



# ***UTILIZZO DEI FILE DI RISORSE NEI PROGETTI***

***Come importare in un progetto Visual Basic  
una risorsa inclusa in un file .res***

***Come utilizzare i file .res per le applicazioni  
multilingua***

***Visual Basic Italia***

## Indice

Premessa.....	3
L'Editor di risorse.....	3
Importazione dei file .ico .....	4
L'output.....	5
Le funzioni.....	6

## Premessa

Spesso si rende utile l'utilizzo di file di risorse ossia file archivio che contengono una serie di stringhe, immagini bitmap file audio, file avi, o altri elementi di svariato tipo. Questo permette ad esempio di includere tutte le icone, o i file .wav o qualsiasi altro dato di cui fa uso l'applicazione in tale file, in modo da renderle invisibili all'utente.

E' questo quindi un modo per fondere un progetto scritto in Visual Basic con le risorse di cui fa uso.

Attraverso il file di risorse è poi possibile creare una o più tabelle di stringhe che rappresentano una stringa di testo in lingue differenti. E' in questo modo molto più agevole la localizzazione dell'applicazione che si andrà a distribuire.

Allo scopo di accedere a tali dati in modo logico, viene assegnato ad ogni elemento che compone il file un identificatore univoco.

In base al tipo di dato archiviato si potrà quindi avere un identificatore sottoforma di intero a 32 bit (Long) oppure in forma di stringa di testo (String).

Nonostante questo archivio sia praticamente illimitato, non è possibile compilare un progetto che faccia riferimento a più files .res.

Vediamo adesso come aggiungere un file di risorse ad un qualsiasi progetto sviluppato in Visual Basic.

A tale scopo dovremo aprire l'Editor di Risorse.

## L'Editor di Risorse

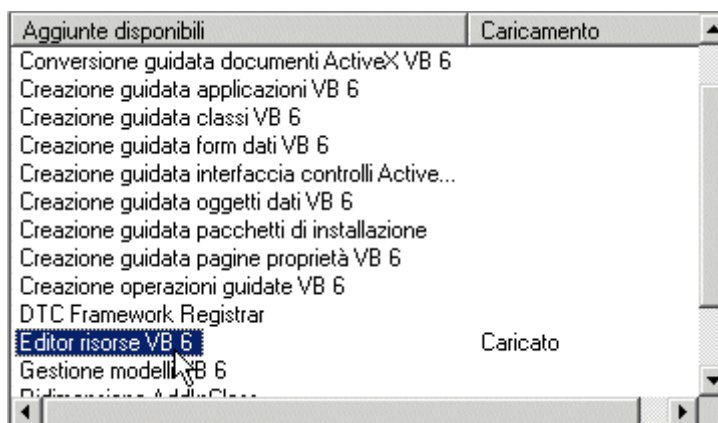
L'Editor di Risorse è richiamabile cliccando sulla voce Aggiunte nel menu principale di Visual Basic come mostrato nella figura sottostante



Nel menu a discesa selezionare quindi Gestione Aggiunte e dalla finestra che apparirà contrassegnare col segno di spunta la casella corrispondente alla voce 'Editor di Risorse'.

Selezionare la casella 'Caricato/scaricato' in modo che accanto alla voce 'Editor di Risorse' appaia il testo 'Caricato' ed infine premere il pulsante OK.

La figura qui sotto rappresenta graficamente questa operazione.

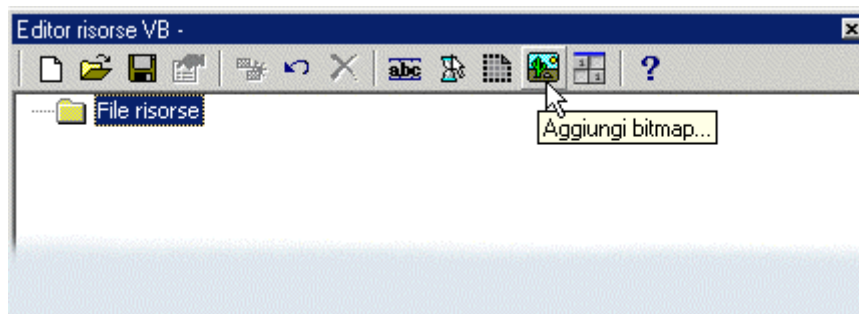


L'Editor di risorse si presenta come una finestra recante l'icona di una cartella.

Inizialmente tale cartella si presenta vuota, in quanto ancora non è stato aggiunto alcun dato al quale si vuole far riferimento nel progetto che si sta sviluppando.

Per fare un'esempio di come integrare un progetto sviluppato in Visual Basic con un file .res, importiamo prima di tutto l'immagine in formato bitmap 'bollicine.bmp' contenuto nella directory "C:\WINDOWS" o "C:\WINNT" per i sistemi basati su NT.

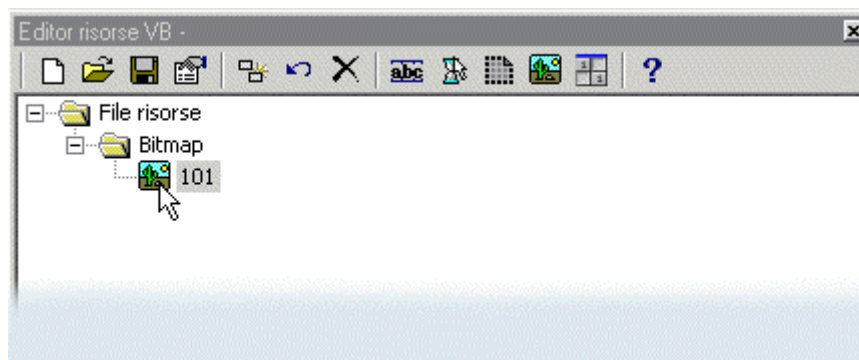
Si dovrà quindi premere il pulsante sulla toolbar dell'Editor di Risorse che rappresenta un'immagine bitmap, come mostrato qui sotto.



Verrà visualizzata la finestra 'Apri' di Windows attraverso la quale si potrà cercare il file immagine che s'intende importare. Nell'esempio che abbiamo fatto il file è contenuto nella directory "C:\WINDOWS". Dopo aver sfogliato la suddetta directory, trovare il file 'bollicine.bmp' e premere il pulsante OK della Commondialog Apri.

Come si noterà la cartella dell'Editor di Risorse sarà collegata ad un'icona che rappresenta la bitmap che abbiamo importato. Accanto a tale icona sarà visibile un numero, nientemeno che l'identificatore univoco della risorsa di cui si parlava all'inizio dell'articolo.

L'immagine sottostante rappresenta quanto apparirà.



Per avere un'idea più precisa della risorsa alla quale corrisponde l'identificatore indicato nell'Editor di Risorse, è possibile cliccare sul dato col tasto destro del mouse e selezionare dal popup-menu che così apparirà la voce 'Proprietà...'. Si noterà che l'intero identificatore corrisponde proprio alla nostra immagine 'bollicine.bmp' che verrà visualizzata in una casella di anteprima.

### Importazione dei file .ico

Lo stesso procedimento può essere compiuto per un qualsiasi tipo di dato. Ad esempio possiamo ripetere tutte le operazioni che abbiamo appena fatto con l'importazione questa volta di un'icona quindi di un'immagine in formato .ico.

A tale scopo sarà sufficiente selezionare l'icona corrispondente dalla toolbar dell'Editor di Risorse.

Bitmaps ed icone (unitamente a qualsiasi altro formato di immagine) rappresentano però il tipo di dato importato.

Che si importi un'immagine, un file di testo, un file audio, la differenza non è rilevante. I dati verranno infatti archiviati allo stesso modo e sarà compito di chi sviluppa l'applicazione richiamare in modo opportuno questo o quello, evitando di fare riferimento all'identificatore di un elemento invece che di un altro. Questo per evitare ad esempio di caricare un file .txt all'interno di un controllo PictureBox...

Diverso è il discorso di una tabella di stringhe che rappresenta un vero e proprio database di dati testuali, anche se di proporzioni di gran lunga ridotte.

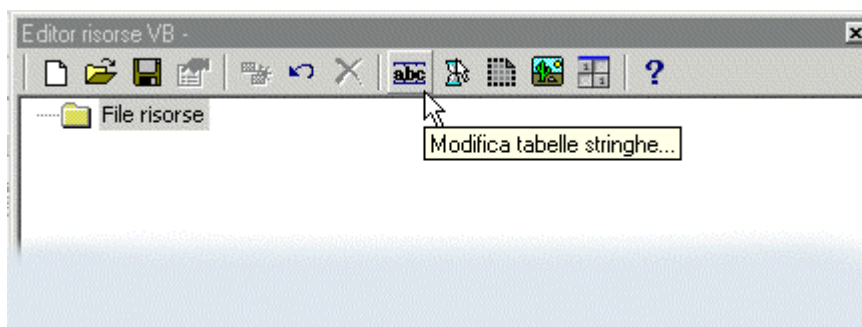
Un tabella delle stringhe non è altro che un semplice insieme di stringhe memorizzate in un file di risorse. Ogni stringa della tabella delle stringhe è identificata da un indice univoco, proprio come accade per ogni altro dato memorizzato nel file .res.

Generalmente si utilizza una sola tabella, ossia una colonna di stringhe rapportate ciascuna al proprio identificatore. E' però possibile avere più tabelle correlate, ossia più colonne ciascuna delle quali contiene stringhe localizzate in lingue diverse. Questo rende molto agevole la distribuzione di un'applicazione in più



lingue senza dover modificare alcuna linea di codice.

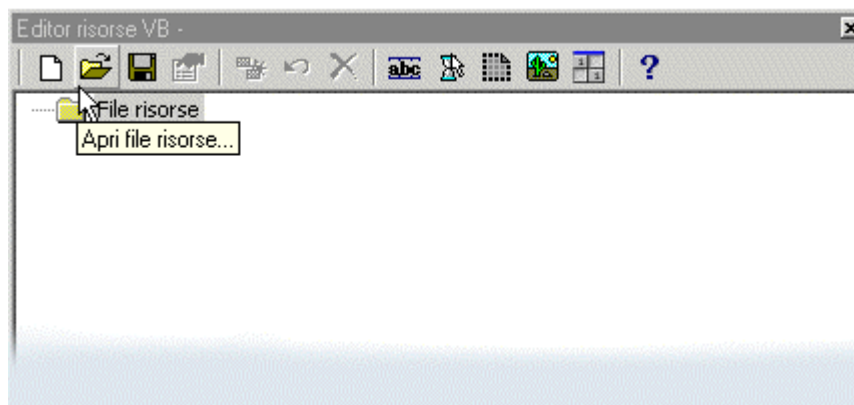
Ecco come creare una tabella delle stringhe. Selezionare dalla toolbar dell'Editor di Risorse il pulsante 'abc', come mostrato qui sotto.



Apparirà come conseguenza la finestra denominata 'Modifica tabella delle stringhe'. Tale tabella può essere gestita come un vero e proprio database. All'apertura della tabella l'identificatore della prima riga sarà già impostato. Proviamo ad esempio ad inserire nella prima riga in corrispondenza dell'intestazione di colonna Italiano (Standard) il testo "Applicazione".

Cliccando due volte consecutive col pulsante del mouse sopra qualsiasi intestazione di colonna è possibile scegliere un secondo identificatore che individua la colonna oppure la lingua, nel caso di una localizzazione dell'applicazione. Adesso aggiungiamo una colonna alla prima riga. Per fare ciò selezioniamo il pulsante della toolbar che indica il testo "abc". La riga avrà dunque una nuova cella. Cliccando due volte sull'intestazione di colonna di questa nuova cella selezioniamo la lingua inglese. Inseriamo nella cella il testo "Application".

Nell'articolo successivo si vedrà come recuperare i dati inseriti nel file .res e nella tabella delle stringhe dalla propria applicazione. Per adesso basterà sapere che per importare un file di risorse all'interno del progetto e quindi per includere tutte le risorse che esso contiene sarà necessario cliccare sulla toolbar dell'Editor di Risorse in corrispondenza dell'icona raffigurante una cartella aperta, come mostra la figura sotto.



## L'output

Seguendo l'esempio del capitolo precedente, siamo rimasti ad un file di risorse contenente la bitmap "bollicine" ed una tabella delle stringhe ad una riga sola e due colonne, una avente l'intestazione di colonna 'Italiano (Standard)' e l'altra 'Inglese (Regno Unito)' anche se in teoria si può scegliere qualsiasi versione della lingua.

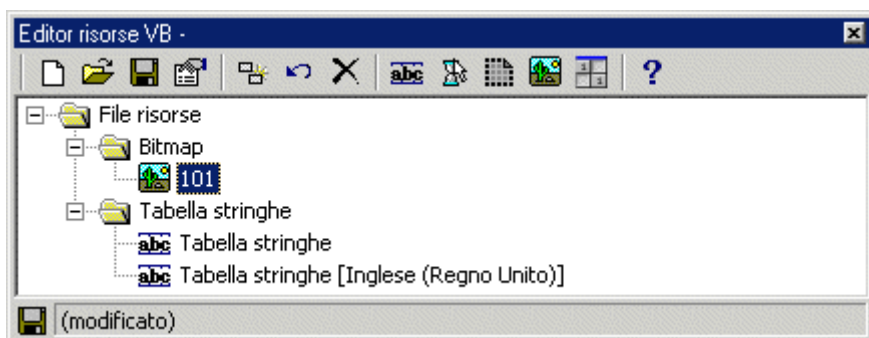
All'interno della cella corrispondente alla lingua italiana abbiamo inserito il testo 'Applicazione' mentre in quella corrispondente alla lingua inglese abbiamo immesso 'Application'.

In linea di massima la tabella delle stringhe, così come è stata appena descritta dovrebbe presentarsi in questo modo:

Modifica tabelle stringhe		
ID	Italiano (standard)	Inglese (Regno Unito)
101	Applicazione	Application

Chiudendo quindi il file finestra 'Modifica tabella delle stringhe' ci si potrà rendere conto dalla struttura del file .res che a tale file si è aggiunto un secondo elemento (secondo rispetto alla bitmap che abbiamo inserito nel corso dell'articolo precedente) ossia la tabella delle stringhe.

Una tabella formata da due tabelle integrate tra loro ossia quella avente come intestazione di colonna 'Italiano (Standard)' e quella avente come intestazione 'Inglese (Regno Unito)'. La struttura del file di risorse così come è stato modificato dovrebbe essere simile a quella mostrata nella figura qui sotto:



Il grosso vantaggio rappresentato dalla possibilità al momento dell'esecuzione di un'applicazione di caricare dal file di risorse solo le stringhe, le bitmap, le icone e i dati necessari anziché tutti contemporaneamente al momento del caricamento di un form o di un modulo.

Inoltre è sempre molto più facile operare esternamente su un file archivio che dover rientrare nella propria applicazione modificandone il codice.

Adesso che il nostro file .res (dove naturalmente .res è l'estensione tipica dei files di risorse. Un file con estensione differente non verrà caricato nell'Editor di Risorse per un'eventuale modifica) è finalmente completo con tutti i dati che vogliamo collegare all'applicazione, possiamo tranquillamente salvare il file.

Quando un file di risorse modificato non è stato salvato, nella barra di stato viene visualizzata l'icona di un dischetto ed un messaggio che indica che la stringa è stata modificata, come si può vedere perfettamente dall'immagine precedente.

Per salvare un file .res basterà premere col tasto sinistro del mouse l'icona recante un dischetto nella toolbar dell'Editor di Risorse.

La CommonDialog Apri permetterà di sfogliare l'hard disk fino a trovare la cartella nella quale s'intende salvare il file.

Naturalmente, se un file .res è associato ad una determinata applicazione, è buona norma inserirla nella stessa cartella che contiene i files del progetto di tale applicazione.

## Le funzioni

Adesso che il lavoro è stato salvato, il file di risorse e tutto il suo contenuto è disponibile. Esistono tre funzioni per il recupero dei dati archiviati in tale file.

Vediamole adesso una per una. La funzione **LoadResPicture** carica una bitmap, un'icona o un cursore da un file risorsa. E' dunque la funzione che dovremo utilizzare per recuperare la bitmap "bollicine" che abbiamo inserito nel file .res.

La sintassi della funzione è la seguente:

*LoadResPicture(indice, formato)*

dove *indice* rappresenta l'identificatore della bitmap nel file di risorse. Corrisponde quindi all'intero che appare alla destra dell'icona della bitmap nell'Editor di Risorse, nel momento in cui viene visualizzata la sua struttura.

Prendendo come esempio la prima immagine di questo articolo, si può notare facilmente che l'intero identificatore della bitmap "bollicine" corrisponde a 101.

Se avessimo inserito una seconda bitmap nel file di risorse, essa avrebbe preso l'identificatore 102 e così via per tutte le bitmap e le icone archiviate.

Sempre riguardo la funzione *LoadResPicture*, *formato* corrisponde ad un valore o ad una costante che specifica il formato dei dati restituiti, come indicato dalla tabella sottostante:

vbResBitmap	rappresenta la costante alla quale è associato il valore 0 ed indica all'applicazione che il tipo di dato da recuperare è una bitmap
vbResIcon	rappresenta la costante alla quale è associato il valore 1 ed indica all'applicazione che il tipo di dato da recuperare è un'icona
vbResCursor	rappresenta la costante alla quale è associato il valore 1 ed indica all'applicazione che il tipo di dato da recuperare è un cursore

Ad esempio, se volessimo inserire in un controllo PictureBox la bitmap "bollicine" presente nel file .res che abbiamo salvato, dovremmo inserire questa linea di codice:

```
Picture1.Picture = LoadResPicture(101, 0)
```

tenendo sempre conto che questo codice è basato sul numero e la tipologia dei dati archiviati nel file .res di esempio.

Se invece "bollicine" fosse stato un'icona il codice sarebbe stato il seguente:

```
Picture1.Picture = LoadResPicture(101, 1)
```

mentre nell'ultimo caso, quello cioè di un cursore:

```
Picture1.Picture = LoadResPicture(101, 2)
```

Altra funzione per il recupero dei dati è la **LoadResString** che recupera una stringa da un file di risorse. La sintassi è la seguente:

*LoadResString(indice)*

dove *indice* rappresenta come per le immagini, l'identificatore della stringa.

Come si vede dalla figura, entrambe le colonne (Italiano ed Inglese) hanno un unico identificatore ossia 101. Questo perché in fase di esecuzione viene selezionata automaticamente a tabella delle stringhe corrispondente all'ID delle impostazioni internazionali di sistema. Sarà quindi il sistema dell'utente a determinare quale colonna caricare, in corrispondenza della stringa con identificatore 101. Quindi per recuperare o la stringa 'Applicazione' nel caso in cui l'utente sia italiano o 'Application' per la lingua inglese:

```
LoadResString(101)
```

La terza funzione da considerare è **LoadResData** che carica dati di vari tipi (ma con una dimensione massima di 64K) da un file di risorse e restituisce una matrice Byte.

La sintassi è la seguente:

```
LoadResData(indice, formato)
```

dove *indice* rappresenta l'identificatore della risorsa che si vuole recuperare dal file mentre *formato* è un valore che specifica il formato originale dei dati restituiti. Tale valore può essere inoltre il nome della stringa di una determinata risorsa.

I tipi di formato sono elencati nella tabella sottostante:

1	Cursore
2	Bitmap
3	Icona
4	Menu
5	Finestra di dialogo
6	Stringa
7	Directory di fonts
8	Font
9	Tabella tasti di scelta
10	Risorsa personalizzata
11	Cursore gruppo
12	Icona gruppo