

## آموزش کار با کنترل Listview - فصل ۱

قبل از اینکه آموزش کار با کنترل Listview را آغاز کنیم، شما به این کنترلها نیاز خواهید داشت:

نوع شی	نام شی	توضیحات
Listview	Listview1	فقط آن را روی فرم قرار دهید
ImageList	imlSmallIcon	یک آیکون با ابعاد ۱۶×۱۶ به آن اضافه کنید
ImageList	ImlLargeIcon	یک آیکون با ابعاد ۳۲×۳۲ به آن اضافه کنید

### بخش اول - اضافه کردن آیتمها به Listview

برای اینکه بتوانیم با Listview هر کاری را انجام دهیم ابتدا باید تعدادی آیتم به آن اضافه کنیم. این کار آسان است، همه چیزی که شما احتیاج دارید این است که یک شی از ListItem بسازید و سپس به یک خط برای ساختن آیتم احتیاج خواهید داشت. به این مثال توجه کنید:

```
Dim itmx As ListItem
```

```
Set itmx = ListView1.ListItems.Add(, , "Item1")
```

این همه آن چیزی است که برای یک آیتم ساده به آن احتیاج دارید. کد فوق را به رویداد Form\_load از فرم برنامه اضافه کنید و آن را اجرا کنید.

شما می توانید آن آیتم را در Listview ببینید، اما آن تا اندازه ای ساده به نظر می رسد. در اینجا هیچ تصویری نمایش داده نشده است. برای نمایش تصاویر در Listview شما باید آن را به یک کنترل ImageList (یا چند کنترل ImageList) الحاق کنید. برای انجام این کار فقط روی خاصیت Custom کلیک کنید و زمانی که صفحه خواص برای Listview نمایش داده شد روی برگه ImageList کلیک کنید. برای گزینه Normal ImageList ، ImlLargeIcon و برای گزینه Small ImageList ، ImlSmallIcons را انتخاب کنید. سپس خارج شوید. شما حالا تصاویری را خواهید داشت که با اجرای VB برای هر کدام از آیتمها خواهید داشت، بزودی تصاویر بسیاری را خواهیم داشت. برای اینکه هنگام اجرا تصویر شماره یک از ImlLargeIcon و تصویر شماره یک از ImlSmallIcon را نمایش دهیم فقط کافی است کد زیر را بنویسیم:

```
Dim itmx As ListItem
```

```
Set itmx = ListView1.ListItems.Add(, , "Item1",1,1)
```

حالا آن را اجرا کنید، شما باید یک آیکون بزرگ را کنار لغت Item1 ببینید.

### بخش دوم تغییر نما

چندین راه برای نمایش اطلاعات در کنترل Listview وجود دارد. مثلاً می توانید به مرورگر ویندوز نگاه کنید که می توانید نمایش اطلاعات را در آن تغییر دهید مثلاً به شکل نمایش آیکون به شکل بزرگ، نمایش آیکون به شکل کوچک، به شکل لیست، یا نمایش بصورت گزارش. برای دیدن هر کدام از حالات مختلف آن را در رویداد Form\_Load از فرم قرار دهید و آن را اجرا کنید.

```
Dim itmx As ListItem  
Dim colx As ColumnHeader
```

```

'Add a bunch of items
Set itm1 = ListView1.ListItems.Add(, , "Item1", 1, 1)
Set itm2 = ListView1.ListItems.Add(, , "Item2", 1, 1)
Set itm3 = ListView1.ListItems.Add(, , "Item3", 1, 1)
Set itm4 = ListView1.ListItems.Add(, , "Item4", 1, 1)
Set itm5 = ListView1.ListItems.Add(, , "Item5", 1, 1)
Set itm6 = ListView1.ListItems.Add(, , "Item6", 1, 1)
Set itm7 = ListView1.ListItems.Add(, , "Item7", 1, 1)

'Force window to be shown.
'This is just so you can see what is happening or else the
'messages would come up and the window would not have
'appeared yet
Me.Show

'Set the listview to icon
ListView1.View = lvwIcon
MsgBox ("You are now viewing the list in Icon View")

'Set the listview to small icon
ListView1.View = lvwSmallIcon
MsgBox ("You are now viewing the list in Small Icon View")

'Set the listview to list
ListView1.View = lvwList
MsgBox ("You are now viewing the list in List View")

'Add a column at runtime
'You can do this ahead of time in the controls
'property pages
Set col1 = ListView1.ColumnHeaders.Add(, , "Name")

'Set the listview to report
ListView1.View = lvwReport
MsgBox ("You are now viewing the list in Report View")

```

در اینجا شما فقط باید خواص ListView1 را تنظیم کنید. ببینید که شما می توانید نمای هر کدام آنها را تغییر دهید. ساده است نه؟

### بخش سوم کار کردن با ستونها

آیا بخاطر دارید نمای Report را در مرورگر ویندوز به شکل می بینید؟ در آن حالت بصورت چند ستونی است و شامل : نام فایل، اندازه، تاریخ و چیزهای دیگری است. ممکن است مشکل به نظر برسد اما من به شما اطمینان می دهم که بسیاری از کدهای آن را قبلاً نوشته ایم! اجازه دهید دوباره برگردیم به آن. اینجا مقداری کد داریم. دوباره آن را اجرا کنید و سپس بررسی کنید که هر کدام آنها بعد از این کد دارای توضیحاتی است:

```

Dim itm As ListItem
Dim col As ColumnHeader

'Add some columns
Set col1 = ListView1.ColumnHeaders.Add(, , "Filename")
Set col2 = ListView1.ColumnHeaders.Add(, , "Type")
Set col3 = ListView1.ColumnHeaders.Add(, , "Size")
Set col4 = ListView1.ColumnHeaders.Add(, , "Date")

```

```

'Add an item. The text here is always the
'First Column (Index 0)
Set itm = ListView1.ListItems.Add(, , "Abstract.exe", 1, 1)

'Here is how we access each of the columns
itm.SubItems(1) = "Program File"
itm.SubItems(2) = "15 KB"
itm.SubItems(3) = "10/10/1999"

'Set the listview to report
ListView1.View = lvwReport

```

شما می توانید ببینید که ما فقط به اضافه کردن تعدادی از ستونها احتیاج داشتیم مانند مثال قبل. تفاوتی که در اینجا وجود دارد این است که ما به SubItem ها در یک ListItem دسترسی داریم. وقتی شما ابتدا یک آیتم متن را اضافه کنید آن را در Column(0) دریافت خواهید کرد. سپس برای قرار دادن متن در ستونهای دیگر ما فقط لازم است که متن را به هر کدام از زیر آیتمها انتقال دهیم. برای مثال فوق من خودم را برای اضافه کردن تصاویر به هدر ستونها در دسک ندام اما اگر شما از VB6 استفاده می کنید می توانید این کار را نیز انجام دهید.

#### بخش چهارم رویدادهای ListView

در اینجا دو تا از رویدادهای اصلی که شما هنگام کار با ListView واقعاً به آنها احتیاج دارید را آورده ایم. رویدادهای ItemClick و ItemCheck. رویداد ItemClick وقتی روی می دهد که روی لیست آیتمها کلیک شده باشد. این رویداد شی ListItem را که روی آن کلیک شده است را برمی گرداند، بنابراین شما می توانید با آن هر کاری را که لازم است انجام دهید. این کد را به مثال قبل اضافه کنید:

```

Private Sub ListView1_ItemClick(ByVal Item As MSComctlLib.ListItem)
    MsgBox Item.Text
End Sub

```

وقتی شما برنامه را اجرا کنید و روی هر کدام از آیتمها در ListView کلیک کنید یک پیغام مبنی حاوی متن آیتمی که روی آن کلیک کرده اید نمایش داده می شود. رویداد دیگر برای وقتی است که شما حالتی را استفاده کرده باشید که کنار هر کدام از آیتمهای ListView یک جعبه چک نمایش داده شده باشد. این کد را به رویداد Form\_Load اضافه کنید:

```

ListView1.Checkboxes = True

```

وقتی شما کد را اجرا کنید متوجه خواهید شد که یک جعبه چک در کنار هر کدام از آیتمها داده شده است. ممکن است تا زمانی که روی ListView کلیک نشود، آن را بروز نکند. ( به نظر می رسد این یکی از باگهای VB باشد ). اما اگر شما مقادیر جعبه چکها را در زمان طراحی تغییر دهید آن را بدرستی نمایش خواهد داد. برای انجام دادن کاری در زمانی که کاربر جعبه چکها را تیک می زند و یا تیک آن را برمی دارد شما باید کد زیر را در رویداد ItemCheck قرار دهید.

```

Private Sub ListView1_ItemCheck(ByVal Item As MSComctlLib.ListItem)
    If Item.Checked Then
        MsgBox "Box is being checked"
    Else
        MsgBox "Box is being unchecked"
    End If
End Sub

```

بخش پنجم مرتب کردن Listview بوسیله ستونها:

در این بخش ما می خواهیم آیتمهایی را در حالت Report به Listview اضافه کنیم و وقتی که کاربر روی عنوان یکی از ستونها کلیک کرد می خواهیم آیتمهای موجود در Listview را بر اساس همان ستون مرتب کنیم. اگر کاربر روی همان ستون دوباره کلیک کرد ما می خواهیم این مرتب سازی بین حالت صعودی و نزولی تغییر کند. ابتدا نگاهی به کد زیر می اندازیم که تاندازه ای واضح به نظر می رسد:

```
Private Sub Form_Load()  
    Dim itm As ListItem  
    Dim col As ColumnHeader  
  
    'Add Some Columns  
    Set col = ListView1.ColumnHeaders.Add(, , "Col1")  
    Set col = ListView1.ColumnHeaders.Add(, , "Col2")  
  
    'Add two items  
    Set itm = ListView1.ListItems.Add(, , "ABC")  
    itm.SubItems(1) = "XYZ"  
  
    Set itm = ListView1.ListItems.Add(, , "XYZ")  
    itm.SubItems(1) = "ABC"  
  
    'Set the view to report  
    ListView1.View = lvwReport  
End Sub  
  
Private Sub ListView1_ColumnClick(ByVal ColumnHeader As  
MSComctlLib.ColumnHeader)  
  
    'Check if the Sortkey is the same as the current one  
    If ListView1.SortKey <> ColumnHeader.Index - 1 Then  
        'When a column is clicked set the sortkey  
        'to the columnheader index -1  
        ListView1.SortKey = ColumnHeader.Index - 1  
        ListView1.SortOrder = lvwAscending  
    Else  
        'If the column is already selected then change the  
        'sortorder to be the opposite of what is currently  
        'being used  
        ListView1.SortOrder = IIf(ListView1.SortOrder = lvwAscending, _  
lvwDescending, lvwAscending)  
    End If  
  
    'Set the sorted property to use the new sortkey  
    'and sort the contents  
    ListView1.Sorted = True  
End Sub
```

بنابر این وقتی کاربر روی یک آیتم کلیک می کند رویداد Column\_Click روی می دهد که مرجعی به شی ColumnHeader که کلیک شده است را برمی گرداند. در این کد ما این را برای دسترسی به خاصیت Index برای مقدار دهی SortKey از کنترل ListView بکار برده ایم ( SortKey فقط می گوید Listview چگونه مرتب شود). وقتی که این کامل شد ما مقدار خاصیت Sorted کنترل ListView را به True مقدار می دهیم. این اطمینان حاصل می کند که کنترل مرتب شده است و بوسیله این تغییرات در SortKey منعکس شده و نمایش داده شود. اگر همان ستون دوباره کلیک شد سپس ما نوع مرتب سازی را تغییر می دهیم و آن را برابر عکس روش فعلی قرار می دهیم

بخش ششم ذخیره کردن عناوین ستونهای مرتب شده:

آیا تاکنون شما برنامه هایی را دیده اید که چیزهایی را در آن مطابق میلتان تنظیم کرده اید و سپس از برنامه خارج شده و وقتی که دوباره به برنامه برگشته اید همه تنظیمات آن به حالت پیش فرض برگشته است و هیچ کدام از تنظیمات مورد نیاز شما را ذخیره نکرده باشد؟ این مسئله بی نهایت آزار دهنده است و نشاندهنده یک برنامه نویسی ضعیف است. برای کنترل ListView این مسئله در رابطه با ColumnHeader صدق می کند. اگر شما اجازه دهید که کاربر اندازه یا نوع مرتب سازی ستونها را تغییر دهد باید این تنظیمات را ذخیره کنید بنابراین دفعه بعد که کاربر از برنامه شما استفاده کند این تنظیمات را در اختیار خواهد داشت. در اینجا کدی را که شما برای انجام دادن این کار نیاز دارید قرار دارد: ( بیشتر کدی که در رویداد Form\_Load قرار دارد فقط برای مقدار دهی چیزهایی است که در بالا گفته شد وگرنه کد اصلی بسیار کوچکتر از چیزی است که در اینجا به نظر می رسد).

```
Private Sub Form_Load()  
    Dim colx As ColumnHeader  
  
    'Let the user reorder the columns  
    ListView1.AllowColumnReorder = True  
  
    'Set view to report  
    ListView1.View = lvwReport  
  
    'Add some columns  
    Set colx = ListView1.ColumnHeaders.Add(, , "Col1")  
    Set colx = ListView1.ColumnHeaders.Add(, , "Col2")  
    Set colx = ListView1.ColumnHeaders.Add(, , "Col3")  
    Set colx = ListView1.ColumnHeaders.Add(, , "Col4")  
    Set colx = ListView1.ColumnHeaders.Add(, , "Col5")  
    Set colx = ListView1.ColumnHeaders.Add(, , "Col6")  
  
    'Loop though each ColumnHeader object and set the  
    'position of it dependent on what the user did  
    'the last time  
    For Each colx In ListView1.ColumnHeaders  
        colx.Position = GetSetting(App.Title, "Settings", "Col" & colx.Index,  
colx.Index)  
        colx.Width = GetSetting(App.Title, "Settings", "ColWidth" &  
colx.Index, colx.Width)  
    Next  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    Dim colx As ColumnHeader  
  
    'Save the Position of each of the ColumnHeader  
    'Objects so we can load them the next time  
    'the user starts the program  
    For Each colx In ListView1.ColumnHeaders  
        SaveSetting App.Title, "Settings", "Col" & colx.Index, _  
        ListView1.ColumnHeaders(colx.Index).Position  
        SaveSetting App.Title, "Settings", "ColWidth" & colx.Index, _  
        ListView1.ColumnHeaders(colx.Index).Width  
    Next  
End Sub
```

رویداد Form\_Unload تا زمانی که همه ColumnHeader ها در کنترل Listview را بدست آورده و مکان و اندازه (عرض) آنها را ذخیره کند، ادامه پیدا می کند. سپس دفعه بعد که برنامه اجرا شود در رویداد Form\_Load این خواص اشاره شده و مقادیر آنها را به اشیای ColumnHeader خاص برمی گردانیم.

من نمی توانم بیان کنم که این پروسس چقدر مهم است. اگر شما اجازه تغییر دادن ستونها را داده باشید لطفاً تنظیمات را برای دفعه بعد ذخیره کنید. شما فقط پروسسی را برای ذخیره تنظیمات می بینید که آسان است و هیچ چیز بیشتر از این کد نیاز ندارد. به من اطمینان کنید، کاربران شما شاید برای این امکان از شما تشکر نکنند اما آنها واضح است که اگر این امکان را نداشته باشند شاد نخواهند شد.