



Il Debug in Visual Basic Editor

Il **Debug** è un aspetto molto importante in programmazione e gli sviluppatori hanno bisogno di identificare e rettificare rapidamente gli errori durante la scrittura del codice. VBA fornisce numerosi strumenti di debug per risolvere i problemi nella fase di sviluppo, e oltre alla possibilità di aggiungere le routine di gestione degli errori, fornisce soluzioni rapide quando si manifestano gli errori durante l'esecuzione.

■ Indentazione del codice

Per poter facilitare la lettura e la comprensione del codice è buona cosa ricorrere all'indentazione del codice che consiste nel precedere le righe di codice con un certo numero di spazi e ha lo scopo di evidenziare i blocchi di codice per permettere di cogliere visivamente la struttura del programma, inoltre si deve indentare il codice mentre lo si sviluppa, non successivamente per renderlo "bello". Il numero di spazi è sempre multiplo di un certo valore scelto come base (in genere 3 o 4) e normalmente si può definire il tasto **Tab** tramite il menu **Strumenti – Opzioni** dell'Editor di VBE, in modo che introduca quel numero di spazi

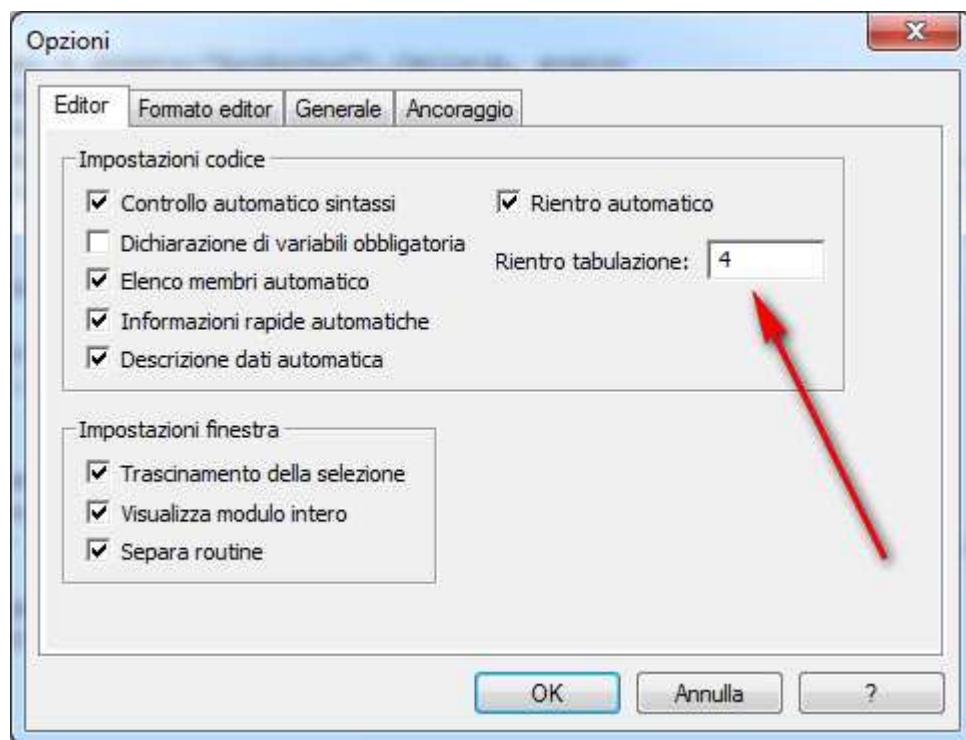


Fig. 1

Indentare il codice è un requisito non obbligatorio e viene spesso messo in secondo piano, tuttavia, è fondamentale per incrementare la leggibilità del codice, in particolare delle strutture di controllo come le condizioni o i loop allo scopo di separare più chiaramente le istruzioni e, in particolare, di rappresentare esplicitamente le relazioni di annidamento e viene considerata come una norma fondamentale di buona programmazione.



Esempio codice NON Indentato

```
Private Sub UserForm_Activate()  
Dim elemento As String  
If ListBox1.ListCount >= 1 Then ListBox1.Clear  
i = 1  
Do Until Sheets("Archivio").Cells(1, i).Value = Empty  
With Sheets("Archivio")  
    elemento = .Cells(1, i).Value  
End With  
ListBox1.AddItem elemento  
i = i + 8  
Loop  
ListBox1.ListIndex = 0  
svuota_box  
End Sub
```

Esempio codice Indentato

```
Private Sub UserForm_Activate()  
    Dim elemento As String  
    If ListBox1.ListCount >= 1 Then ListBox1.Clear  
    i = 1  
    Do Until Sheets("Archivio").Cells(1, i).Value = Empty  
        With Sheets("Archivio")  
            elemento = .Cells(1, i).Value  
        End With  
        ListBox1.AddItem elemento  
        i = i + 8  
    Loop  
    ListBox1.ListIndex = 0  
    svuota_box  
End Sub
```

Per indentare il codice è possibile usare un Add-In che potete trovare [a questo link](#).



Non è ancora stata rilasciata una versione che supporti Office 64bit, per cui la versione 2013 non viene supportata, ma è possibile usare un indentatore on-line [a questo Link](#)



■ Debug utilizzando una finestra di messaggio

Uno dei metodi più elementari e spesso utilizzati durante la scrittura di codice è quello di usare una finestra di messaggio per controllare i valori mutevoli di una variabile (MsgBox). Generalmente si utilizza una finestra di messaggio immediatamente dopo la riga di codice in cui la variabile assume un valore, per verificare come la procedura viene eseguita con i valori delle variabili che cambiano dinamicamente. Per visualizzare la finestra di messaggio per ottenere il valore di una variabile si utilizza la sintassi:

MsgBox NomeVariabile

Questa riga di codice verrà rimossa dopo aver terminato la verifica del valore. Di seguito vediamo alcuni esempi che mostrano come utilizzare la funzione MsgBox per verificare i valori delle variabili che cambiano mentre viene eseguito il codice.

Esempio: Verifica dei valori delle variabili alla fine del Ciclo

```
Sub Prova1()  
    Dim I, tot As Integer  
    i = 0  
    Do  
        i = i + 1  
        MsgBox i  
'Restituisce 1, 2, 3  
        tot = tot + i  
    Loop Until i > 2  
'restituisce 6  
    MsgBox tot  
End Sub
```

Esempio: Verifica dei valori delle variabili all'inizio del Ciclo

```
Sub Prova1()  
    Dim i , tot As Integer  
    For i = 1 To 2  
        Do Until tot > 3  
            tot = tot + 2  
            MsgBox " i = " & i  
'Restituisce i=1  
            MsgBox "tot =" & tot  
'Restituisce tot=2  
        Loop  
    Next i  
    MsgBox " Totale Complessivo = " & tot  
'restituisce 4  
End Sub
```



■ Utilizzare i punti di interruzione

È possibile inserire uno o più punti di interruzione su qualsiasi riga del codice, che permette di fermare temporaneamente l'esecuzione della macro in quel punto. A questo punto la macro è in modalità interruzione e consente di vedere il valore corrente delle variabili spostando il cursore del mouse su di loro. Un punto di interruzione può essere posizionato su qualsiasi riga del codice, ma non sulle righe che definiscono le variabili o nella sezione generale delle dichiarazioni. I punti di interruzione sono generalmente fissati a una riga di codice specifico in cui si immagina un errore e che vengono eliminati quando sono stati risolti gli errori.

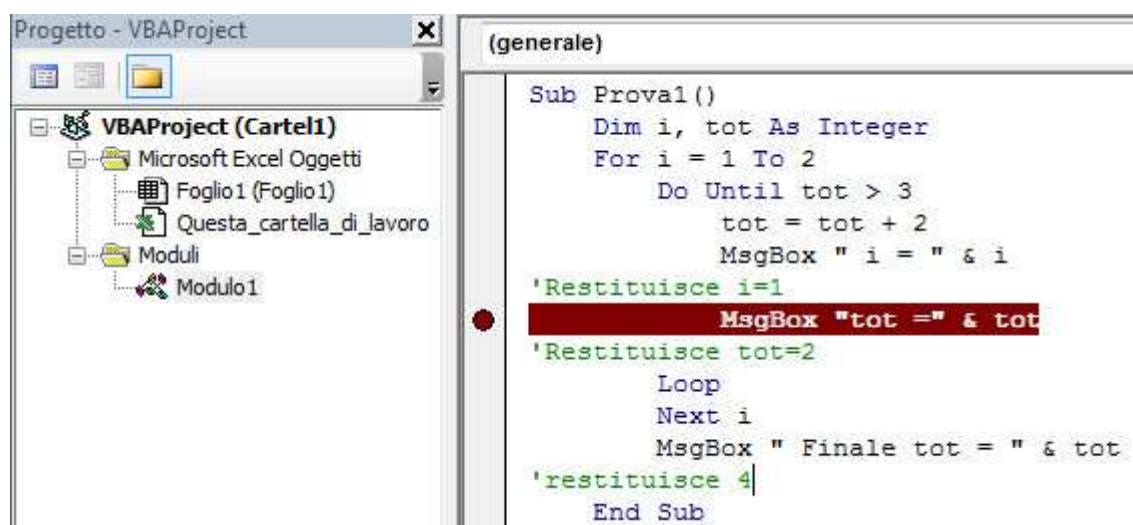


Fig. 2

■ Aggiungere o Cancellare un punto di interruzione

Per aggiungere un punto di interruzione, si deve operare in questo modo:

- Cliccare sul bordo sinistro della riga di codice in cui si desidera inserirlo
- Cliccare sulla riga e premere F9
- Dal menu Debug – Imposta/rimuovi punto di interruzione

Per cancellare un punto di interruzione, si deve ripetere l'operazione appena descritta, mentre se sono presenti vari punti di interruzione per rimuoverli tutti si deve premere **Ctrl + Maiusc + F9** oppure dal menu **Debug - Rimuovi punti di interruzione**.

■ Utilizzo della modalità Interruzione o Pausa

La macro va in modalità interruzione quando si interrompe l'esecuzione del codice e si mette in pausa temporaneamente, oppure quando

- Si incontra un punto di interruzione
- Premendo i tasti **Ctrl + Pausa** durante l'esecuzione del codice
- Se si incontra una istruzione Stop nel codice
- Al verificarsi di un errore di sintassi o di un errore Run-Time.

Interrompendo questa modalità è possibile visualizzare lo stato corrente della macro e controllare il valore delle variabili portando il cursore del mouse sulla variabile stessa. Una volta che il codice è in modalità interruzione, è possibile scegliere di continuare con l'esecuzione premendo il tasto **F5**, oppure cliccando su **Ripristina** dal menu **Esegui**. Al verificarsi di un errore, è possibile correggere l'errore e scegliere di continuare, oppure terminare l'esecuzione del codice e riavviare la macro.



■ Esecuzione Passo - Passo del codice

Per attivarla si deve fare clic su **Esegui istruzione** dal menu **Debug** o premere **F8**. In fase di progettazione, all'interno di una procedura, l'esecuzione del codice parte dall'inizio della macro ed entra in modalità pausa prima di eseguire la prima riga di codice. Questo è un modo per entrare in modalità pausa passando attraverso il codice che verrà eseguito una riga alla volta per poi passare alla riga successiva.

■ Finestra Immediata

Si può visualizzare la Finestra Immediata dal menu **Visualizza** oppure premere i tasti **Ctrl + G**. La finestra Immediata è una zona di debug primaria, ed è utilizzato per:

- Visualizzare i risultati delle istruzioni della macro
- Digitare un'istruzione o una riga di codice direttamente nella finestra e premere Invio per eseguirlo
- Modificare il valore di una variabile durante l'esecuzione di una macro

Quando la macro è in modalità di pausa, è possibile assegnare un nuovo valore alla variabile nella finestra Immediata come si farebbe nella macro stessa, inoltre nella finestra Immediata è possibile valutare le dichiarazioni VBA o le espressioni che possono essere correlate o meno alla macro. Quando la macro è in modalità di interruzione, una dichiarazione VBA nella finestra immediata viene eseguita nel contesto di tale macro, per esempio se si digita `MsgBox i`, dove `i` è il nome di una variabile utilizzata nella macro, nella finestra Immediata, si otterrà il valore corrente della variabile `i` come se il comando fosse stato utilizzato all'interno della macro in esecuzione.

Per restituire un valore, si deve precedere l'espressione con un punto interrogativo, per esempio, se digitiamo un'espressione come `? 100/2`, viene riportato il valore della divisione dei due numeri, come è possibile vedere nell'immagine sotto riportata



Fig. 3

Per visualizzare il valore di una variabile nella finestra Immediata è possibile usare il comando **Debug.Print** che permette di vedere lo stato delle variabili e le decisioni che il nostro programma prende, in quanto con l'aggiunta di questa dichiarazione vengono stampate informazioni utili a correggere i nostri programmi nella finestra immediata. La sintassi è la seguente:

Debug.Print Expression

Ad esempio, se nel foglio di Excel nella cella A1 abbiamo il testo "Prova debug.print" e ci posizioniamo su quella cella possiamo conoscerne il contenuto scrivendo nella finestra immediata il comando `?activecell.Value` che riporterà:

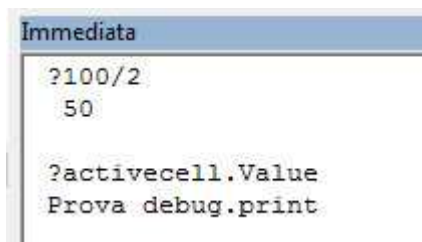


Fig. 4

Questo comando può essere usato in abbinamento alla punteggiatura in questo modo: `?activecell.Value,` se inseriamo una virgola alla fine del comando, alla prossima istruzione `Debug.Print` il risultato viene posto sulla stessa riga usando una tabulazione predefinita, mentre se inseriamo `?activecell.Value;` con un punto e virgola alla fine al prossimo comando il risultato viene inserito sulla stessa riga attaccato a quello precedente

Finestra Locale

Possiamo utilizzare la finestra Locale per vedere tutte le variabili dichiarate nella routine corrente e il loro tipo e valore corrente. La finestra Locale visualizzerà per le variabili locali dichiarate nella routine corrente e le variabili dichiarate nella sezione di dichiarazione moduli - il display è in 3 colonne, *Espressione*, *Valore* e *Tipo*. Si noti che la finestra Locale viene usata solo quando la macro è in modalità Pausa.

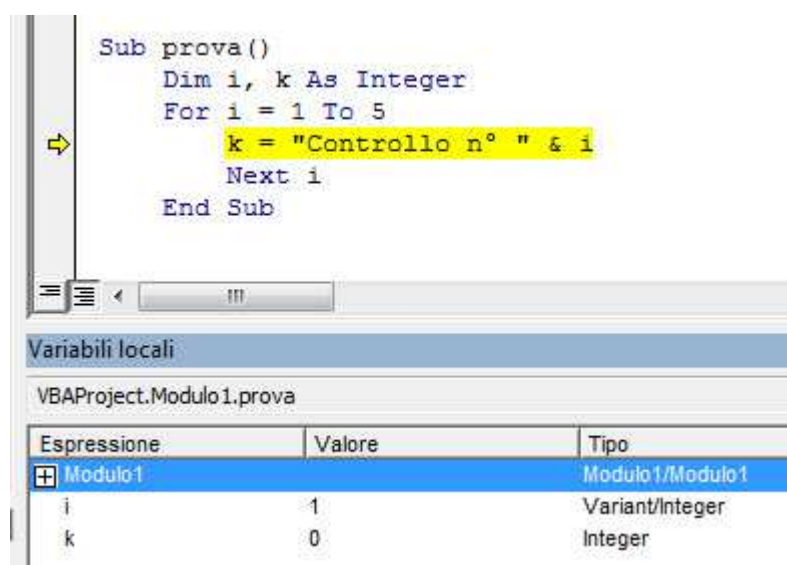


Fig. 5

Il valore di una variabile locale può essere modificata nella finestra Locale facendo clic sulla colonna Valore

Finestra Espressioni di controllo

Tramite questa finestra è possibile monitorare i valori delle variabili in modalità di interruzione, mentre la finestra Locale automaticamente tiene traccia di tutte le variabili dichiarate nella routine corrente, nella finestra di controllo è necessario specificare le variabili che si desidera tenere traccia. Nella finestra di controllo, indipendentemente da quella attuale, è possibile monitorare i valori delle variabili attraverso moduli e procedure. È possibile aggiungere manualmente le variabili e persino le espressioni alla finestra di controllo, quando è in modalità di interruzione.

Per aggiungere una variabile alla finestra di controllo, si deve cliccare col destro del mouse sulla variabile e selezionare *Aggiungi Espressione di Controllo* dal menu del tasto destro del mouse, oppure posizionare il cursore sulla variabile e selezionare *Aggiungi Espressione di Controllo* dal menu *Debug*, e in entrambi i casi, verrà visualizzata una finestra di dialogo in cui compare il campo *Espressione* e verrà visualizzato il nome della variabile.



Fig. 6

Per modificare o eliminare il controllo, si deve selezionare la variabile nella finestra di controllo, fare clic col destro del mouse e selezionare Elimina Espressione di controllo