

# Tricks & Tips

[Access]

- *Michele de Nittis* -

---

## Tutorial - ActiveX - Controllo Tree View

---

Versione: 1

Data Versione: venerdì 30 agosto 2002

## **SOMMARIO**

<b>PREMESSA.....</b>	<b>3</b>
<b>INSERIMENTO DEL CONTROLLO.....</b>	<b>3</b>
<b>AGGIUNTA DELLE VOCI AL CONTROLLO .....</b>	<b>4</b>
<b>L'EVENTO NODECLICK().....</b>	<b>7</b>
<b>CONSIDERAZIONI SULL'INSTALLAZIONE DEL CONTROLLO TREEVIEW .....</b>	<b>9</b>

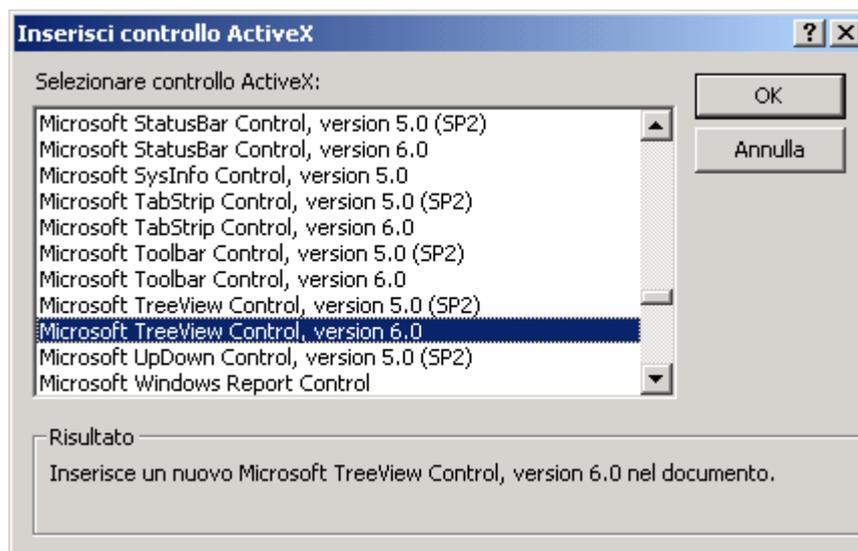
## Premessa

Questo breve tutorial si propone di illustrare l'impiego del controllo ActiveX Microsoft Tree View Control Versione 6 all'interno di una maschera di un applicativo Access. Questo controllo si presenta come un insieme di voci organizzate secondo una struttura ad albero. Ogni voce corrisponde ad un nodo dell'albero.

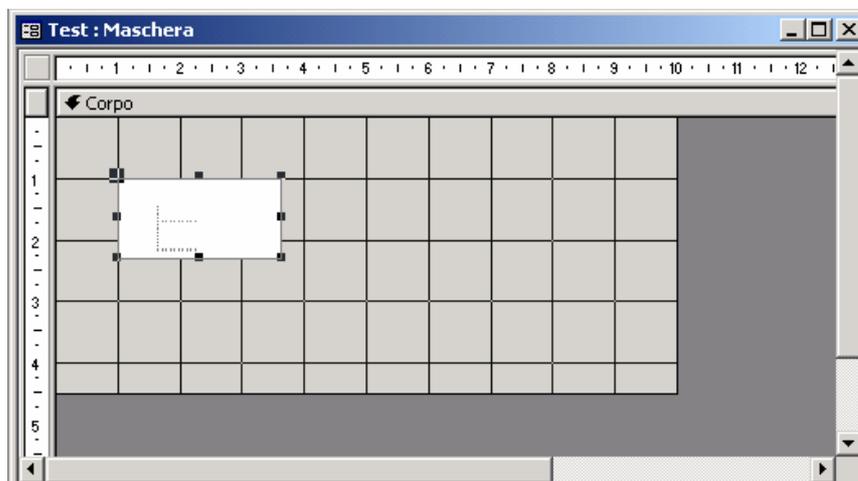
Quanto verrà detto sul componente in questione, ovviamente, è di validità generale potendosi applicare, con pochissime differenze, anche in altri ambienti di programmazione.

## Inserimento del Controllo

Entrare in modalità *Struttura* della maschera dove si vuol collocare il controllo ActiveX. Selezionare la voce "Controllo ActiveX" dal menu "Inserisci" e scegliere la voce "*Microsoft TreeView Control, version 6.0*" dalla lista della finestra "Inserisci controllo ActiveX". Premere il bottone OK.



Trascinare la sagoma del controllo nella posizione desiderata.



## Aggiunta delle voci al controllo

Come già detto in premessa, ad ogni voce del controllo corrisponde un nodo dell'albero. Inizialmente l'albero risulta privo di nodi, cosicché è necessario sfruttare un evento per eseguire un procedura (Sub) di popolamento dell'albero. Possiamo, ad esempio, sfruttare l'evento **Form\_Load()** della maschera che contiene il componente.

L'oggetto di classe *TreeView*, appartenente alla libreria *MSComCtlLib*, è in relazione *Has Many* con l'oggetto di classe *Node* appartenente alla medesima libreria. Infatti tra le sue proprietà figura la collezione *Nodes* di oggetti *Node* che stabilisce la predetta relazione.

Esaminiamo alcuni dettagli della classe *Node* e della collezione *Nodes*. Le proprietà più significative dell'oggetto *MSComCtlLib.Node* sono le seguenti:

<b>Checked</b>	Boolean	Determina (r/w) se l'oggetto risulta selezionato, nel qual caso restituisce il valore true.
<b>Child</b>	(Property) Node	E' l'oggetto classe <i>Nodo</i> gerarchicamente inferiore all'oggetto corrente. Disponibile solo a tempo di esecuzione.
<b>Expanded</b>	Property (Boolean)	Determina (r/w) se il nodo è collassato o espanso nei suoi sottonodi: true per espanso, false per collassato.
<b>Index</b>	(Property) Integer	E' l'identificatore numerico univoco per l'oggetto di classe <i>Nodo</i> corrente. Coincide con il numero d'ordine del nodo all'interno della collezione <i>Nodes</i> e può variare dinamicamente con l'inserimento e la rimozione di nodi dalla predetta collezione.
<b>Key</b>	(Property) String	E' una stringa identificativa univoca di un nodo all'interno di un controllo <i>TreeView</i> .
<b>Next</b>	(Property) Node	E' l'oggetto classe <i>Nodo</i> che segue l'oggetto corrente nel medesimo livello dell'albero.
<b>Parent</b>	(Property) Node	E' l'oggetto classe <i>Nodo</i> gerarchicamente superiore all'oggetto corrente. Disponibile solo a tempo di esecuzione.
<b>Previous</b>	(Property) Node	E' l'oggetto classe <i>Nodo</i> che precede l'oggetto corrente nel medesimo livello dell'albero.
<b>Root</b>	(Property) Node	E' il nodo radice dell'albero di cui l'oggetto di classe <i>Nodo</i> corrente è parte.
<b>Text</b>	String	E' la stringa di testo con cui l'oggetto viene graficamente visualizzato nell'albero.
<b>Visibile</b>	Boolean	Determina se l'oggetto risulta visibile (true) o nascosto (false)

I metodi più significativi della collezione *Nodes* sono i seguenti:

<b>Add( ... )</b>	Node	Aggiunge un nuovo nodo alla collezione di nodi ' <i>Nodes</i> ' del componente TreeView
<b>Remove(item)</b>	Void	Rimuove un l'elemento indicato dal parametro <item> dalla collezione di nodi ' <i>Nodes</i> ' del controllo TreeView. <item> è l'indicatore univoco del nodo da rimuovere e può essere un numero (prop. Index) o una stringa (prop. Key).

Esaminiamo la sintassi del metodo **Add()** che è quello che ci consente di aggiungere voci al controllo TreeView:

TreeView.Nodes.**Add**(*relative, relationship, key, text, image, selectedimage*)

Questo metodo consente di creare un nuovo nodo e di aggiungerlo alla collezione *Nodes*, restituendo al chiamante un riferimento al nodo stesso.

Non si deve, quindi, creare un nuovo oggetto di classe *Node* ed aggiungerlo alla collezione perché a questo provvede già il metodo **Add**.

### **NOTA**

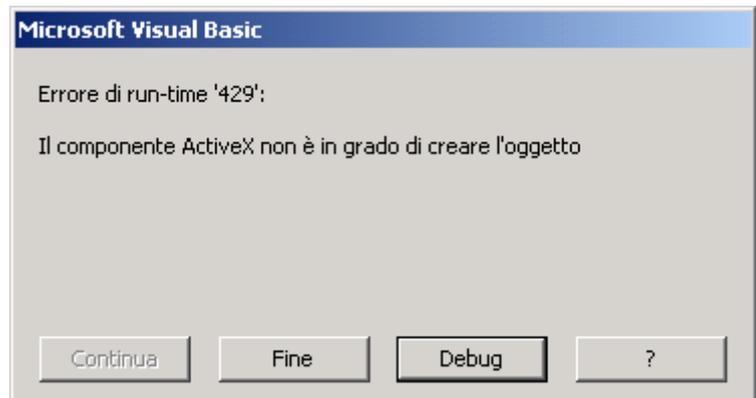
Se si tentasse di creare direttamente un nuovo oggetto di classe *Node* e di copiarne il riferimento in una variabile, Visual Basic reagirebbe con un errore di tipo '429'.

### ***Esempio:***

Il seguente frammento di codice genera un errore 429.:

```
Option Compare Database

Private Sub Form_Load()
Dim provaNode As MSComctlLib.Node
Set provaNode = New Node
End Sub
```



Tra i parametri che devono essere forniti al metodo, quindi, ve ne sono alcuni necessari per identificare univocamente un oggetto di classe *Node* e per stabilirne il grado di correlazione con gli altri nodi della collezione *Nodes*.

Il dettaglio dei parametri formali del metodo **Add()** è il seguente:

<i>Relative</i>	Opzionale. Rappresenta l'identificatore (numero o stringa) univoco di un nodo già presente nella collezione <i>Nodes</i> con il quale quello che da aggiungere è legato da una relazione (vedi parametro <i>Relationship</i> ).
<i>Relationship</i>	Opzionale. Rappresenta il tipo di relazione che lega il nodo da aggiungere ad uno o più nodi della collezione <i>Nodes</i> ( <i>vedere tabella successiva</i> ).
<i>Key</i>	Opzionale: corrisponde alla omonima proprietà della classe <i>Node</i> .
<i>Text</i>	Obbligatorio: corrisponde alla omonima proprietà della classe <i>Node</i>
<i>Image</i>	Opzionale: è l'indice dell'immagine, presente nel controllo <b>ImageList</b> associato all'albero, che viene usata per visualizzare il nodo nell'albero;
<i>Selectedimage</i>	Opzionale: come il parametro precedente ma l'immagine è quella che viene mostrata quando il nodo è selezionato.

I valori ammissibili per il parametro *Relationship* sono:

<b>tvwFirst</b>	0	Il nodo viene posizionato nell'albero <u>prima</u> di tutti gli altri nodi di livello pari a quello del nodo specificato nel parametro <i>Relative</i> .
<b>tvwLast</b>	1	Il nodo viene posizionato nell'albero <u>dopo</u> tutti gli altri nodi di livello pari a quello del nodo specificato nel parametro <i>Relative</i> .
<b>tvwNext</b>	2	Il nodo viene posizionato nell'albero <u>dopo</u> il nodo specificato nel parametro <i>Relative</i> e nello stesso livello.
<b>tvwPrevious</b>	3	Il nodo viene posizionato nell'albero <u>prima</u> del nodo specificato nel parametro <i>Relative</i> e nello stesso livello.
<b>tvwChild</b>	4	Il nodo diventa uno dei figli del nodo specificato nel parametro <i>Relative</i> .

Nella sottostante figura, è riportato un esempio di applicazione del metodo **Add**.

```
Option Compare Database
Public tmpnode As MSComctlLib.Node
Private Sub Form_Load()
' Aggiunta delle finanziarie
Set tmpnode = Me.TreeView2.Nodes.Add(, , "Fin01", "Finanziaria 1")
tmpnode.Expanded = True
Set tmpnode = Me.TreeView2.Nodes.Add(, , "Fin02", "Finanziaria 2")
tmpnode.Expanded = True
Set tmpnode = Me.TreeView2.Nodes.Add(, , "Fin03", "Finanziaria 3")
tmpnode.Expanded = True
```

Vediamo adesso un semplice esempio di controllo TreeView:



il codice per ottenere il controllo in figura è il seguente:

#### Option Compare Database

```
Private Sub Form_Load()
' Aggiungo tre voci di primo livello
Dim tmpnode As MSComctlLib.Node
Set tmpnode = Me.TreeView0.Nodes.Add(, , "Fin01", "Finanziaria 1")
tmpnode.Expanded = True
Set tmpnode = Me.TreeView0.Nodes.Add(, , "Fin02", "Finanziaria 2")
tmpnode.Expanded = True
Set tmpnode = Me.TreeView0.Nodes.Add(, , "Fin03", "Finanziaria 3")
tmpnode.Expanded = True

' Aggiungo a Fin01 una sottovoce
Set tmpnode = Me.TreeView0.Nodes.Add("Fin01", tvwChild, "Mattoni1Fin01", "Mattoni")
tmpnode.Expanded = False

End Sub
```

Si osservi che, qualora non venga specificato alcun valore per il parametro *Relative*, il nodo creato viene aggiunto come ultimo tra i nodi del livello più alto.

## L'Evento NodeClick()

Questo evento occorre ogni volta che l'utente *seleziona* un nodo del controllo TreeView. La procedura (Sub) che intrappola e gestisce l'evento ha la seguente sintassi dichiarativa:

```
Private Sub TreeViewObject_NodeClick(ByVal Node as Object)1
```

### **Esempio:**

Il seguente frammento di codice implementa la procedura '*NodeClick()*', determina il livello del nodo selezionato e scrive, in altri appositi controlli della maschera di appartenenza, il valore dei nodi gerarchicamente superiori.

<sup>1</sup> L'help in linea mostra la seguente sintassi, leggermente differente e scorretta (Art. Microsoft Knowledge Base n° Q172167):

```
Private Sub TreeViewObject_NodeClick(ByVal Node as Node)
```

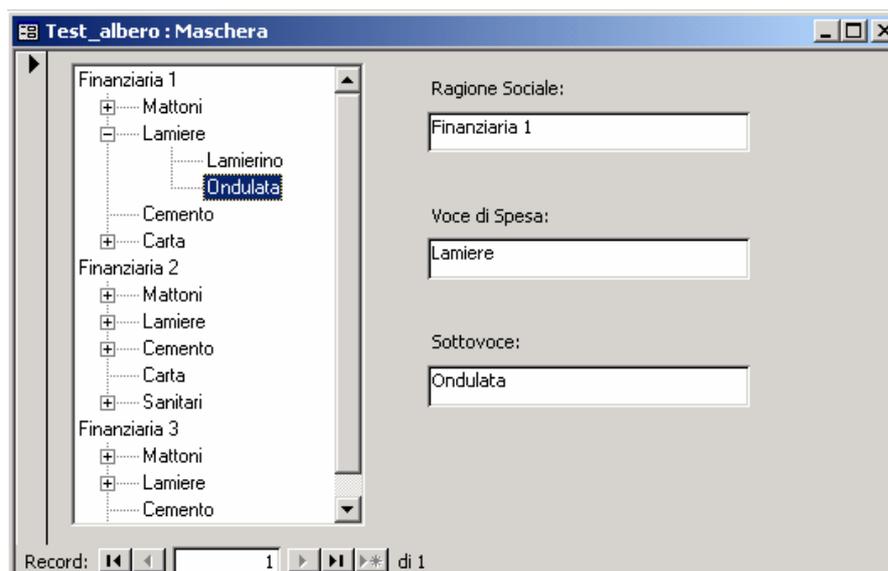
```

Private Sub TreeView2_NodeClick(ByVal Nodo As Object)
Dim level As Integer
Dim tmpnode As MSComctlLib.Node
level = 0
' individuazione del tipo di nodo in base al suo livello nell'albero
Set tmpnode = Nodo
While Not (tmpnode.Parent Is Nothing) = True
    level = level + 1
    Set tmpnode = tmpnode.Parent
Wend

Set tmpnode = Nodo
Select Case level
    Case 2
        Me.R_Sottovoce.Value = tmpnode.Text
        Set tmpnode = tmpnode.Parent
        Me.R_Voce.Value = tmpnode.Text
        Set tmpnode = tmpnode.Parent
        Me.R_Finanziaria.Value = tmpnode.Text
    Case 1
        Me.R_Voce.Value = tmpnode.Text
        Set tmpnode = tmpnode.Parent
        Me.R_Finanziaria.Value = tmpnode.Text
        Me.R_Sottovoce.Value = ""
    Case 0
        Me.R_Finanziaria.Value = tmpnode.Text
        Me.R_Sottovoce.Value = ""
        Me.R_Voce.Value = ""
End Select
End Sub

```

Il risultato del precedente frammento è visualizzato nella seguente figura:



## Considerazioni sull'installazione del controllo TreeView

Affinché l'applicativo che fa uso del controllo TreeView funzioni correttamente è necessario che nel sistema sia installato il gruppo di controlli ActiveX MSComCtl. Questo si trova nel file **comctl32.ocx** che deve essere presente nella directory **System** o **System32**.