

# **Tutorial**

**[Access]**

*- Michele de Nittis -*

---

## **Tutorial - Activex - Controllo List View**

---

Versione: 1

Data Versione: martedì 21 dicembre 2004

## SOMMARIO

<b>PREMESSA.....</b>	<b>3</b>
<b>INSERIMENTO DEL CONTROLLO .....</b>	<b>3</b>
<b>ARCHITETTURA DEL CONTROLLO LIST VIEW .....</b>	<b>4</b>
<b>AGGIUNTA DELLE VOCI AL CONTROLLO .....</b>	<b>5</b>
Modalità Icon View, Small Icon View e List View.....	5
La modalità Report View.....	8
<b>ALCUNE IMPORTANTI PROPRIETÀ DEL CONTROLLO LISTVIEW .....</b>	<b>10</b>
<b>COME LEGGERE I VALORI DEL CONTROLLO SELEZIONATI DALL'UTENTE .....</b>	<b>14</b>

## Premessa

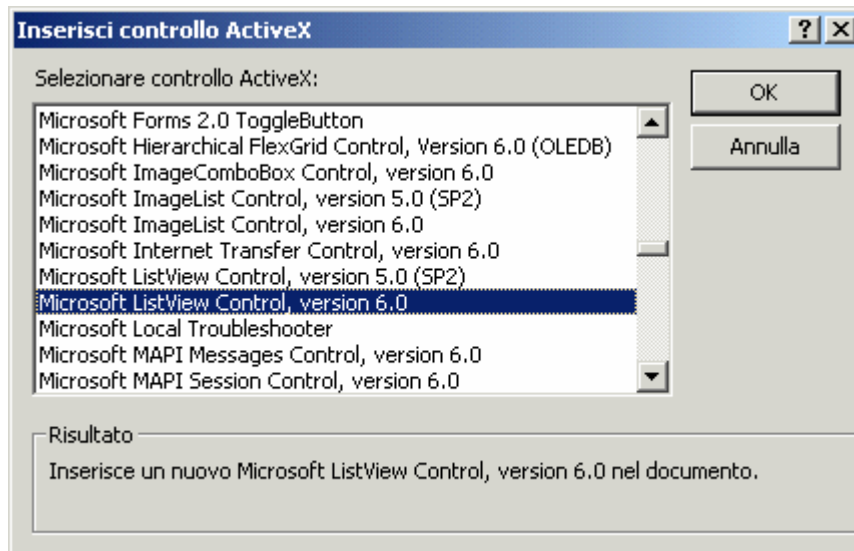
Questo breve tutorial si propone di illustrare l'impiego del controllo ActiveX Microsoft List View Control Versione 6 all'interno di una maschera di un applicativo Access. Questo controllo si presenta come un insieme di voci organizzate in uno dei seguenti quattro modi:

Modo	Descrizione
Icon view	Ogni voce è un elemento grafico costituito da un'icona e da un'etichetta descrittiva. L'utente può spostare ogni elemento in un qualsiasi punto del controllo.
Small icon view	Come il precedente modo ma le icone sono più piccole.
List view	Ogni voce è rappresentata da un elemento testuale accompagnato da un'eventuale piccola icona. Gli elementi vengono mostrati incolonnati dall'alto verso il basso e da sinistra verso destra ed allineati a sinistra.
Report view	Questo modo mostra le voci in modo tabellare. Ogni voce è una riga di una tabella ed è costituita da molteplici informazioni testuali e grafiche disposte lungo le colonne.

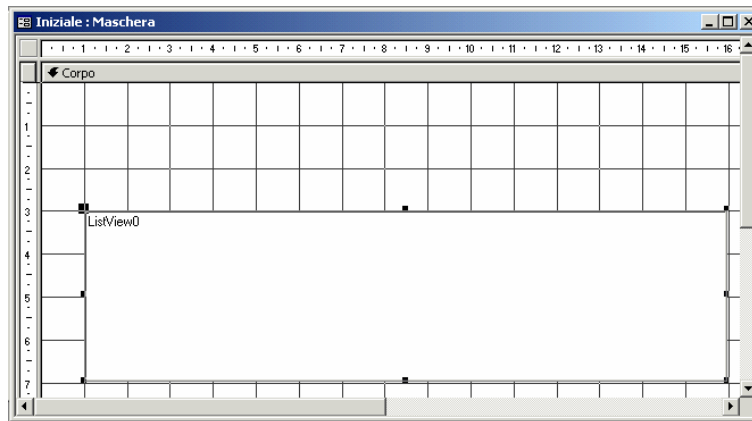
Quanto verrà detto sul componente in questione, ovviamente, è di validità generale potendosi applicare, con pochissime differenze, anche in altri ambienti di programmazione.

## Inserimento del Controllo

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

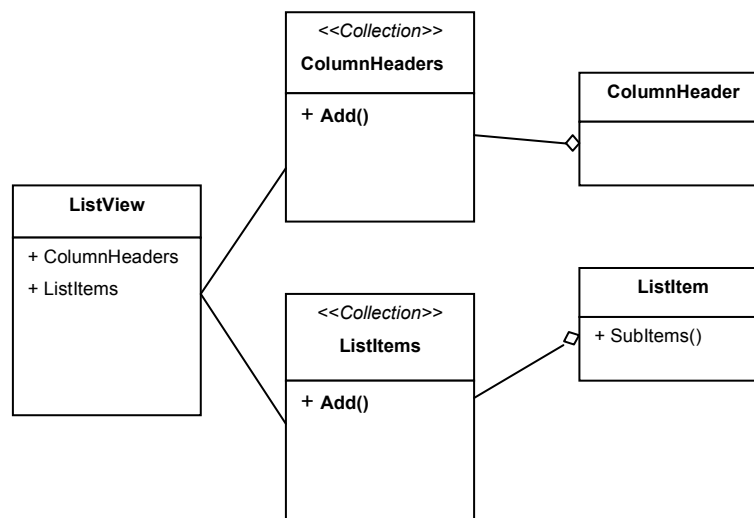


Trascinare la sagoma del controllo nella posizione desiderata e farli assumere le dimensioni preferite.



## Architettura del controllo List View

L'architettura del controllo si può schematizzare come segue:



Come si vede dal suddetto schema, il controllo `ListView` usa due collezioni: `ColumnHeaders`, che è la collezione di oggetti di classe `ColumHeader` che rappresentano le intestazioni delle colonne del controllo in modalità `Report View`, e `ListItems`, che è la collezione di oggetti `ListItems`, cioè gli elementi testuali e grafici che rappresentano le voci da visualizzare. In modalità `Report View`, inoltre, è possibile specificare, per ogni voce, tutte le rispettive sottovoci mediante l'array `SubItems()`.

Nel seguito illustreremo, mediante un esempio, come popolare un controllo `ListView` con delle informazioni testuali e grafiche. Per quanto concerne i dettagli sulle modalità di inserimento di informazioni grafiche (icone), si consiglia la lettura del tutorial [2] di questa stessa serie per ulteriori approfondimenti.

## Aggiunta delle voci al controllo

Esaminiamo prima le modalità `Icon View`, `Small Icon View` e `List View`, più semplici, per poi approfondire la modalità `Report View` più articolata e completa. Per impostare la modalità di visualizzazione si deve assegnare alla proprietà pubblica `View` uno dei seguenti valori costanti:

Costante	Valore	Visualizzazione
<code>lvwReport</code>		Report View
<code>lvwIcon</code>		Icon View
<code>lvwList</code>		List View
<code>lvwSmallIcon</code>		Small Icon View

### Modalità `Icon View`, `Small Icon View` e `List View`

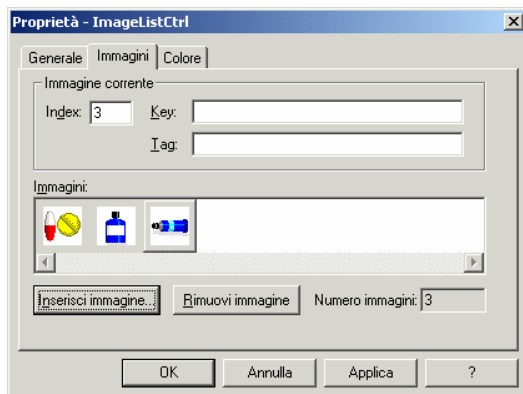
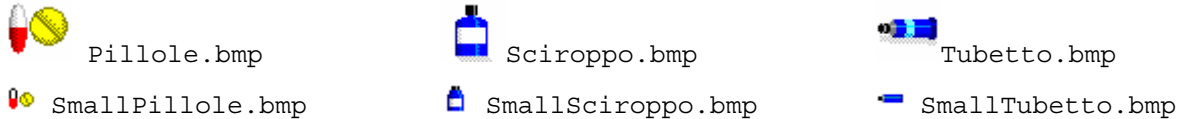
Per aggiungere un elemento di tipo `ListItem` al controllo `ListView` si deve usare il metodo `Add()` della collezione `ListItems`, la cui segnatura è la seguente:

```
Object.Add(index, key, text, icon, smallicon)
```

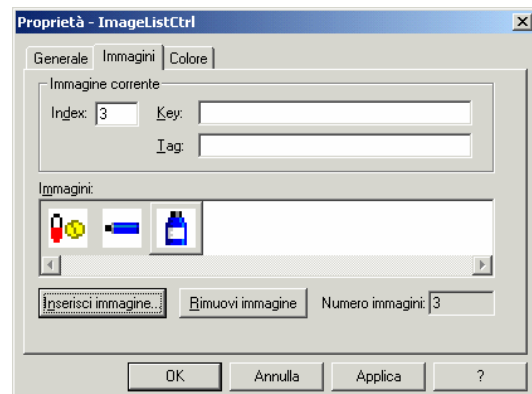
<code>Index</code>	<b>Opzionale.</b> È un numero che indica la posizione della voce all'interno della collezione di voci. Se omissso, la voce viene inserita in coda alla collezione. Il suo valore varia in funzione di successivi inserimenti e cancellazioni.
<code>Key</code>	<b>Opzionale.</b> E' una stringa che rappresenta la chiave identificativa <b>univoca</b> di ogni voce nella collezione. Utile per la ricerca di una voce.
<code>Text</code>	<b>Opzionale.</b> Rappresenta la parte testuale della voce. Viene mostrata nel controllo, eventualmente associata ad una icona.
<code>Icon</code>	<b>Opzionale.</b> Può essere un numero o una stringa e rappresenta, rispettivamente, l'indice o la chiave identificativa per fare riferimento ad un'immagine del controllo <code>ImageList</code> associato all'oggetto <code>ListView</code> (proprietà <b>Icon</b> ).
<code>SmallIcon</code>	<b>Opzionale.</b> Come il precedente parametro ma fa riferimento alla proprietà <b>SmallIcon</b> .

## Esempio.

Aggiungiamo alla maschera di prova "Iniziale" due controlli di tipo ImageList (Medicinali e SmallMedicinali) contenenti, rispettivamente, i seguenti due gruppi di icone:



Controllo Medicinali



Controllo SmallMedicinali

Per popolare il controllo di voci impieghiamo l'evento **Form\_Load()** (su Caricamento) della finestra "Iniziale":

```
Private Sub Form_Load()
Dim LI As ListItem
' Aggiunta di uno sciroppo
Set LI = ListView0.ListItems.Add(, "Glu001", "Glutax", "Sciropo", "SmallSciropo")
' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "Asp001", "Aspirina", "Pillole", "SmallPillole")
' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "VtCO01", "Vitamina C", "Pillole", "SmallPillole")
' Aggiunta di una Pomata
Set LI = ListView0.ListItems.Add(, "Tmt001", "Traumatone", "Pomata", "SmallPomata")
' Aggiunta di una Pomata
Set LI = ListView0.ListItems.Add(, "Antm001", "Antimicosix", "Pomata", "SmallPomata")
' Aggiunta di uno sciroppo
Set LI = ListView0.ListItems.Add(, "Flu001", "FluidTox", "Sciropo", "SmallSciropo")
' modo di visualizzazione di default - Icon View
Me.ListView0.View = lvwIcon
End Sub
```

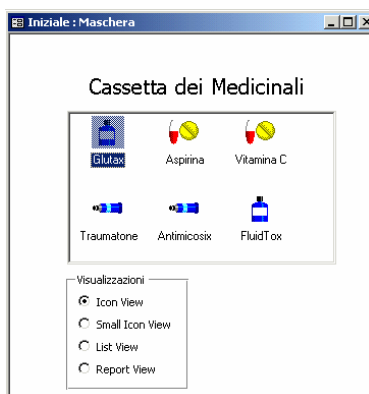
Il risultato del predetto codice è mostrato nella seguente figura:



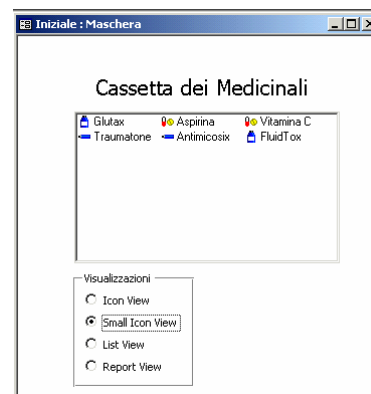
Per visualizzare i diversi modi con cui il controllo ListView mostra le informazioni, inseriamo nella maschera "Iniziale" un gruppo di pulsanti d'opzione e gestiamo l'evento *AfterUpdate* (Dopo Aggiornamento).

```
Private Sub Cornice4_AfterUpdate()
Dim scelta As Integer
scelta = Me.Cornice4.Value
Select Case scelta
Case 1
' Icon View
Me.ListView0.View = lvwIcon
Case 2
' Small Icon View
Me.ListView0.View = lvwSmallIcon
Case 3
' List View
Me.ListView0.View = lvwList
Case 4
' Report View
MsgBox "Non ancora implementato"
End Select
End Sub
```

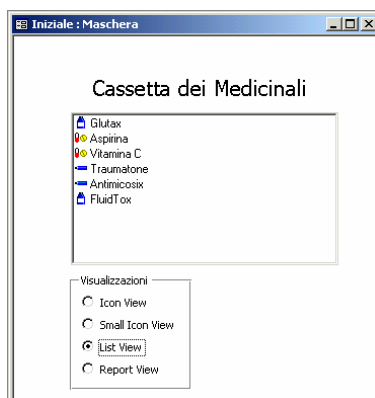
Selezionando le diverse voci si cambia la visualizzazione del controllo List View.



Icon View



Small Icon View



List View

### ***La modalità Report View***

Questa modalità di visualizzazione è la più articolata ed è anche quella con maggior potere espressivo. In sintesi è possibile mostrare ciascuna delle voci del controllo mediante informazioni di dettaglio testuali e grafiche.

Come abbiamo visto nei precedenti modi di visualizzazione, per inserire una nuova voce nel controllo ListView è necessario creare un apposito oggetto **ListItem** ed aggiungerlo alla collezione **ListItems** mediante il metodo **Add()**. Per aggiungere informazioni di dettaglio ad una voce è possibile usare l'array **SubItems** del corrispondente oggetto **ListItem**.

Ad esempio, se volessimo visualizzare informazioni di dettaglio, quali la tipologia farmaceutica e la data di scadenza di ogni farmaco visualizzato nella farmacia domestica dell'esempio precedente, si dovrebbero inserire tali informazioni nell'array SubItems di ogni oggetto **ListItem**, come illustra il seguente frammento di codice.



```

' Aggiunta di uno sciroppo
Set LI = ListView0.ListItems.Add(, "Glu001", "Glutax", "Sciroppo", "SmallSciroppo")
LI.SubItems(1) = "Lassativo"
LI.SubItems(2) = "12/12/2004"

' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "Asp001", "Aspirina", "Pillole", "SmallPillole")
LI.SubItems(1) = "Febbrifugo"
LI.SubItems(2) = "04/09/2003"

' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "VtC001", "Vitamina C", "Pillole", "SmallPillole")
LI.SubItems(1) = "Ricostituente"
LI.SubItems(2) = "19/05/2005"

' Aggiunta di una Pomata
Set LI = ListView0.ListItems.Add(, "Tmt001", "Traumatone", "Pomata", "SmallPomata")
LI.SubItems(1) = "Unguento anti trauma"
LI.SubItems(2) = "25/02/2005"

' Aggiunta di una Pomata
Set LI = ListView0.ListItems.Add(, "Antm001", "Antimicosix", "Pomata", "SmallPomata")
LI.SubItems(1) = "Antimicotico"
LI.SubItems(2) = "25/02/2005"

' Aggiunta di uno sciroppo
Set LI = ListView0.ListItems.Add(, "Flu001", "FluidTox", "Sciroppo", "SmallSciroppo")
LI.SubItems(1) = "Fluidificatore Muco"
LI.SubItems(2) = "14/11/2003"

```

Si devono, poi, aggiungere alla collezione **ColumnHeaders** dell'oggetto **ListView**, mediante il metodo **Add()**, tanti oggetti di classe **ColumnHeader** quante sono le colonne che si vogliono mostrare. Il metodo **Add()** della collezione **ColumnHeaders** ha la seguente sintassi:

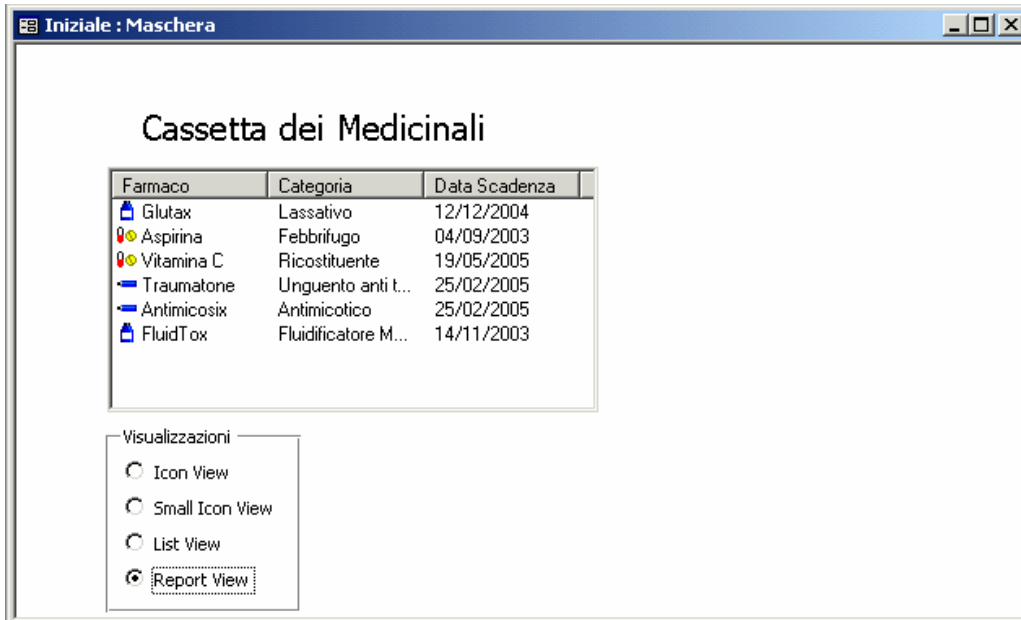
`Object.Add(indice, chiave, testo, larghezza, allineamento, icona)`

<b>indice</b>	<b>Opzionale.</b> È un numero che indica la posizione della colonna.		
<b>chiave</b>	<b>Opzionale.</b> E' una stringa che rappresenta la chiave identificativa <b>univoca</b> di ogni voce nella collezione. Utile per la ricerca di una voce.		
<b>testo</b>	<b>Opzionale.</b> Rappresenta la parte testuale di un'intestazione di colonna. Può essere associata ad una icona.		
<b>larghezza</b>	<b>Opzionale.</b> Espressione numerica che specifica la larghezza dell'oggetto, in base all'unità di scala utilizzata dal contenitore del controllo.		
<b>allineamento</b>	<b>Opzionale.</b> Facoltativa. Valore intero che determina l'allineamento del testo nell'oggetto <b>ColumnHeader</b> .		
	<b>lvwColumnLeft</b>	0	(Predefinita) Sinistra. Il testo viene allineato a sinistra.
	<b>lvwColumnRight</b>	1	Destra. Il testo viene allineato a destra.
	<b>lvwColumnCenter</b>	2	Centrato. Il testo viene centrato.
<b>icona</b>	<b>Opzionale.</b> Può essere un numero o una stringa e rappresenta, rispettivamente, l'indice o la chiave identificativa di un'immagine del controllo <b>ImageList</b> associato all'oggetto <b>ListView</b> (proprietà <b>ColumnHeaderIcons</b> ).		

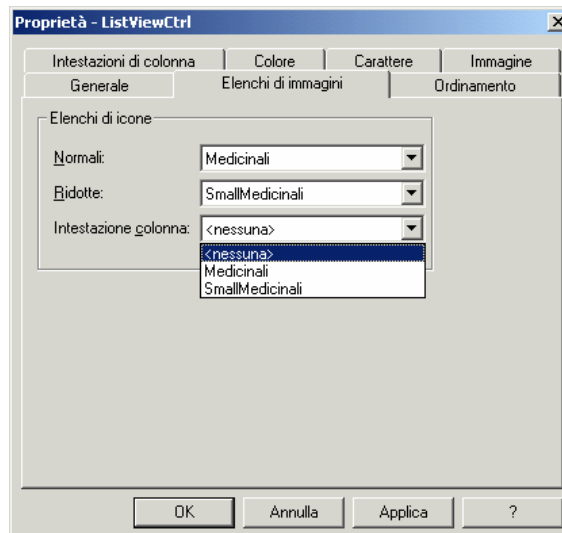
**Esempio:**

```
' impostiamo tre intestazioni di colonna
Set CH = ListView0.ColumnHeaders.Add( , , "Farmaco")
Set CH = ListView0.ColumnHeaders.Add( , , "Categoria")
Set CH = ListView0.ColumnHeaders.Add( , , "Data Scadenza")
```

Il risultato del nuovo codice aggiunto è il seguente:

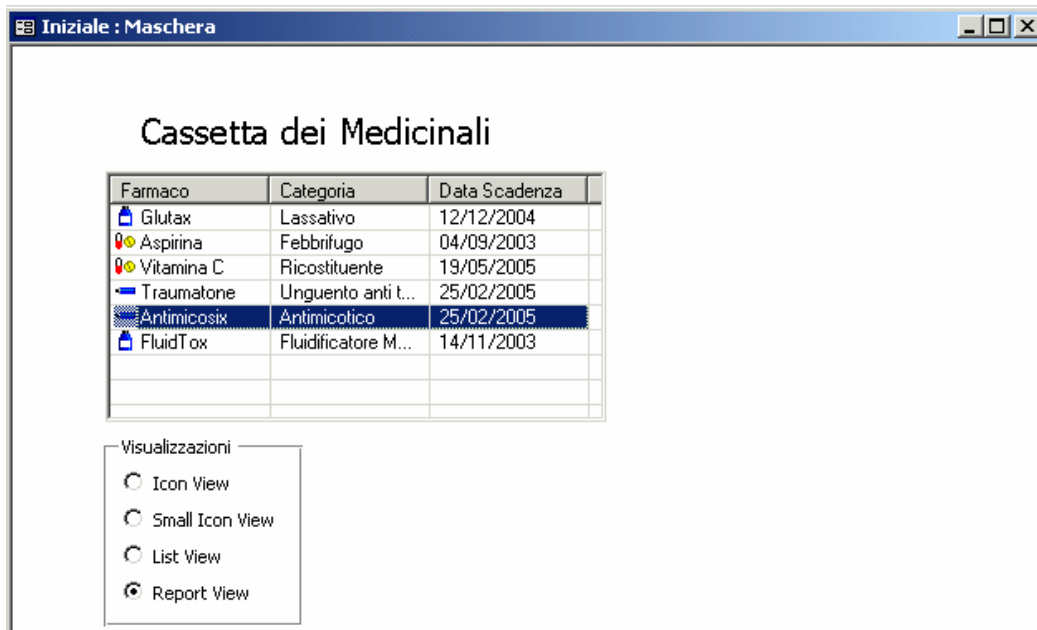


Per associare a ciascun'intestazione un'icona, si deve impostare la proprietà **ColumnHeaderIcons** del controllo **ListView** in fase di definizione della struttura della maschera (*Design Time*).

**Alcune importanti Proprietà del controllo ListView**

L'oggetto ListView, dalla versione 6, è dotato di molte proprietà che ne migliorano il grado d'usabilità.

- **AllowColumnReorder:** Booleano. Se impostato a TRUE consente all'utente di riordinare le colonne a tempo d'esecuzione mediante il mouse. Si provi a selezionare l'intestazione di una colonna e, tenendo premuto il tasto sinistro del mouse, la si trascini sopra una delle altre intestazioni.
- **FullRowSelect:** Booleano. Se impostato a TRUE consente all'utente di selezionare un'intera riga, se a FALSE solo la voce della prima colonna.
- **GridLines:** Booleano. Se impostato a TRUE, mostra una griglia nel controllo.



- **CheckBoxes:** Booleano. Specificando True è possibile far comparire una *check box* per ogni voce inserita nel controllo, come mostra la seguente figura.



Per far comparire la *check box* di una determinata voce (nell'esempio di figura "Vitamina C") nello stato selezionato, basta impostare a *True* il valore della proprietà *Checked* del corrispondente oggetto di classe **ListItem**, come mostra il seguente listato.

```

' imposto visualizzazione con checkbox
Me.ListView0.CheckBoxes = True

' Aggiunta di uno sciroppo
Set LI = ListView0.ListItems.Add(, "Glu001", "Glutax", "Sciroppo", "SmallSciroppo")
LI.Checked = False
LI.SubItems(1) = "Lassativo"
LI.SubItems(2) = "12/12/2004"

' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "Asp001", "Aspirina", "Pillole", "SmallPillole")
LI.Checked = False
LI.SubItems(1) = "Febbrifugo"
LI.SubItems(2) = "04/09/2003"

' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "VtC001", "Vitamina C", "Pillole", "SmallPillole")
LI.Checked = True
LI.SubItems(1) = "Ricostituente"
LI.SubItems(2) = "19/05/2005"

```

E' possibile ricorrere alla proprietà *Checked* dell'oggetto di classe **Listitem** anche per leggere lo stato del controllo *Check Box* associato ad una determinata voce.

- **HotTracking**: Booleano. Questa proprietà agisce sulla visualizzazione del controllo **ListView** rendendolo più dinamico. L'effetto è simile a quello riscontrabile in alcuni dei più comuni browser web: quando il puntatore si sposta sul controllo cambia forma e l'eventuale riga selezionata cambia colore al suo passaggio.
- **MultiSelect**: Booleano. Se impostato a TRUE consente la selezione multipla di più voci.

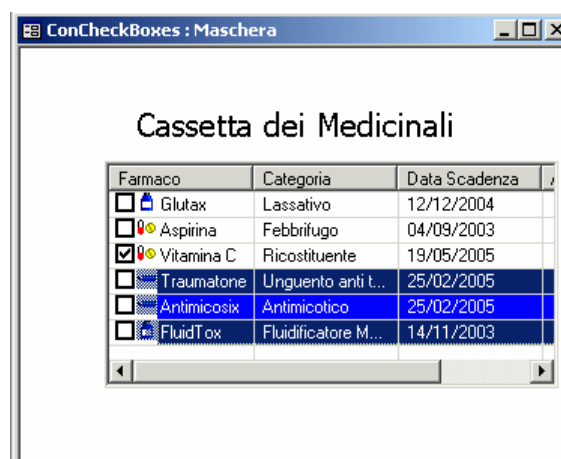
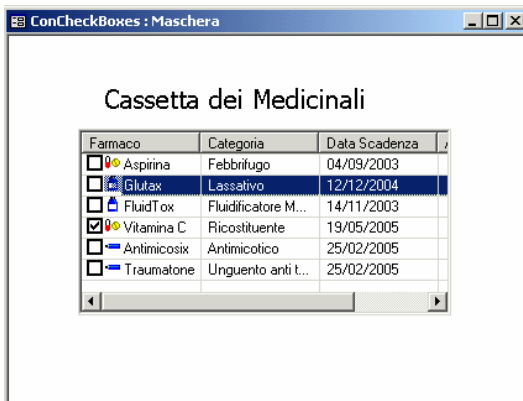


Figura 1: Controllo ListView con le proprietà HotTracking e MultiSelect impostate a True.

- **FlatScrollBar**: Booleano. Impostando a TRUE questa proprietà si cambia lo stile grafico delle eventuali barre di scorrimento laterali.

- **Sorted:** Booleano. Se impostato a TRUE indica che le voci visualizzate nel controllo di tipo **ListView** devono essere ordinate. Impostando, inoltre, le proprietà **SortOrder** e **SortKey** è possibile specificare un particolare tipo di ordinamento (crescente [**lvwAscending=0**] o decrescente [**lvwDescending=1**]) ed una colonna come chiave dell'ordinamento (un numero intero positivo che rappresenta l'indice di quella colonna). Impostando **SortKey=0** (valore di default), l'ordinamento (crescente per default) viene eseguito sulla proprietà **Text** degli oggetti **ListItem** della collezione **ListItems**. Nella seguente figura è illustrato l'ordinamento crescente delle voci del controllo **ListView**, assumendo come chiave dell'ordinamento la terza colonna (OrderKey=2), ovvero la data di scadenza dei medicinali.



```
Me.ListView0.Sorted = True|
Me.ListView0.SortKey = 2
Me.ListView0.SortOrder = lvwAscending
```

Figura 2: Ordinamento sulla data di scadenza

### ATTENZIONE:

L'ordinamento non è stato eseguito sui valori dei dati contenuti nella terza colonna ma sulle stringhe rappresentative di quei dati. L'ordinamento, pertanto, è testuale.

- **SelectedItem:** è una proprietà in lettura e scrittura. Consente di **leggere** l'oggetto di classe **ListItem** che è stato selezionato dall'utente o di **impostare** una particolare voce del controllo come selezionata per default.

```

' Aggiunta di Pillole
Set LI = ListView0.ListItems.Add(, "VtC001", "Vitamina C", "Pillole", "SmallPillole")
LI.Checked = True
LI.SubItems(1) = "Ricostituente"
LI.SubItems(2) = "19/05/2005"
Set Me.ListView0.SelectedItem = LI

```

### Listato 1: Uso della proprietà SelectedItem in scrittura.

Molte altre sono le proprietà del controllo ListView che non riportiamo nel presente tutorial, il cui scopo non è quello di fare una trattazione esaustiva di questo particolare controllo ma quello di fornire una panoramica generale del suo funzionamento e delle modalità d'impiego. Ulteriori e più dettagliate informazioni sono reperibili nell'help in linea di Visual Basic.

## Come leggere i valori del controllo selezionati dall'utente

Per accedere ai valori della voce del controllo (riga) selezionata dall'utente è necessario recuperare il corrispondente oggetto di classe ListItem.

```
Set LI= ListView0.selectedItem
```

Una volta ottenuto l'oggetto di classe ListItem è possibile leggere i valori delle sue proprietà:

```
LI.text;
```

```
LI.SubItems(x), con  $x \in \{1, 2, 3, \dots, N\}$ 
```

Per conoscere il numero N di oggetti di classe ListItem associati ad un oggetto di classe ListView, basta ricordare che la proprietà ListItems dell'oggetto ListView è un oggetto di classe Collection. Dunque:

```
N = ListView0.ListItems.count()
```

## Vedere Anche

[1] Tutorial **Il Controllo TreeView (ActiveX)**;

[2] Tutorial **Oggetto Imegelist: Aggiungere immagini ad un controllo 'Common Control'**.