




[Home Page](#) 
[Informazioni](#) 
[Aiuto](#) 

Informazioni aggiuntive sui files di risorse

http://www.vbsimple.net/info/info_10.htm

I files di risorse  sono dei particolari files di testo con [estensione .RC](#), scritti con un qualunque editor di testo *pulito*, ad esempio il Blocco Note di Windows. Questi files verranno poi [compilati](#) con un programma di nome RC (Resource Compiler) che genererà un file [binario](#) con [estensione](#) .RES.

L'utilizzo dei files di risorse all'interno di un progetto Visual Basic permette la localizzazione separata di alcuni elementi di un progetto. Ad esempio è possibile inserire tutte le immagini, le icone e le scritte informative all'interno di un file di risorse e poi caricarle da esso quando il programma ne necessita.

Il vantaggio di questo genere di operazioni si manifesta nel momento in cui si vogliono effettuare traduzioni di un progetto in altre lingue. Infatti basterà cambiare il contenuto del file di risorse contenente i testi o le immagini e ricompilare il progetto per ottenere la nuova versione del programma.

C'è un *HowTo* dedicato proprio all'[utilizzo dei files di risorse per progetti multilingua](#).

Un altro vantaggio dato dai files di risorse è quello di poter "*spostare*" gli elementi *inutili* di un progetto, quali i testi e le immagini, all'interno di una [libreria](#) e poterli richiamare attraverso d'essa. Ciò rende i files eseguibili più piccoli e più facilmente modificabili.

Per esempio se sviluppiamo un progetto contenente molta grafica, lo inviamo a qualche nostro amico e, dopo di questo, notiamo che sono presenti errori o [bugs](#) all'interno del codice del programma, dopo averli corretti saremmo costretti a reinviare l'intero programma, compresa quindi la grafica che non ha bisogno di essere rivista, al nostro amico. L'utilizzo di un file di risorse - e quindi la localizzazione di elementi *inutili* in un file di libreria - porta il vantaggio che dopo aver corretto il nostro progetto non dovremo reinviare al nostro amico l'intero programma. Ci basterà invece reinviargli solo i files che abbiamo modificato e rimarranno intoccate le librerie contenenti le risorse.

Ogni dato all'interno del file di risorse è identificato da un numero univoco e dal suo gruppo di appartenenza. Tuttavia è possibile definire delle costanti simboliche che puntano ad un numero di risorsa. La dichiarazione delle costanti simboliche è identica a quella utilizzata nel linguaggio C.

Ad esempio definiamo il numero di risorsa 1001 mediante il nome simbolico TITOLO:

```
#define TITOLO      1001
```

da questa riga in poi sarà possibile, all'interno del file di risorse, fare riferimento al numero di risorsa 1001 mediante il nome simbolico TITOLO.

Poiché le risorse sono organizzate a gruppi - chiamati tipi - è possibile avere più di una risorsa con numero 1001: avremo una risorsa di tipo BITMAP, un suono di tipo SOUND e un piccolo video di tipo VIDEO, tutti con lo stesso numero di risorsa. Ciò che li

differenzia in questo caso è il loro gruppo.

I file di risorse possono contenere ogni tipologia di risorsa: testi, immagini, icone, suoni e dati in genere. Le risorse si dividono in tre categorie:

1. **Stringhe** - caricabili mediante l'istruzione *LoadResString*
 2. **Immagini** - caricabili mediante l'istruzione *LoadResPicture*
 3. **Altri dati** - caricabili mediante l'istruzione *LoadResData*
-

Le stringhe sono la tipologia più semplice di risorsa. Vengono memorizzate nel file di risorse a doppio byte, ovvero ogni carattere di una stringa è formato da due [bytes](#). Tuttavia, l'istruzione di lettura *LoadResString* provvede automaticamente la conversione nel formato originale.

Sono organizzate in tabelle di stringhe (STRINGTABLE) che tengono assieme più stringhe in un'unico segmento, allo scopo di rendere più rapido l'accesso alle stringhe dello stesso segmento.

La sintassi di una tabella di stringhe è la seguente:

```
STRINGTABLE [caricamento] [memoria]  
BEGIN  
    ID1    stringa1  
    ID2    stringa2  
    ID3    stringa3  
END
```

Il loro richiamo attraverso Visual Basic è semplicissimo:

```
variabile = LoadResString(numero)
```

Le immagini sono la prima tipologia di risorsa binaria; tra esse sono incluse sia le bitmap, sia le icone, sia i cursori del mouse.

A differenza delle stringhe, le risorse binarie non sono contenute nel file di definizione .RC, ma è presente soltanto un riferimento alla posizione della risorsa da inserire nel file.

Il compilatore andrà a cercarsi il file indicato nella definizione .RC e lo inserirà nel file di risorse compilato .RES.

Nel caso di immagini il compilatore effettuerà una conversione di formato per trasformarlo nel formato immagine di Visual Basic.

La sintassi delle immagini è:

```
ID tipologia [caricamento] [memoria] nomefile
```

La **tipologia** può essere: **BITMAP** per le immagini bitmap, **ICON** per le icone e **CURSOR** per i cursori del mouse.

Ecco degli esempi di risorse valide:

```
2000 BITMAP figural.bmp
3000 ICON iconal.ico
4000 CURSOR cursore.cur
```

Il nome del file deve essere di tipo [MS-DOS 8.3](#).

L'utilizzo di un'immagine da un file di risorse si effettua nel seguente modo:

```
Set Immagine = LoadResPicture(numero, tipo)
```

Dove *Immagine* è un oggetto di tipo **IPictureDisp**, capace di contenere immagini, *numero* è il numero della risorsa e *tipo* è la tipologia della risorsa. La tipologia, in Visual Basic, è uno di questi tre valori: **vbResBitmap** per le immagini bitmap, **vbResIcon** per le icone e **vbResCursor** per i cursori del mouse. Questi tre valori fanno parte dell'[enumerazione LoadResConstants](#).

La funzione *LoadResPicture* restituisce infatti un [puntatore](#) ad un oggetto **IPictureDisp**.

Ma i file di risorse possono contenere anche altri dati che non sono né stringhe né immagini, come ad esempio suoni, video o altre specie di dati binari.

I dati binari subiscono, in fase di compilazione del file di risorse, un processo di conversione di formato. È pertanto necessario, prima di riutilizzare tali dati, effettuare una riconversione mediante l'istruzione *StrConv*.

Come per le immagini, il file di definizione .RC non conterrà le risorse stesse, ma conterrà il percorso al file binario da inserire nelle risorse.

La sintassi è identica a quella delle risorse immagine:

```
ID tipologia [caricamento] [memoria] nomefile
```

Dove *tipologia* è un nome qualunque, formato da una sola parola, che, in combinazione con il numero ID identifica univocamente una risorsa all'interno del file. Ad esempio:

```
2000 MUSICA musical.mid
2000 DATI datil.dat
```

Le due risorse, sebbene abbiano il medesimo numero ID, si contraddistinguono per la loro tipologia differente. Il richiamo di tali dati attraverso Visual Basic si effettua nel seguente modo:

```
Variabile = LoadResData(numero, tipo)
```

ma per ottenere i dati originali è necessario effettuare una conversione tramite *StrConv*:

```
Variabile = StrConv(LoadResData(numero, tipo), vbUnicode)
```

Tipo deve essere una stringa che identifica il nome assegnato al gruppo dati nel file di risorse.

Nella definizione delle risorse è possibile definire due parametri che modificano le modalità con le quali tali dati vengono inseriti e rimangono in memoria.

Il parametro *caricamento* è una modalità di caricamento che può essere:

- **PRELOAD**: caricamento immediato
- **LOADONCALL**: (default) la risorsa viene caricata quando richiesto

Il parametro *memoria* invece definisce la maniera in cui i dati rimangono in memoria; può essere:

- **FIXED**: i dati rimangono sempre alla stessa posizione della memoria
- **MOVEABLE**: le risorse possono essere spostate per compattare la memoria
- **DISCARDABLE**: possono essere scaricate quando non servono più

Un esempio, abbastanza completo, di file di risorse è il seguente:

```
#define STR_OK      1001
#define STR_ERR     1002
#define STR_CHIUDI  1003
#define STR_APRI    1004
#define STR_SALVA   1005

#define ICO_CHIUDI  2001
#define ICO_APRI    2002
#define ICO_SALVA   2003

#define MUS_INTRO   3001
#define MUS_FINE    3002

STRINGTABLE PRELOAD FIXED
BEGIN
    STR_OK      "L'operazione è stata eseguita!"
    STR_ERR     "L'operazione ha generato un errore!"
    STR_CHIUDI  "Sicuro di voler chiudere?"
    STR_APRI    "Seleziona il file da aprire"
    STR_SALVA   "Seleziona il nome con cui salvare"
END

ICO_CHIUDI LOADONCALL MOVEABLE "CHIUDI.ICO"
ICO_APRI   LOADONCALL MOVEABLE "APRI.ICO"
ICO_SALVA  LOADONCALL MOVEABLE "SALVA.ICO"

MUS_INTRO  LOADONCALL DISCARDABLE "INTRO.MID"
MUS_FINE   LOADONCALL DISCARDABLE "FINE.MID"
```

Prima di lasciare l'argomento è importante accennare dove si trovi il compilatore RC per generare i files di risorse .RES; la sua posizione è all'interno della cartella RESOURCE, che si trova all'interno della cartella WIZARDS.

Per cui, se Visual Basic è installato nella cartella
 C:\PROGRAMMI\DEVSTUDIO\VB
 il compilatore RC si troverà all'interno della cartella
 C:\PROGRAMMI\DEVSTUDIO\VB\WIZARDS\RESOURCE

Il suo nome è RC.EXE ed è un programmino senza interfaccia grafica. Ciò significa che il suo utilizzo deve essere effettuato all'interno di una finestra MS-DOS.

[Fibia FBI](#)

8 Gennaio 2001
