




Utilizzare i Watch e l'Object Browser per analizzare un oggetto WebBrowser

http://www.vbsimple.fbi/news/news_13.htm

Quest'articolo nasce da [una domanda](#) posta nel [Forum di VBSimple](#) circa il recupero del testo attualmente selezionato all'interno di una pagina web caricata in un controllo **WebBrowser**. Personalmente non conoscevo la soluzione ma sono arrivato al recupero delle informazioni necessarie con la semplice analisi riportata in quest'articolo. Ho ritenuto utile e costruttivo scrivere l'articolo perchè la coppia **Watch** e **Object Browser** (chiamati in italiano rispettivamente *Espressioni di controllo* e *Visualizzatore oggetti*) consente un'analisi molto approfondita dei contenuti delle librerie ActiveX esterne, in grado di supplire alla troppo spesso frammentaria documentazione delle librerie.

Per capire quanto spiegato in quest'articolo si raccomanda di seguire passo passo il progetto sviluppato; a tal fine sarà necessario inserire un riferimento al controllo **Microsoft Internet Controls (SHDOCVW.DLL)**, aggiungere un controllo **WebBrowser**  di nome **wb1** sopra un form ed un semplice **CommandButton** di nome **cmdTesto**. Il codice inizialmente includerà due sole righe, la prima all'interno della routine di gestione dell'evento **Load** del Form e la seconda legata al **Click** sul pulsante:

```
1. Option Explicit
2.
3. Private Sub Form_Load()
4.     wb1.Navigate "http://www.vbsimple.net/forum"
5. End Sub
6.
7. Private Sub cmdTesto_Click()
8.     Stop
9. End Sub
```

La prima delle due istruzioni richiamerà il forum di VBSimple all'interno del controllo WebBrowser, mentre la seconda interromperà l'esecuzione del codice per trasferire il controllo all'ambiente di Visual Basic. Naturalmente al posto di quello Stop andrà inserito il codice che vedremo fra poco. L'interruzione ci servirà per capire come recuperare le informazioni che ci servono.



Abbiamo scelto di usare il Forum di VBSimple perché si tratta di una pagina semplice, senza frames (le divisioni dell'area in più parti), a differenza della pagina principale di VBSimple che contiene tre frames al suo interno. La soluzione qui spiegata infatti non funziona correttamente in presenza di frames, ma non è questo scopo lo scopo dell'articolo.

Eeguire il progetto, attendere il caricamento della pagina e selezionare una o più parole nella pagina, come faremmo solitamente all'interno del browser, trascinando cioè il cursore del mouse. Nel nostro esempio supponiamo una situazione del genere:

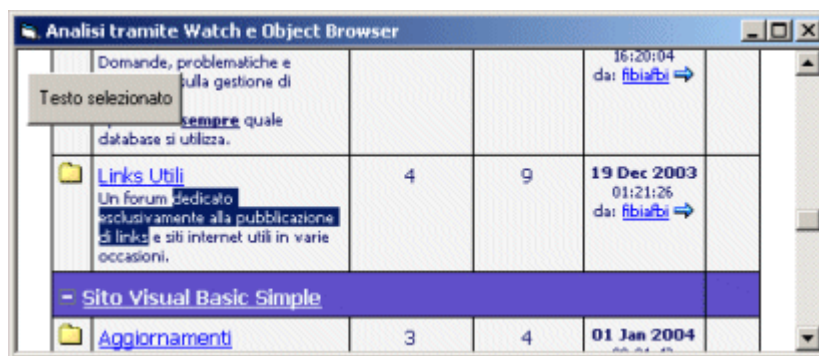


Figura 1

Come mostrato nella Figura 1 abbiamo selezionato il testo "**dedicato esclusivamente alla pubblicazione di links**", disposto su tre righe differenti ma continue. Il click sopra il pulsante Testo selezionato dovrà restituirci proprio il suddetto testo. Attualmente al posto del codice necessario c'è solo uno Stop; clicchiamo il pulsante e quando il programma verrà interrotto selezioniamo wb1 dall'istruzione che carica la pagina, clicchiamo il tasto destro e scegliamo l'opzione *Aggiungi Espressione di Controllo (Watch)* come evidenziato nella Figura 2:

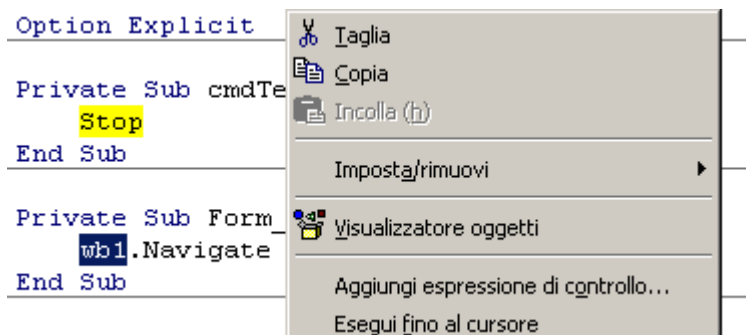


Figura 2

Sarà mostrata la finestra di dialogo *Aggiungi espressione di controllo (Add Watch)*, selezionare la voce (**tutte le routine**) a fianco dell'etichetta Routine, quindi premere OK.

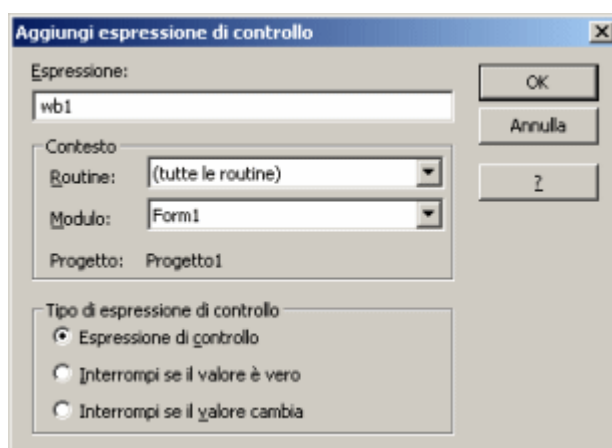


Figura 3

Si aprirà una finestrella a parte, solitamente posta in basso, contenente il Watch attivato in precedenza, con una struttura ad albero da sfogliare. Al suo interno sono elencate tutte le proprietà leggibili del controllo WebBrowser (vale naturalmente anche per qualsiasi altro oggetto). In particolare per il nostro scopo interessa la proprietà **Document**, possiamo

intuirlo sfogliando le proprietà una per una ed andando per esclusione. Apriamo la struttura e osserviamo le altre proprietà all'interno.



Particolare attenzione deve essere dedicata alle proprietà circolari, quali *Application*. Questa infatti punta allo stesso oggetto, così l'esplorazione risulta infinita: dentro *Application* c'è un'altra *Application* (se stessa in realtà) e dentro questa un'altra ancora e ancora e così via...

Espressione	Valore	Tipo	Contenitore
wb1	"Controllo Microsoft Web Br"	Object/WebBrowser	Form1
AddressBar	Vero	Boolean	Form1
Application		Object/WebBrowser	Form1
Busy	Falso	Boolean	Form1
Container		Object/Form1	Form1
DataBindings		DataBindings/DataBindings	Form1
Document		Object/HTMLDocument	Form1
activeElement		IHTMLElement/HTMLFrameElement	Form1
alinkColor	"#0000ff"	Variant/String	Form1
all		IHTMLElementCollection/DispHTMLElementCollec	Form1
anchors		IHTMLElementCollection/DispHTMLElementCollec	Form1
applets		IHTMLElementCollection/DispHTMLElementCollec	Form1
attributes	Nothing	Object	Form1

Figura 4

Sfogliamo le proprietà fino a quando non arriviamo ad una che potrebbe fare al caso nostro: **Selection**, che in inglese significa appunto selezione. Al suo interno non sembra esserci alcuna proprietà che faccia riferimento al nostro testo scelto nella finestra, potremmo procedere oltre ma nessun'altra proprietà della struttura riporta ciò che cerchiamo.


Espressione	Valore	Tipo
selection		IHTMLSelectionObject/IHTMLSelectionObject
type	"None"	String

Figura 5

Approfondiamo perciò la proprietà *Selection* che risulta essere di tipo **IHTMLSelectionObject** (la presenza del doppio tipo di dato è giustificata da alcuni fattori complessi, quali le interfacce polimorfiche o quelle doppie, dette altrimenti *dual interfaces*). In questi casi il tipo di dato reale è comunque sempre il secondo.

Sorge qui una complicazione che può sembrare strana e rendere più difficile la nostra ricerca: tra le classi native di VB non esiste nessuna *IHTMLSelectionObject*; la stessa non appare neanche nella libreria contenente il *WebBrowser*.

La classe è definita nella libreria *Microsoft HTML Object Library* (MSHTML.TLB) che dovremo aggiungere manualmente al progetto dalla voce **Riferimenti** nel menu *Progetto*. Questo genere di notizia non è scritta da alcuna parte e tocca al fiuto del programmatore capire dove recuperare tale informazione. 😊

Per investigare il contenuto completo della classe *IHTMLSelectionObject* e quindi della proprietà *Selection* sarà necessario dare uno sguardo al secondo strumento d'analisi: il Visualizzatore di oggetti  (**Object Browser**) richiamabile tramite il menu *Visualizza* o premendo semplicemente il tasto F2. Il funzionamento del Visualizzatore di oggetti è

[spiegato nel Corso Base](#).

Selezionare dalla ComboBox in alto a sinistra libreria MSHTML e cercare tra le classi (la colonna di sinistra) **IHTMLSelectionObject**. Troveremo al suo interno 1 proprietà (la stessa che era riportata nella struttura ad albero indicata prima) e 3 metodi.

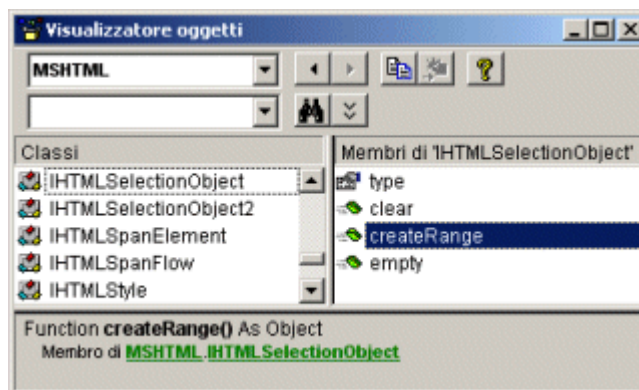


Figura 6

Clear ed Empty difficilmente serviranno al nostro scopo, ma piuttosto il metodo **CreateRange** desta il nostro interesse; cliccando e leggendo la sua definizione in fondo scopriamo che restituisce un oggetto dichiarato come Object, ma non sappiamo di quale tipo esattamente e cosa conterrà.

Con la stessa tecnica di prima interrogheremo il risultato della chiamata al metodo CreateRange; per far ciò richiameremo la finestra *Aggiungi Espressione di controllo*, avendo cura di specificare il nome del nuovo oggetto da analizzare, sotto l'etichetta *Espressione*, come mostrato nella Figura 7.

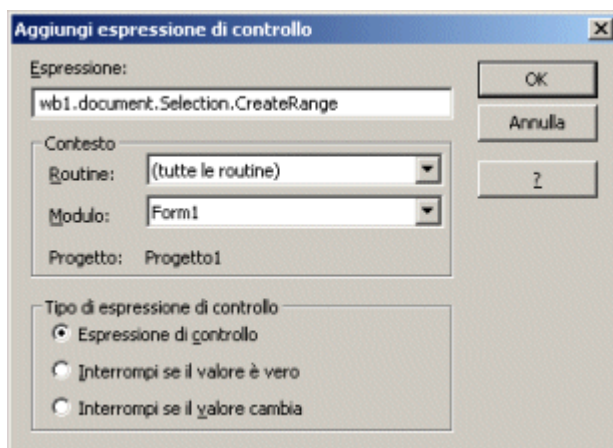


Figura 7

Verrà aggiunto un altro watch, e scopriremo subito che si tratta di un oggetto di classe **IHTMLTxtRange**.



Figura 8

Possiamo anche notare subito che al suo interno sono contenute due proprietà di tipo String

che contengono esattamente il testo che avevamo selezionato in precedenza. Il nostro viaggio sembra terminato ma vogliamo procedere oltre, e conoscere il contenuto dell'oggetto di classe IHTMLTxtRange.

Cerchiamo perciò detta classe nell'Object Browser e troveremo due proprietà di nome **text** e **htmltext** (le stesse mostrate all'interno del Watch), assieme a tante altre funzioni che il Watch non presenta. Ciascuna di queste potrebbe richiedere ulteriori analisi combinazioni da Watch + Object Browser.

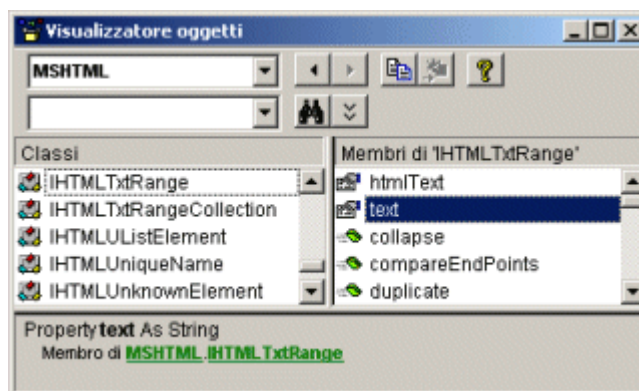


Figura 9

Possiamo quindi modificare il nostro codice, rimuovendo la riga Stop e sostituendola ad esempio con la seguente:

```
MsgBox wbl.Document.selection.createRange.Text, vbInformation
```

Ottenendo così la visualizzazione del testo evidenziato in precedenza.

Lo scopo di questo tutorial non è quello di mostrare in una finestra il testo selezionato in un oggetto WebBrowser ma piuttosto quello di insegnare come recuperare queste informazioni, come scoprire i contenuti degli oggetti complessi e come orientarsi di fronte lo smarrimento dovuto ad un componente sconosciuto.

Watch ed Object Browser accoppiati possono divenire degli strumenti indispensabili per l'analisi delle librerie ActiveX e degli oggetti sia in fase di progettazione (Object Browser) che di esecuzione (Watch ed Object Browser). Sono convinto che **oltre il 60%** dei problemi legati alla difficile implementazione di una soluzione già nota possono essere risolti conoscendo l'uso di questi due potentissimi strumenti.

[Fibia.FBI](#)

5 Dicembre 2003

Ripubblicato il 21 Gennaio 2004



[Torna all'indice degli articoli](#)