella pivot è scollegata dalla sorgente dati e il parametro Target corrisponde alla tabella offline.

■ **Private Sub Workbook\_RowsetComplete** (ByVal Description As String, ByVal Sheet As String, ByVal Success As Boolean)

Si verifica quando l'utente esamina il recordset o chiama il set di righe per un oggetto tabella pivot (da Excel2007). Il parametro *Description* restituisce una breve descrizione dell'evento e il parametro Sheet restituisce il nome del foglio che contiene il Recordset creato.





### ■ **Private Sub Workbook\_Sync** (ByVal SyncEventType As Office.MsoSyncEventType)

Viene generato quando la copia in locale di un foglio di lavoro che fa parte del lavoro di un documento è sincronizzato con la copia sul server (da Excel2003) Il parametro SyncEventType può assumere i seguenti valori:

- *msoSyncEventDownloadFailed*
- *msoSyncEventDownloadInitiated*
- *msoSyncEventDownloadNoChange*
- *msoSyncEventDownloadSucceeded*
- *msoSyncEventOffline*
- *msoSyncEventUploadFailed*
- *msoSyncEventUploadInitiated*
- *msoSyncEventUploadSucceeded*

```
Private Sub Workbook_Sync(ByVal SyncEventType As Office.MsoSyncEventType)
    If SyncEventType = msoSyncEventDownloadFailed Or _
        SyncEventType = msoSyncEventUploadFailed Then _
        MsgBox "La sincronizzazione è fallita"
End Sub
```



*Tutti gli eventi il cui nome inizia con **Workbook\_Sheet** hanno il loro equivalente in ogni modulo foglio e il principio di funzionamento è identico. Si utilizza la procedura nel modulo **ThisWorkbook** quando si necessita di scrivere un unico processo per la gestione di tutti i fogli della cartella di lavoro.*

### ■ **Private Sub Workbook\_SheetActivate** (ByVal Sh As Object)

Identifica l'attivazione di un foglio nella cartella di lavoro e il parametro Sh corrisponde al foglio selezionato

```
Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    `restituisce il nome del foglio selezionato.
    MsgBox Sh.Name
End Sub

Private Sub Workbook_SheetActivate(ByVal Sh As Object)
    MsgBox "Nome del Foglio : " & Sh.Name
End Sub

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



### ■ **Private Sub Workbook\_SheetDeactivate** (ByVal Sh As Object)

Si verifica quando un foglio nella cartella di lavoro viene disattivato e il parametro Sh corrisponde al foglio disattivato

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

```
Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)
    MsgBox ("Foglio Disattivato")
End Sub
```

```
Private Sub Workbook_SheetDeactivate(ByVal Sh As Object)
    `parcheggiare la finestra di excel
    ThisWorkbook.Windows(1).WindowState = xlMinimized
End Sub
```

### ■ **Private Sub Workbook\_SheetChange** (ByVal Sh As Object, ByVal Target As Range)

Questo evento viene attivato quando avvengono dei cambiamenti in una cella della cartella di lavoro, il parametro Sh corrisponde al foglio contenente la cella modificata e il parametro Target corrisponde alla cella modificata. Questo esempio identifica la cella che avete appena modificato.

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    `identifica la cella modificata
    If Target.Count > 1 Then Exit Sub
    MsgBox "Hai modificato la cella " & Target.Address & _
        " (" & Target.Value & ")" & _
        " Nel foglio denominato " & Sh.Name
End Sub
```

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    `incrementare la cella A1
    Application.EnableEvents = False
    Sheets(1).Range("A1") = Sheets(1).Range("A1") + 1
    Application.EnableEvents = True
End Sub
```

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    `inserire il carattere grassetto e il colore rosso nella cella modificata
    Target.Font.Bold = True
    Target.Font.Color = vbRed
End Sub
```



■ **Private Sub Workbook\_SheetBeforeRightClick** (ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)

Questo evento si verifica quando si utilizza il pulsante destro del mouse in uno dei fogli della cartella di lavoro e viene attivato dopo *Worksheet\_BeforeRightClick*. Il parametro Sh corrisponde al foglio attivo e il parametro Target è la cella che riceve il tasto destro del mouse, mentre il parametro Cancel disabilita l'azione associata della macro.

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

```
    'disabilita la visualizzazione del menu contestuale in tutti i fogli della cartella di lavoro
```

```
    Dim x As Long, lngColorIndex As Long
```

```
    Cancel = True
```

```
    lngColorIndex = Application.Dialogs(xlDialogPatterns).Show
```

```
    x = ActiveCell.Interior.ColorIndex
```

```
    If lngColorIndex = xlColorIndexAutomatic Then x = xlColorIndexNone
```

```
    ActiveCell.Interior.ColorIndex = x
```

```
End Sub
```

■ **Private Sub Workbook\_SheetBeforeDoubleClick** (ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)

Identifica il doppio click in una cella. Questo evento viene attivato dopo *Worksheet\_BeforeDoubleClick*. Parametro Sh corrisponde alla scheda attiva e il parametro Target è la cella che riceve un doppio clic. il parametro Cancel disabilita l'azione della macro associata

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

```
    If Sh.Name = "Foglio1" Then
```

```
        Target.Interior.Color = RGB(255, 108, 0) 'Arancione
```

```
    Else
```

```
        Target.Interior.Color = RGB(136, 255, 0) 'Verde
```

```
    End If
```

```
End Sub
```

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

```
    'disabilitare il doppio click
```

```
    Cancel = True
```

```
End Sub
```

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

```
    MsgBox "Hai fatto doppio click nel Foglio: " & Sh.Name & vbCrLf & "e nella Cella:  
    " & Target.Address
```

```
End Sub
```



#### ■ **Private Sub Workbook\_SheetCalculate** (ByVal Sh As Object)

Questo evento viene attivato quando si esegue il ricalcolo (formule di convalida o di aggiornamento) in uno dei fogli della cartella di lavoro. L'evento si verifica dopo [Worksheet\\_Calculate](#) e il parametro Sh corrisponde al foglio che contiene la formula.

#### ■ **Private Sub Workbook\_SheetFollowHyperlink** (ByVal Sh As Object, ByVal Target As Hyperlink)

L'evento si verifica quando viene attivato un collegamento ipertestuale e il parametro Sh corrisponde al foglio che contiene il collegamento, mentre il parametro Target è il collegamento ipertestuale dell'oggetto

```
Private Sub Workbook_SheetFollowHyperlink(ByVal Sh As Object, ByVal Target
As Hyperlink)
    'mostra l'indirizzo del collegamento attivato
    MsgBox Target.Address & vbCrLf & Target.SubAddress
End Sub
```

#### ■ **Private Sub Workbook\_SheetPivotTableUpdate** (ByVal Sh As Object, ByVal Target As pivot)

Questo evento si verifica quando si aggiorna una tabella pivot, il parametro Sh corrisponde a foglio che contiene la tabella e il parametro Target è l'oggetto Tabella pivot aggiornato.

```
Private Sub Workbook_SheetPivotTableUpdate(ByVal Sh As Object, ByVal Target
As PivotTable)
    MsgBox "E' stata aggiornata " & Sh.Name & " / " & Target.Name
End Sub
```

#### ■ **Private Sub Workbook\_SheetSelectionChange** (ByVal Sh As Object, ByVal Target As Range)

Questo evento si verifica quando viene modificata la selezione di una cella in uno dei fogli della cartella di lavoro e viene attivato dopo [Worksheet\\_SelectionChange](#). Il parametro Sh è il foglio attivo e il parametro Target corrisponde alla cella selezionata.

```
Private Sub Workbook_SheetSelectionChange(ByVal Sh As Object, ByVal Target
As Range)
    'recuperare il nome del foglio e l'indirizzo della cella
    MsgBox "Hai selezionato la cella " & Target.Address & _
        " nel foglio denominato " & Sh.Name
End Sub
```