



Array Pubblici in un Form

http://www.vbsimple.net/howto/ht_053.htm

Difficoltà: ► 1 / 5

Un limite davvero pesante, noioso e del tutto ingiustificato è quello imposto dall'impossibilità di dichiarare un [array](#) pubblico di qualsiasi tipo all'interno di un modulo di oggetto, quale un Form.

Ad esempio una dichiarazione del genere non è valida e genera un errore in fase di compilazione se posto all'interno di un Form:

```
Public arr1(2, 1) As String
```

Vedremo in quest'articolo come superare questo limite che avrà creato delle grane a più di una persona, senza obbligare il programmatore a porre tali dati all'interno di un modulo standard.

Accontentiamo il compilatore e dichiariamo tale vettore con Private, rendendolo quindi inaccessibile dall'esterno. Per consentire l'uso indiretto dell'array da parte di altre parti del codice utilizzeremo una proprietà [wrapper](#) in lettura e scrittura che maneggi i dati del vettore come se fossero accessibili direttamente dall'esterno.

Sviluppiamo un progettino composto da due soli form, senza interfaccia. Il primo form conterrà l'array privato e la proprietà pubblica.

```
1. Option Explicit
2. Private arr1(2, 1) As String
3.
4. Public Property Get Array1(ByVal Indice1 As Integer, ByVal Indice2 As Integer) As String
5.     Array1 = arr1(Indice1, Indice2)
6. End Property
7.
8. Public Property Let Array1(ByVal Indice1 As Integer, ByVal Indice2 As Integer,
9.     ByVal Valore As String)
10.     arr1(Indice1, Indice2) = Valore
11. End Property
```

La proprietà in questione si chiama **Array1** e richiede il passaggio di due argomenti, corrispondenti alle due dimensioni dell'array **arr1**. La proprietà in lettura (righe 4-6) recupera i dati dall'array utilizzando i due indici forniti alla proprietà e li restituisce in uscita, mentre la proprietà in scrittura (righe 8-10) effettua l'operazione inversa.



Nota bene!

L'uso di proprietà che richiedono il passaggio di almeno un argomento durante la lettura necessitano una cura maggiore. Infatti la controparte in scrittura richiede gli stessi argomenti (nell'esempio **Indice1** ed **Indice2**) ed il passaggio del nuovo valore da assegnare. L'ordine con cui questi argomenti sono dichiarati potrebbe generare confusione: in tali situazioni è fondamentale ricordare che il nuovo valore della proprietà **va**

dichiarato per ultimo, in fondo all'elenco degli argomenti obbligatori della proprietà.

Aggiungiamo un paio di righe giusto per effettuare un controllo di funzionamento:

```
12. Private Sub Form_Load()  
13.     arr1(0, 0) = "Ciao"  
14.     arr1(0, 1) = "Ciccio"  
15.     arr1(1, 0) = "Hello"  
16.     arr1(1, 1) = "Paperino"  
17.     arr1(2, 0) = "Bonjour"  
18.     arr1(2, 1) = "Topolino"  
19.     Load Form2  
20.     Unload Form2  
21.     Unload Me  
22. End Sub
```

Le righe 13-18 inizializzano l'array all'interno del primo form con dei valori.

Alla riga 19 è caricato il secondo form, che si occuperà di leggere e modificare i valori della matrice dall'esterno. poiché i due form non ci servono ad altro saranno normalmente scaricati.

All'interno del secondo form abbiamo un codice ancora più semplice:

```
1. Option Explicit  
2.  
3. Private Sub Form_Load()  
4.     MsgBox Form1.Array1(0, 1)  
5.     Form1.Array1(0, 1) = "Ciccino"  
6.     MsgBox Form1.Array1(0, 1)  
7. End Sub
```

Al caricamento del form viene inizialmente presentato un valore della matrice nel primo form (dimostrando quindi la possibilità di leggerne i valori), corrispondente a **"Ciccio"**.

Alla riga 5 tale valore è modificato in **"Ciccino"** (dimostrando quindi la possibilità di modificare i valori). Alla riga 6 lo stesso valore è riletto (per dimostrare ancora l'avvenuta scrittura).

La stessa soluzione può essere applicata per accedere a variabili e stringhe a lunghezza fissa.

Non si tratta di un'idea geniale per evitare una limitazione del linguaggio. È piuttosto l'applicazione di un concetto di base della [programmazione ad oggetti](#) ed in linea generale teorica nessun dato all'interno di una classe dovrebbe essere accessibile dall'esterno se non mediante l'uso di un [metodo](#) o di una proprietà della classe stessa.

[Fibia FBI](#)

21 Dicembre 2002



[Torna all'indice degli HowTo](#)
