



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


Utilizzare un file di risorse per progetti multilingua

http://www.vbsimple.net/howto/ht_016.htm

- [Creare i files di risorse.](#)
- [Compilare i files di risorse.](#)
- [Utilizzo dei files di risorse.](#)
- [Seconda versione del progetto.](#)

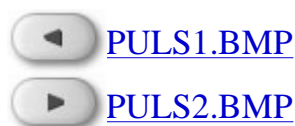
Difficoltà:  3 / 5

Avete mai pensato di scrivere un programma multilingua?

Ovvero un programma in due o più versioni: una per la lingua italiana ed un'altra per la lingua inglese? Il tutto, ovviamente, senza dover riscrivere tutte le Caption ed i testi presenti all'interno del progetto originale. Ebbene, per realizzare un'operazione del genere è fondamentale (e più semplice) utilizzare un [file di risorse](#)  differente per ogni versione.

Un file di risorse è un un file di testo, con [estensione RC](#) che deve essere compilato e trasformato in un file [binario](#) con estensione RES. La compilazione viene effettuata mediante un piccolo compilatore di risorse, fornito assieme a Visual Basic 5.

Prima di cominciare un progetto vero e proprio è necessario possedere due piccole immagini che utilizzeremo in questo progetto. Le due immagini sono le seguenti:



Basterà scaricarle (premere il tasto destro del mouse sopra i due collegamenti e scegliere la voce "Salva Oggetto con nome") ed inserirle nella stessa cartella del nostro progetto, avendo cura di mantenere i nomi originali.

Sicuri di possedere le due immagini sopra prepariamo due semplici file di risorse, uno in lingua italiana ed uno in lingua inglese.

Apriamo una finestra di Blocco Note e scriviamoci dentro queste righe:

```
1. #define STRTITOLOPROGRAMMA 1001
2. #define STRPULSANTE1 1002
3. #define STRPULSANTE2 1003
4. #define IMG_PULSANTE1 2001
5. #define IMG_PULSANTE2 2002
6.
7. STRINGTABLE
8. BEGIN
9.     STRTITOLOPROGRAMMA, "Utilizzo di un file di risorse"
10.    STRPULSANTE1, "Primo pulsante"
11.    STRPULSANTE2, "Secondo pulsante"
12. END
13.
14. IMG_PULSANTE1 BITMAP "PULS1.BMP"
```

```
15. IMG_PULSANTE2 BITMAP "PULS2.BMP"
```

Il contenuto di questo file si riassume in pochi semplici concetti: alle prime cinque righe vengono dichiarati cinque nomi simbolici ai numeri 1001, 1002, 1003, 2001 e 2002.

Alla riga 7 inizia una sezione di stinghe, definite alle righe 9, 10 e 11. Ogni costante simbolica rappresenta il proprio numero ed ai tre numeri viene associata una stringa.

Alle ultime due righe vengono definite due immagini, PULS1.BMP e PULS2.BMP. Anch'esse sono legate al loro nome simbolico indicante i numeri 2001 e 2002.

Per una migliore comprensione di questi concetti si rimanda alle [informazioni aggiuntive sui file di risorse](#).

Scritte le informazioni salviamo il documento utilizzando il comando *"Salva con nome"* del menu **File** del Blocco Note, come mostrato nella figura 1.

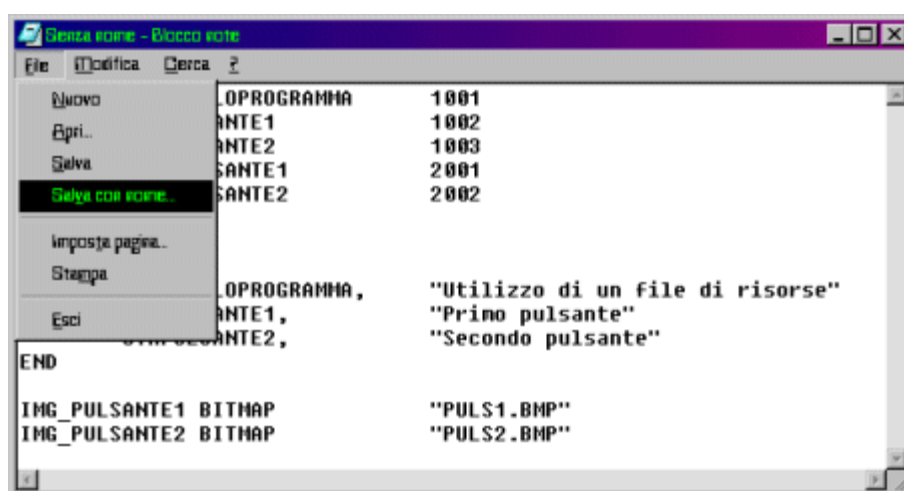


Figura 1

Fatto questo, apparirà la finestra di dialogo per la scelta del nome di file da assegnare al documento. Inserire nella casella apposita, il nome **RIS_ITA**, come visibile nella figura 2.

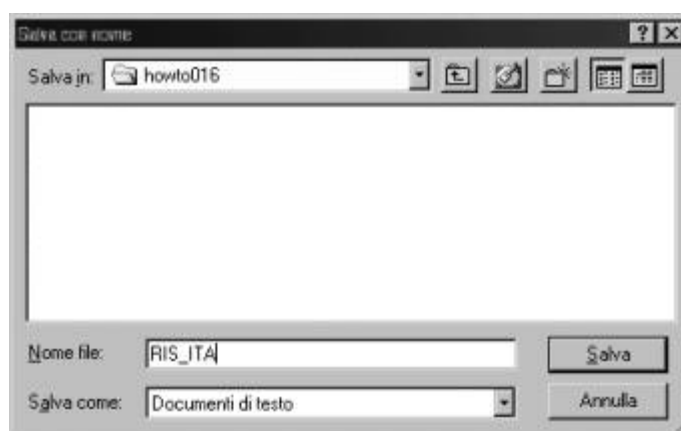


Figura 2

Eseguiamo un'operazione simile per il file di risorse in lingua inglese. Apriamo un'altra finestra di Blocco Note e digitiamo il codice seguente:

```

1. #define STRTITOLOPROGRAMMA 1001
2. #define STRPULSANTE1 1002
3. #define STRPULSANTE2 1003
4. #define IMG_PULSANTE1 2001
5. #define IMG_PULSANTE2 2002
6.
7. STRINGTABLE
8. BEGIN
9.     STRTITOLOPROGRAMMA, "How to use a resource file"
10.    STRPULSANTE1, "First Button"
11.    STRPULSANTE2, "Second Button"
12. END
13.
14. IMG_PULSANTE1 BITMAP "PULS1.BMP"
15. IMG_PULSANTE2 BITMAP "PULS2.BMP"

```

Questo codice differisce dal precedente solo per il contenuto delle righe 9, 10 e 11. Salviamo questo documento come **RIS_ING**.

Al termine di queste due operazioni dovremmo avere all'interno della nostra cartella 4 files:

- **PULS1.BMP**
- **PULS2.BMP**
- **RIS_ITA.TXT**
- **RIS_ING.TXT**

Generati i due files di risorse è necessario [compilarli](#) con il compilatore apposito fornito con Visual Basic. Prima di effettuare questa operazione è necessario ridenominare i due files di risorse con la loro estensione corretta: RC.

Per far questo apriamo un Prompt di MS-DOS, posizioniamoci nella cartella del nostro progetto (nel nostro caso C:\Windows\Desktop\HowTo016) e ridenominiamo i due files utilizzando il comando REN come mostrato nella figura 3.

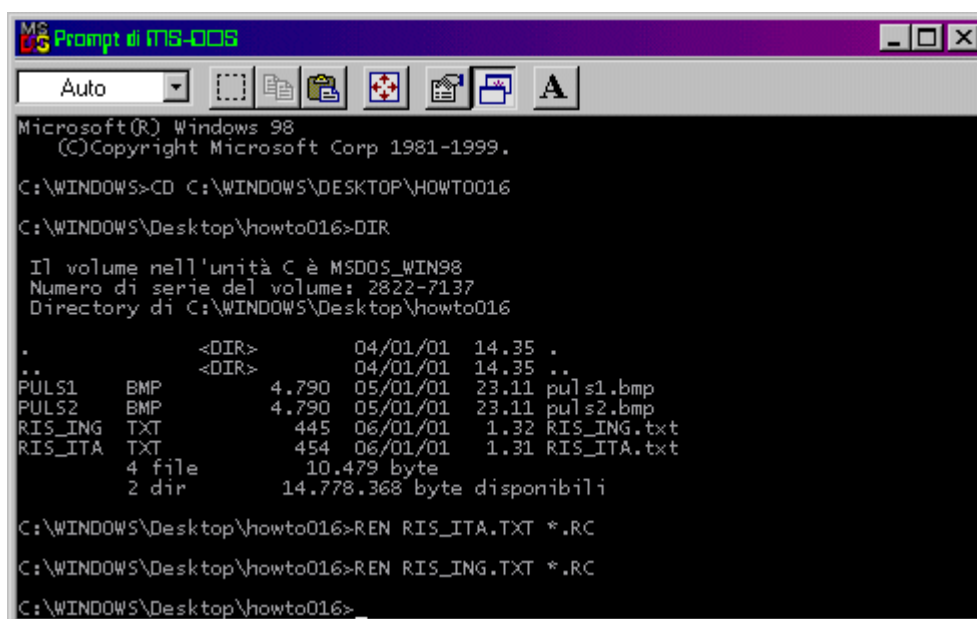


Figura 3

I comandi che abbiamo eseguito sono:

- `CD C:\WINDOWS\DESKTOP\HOWTO016`
Per posizionarci sulla cartella del progetto
- `DIR`
Per dare uno sguardo al contenuto della cartella
- `REN RIS_ITA.TXT *.RC`
Ridenominare il file RIS_ITA.TXT in RIS_ITA.RC
- `REN RIS_ING.TXT *.RC`
Ridenominare il file RIS_ING.TXT in RIS_ING.RC

Da questo momento in poi la nostra cartella conterrà questi files:

- **PULS1.BMP**
- **PULS2.BMP**
- **RIS_ITA.RC**
- **RIS_ING.RC**

Solo adesso possiamo provvedere alla compilazione dei due file di risorse RC. Prima di effettuarla, però, dobbiamo conoscere la posizione della cartella ove risiede Visual Basic: nel nostro esempio è "**C:\Programmi\DevStudio\Visual Basic 5**". Trovata la cartella, sempre nella stessa finestra DOS, scriviamo un paio di comandi:

- `CD "C:\PROGRAMMI\DEVSTUDIO\VISUAL BASIC 5"`
Per posizionarci nella cartella di Visual Basic
- `CD WIZARDS`
Per entrare nella cartella WIZARDS
- `CD RESOURCE`
Per entrare nella cartella RESOURCE
- `RC /R C:\WINDOWS\DESKTOP\HOWTO016\RIS_ITA.RC`
Compilare il file RIS_ITA.RC
- `RC /R C:\WINDOWS\DESKTOP\HOWTO016\RIS_ING.RC`
Compilare il file RIS_ING.RC

Al termine dell'operazione il contenuto della finestra DOS dovrebbe essere simile a quello mostrato nella figura 4.

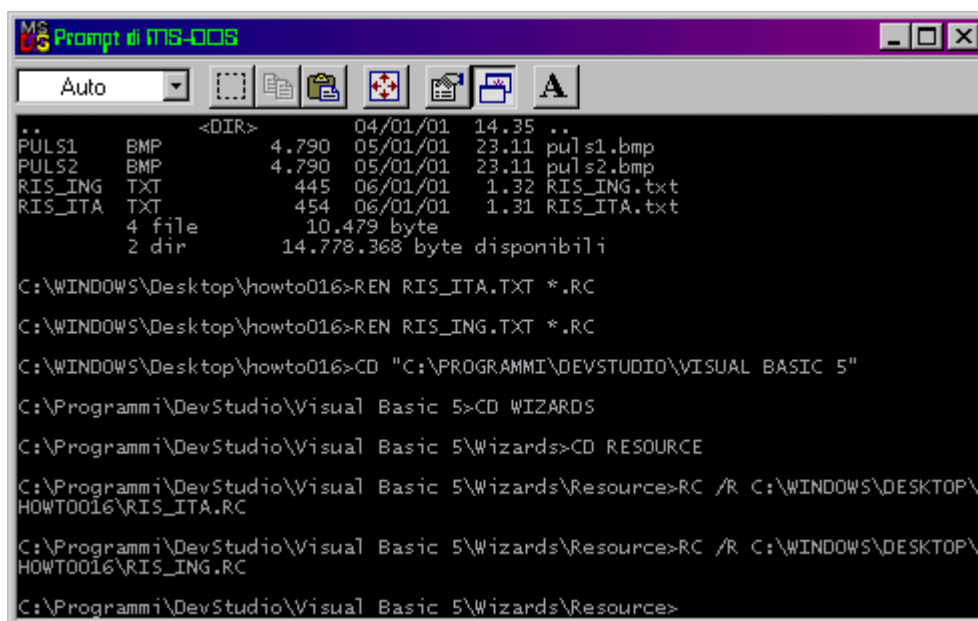



Figura 4

A seguito del comando di compilazione non dovrebbe comparire alcun messaggio. La presenza di qualche messaggio indica la presenza di errori nel codice del file di risorse o la mancanza delle immagini PULS1.BMP o PULS2.BMP.

Per avere la completa sicurezza che la compilazione è andata a buon fine dovremmo avere all'interno della cartella del progetto i seguenti files:

- PULS1.BMP
- PULS2.BMP
- RIS_ITA.RC
- RIS_ING.RC
- RIS_ITA.RES
- RIS_ING.RES

Adesso che abbiamo creato e compilato i due files di risorse .RES 🗄️ è necessario creare i due progetti per sfruttare tali files.

Creato un nuovo progetto, inseriamo sopra un form due *CommandButton*  di nome **Pulsante1** e **Pulsante2**. Impostiamo per essi la proprietà *Style* a 1-Graphical. Questa proprietà comunica al [compilatore](#) che il pulsante ospiterà sopra d'esso un'immagine grafica.



Prima di scrivere il codice che gestisce le risorse inseriamo un riferimento al primo file di risorse, quelle italiane (**RIS_ITA.RES**). Selezioniamo dal menu **Progetto** la voce *Inserisci file* ed apparirà la finestra di dialogo per la selezione del file da inserire, che vediamo nella figura 6. Basterà selezionare il file di risorse voluto e premere il pulsante **Apri**.

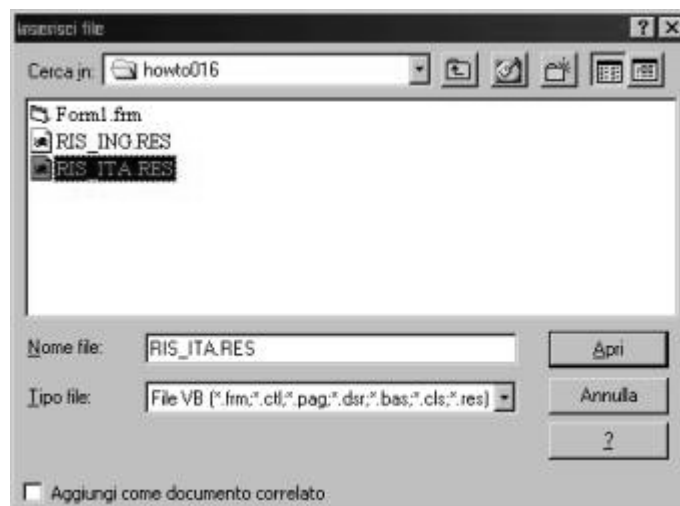


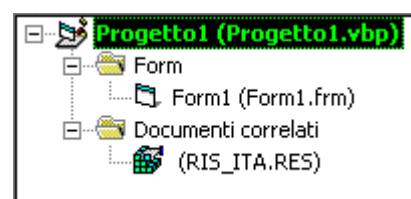



Figura 6

Fatto questo, apparirà nella finestra di gestione dei progetti  il riferimento al file di risorse  nella categoria *Documenti correlati*.




È importante ricordare che non è possibile inserire più di un file di risorse per ogni progetto. Pertanto, per utilizzare le risorse in lingua inglese bisognerà creare un nuovo processo che conterrà il riferimento al file di risorse in lingua inglese.

Una volta inserito il riferimento al file di risorse, possiamo dedicarci alle poche righe di codice di questo progetto. Innanzitutto, per comodità, dichiariamo delle costanti  corrispondenti a quelle specificate nel file di risorse:

```
1. Option Explicit
2.
3. Private Const STRTITOLOPROGRAMMA = 1001
4. Private Const STRPULSANTE1 = 1002
5. Private Const STRPULSANTE2 = 1003
6. Private Const IMG_PULSANTE1 = 2001
7. Private Const IMG_PULSANTE2 = 2002
8.
```

Le costanti alle righe 3-7 sono le medesime di quelle dichiarate nel file di risorse. È una scelta implementativa molto comoda. Infatti, ad ogni risorsa è associato un numero, ma a livello di codice è più facile e più chiaro stabilire dei termini letterali indicanti il numero univoco della risorsa.

```
9. Private Sub Form_Load()
10.     Me.Caption = LoadResString(STRTITOLOPROGRAMMA)
11.     Pulsante1.Caption = LoadResString(STRPULSANTE1)
12.     Pulsante2.Caption = LoadResString(STRPULSANTE2)
13.     Set Pulsante1.Picture = LoadResPicture(IMG_PULSANTE1, vbResBitmap)
14.     Set Pulsante2.Picture = LoadResPicture(IMG_PULSANTE2, vbResBitmap)
15. End Sub
```

E questo è il codice del programma. L'abbiamo inserito nell'evento  *Load* del form, cosicché i valori vengano subito caricati dal file di risorse.

Alla riga 10 impostiamo la *Caption* del form: il suo valore è caricato dal file di risorse

mediante l'istruzione **LoadResString** ed è identificato dal numero 1001 (riga 3).

Alle righe 11 e 12 effettuiamo un'operazione simile: impostiamo le *Caption* dei due pulsanti caricando i valori con numero 1002 e 1003 (righe 4 e 5) dal file di risorse.

Alle righe 13 e 14, invece, carichiamo due immagini dal file di risorse mediante **LoadResPicture** ed impostiamo la proprietà *Picture* dei due pulsanti con tali immagini.

A differenza dell'istruzione *LoadResString*, la *LoadResPicture* richiede due parametri: il primo indica il numero della risorsa, mentre il secondo indica il tipo di immagine da caricare. Tale valore deve essere uno di quelli specificati nell'[enumerazione LoadResConstants](#): **vbResBitmap** per le immagini bitmap, **vbResIcon** per le icone e **vbResCursor** per i cursori del mouse.

Oltre alle due citate istruzioni per caricare risorse da un file, ne esiste una terza: *LoadResData* viene infatti utilizzata per caricare dati generici, non ben definiti, da un file di risorse, ad esempio suoni, video, etc...

Possiamo adesso provare il programma.

In fase di esecuzione troviamo cambiate le *Caption* del form e dei due pulsanti e le immagini sopra i pulsanti.



Ma i nostri sforzi sarebbero stati vani se il risultato si limitasse a questo. E infatti non è così. Passiamo alla creazione dello stesso progetto in lingua inglese?

Possedendo già il file di risorse .RES compilato ci basterà creare un nuovo progetto e, sfruttando la funzione *Inserisci file* del menu **Progetto**, inserire prima un riferimento al Form1 e poi al file di risorse in lingua inglese.

Fatto questo, basterà eseguire il progetto per ritrovarsi il testo già tradotto - nel nostro caso quindi le tre *Caption* - senza dover apportare nessuna modifica al codice.



L'utilizzo dei files di risorse semplifica enormemente la revisione di un progetto per effettuare traduzioni o cambiamenti linguistici o grafici. Infatti il codice non viene minimamente influenzato: basterà fare riferimento al nuovo file di risorse e [ricompilare](#) il progetto per avere una nuova versione del programma.

La stessa [Microsoft](#), per i suoi progetti multilingua, sfrutta i files di risorse ed inserisce tutte le risorse in un file di [libreria](#). Per cambiare lingua ad un programma basterà sostituire la DLL contenente le risorse, et voilà, il gioco è fatto.

Fibia FBI
6 Gennaio 2001



[Torna all'indice degli HowTo](#)