



## Lettura ed utilizzo dei parametri passati nella riga di comando del programma

[http://www.vbsimple.net/howto/ht\\_018.htm](http://www.vbsimple.net/howto/ht_018.htm)

Difficoltà: 3 / 5

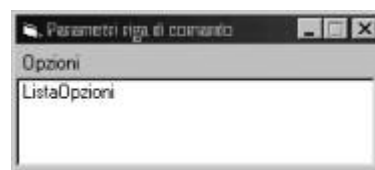
Da quando è nato il sistema a finestre Windows le righe di comando di un programma hanno cominciato a perdere senso, ma sono ancora molto utilizzate.

Infatti, nel momento in cui clicchiamo due volte sopra un documento, viene caricato il programma associato all'estensione del documento e viene passato come parametro al programma il nome del documento che si desidera aprire.

Altresì, ogni volta che utilizziamo il Prompt di MS-DOS o la voce Esegui del menu **Start** (**Avvio**) è possibile fornire al programma che stiamo per caricare, una serie di comandi o parametri, per indicare ad esso di effettuare una serie di operazioni.

In questo semplice esempio vedremo come analizzare la riga di comando ed estrarre i parametri passati ad un nostro programma.

Inseriamo sopra un form una **Label A** di nome **Label1** ed una **ListBox** di nome **ListaOpzioni**.



Il programma, al caricamento, estrarrà i parametri dalla riga di comando e li scriverà nella **ListBox** **ListaOpzioni**.

Il codice in questione è stato posto in una Sub che, partendo da una stringa, inserisce nella **ListBox** i singoli [token](#) (elementi). Il codice è il seguente:

```
1. Option Explicit
2.
3. Private Sub LeggiParametri(ByVal Riga As String)
4.     Dim LungSez As Integer
5.     Dim LungRiga As Integer
```

La Sub in questione si chiama **LeggiParametri** e richiede un parametro stringa di nome **Riga**. Esso sarà trattato [per valore](#), per cui tutte le modifiche apportate sulla stringa non si rifletteranno all'esterno della funzione.

Infatti, abbiamo intenzione di estrarre i singoli token dalla riga ed eliminarli da essa subito dopo la loro estrazione. Questo farà sì che la riga diventi sempre più corta, velocizzando e semplificando il processo. I token che possiamo estrarre dalla riga sono di due tipologie: quelli semplici, delimitati da uno spazio, e quelli a parole multiple, delimitati da una coppia di virgolette a sinistra ed una a destra.

All'interno della Sub abbiamo dichiarato due variabili, entrambe di tipo Integer: **LungSez**

indica la lunghezza di una sezione, ovvero la posizione all'interno di una stringa dove finisce un token; **LungRiga** indica semplicemente la lunghezza della stringa.

```
6.      Do While LungSez < LungRiga
```

Alla riga 6 inizia un ciclo che dura fintanto che la lunghezza della riga non è maggiore della sezione; ovvero la sezione è minore della lunghezza della riga.

Poiché, ogni volta, la riga contenente i vari parametri, viene accorciata mediante eliminazione dell'ultimo parametro utilizzato, essa diviene sempre più corta. Nel momento in cui l'ultima sezione estratta (LungSez) è uguale (quindi non è minore) alla lunghezza della riga, il ciclo finisce.

```
7.      Riga = Trim(Right(Riga, LungRiga - LungSez))
8.      LungRiga = Len(Riga)
```

Alla riga 7 viene effettuata l'eliminazione da Riga dell'ultimo parametro trovato, estraendo LungRiga - LungSez caratteri partendo dalla destra. In ogni caso gli spazi all'inizio ed alla fine saranno eliminati poiché essi fungono da separatori dei vari parametri.

Alla riga 8 viene effettuato ogni volta il ricalcolo della lunghezza della riga.

```
9.      If Left(Riga, 1) = "" Then
10.         LungSez = InStr(2, Riga, "", 1)
11.         If LungSez = 0 Then LungSez = LungRiga
```

Se il primo carattere della riga è un " ci troviamo davanti un parametro a parole multiple. Abbiamo per cui la necessità di trovare il secondo " che delimita la fine di tale token.

Alla riga 10 abbiamo, per cui, la ricerca del secondo delimitatore, partendo ovviamente dal secondo carattere, poiché il primo, come già detto, contiene il primo delimitatore.

Se il delimitatore viene trovato, LungSez conterrà la sua posizione, altrimenti conterrà 0. Alla riga 11 viene verificato se LungSez contiene 0. In tal caso ci troviamo davanti un parametro a parole multiple senza delimitatore finale. Non conoscendo la posizione dove tale token dovrebbe terminare, facciamo in modo che esso contenga tutta la rimanente riga, ponendo uguali LungSez e LungRiga.

Questo farà sì che il token venga estratto alla riga 20.

```
12.     Else
13.         LungSez = InStr(2, Riga, " ", 1)
14.         If LungSez = 0 Then
15.             LungSez = LungRiga
16.         Else
17.             LungSez = LungSez - 1
18.         End If
19.     End If
```

Se, invece il token non inizia per " abbiamo davanti un token semplice, delimitato da uno spazio.

Alla riga 13 ricerchiamo la posizione del delimitatore e la memorizziamo all'interno di **LungSez**. Infatti la posizione del delimitatore indica anche la lunghezza della sezione.

Se la ricerca del delimitatore ha dato risultato 0, abbiamo trovato l'ultimo parametro, al cui seguito non appare uno spazio. Per tale ragione il nostro token sarà uguale alla stringa rimanente (riga 15).

Se, invece, abbiamo un delimitatore, escludiamo lo spazio, che non farà parte del token stesso, ma lo delimerà soltanto (riga 17).

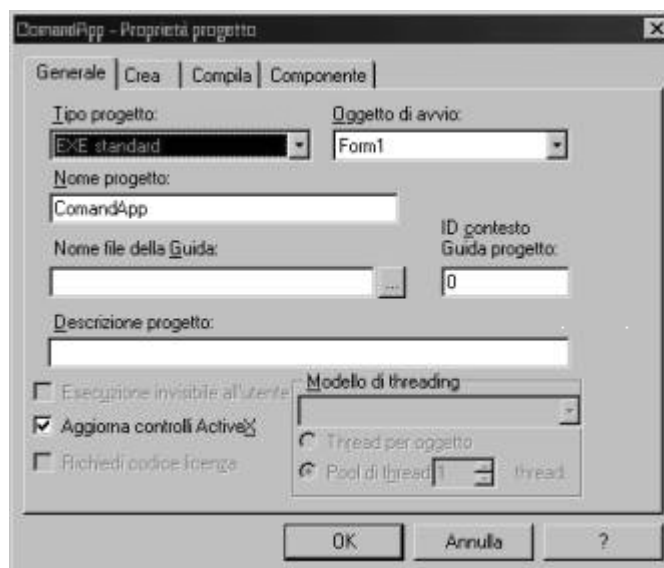
```
20.          ListaOpzioni.AddItem Left(Riga, LungSez)
21.      Loop
22. End Sub
```

Così, ad ogni passaggio, può essere estratto un token ed inserito nella *ListBox* **ListaOpzioni**.

Al ciclo successivo il token appena trovato sarà cancellato dalla riga ed il processo reinizierà da capo.

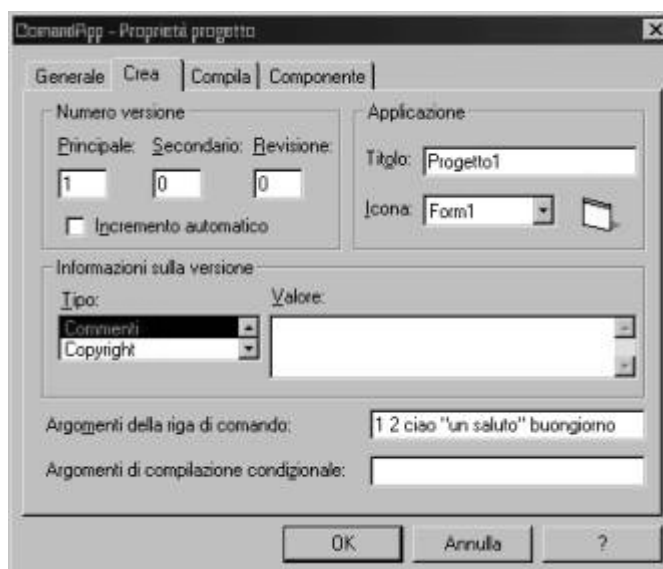
Per provare il programma abbiamo due possibilità: [compilare](#) il progetto in un file eseguibile e caricare esso tramite finestra DOS oppure inserire i parametri direttamente all'interno dell'[IDE](#). Per semplificare la cosa abbiamo scelto proprio quest'ultima possibilità.

Apriamo il menu **Progetto** di Visual Basic, scegliamo la voce "*Proprietà di .....*" (l'ultima voce) ed apparirà la finestra mostrata nella figura 2.



**Figura 2**

Fatto questo basterà cliccare sulla voce **Crea** in alto ed apparirà un'altra finestra.

**Figura 3**

In questa finestra sarà possibile definire alcuni parametri di esecuzione.

In basso abbiamo una casella di testo, alla cui sinistra c'è scritto **"Argomenti della riga di comando"**, in cui è possibile inserire alcuni parametri, come se eseguiamo il progetto da una finestra DOS.

Inseriamo questi 5 parametri, separandoli da uno spazio:

- 1
- 2
- ciao
- "un saluto"
- buongiorno

Fatto questo, basterà premere il pulsante OK e provare il progetto.

Appariranno nella *ListBox* i parametri poco prima inseriti.



Ovviamente in questo caso abbiamo semplicemente voluto elencarli, ma avremmo potuto pure svolgere operazioni particolari per ogni parametro.

Si consiglia di concedere all'utente quasi sempre l'utilizzo della riga di comando, per permettere di eseguire alcune operazioni, per esempio aprire un documento semplicemente trascinando l'icona del documento sopra il nostro programma, oppure l'utilizzo di procedure automatiche tramite l'impostazione di determinati parametri sulla riga di comando.

Giuseppe Della Bianca  
11 Gennaio 2001



[Torna all'indice degli HowTo](#)