

# Costruire un file di help

*Tutte le migliori applicazioni per Windows sono correlate da un file di guida: ma come si può creare un file HLP e collegarlo ad un'applicazione Visual Basic?*

di Roberto Scano

Ormai sono più di tre anni che utilizzo Visual Basic per Windows: le mie prime applicazioni non erano fornite di un help in linea e per questo dovevo correlare il programma di un file in formato Write, così l'utente poteva apprendere (previa stampa del documento) come funzionasse il programma.

Poi un giorno ho deciso di "addentrarmi" nella creazione di files in formato HLP, in modo da rendere più professionali le mie applicazioni e in modo da offrire all'utente la possibilità di avere l'aiuto giusto al momento giusto, tramite la pressione del tasto F1. Alla fine ho pensato di scrivere quest'articolo, anche considerando che potrà essere utile sia agli utenti di Visual Basic, che a qualsiasi altro programmatore.

## COME COSTRUIRE UN FILE DI HELP

Per poter creare un file HLP è necessario innanzitutto possedere un elaboratore testi come Word o WordPerfect, che offra la possibilità di salvare un documento in formato RTF (Rich Text Format). Bisognerà poi costruire un file di progetto (con estensione HPJ) in formato ASCII e compilarlo con HC31.EXE, un eseguibile Dos fornito assieme a Visual Basic. A compilazione terminata si avrà un file con estensione HLP.

Normalmente utilizzo Word 2.0 per la creazione delle pagine di help. La scrittura di una pagina è da effettuarsi come fosse una qualsiasi lettera, curando i minimi particolari (immagini grafiche, tabulazioni, ecc.).

Per ciascun argomento, come in un libro, si deve assegnare un **titolo** e, se si desidera un **capitolo** e alcuni **paragrafi**, come evidenziato qui:

### Assegnazione del TITOLO

Dopo aver scritto il titolo di una pagina (ad esempio, "Introduzione") bisogna posizionarsi alla sinistra del primo carattere ed inserire una nota a piè di pagina assegnando "#" come segno di nota personalizzata. Di seguito, in corrispondenza del simbolo "#" a piè di pagina bisogna digitare una stringa di riconoscimento, la quale verrà poi richiamata nel file HPJ (ad esempio "INTRO", vedi la figura 1).

### Assegnazione del CAPITOLO

Bisogna posizionarsi alla destra del carattere # ed inserire una nota a piè di pagina assegnando K come segno di nota personalizzata e quindi, spostandosi sul piè di pagina, in corrispondenza del suddetto simbolo bisogna digitare una stringa che verrà visualizzata quando si attiva la funzione di ricerca dell' help. (nell'esempio "INTRODUZIONE").

### Assegnazione del PARAGRAFO

Bisogna posizionarsi alla destra del carattere K ed inserire una nota a piè di pagina assegnando \$ come segno di nota personalizzata e quindi, a piè di pagina e in corrispondenza del suddetto simbolo bisogna digitare una stringa che verrà visualizzata quando si attiva la funzione di ricerca dell' help dopo aver cliccato sul capitolo prescelto. (nell'esempio "INTRODUZIONE AL PROGRAMMA").

Una volta impostata la prima pagina bisogna ripetere la stessa operazione per ogni pagina che si desidera creare, ricordandosi di separare le pagine l'una dall'altra mediante un salto pagina for-

zato. Quando, in un punto qualsiasi del testo, si desidera saltare da un argomento ad un altro (ad esempio, se si ha una pagina che spiega come estrarre dei dati da un database e si ha un'altra pagina che descrive come funziona SQL) è necessaria una *doppia sottolineatura* della parola che si desidera utilizzare come "richiamo di argomento", la quale deve essere seguita immediatamente - senza spazi e in formato nascosto - dal nome del titolo destinazione del salto. Se si desidera invece associare una parola ad una window **pop-up** è necessario marcare la parola con la *sottolineatura singola*, e farla seguire dal titolo della pagina in questione, formattato come testo nascosto.

Per inserire delle immagini è possibile operare in due modi distinti: (a) inserendo l'immagine direttamente da file o da clipboard, oppure (b) creando un comando per il caricamento del file in fase di compilazione. Quest'ultima opzione è consigliata perché permette di lavorare con file molto piccoli senza creare dei documenti RTF di grandi dimensioni. Per segnalare il caricamento di una immagine occorre inserire una riga di questo

```
{bmc c:\immagini\nomefile.bmp}
```

questo comando carica l'immagine nel punto preciso in cui si trova il testo; in alternativa è possibile utilizzare i comandi *bml* e *bmr*, che allineano l'immagine rispettivamente a sinistra e a destra.

## UTILIZZO DI 'HOTSPOT EDITOR

A questo punto mi sembra doveroso aprire una parentesi sull' HOTSPOT EDITOR, un programma che associa una zona in una immagine ad una determi-

nata azione (jump, pop-up o macro). Questo programma, fornito con Visual Basic, può essere utile se ad esempio si desidera creare un file di help che offra all'utente la possibilità di cliccare su una parte di una immagine rappresentante un form per ottenere informazioni sull'utilizzo di quel particolare tasto, etichetta o altro. Per creare un file SHG che spieghi il funzionamento di una schermata del programma, è necessario catturare l'immagine del form (tramite printscreen e un editor grafico come PaintBrush) e caricarla dal programma HOTSPOT EDITOR.

A questo punto è sufficiente cliccare sulla parte dell'immagine a cui si vuole associare una spiegazione e delimitarla con un rettangolo. Con un doppio click su questa figura apparirà una finestra simile a quella raffigurata in figura 2. Nel campo CONTEXT STRING è necessario segnare il titolo della pagina dove si vuol saltare. Il campo TYPE può assumere i seguenti valori:

- Pop-up:** la pagina con il titolo indicato verrà visualizzata come finestra di pop-up
- Jump:** la pagina con il titolo indicato verrà raggiunta con un salto
- Macro:** sarà eseguita una macro

Per quanto riguarda l'inserimento nel file DOC del file SHG creato in questo modo, si opera come per l'inserimento di un file BMP. Una volta completata questa operazione è necessario inserire nel file DOC delle pagine avanti per intestazione (ossia una nota a pie' di pagina personalizzata "#") i valori assegnati ai singoli context string.

Ad esempio, se abbiamo evidenziato un'area dell'immagine BMP che rappresenta un campo di inserimento dati particolarmente importante e gli abbiamo assegnato come context string il valore CAMPO\_IMPORTANTE, nel file DOC si avrà una situazione simile a quella che compare in figura 3: per prima cosa è necessario richiamare il file SHG tramite il comando:

```
{bmc c:\immagini\nomefile.SHG}
```

Anche in questo caso sono permesse le varianti **bml** e **bmr** per modificare l'allineamento dell'immagine. A questo punto è necessario inserire una interruzione di pagina ed inserire una nota a pie' di pagina personalizzata (#) a cui si assegna il valore CAMPO\_IMPORTANTE. Effettuata questa operazione è sufficiente digitare il testo di spiegazione corrispondente, con l'accortezza di limitarsi a non più di due-tre righe di spiegazione se come

Tabella 1

*I valori che possono essere assegnati alla proprietà HelpCommand di un control CMDIALOG.VBX e le azioni corrispondenti*

1 HELP_CONTEXT	visualizza il contenuto della pagina indicata dalla proprietà HelpContext
2 HELP_HELPONHELP	visualizza la Guida a "WinHelp", necessaria per i principianti di Windows
3 HELP_INDEX	visualizza la pagina di indice del file di guida
4 HELP_KEY	visualizza il contenuto della pagina indicata dalla proprietà HelpKey
5 HELP_QUIT	chiude il file di guida
6 HELP_SETINDEX	indica che il valore di Help Context è il nuovo indice
7 HELP_PARTIALKEY	attiva la funzione di ricerca

valore per TYPE dell'hot-spot si è assegnato POP-UP; per hot-spot di tipo JUMP non vi sono particolari limitazioni. Una volta terminata la creazione del file DOC è necessario salvarlo in formato RTE.

### CREAZIONE DEL FILE HPJ

Per ogni file di help deve essere presente un file di progetto, che racchiude una serie di informazioni utili per il compilatore di help. Il file di progetto deve essere creato in formato ASCII (quindi può essere facilmente creato con l'EDIT di MS-DOS). Il file è suddiviso in sezioni, ciascuna delle quali comincia con il titolo racchiuso tra parentesi quadre:

#### [OPTIONS]

Questa sezione contiene le informazioni generali del file di HELP: generalmente i valori più utilizzati sono i seguenti:

- TITLE:** il valore assegnato compare come titolo della finestra della Guida di Windows
- COPYRIGHT:** il valore assegnato compare nella finestra di About della Guida di Windows
- COMPRESS:** può assumere tre valori: FALSE, TRUE, HIGH i quali indicano al compilatore rispettivamente di non compattare il file di help, di compattarlo in modo normale o in modo elevato
- REPORT:** se il valore assegnato è ON, viene visualizzato a video lo stato di progresso della compilazione
- ICON:** assegna un'icona al file di help: se questo campo non viene assegnato, il file HLP acquisterà l'icona di default della Guida di Windows

#### [FILES]

contiene l'indicazione del percorso del file o dei file RTF da compilare.

#### [MAP]

contiene l'elenco dei TITOLI, ovvero dei riferimenti alle pagine del file RTF indicati nelle note a pie' di pagina personalizzate. Accanto al TITOLO, vi è un valore chiamato HELPCONTEXTID, il quale è utilizzato per richiamare una determinata pagina di help da un'applicazione Visual Basic, tramite la proprietà HELPCOMMAND di CMDIALOG.VBX (vedi il paragrafo "Attivazione del file di Help da Visual Basic").

#### [WINDOWS]

contiene le caratteristiche della finestra del file di help (titolo, posizione, dimensioni, colore).

#### [CONFIG]

contiene le macro dell'Help Compiler. Si tratta di una feature avanzata che non verrà discussa in questo articolo.

A titolo di esempio, il listato 1 riporta il contenuto del file HPJ relativo al mio programma 'ETICHETTE'.

Notate il comando **BrowseButtons()** nella sezione [CONFIG], che aggiunge i pulsanti << e >> alla barra dei pulsanti di WinHelp.

Se ad esempio si volesse aggiungere un pulsante che richiami la calcolatrice di Windows, è necessario aggiungere la seguente riga:

```
CreateButton("&Calcolatrice", "Exec  
Program('calc', 0)")
```

Arrivati a questo punto è necessario avviare una shell DOS (ma perché non hanno creato un compilatore in ambiente Windows? Bah!!) ed eseguire il seguente comando:

A questo punto il file HLP è pronto e può essere visualizzato mediante WinHelp. Dopo aver testato i jump, i pop-up e le eventuali macro e controllato che tutto funzioni a dovere, non ci resta che collegarlo alla nostra applicazione VB.

**FILE HLP: ATTIVAZIONE DA VISUAL BASIC**

La prima cosa da fare è dichiarare il nome del file di help appena creato nel progetto (menu Options, Project): in questo modo, quando l'utente preme il tasto F1 viene caricato il file indicato. Se si desidera un help molto accurato, di tipo *context sensitive* e con spiegazioni per ogni controllo del form, è necessario impostare la proprietà **HelpContextID**, assegnando il numero corrispondente ad una pagina del file di help; questa corrispondenza è stabilita nella sezione [MAP].

Per collegare il file HLP alla nostra applicazione, è possibile procedere utilizzando CMDIALOG.VBX oppure richiamando la funzione **WinHelp** della libreria "User". Nel primo caso si deve procedere nel modo seguente:

1. Aggiungere un pulsante al form (**Guida** nell'esempio che segue)
2. Aggiungere un CMDIALOG.VBX al form, (**Aiuto** nell'esempio)
3. Nell'evento Guida\_Click aggiungere le seguenti righe di programma

```
'asigna il nome del file di help
aiuto.HelpFile = app.path + "\NOMEFILE.HLP"
'visualizza pagina indicata sotto
aiuto.HelpCommand = HELP_CONTEXT
'asigna HELPCONTEXTID
```



**Figura 1**

Assegnazione del titolo di una pagina

```
aiuto.HelpContext = 204
'avvia la guida
aiuto.Action = 6
```

La pagina di guida da visualizzare è indicata da quattro proprietà:

**HelpFile** è il nome del file di guida  
**HelpCommand** indica il tipo di guida richiesto; i possibili valori sono riportati nella tabella 1:

**HelpContext** indica un valore corrispondente ad un argomento della guida (in questo caso HelpCommand deve essere HELP\_CONTEXT).

**HelpKey** indica una parola chiave corrispondente ad un argomento della guida (in questo caso Help Command deve essere HELP\_KEY).

**CREAZIONE DI UN MENU DI GUIDA**

In ogni applicativo che si rispetti è necessario creare un menu contenente la guida. Io l'ho organizzato in questo modo:

Caption	Name
&Guida	MenuGuida
&Sommario	MenuSommario
C&erca Argomento...	MenuCerca (shortcut-key = F1)
-	MenuBar1 (Separator=True)
&Uso della Guida	MenuUsoGuida
-	MenuBar2
(Separator=True)	
&About...	MenuAbout

Per prima cosa è necessario allineare a destra **MenuGuida** tramite la seguente istruzione, inserita in Form\_Load

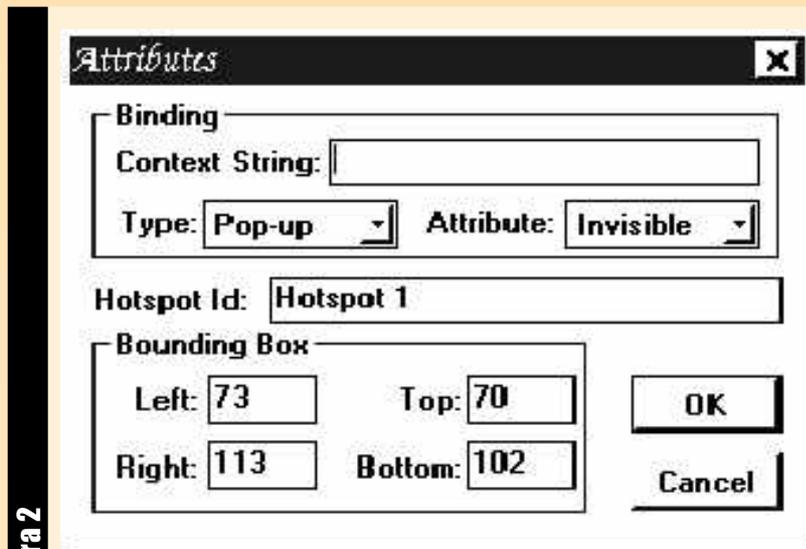
```
MenuGuida.Caption = Chr$(8) + MenuGuida.Caption
```

Ora è necessario assegnare dei richiami all'help di Windows per le altre voci di menu.

Se non si desidera utilizzare il Custom Control CMDIALOG.VBX, è necessario inserire in GLOBAL.BAS la seguente dichiarazione:

```
Declare Function WinHelp Lib "User" (ByVal hWnd As Integer, ByVal lpHelpFile As String, ByVal wCommand As Integer, dwData As Any) As Integer
```

È necessario inoltre dimensionare una variabile *DummyVal\$* ed una temporanea *Temp%* nelle dichiarazioni generali del form in questo modo, in modo che siano globali a livello di form.



**Figura 2**

L'Hotspot Editor in azione

*Il file di progetto relativo al programma etichette, un applicativo shareware per la creazione e stampa di etichette di dischetti*

```
[OPTIONS]
Title=Guida ETICHETTE 5.0           ; compare nella finestra
Copyright=c 1995 ScanoSoft         ; compare nell'About Box di WinHelp
Compress=HIGH                     ; comprime il file di help
Report=On                          ; visualizza l'operazione di compressione
icon=help.ico                     ; assegna l'icona help.ico al file di
help

[FILES]
eti50.RTF                          ; nome del sorgente da compilare

[WINDOWS]
main="Guida ETICHETTE 5.0",,,, (192,192,192) ; titolo e posizione e colore finestra

[MAP]
intro 200
mnu file 201
file_apri 202
file_salva 203
file_stampa 204
file_imposta 205
file_esci 206
mnu_modifica 207
restore_def 208
save_def 209
imposta_lbl 210
imposta_sheet 211
crea_lbl 212
mod_txt 213
mod_ico 214
mod_barra 215
registra 216
ringrazia 217
file_salvaeti 218
mnu_nuovo 219
uninstall 220

[CONFIG]
BrowseButtons()
```

A questo punto è possibile richiamare la funzione **Winhelp** dai vari menu, modificando soltanto il valore che segue il nome del file HLP:

```
MenuSommarario_Click( )
DummyVal$ = " "
Temp% = WinHel(FORM1.hWnd, App.Path
+ "\NOMEFILE.HLP", HELP_INDEX,
DummyVal$)
End Sub

MenuCerca_Click( )
DummyVal$ = " "
Temp% = WinHelp(FORM1.hWnd, App.Path
+ "\NOMEFILE.HLP", HELP_PARTIAL
KEY, DummyVal$)
End Sub

MenuUsoGuida_Click( )
DummyVal$ = " "
Temp% = WinHelp(FORM1.hWnd, App.Path
+ "\NOMEFILE.HLP", HELP_HELPON
HELP, DummyVal$)
End Sub
```

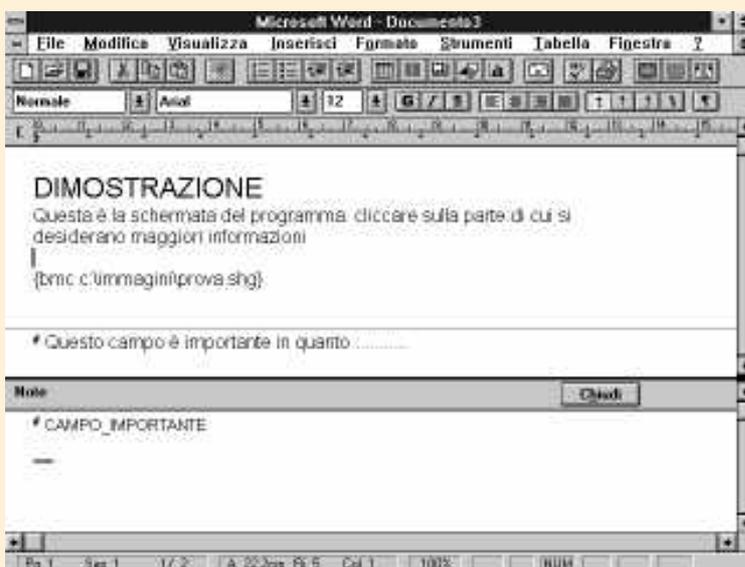
## CONCLUSIONI

Creare un file di help e collegarlo al programma Visual Basic dovrebbe essere uno dei passi obbligati nella costruzione di una applicazione professionale. Per creare un buon file di help vi sono alcune "regole" da rispettare, altrimenti si rischia di rendere ancora più difficoltosa la comprensione del programma da parte dell'utente. La procedura consigliata è la seguente

1. Preparare su carta un progetto di organizzazione del file di help (ad esempio, introduzione, menu principale, sottomenu1, ecc.), assegnando ad ogni pagina un titolo, un capitolo, un paragrafo ed un ContextId (da utilizzare nel progetto in VB)
2. Cercare di scrivere nel modo più semplice possibile, senza usare troppi termini tecnici e senza esagerare con le illustrazioni grafiche
3. Assegnare ad ogni controllo il proprio ContextId

E ora che siamo giunti alla fine dell'articolo, non mi resta che augurarvi buon lavoro!

.....  
*Roberto Scano, vent'anni, ha conseguito il Diploma di Ragioniere e Perito Commerciale lo scorso anno. Ha esperienza di produzione e vendita di software sia nel mercato shareware che commerciale. Collabora spesso con riviste informatiche e da due anni si occupa di programmazione Windows in Visual Basic. Può essere raggiunto tramite tel o fax al numero 041-5263540*



*Inserimento di un file SHG nel documento*