

بهبود نمایش رشته های متنی طولانی در ListBox و ComboBox

جان کالورت این مقاله فرض را بر این گذاشته که شما با ویندوز و ویژوال بیسیک بصورت حرفه ای آشنا هستید. برنامه مربوط به این مقاله را می توانید از آدرس زیر دانلود کنید:

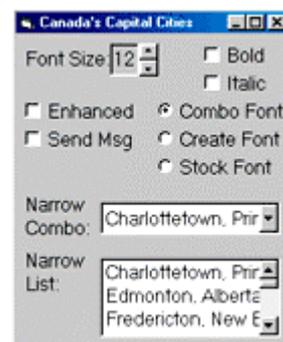
<http://download.microsoft.com/download/2/8/c/28c4ace3-f5ed-4e14-bc64-3d563b807dfb/Calvert.exe>

خلاصه

کنترل‌های ComboBox و ListBox در ویژوال بیسیک بطور پیش فرض نمی توانند رشته های متنی که طولانی تر از اندازه خود کنترل هستند، نمایش دهند. این مقاله چندین روش را برای بهبود توانایی خواندن رشته های طولانی را در هر کنترل شرح می دهد. اولین روش استفاده از Tooltip برای نمایش یک آیتم در ComboBox و ListBox است. روش دیگر استفاده از توابع API مختلف جهت اضافه کردن یک ScrollBar افقی برای ListBox و مشخص کردن عرض کافی برای طولانی ترین رشته با استفاده از صفات فونت در ComboBox می باشد. مراحل لازم برای هر دو کنترل ListBox و ComboBox شرح داده شده اند.

کنترل‌های ذاتی ویژوال بیسیک و کنترل‌های اکتیویکس که متن‌های طولانی را در خود جای دهند وجود دارد. برای مثال کنترل ذاتی TextBox می تواند بصورت چند خطی و با هر دو ScrollBar عمودی و افقی باشد. این به کاربر اجازه می دهد که یکی از دو حالت Wrap یا پیمایش را انتخاب کند که در آن تمام متن بی کم و کاست قابل رویت است. علاوه بر آن کنترل‌های Grid ستون‌هایی با قابلیت تغییر اندازه، خانه هایی چند خطی و گاهی یک Tooltip کامل دارند که اجازه می دهد بطور بسیار ساده ای همه محتویات خانه فعال را ببیند.

کنترل‌های ذاتی ComboBox و ListBox در ویژوال بیسیک بطور مناسب داده هایی را که طولانی تر از اندازه محدوده قابل رویت هر کدام از کنترل‌ها هستند، نمایش نمی دهند. فقط قسمتی از داده ها می تواند قابل رویت باشد. اگرچه یک فرم دارای قابلیت تغییر اندازه است، آرایش کنترل‌ها و اندازه صفحه نمایش کاربران ممکن است اجازه ندهد یک ComboBox یا ListBox برای موثر بودن و تطبیق با طولانی ترین آیتم در لیست آنها تغییر اندازه دهد، همانطور که شما می توانید در شکل ۱ ببینید.



شکل ۱- کنترل‌های بدون پیمایش

بعد از مطالعه روش استفاده از Tooltip برای نمایش محتویات یک ComboBox یا ListBox با استفاده از API های ویندوز، ما به شما راه‌هایی برای قابلیت خواندن داده های متنی طولانی در این دو کنترل نشان خواهیم داد. به شما درباره چگونگی تعیین حداکثر طول داده ای که در یک برنامه برای نمایش لازم است و چگونگی اضافه کردن ScrollBar های افقی برای نمایش متن‌های طولانی بصورت موثر توضیح خواهیم داد. پروژه های ویژوال بیسیک در دسترس برای دانلود از لینکی که در بالا ذکر شد این تکنیک‌ها را برای شما آشکار می کنند.

Tooltip برای یک ComboBox یا ListBox

یکی از راه‌های نمایش همه داده های متنی در ComboBox یا ListBox استفاده از Tooltip برای نمایش رشته بصورت کامل است، اگرچه ممکن است آن طولانی باشد. در یک حالت ساده، وقتی خاصیت Style یک ComboBox روی Dropdown Combo

یا DropDown List تنظیم شده باشد، در آنها فقط یک آیتم از لیست در یک زمان قابل نمایش است، برعکس این دو حالت وقتی است که کنترل روی Dropped down تنظیم شده باشد. برای نمایش Tooltip باید متن خاصیت Tooltip را به مقدار جاری ComboBox بوسیله قرار دادن کد زیر در رویدادهای Click و Change کنترل قرار دهیم:

```
Combo1.ToolTipText = Combo1
```

شما می توانید چگونگی کار این کد را در پروژه تهیه شده برای این مقاله ببینید. وقتی خاصیت Style به Simple Combo تنظیم شده است بیشترین پیچیدگی را به خود می گیرد زیرا کنترل ComboBox مانند ترکیب یک TextBox با ListBox به نظر می رسد. بعد از اینکه من چگونگی بکار بردن Tooltip را برای کنترل ListBox شرح دادم دوباره به این قسمت برمی گردیم.

کنترل ListBox ممکن است در یک زمان آیتمهای زیادی را در محدوده نمایش خود داشته باشد، بنابراین شما نمی توانید فقط مقدار جاری یک ListBox را بصورت Tooltip نمایش دهید و تصور کنید که همه متن داخل آن را ببینید. به بیان دیگر، یک Tooltip باید هر کدام از آیتمها را که ماوس روی آن قرار دارد نمایش دهد. همانطور که کاربر ماوس را روی یک سری از آیتمهای لیست حرکت می دهد، متن کامل برای هر آیتم بعدی در Tooltip ظاهر شود. پیغام ویندوز که این امکان را فراهم می کند LB_ITEMFROMPOINT است. این پیغام مختصات x و y یک نقطه از ListBox را می گیرد و اندیس آیتم موجود در آن نقطه از لیست را برمی گرداند. فراخوانی SendMessage از رویداد MouseMove مربوط به ListBox مقداری را که می تواند برای نمایش یک آیتم درون خاصیت TooltipText بکار رود را بر می گرداند.

قسمت مشکل فقط اندازه گرفتن مختصات x و y است و دسته بندی کردن آنها بصورتی که مورد نیاز پیغام LB_ITEMFROMPOINT می باشد. با وجود اینکه مقدار پارامترهای موجود در رویداد MouseMove بنا به خاصیت ScaleMode فرم است، مقادیر پارامترهای x و y برای رویداد MouseMove در واحد twip (یک واحد اندازه که برای چاپ استفاده می شود، برابر تقریباً 1/72 اینچ می باشد) می باشند. در صورتی که LB_ITEMFROMPOINT انتظار دارد که مختصات بر حسب پیکسل باشد. در نتیجه آنها باید تبدیل شوند. بعلاوه مختصات x و y باید به ترتیب در کلمه با ارزش و کلمه کم ارزش از یک متغیر از نوع Long به آن ارسال شوند. این تبدیلات برای ویژوال بیسیک بصورت کد زیر است:

```
lParam = (CInt(Y / Screen.TwipsPerPixelY) * 2 ^ 16) _  
+ CInt(X / Screen.TwipsPerPixelX)  
lResult = SendMessage(List1.hwnd, LB_ITEMFROMPOINT, _  
0, ByVal lParam)
```

سادگی مقدار برگشتی تابع SendMessage حاوی دو عدد Integer بسته بندی شده در یک Long است. برای تفسیر این نتیجه از کدی که در زیر مشاهده می کنید استفاده کنید:

```
'(X,Y) is outside listbox  
If (lResult \ 2 ^ 16) <> 0 Then Exit Sub
```

```
nIndex = CInt(lResult)  
List1.ToolTipText = List1.List(nIndex)
```

برای اختصار بررسی وضعیت خطا بعد از هر فراخوانی تابع API در این کد نمونه کوتاه، حذف شده است. هر چند در یک محیط تولید شما باید همیشه مقادیر بازگشتی توابع API را بررسی کنید. هرگز شما نمی دانید که در موقع اجرا چه وقت و چرا یک فراخوانی API شکست خواهد خورد. SDK ویندوز را برای دیدن جزئیات هر پیغام و شماره خطاهای بازگشتی از تابع SendMessage بررسی کنید.

قبلا من بخشی را درباره یک ComboBox که خاصیت Style آن به Simple Combo مقداردهی شده است توضیح دادم. در برنامه برای بکار بردن روش Tooltip برای ListBox فقط در مورد این حالت از ComboBox شرح دادیم، کد باید در رویداد MouseMove از قسمت ListBox موجود در ComboBox نیز جواب دهد. متأسفانه ویژوال بیسیک رویدادهای ماوس را برای ComboBox در معرض دید قرار نمی دهد. البته این امکان وجود دارد که برای بدام انداختن رویدادها از بدام اندازی پیغامها استفاده کنیم، اما این مسئله از خارج از محدوده این مقاله قرار دارد.

حالا که شما حالات مختلف استفاده از Tooltip را دیدید، اجازه دهید نظری بر چگونگی نمایش متنهای طولانی با استفاده از ScrollBar ها بیندازیم.

ScrollBar های افقی برای کنترلهای ListBox

کنترل ذاتی ListBox ، برخلاف یک TextBox چند خطی، خاصیت ScrollBar برای کنترل ظاهر ScrollBar های عمودی و افقی ندارد. اگر در آن آیتمهایی وجود داشته باشد که بتوانند بصورت عمودی در داخل کنترل، ListBox بطور خودکار یک ScrollBar عمودی به آن اضافه می کند. با این وجود اگر آیتمهای موجود در لیست طولانی تر از اندازه افقی داخل کنترل باشد، ListBox هیچ کاری برای شما انجام نمی دهد. در مقابل کنترل TextBox به شما خیلی کمک می کند. اگر یک TextBox چند خطی ScrollBar افقی نداشته باشد، رشته های متنی طولانی را بطور خودکار می شکند و برای شما در جاهایی که خط شکسته می شود جای خالی تولید می کند. ListBox این وظیفه را انجام نمی دهد.

خوشبختانه با استفاده از کد می توانید به ListBox یک ScrollBar افقی اضافه کنید. این امکان پذیر است زیرا در ظاهر ScrollBar ها جزو خواص پنجره ای است که می تواند بعد از ساخته شدن، اصلاح شود. در ابتدا شما باید ویژگی پنجره (Window Style) کنترل را با استفاده از توابع GetWindowLong و SetWindowLong برای اضافه کردن WS_HSCROLL به خاصیت GWL_STYLE همانطور که در کد زیر می بینید، اصلاح کنید:

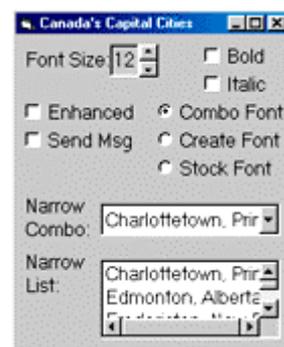
```
Private Const WS_HSCROLL = &H100000
Dim lWindowState As Long

lWindowState = GetWindowLong(List1.hwnd, GWL_STYLE)
lWindowState = lWindowState Or WS_HSCROLL

SetLastError 0

lWindowState = SetWindowLong(List1.hwnd, GWL_STYLE, lWindowState)
```

هر چند، این کد کار نمی کند. اگر شما رشته های طولانی را به ListBox اضافه کنید، یک ScrollBar نمایان خواهد شد اما این ScrollBar هیچ امکان پیمایشی را ندارد. جعبه پیکان عریض تر از ScrollBar خواهد بود و این پیمایش را غیر ممکن می سازد (شکل ۲ را نگاه کنید). بعلاوه اگر شما ScrollBar را بعد از ظاهر شدن آیتمها در ListBox اضافه کنید، ScrollBar افقی به هیچ وجه ظاهر نخواهد شد.



شکل ۲- جعبه عریض شده

در حال حاضر ظاهر کردن یک ScrollBar افقی با استفاده از پیغام LB_SETHORIZONTALTEXTENT انجام می شود. این پیغام عرضی را که ListBox می تواند بصورت افقی پیمایش شود را در واحد پیکسل مقداردهی می کند. این مقدار به طول طولانی ترین رشته موجود در ListBox در واحد پیکسل اشاره می کند. اگر عرض قابل پیمایش از عرض واقعی کنترل بیشتر باشد، یک ScrollBar افقی نمایش داده می شود. اگر با تغییر اندازه فرم اصلی کنترل ListBox عریض تر از عرض قابل پیمایش شد، ScrollBar افقی پنهان خواهد شد تا زمانی که کنترل دوباره تغییر اندازه داده شود. این پیغام را می توانید بسادگی مانند زیر ارسال کنید:

```
lResult = SendMessage(List1.hwnd, LB_SETHORIZONTALTEXTENT, lLength, ByVal 0)
```

قسمت دشوار آن تشخیص دادن عرض لازم در واحد پیکسل برای جا دادن طولانی ترین رشته در لیست می باشد. برای انجام این دستور شما باید برای پیدا کردن طولانی ترین رشته و پیدا کردن عرض آن، مشخصات فونت استفاده شده در آن کنترل را بگیرید.

تشخیص طول رشته

فرض کنید که طولانی ترین آیتم موجود در ListBox در متغیر رشته ای sString است. کد نشان داده شده در لیست کد ۱ عرض این رشته را در واحد پیکسل تشخیص خواهد داد. تابع کلیدی در لیست کد ۱، تابع GetTextExtentPoint32 است. آن تابع یک محتوای ابزار و یک رشته را به عنوان آرگومان می گیرد و عرض و ارتفاع رشته را تشخیص می دهد. یک محتوای ابزار محتوای چاپ و یا نمایش را در هر یک که رشته ظاهر خواهد شد، تعریف می کند. این شامل مشخصات فونت، که بر ظاهر و در نتیجه بر عرض رشته تاثیر می گذارد، می باشد. تابع GetTextExtentPoint32 می تواند برای اندازه یک نقطه، فاصله گذاری متناسب، ضخامت، ایتالیک و فاصله گذاری میان کاراکترها از فونت، و بسیاری چیزهای دیگر پاسخگو باشد. از پارامتر محتوای ابزار بسیار متشکریم.

لیست کد ۱

```
Private Type SIZE
    cx As Long
    cy As Long
End Type

Dim uSize As SIZE

With List1
    'Get a handle to the device context for the control
    hDC = GetDC(.hwnd)

    lFont = SendMessage(.hwnd, WM_GETFONT, 0, ByVal 0)

    'Select the font in to the device context, and retain prior font
    lFontOld = SelectObject(hDC, lFont)

    'Determine the width of the string
    lResult = GetTextExtentPoint32(hDC, sString, Len(sString), uSize)

    'Return the string length
    lStringLenInControl = uSize.cx
End With

'Reset the device context font
SelectObject hDC, lFontOld

'Release the device context handle
ReleaseDC List1.hwnd, hDC
```

قبل از اینکه تابع GetTextExtentPoint32 را فراخوانی کنید، به یک شی محتوای ابزار ساخته شده با خواص فونت مشخص شده متعلق به ListBox احتیاج دارید. چند خط اول از کد نمایش داده شده در لیست کد ۱ به این مسئله پرداخته است. اگر چه شما می توانید شی محتوای ابزار کنترل را مستقیماً با استفاده از تابع GetDC(List1.hwnd) بدست آورید، اطلاعات فونت محتوای ابزار باید صریحاً به محتوای ابزار بصورت مراحل جداگانه اشاره کند. ساده ترین راه برای انجام این کار این است که یک Handle برای فونت با استفاده از پیغام ویندوز WM_GETFONT بدست آوریم، سپس فونت را به محتوای ابزار با استفاده از SelectObject(hDC,IFont) واگذار کنیم. اگر بنا به دلایلی بخواهید از فونتی به غیر از فونت جاری در ListBox در زمان فراخوانی

تابع `GetTextExtentPoint32` استفاده کنید، می توانید با استفاده از تابع `CreateFont` یک `Handle` برای فونت دلخواه مناسب بدست آورید. این تابع به شما اختیارات زیادی را در رابطه با فونتها می دهد.

وقتی که محتوای ابزار `Handle` های فونت ساخته شده باشند، شما باید کدی را برای آزاد کردن و حذف آنها بنویسید. این می تواند کمی گیج کننده باشد، چون کاری که شما با `Handle` های فونت انجام می دهید بستگی دارد به این که شما آن را با استفاده از پیغام `WM_GETFONT` بدست آورده اید یا بوسیله استفاده از تابع `CreateFont`. اگر از تابع پیغام `WM_GETFONT` استفاده کرده اید، احتیاجی نیست که هیچ کار اضافی را انجام دهید. اگر `Handle` فونت را با استفاده از تابع `CreateFont` بدست آورده اید، باید آن را با استفاده از تابع `DeleteFont` آزاد کنید. این مسئله خیلی مهم است زیرا جمع آوری خودکار مجموعه اشاره گرهای (یا `Handle` ها) بی مصرف در ویژوال بیسیک معمولاً محدود به متغیرهایی از نوع `Object` است که این `Handle` ها از آن نوع نیستند. شما باید مطمئن باشید که `Handle` هایی را که اختصاص داده اید بدرستی حذف شده اند.

هرگاه طول طولانی ترین رشته موجود در محتوای ابزار محاسبه شد، من بسادگی پیغام `LB_SETHORIZONTALEXTENT` را برای `ListBox` ارسال می کنم تا `ScrollBar` افقی آن فعال شود.

در اینجا بیش از یک راه برای محاسبه و بدست آوردن طولانی ترین رشته موجود در `ListBox` در لیست کد ۱ وجود دارد. ساده ترین پیشنهاد قرار دادن حلقه روی همه آیتمهای درون لیست برای آیمتی با بزرگترین مقدار برگشتی از تابع `Len` می باشد:

```
For I = 0 To List1.ListCount - 1
    If (Len(List1.List(I)) > lItemMaxLen) Then
        sItemText = List1.List(I)
        lItemMaxLen = Len(List1.List(I))
    End If
Next I
```

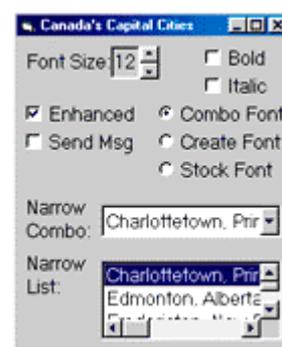
این کد دو اشکال دارد. اول، این روش آیمت مورد نظر را با استفاده از بیشترین تعداد کاراکترها تشخیص می دهد، نه با بزرگترین طول فونت جاری. این دو معنی یکسانی ندارند. به تفاوت بین دو رشته و توجه کنید. به تناسب فضای فونت، رشته حاوی کاراکترهای `I` کوتاهتر از همان تعداد از `w` ها است. هر چند تفاوت حروف تکی معمولاً آشکار است، بنابراین نگرانی درباره آن نداشته باشید.

دومین مشکل در کد ارائه شده این است که چندین دفعه به `List1.List(I)` مراجعه می کند. هر ارجاع و یا استفاده از نوع داده رشته ای طولانی می تواند کارایی را کاهش دهد، مخصوصاً وقتی که در یک حلقه تکرار شود. من بعضی آزمایشها را برای مقایسه عبارت `Len(List1.List(I))` با پیغام `LB_GETTEXTLEN` که طول `I` امین آیمت لیست را بر می گرداند، انجام داده ام. استفاده از پیغام `LB_GETTEXTLEN` در مجموع سریعتر از فراخوانی تابع `Len` بود. در اینجا نسخه اصلاح شده از حلقه قبلی که با استفاده از پیغام `LB_GETTEXTLEN` پیاده سازی شده است را نشان می دهیم:

```
For i = 0 To List1.ListCount - 1
    lResult = SendMessage(List1.hwnd, LB_GETTEXTLEN, i, ByVal 0)

    If (lResult > lItemMaxLen) Then
        sItemText = List1.List(i)
        lItemMaxLen = lResult
    End If
Next I
```

هر گاه شما همه این کدها را با هم قرار دهید، در `ListBox` یک `ScrollBar` افقی ظاهر خواهد شد، مانند شکل ۴.



شکل ۴- یک `ListBox` با قابلیت پیمایش

نمایش رشته های طولانی در ComboBox ها

حالا اجازه دهید به چگونگی نمایش رشته های طولانی در قسمت پایین آمدنی ComboBox رسیدگی کنیم. وقتی یک ComboBox حاوی رشته های طولانی است، این رشته ها ممکن است کاملا در قسمت پایین آمدنی آن بطور کامل قابل نمایش نباشند. (شکل ۵ را ببینید).



شکل ۵- رشته های منی طولانی

ظاهرا ویژگی بیسک اندازه قسمت پایین آمدنی را به اندازه آیتمهای طولانی آن تغییر اندازه نمی دهد و شما برای انجام دادن این کار باید کد نویسی کنید. اگر شما اندازه مورد نظرتان را بدانید، تغییر اندازه قسمت پایین آمدنی در یک ComboBox ساده خواهد بود. این کار بسادگی ارسال پیام `CB_SETDROPPEDWIDTH` با عرض مورد نظر شما در واحد پیکسل به کنترل مورد نظر است:

```
lResult = SendMessage(cboCombo.hwnd, CB_SETDROPPEDWIDTH, lDropdownWidth, ByVal 0)
```

برای تشخیص عرض صحیح از قسمت پایین افتادنی، ابتدا طول بلندترین رشته داخل ComboBox را محاسبه می کنیم. این دقیقا به همان روشی که قبلا برای ListBox شرح دادیم انجام می شود (شکل ۳ را نگاه کنید). اگر چه طول رشته برای تشخیص عرض مورد نیاز برای قسمت پایین افتادنی کافی نیست. شما باید همچنین باید عرض لبه های قسمت پایین افتادنی را نیز محاسبه کنید که یک پیکسل در هر طرف می باشد، و عرض ScrollBar عمودی را هم باید محاسبه کنید. کد زیر این دو مطلب مورد بحث را نیز مد نظر قرار داده است:

```
'Account for the dropdown window's  
' border (left and right)  
lDropdownWidth = lStringLenInControl + 2  
  
'Account for the scrollbar width  
lVerticalScrollbarWidth =  
    GetSystemMetrics(SM_CXVSCROLL)  
If (lVerticalScrollbarWidth <> 0) Then  
    lDropdownWidth = lDropdownWidth + lVerticalScrollbarWidth  
End If
```

عرض لبه یک پنجره استاندارد ثابت نیست. مقدار آن می تواند از طریق اپلت Display در Control Panel تنظیم شود و شامل لبه خودش و اندازه قاب آن است. اگر چه هیچ کدام از این کاربردها برای پنجره پایین افتادنی نیست، بنابراین شما بدون ترس از اشتباه از مقدار یک پیکسل برای عرض هر کدام از چپ و راست لبه ها استفاده کنید.

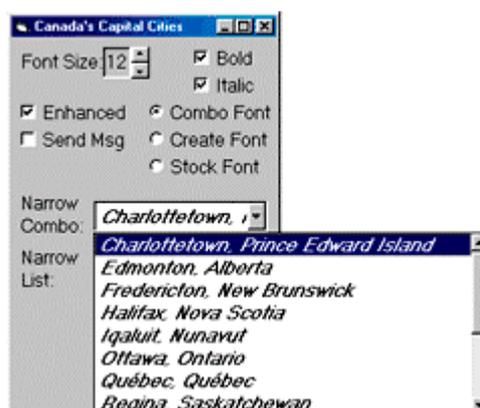
برای ساختن این محاسبات، شما باید طولانی ترین رشته در ComboBox را بشناسید. خوشبختانه می توانید از همان پیشنهاد قسمت قبلی برای پیدا کردن طولانی ترین رشته در ListBox برای ComboBox هم استفاده کنید. فقط باید مطمئن باشید که کد

شما برای بازیابی طول آیتمهای درون لیست از پیغامهای ComboBox استفاده می کنید. این پیغام CB_GETLBTEXTLEN می باشد. کد حاصل مانند زیر است:

```
For I = 0 To Combo1.ListCount - 1
    lResult = SendMessage(Combo1.hwnd, CB_GETLBTEXTLEN, I, ByVal 0)

    If (lResult > lItemMaxLen) Then
        sItemText = Combo1.List(I)
        lItemMaxLen = lResult
    End If
Next I
```

حالا شما می توانید تابع SendMessage را با استفاده از پیغام CB_SETDROPPEDWIDTH فراخوانی کنید و مقدار محاسبه شده در IDropdownWidth را به آن بفرستید. در صورتی که نام فونت، اندازه، ضخامت، ایتالیک و خواص دیگر محاسبه شده برای IDropdownWidth تغییر کند، برای همه این فاکتورها دوباره محاسبه خواهد شد. نتیجه در شکل ۶ نمایش داده شده است.



شکل ۶- یک ComboBox برای نمایش رشته های متنی طولانی

من در کد قبلی یک بررسی اضافه برای اطمینان از اینکه قسمت پایین افتادنی از لبه صفحه بیرون نمی رود، انجام داده ام. در غیر اینصورت ScrollBar عمودی در ComboBox دیده نخواهد شد. برای محاسبه به این حقیقت نیاز داریم که سیستم عامل مکان قسمت پایین افتادنی را بطور خودکار در سمت چپ ComboBox رسم خواهد کرد، که ممکن است برای قسمت پایین افتادنی عرض کمتر از صفحه را اجازه دهد. من اغلب تنظیمات عرض قسمت پایین افتادنی را با اضافه کردن یک ScrollBar افقی به آن با استفاده از پیغام CB_SETHORIZONTALEXTENT بکار می برم. من اعمال این امکان را به خود خواننده واگذار می کنم.

نتیجه گیری

کنترلهای ذاتی ListBox و ComboBox در ویژوال بیسیک هنگامی که به نمایش داده های طولانی در آنها نیاز دارید کمبودهایی دارند. خوشبختانه با فراخوانی توابع API آسان که در این مقاله نام بردم، شما می توانید وظایف این کنترلها را با نمایش Tooltip، ScrollBar افقی و تغییر اندازه خودکار لیست پایین افتادنی، افزایش دهید.

با این وجود به دریا نروید. اگر شما دریافتید که بسیاری قسمتهای داده های طولانی در ComboBox یا ListBox دارید، عقلانه تر آن است که طراحی برنامه تان را مجدداً انجام دهید. صفحات نمایش ممکن است برای ارائه داده ها بصورتی که براحتی و بدون پیمایشها و قسمتهای پایین افتادنی بزرگ، قابل نمایش باشد دوباره پیشرفته شده باشند. این پیشرفتهها را در برنامه بصورت عاقلانه استفاده کنید.

جان کالورت: در اوتاوا، اونتاریو زندگی می کند و یک مشاور فنی تخصصی مستقل در زمینه برنامه های Client/Server و طبقه سه در IT می باشد. او با محصولات مایکروسافت بطور وسیعی کار می کند و یک متخصص MCSD است. شما می توانید با آدرس زیر با او مکاتبه کنید:

john-m-calvert@home.com

نویسنده: بادوگر ویزوال بیسیکا

وبلاگ: بادوگر ویزوال بیسیکا <http://VbLog.PersianBlog.com>

پست الکترونیک: mam_programmer@yahoo.co.uk

توجه: لطفا هنگام ذکر مطالب منبع آن را هم ذکر کنید!