



تغییر سایز مکان نمای متن در یک Text Box

| نگارش ویندوز | نگارش ویژوال بیسیک | تاریخ | نام فایل |
|--------------|--------------------|----------------|-----------|
| 95 / 98 / NT | 32 bit | ۲۱ ژانویه ۱۹۹۵ | 94318.TXT |

هیچ روشی برای تغییر شکل مکان نمای متن مربوط به یک TextBox در ویژوال بیسیک پیش بینی نشده است، اما انجام این کار از طریق توابع API امکان پذیر است. تابع CreateCaret() شکل جدیدی برای یک مکان نمای متن ایجاد کرده و آن را به یک پنجره خاص نسبت می دهد. شکل مکان نما می تواند یک خط، یک بلوک، یا یک تصویر Bitmap باشد. نحوه فراخوانی این تابع به شکل زیر است:

```
Void CreateCaret(hwnd, hbmp, nwidth, nheight)
```

که در آن:

hwnd از نوع HWND = مشخص کننده شناسه پنجره مورد نظر

hbmp از نوع HBITMAP = مشخص کننده شناسه Bitmap مورد نظر برای شکل مکان نما

nwidth از نوع int = مشخص کننده عرض مکان نما

nheight از نوع int = مشخص کننده ارتفاع مکان نما

تابع ShowCaret() مکان نما را روی صفحه در محل فعلی مکان نما نمایش می دهد. مکان نما پس از اینکه نمایش داده شد، بطور خود کار شروع به چشمک زدن می کند.

```
Void ShowCaret(hwnd)
```

hwnd از نوع HWND = مشخص کننده شناسه پنجره حاوی مکان نما

تابع GetFocus() شناسه پنجره ای را که در حال حاضر کنترل اجرا را در دست دارد، مشخص

می کند.

```
HWND GetFocus(void)
```

مثال

۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید. Form1 بطور پیش فرض موجود می آید.

۲- دو TextBox روی Form1 قرار دهید.

۳- کد زیر را در بخش Declarations از Form1 وارد کنید:

```
Private Declare Sub CreateCaret Lib "user32" (ByVal w%, ByVal x%, _
    ByVal y%, ByVal z%)
Private Declare Function ShowCaret Lib "user32" ( ByVal hwnd _
    As Long) As Long
Private Declare Function GetFocus Lib "user32" () As Long

Sub Text1_GotFocus ()
    h% = getfocus()           ' get the handle to the text box
    Call createcaret(h%, 0, 3, 24) ' create new caret size
    x% = showcaret(h%)        ' show the new caret
End Sub
```

۴- برنامه را اجرا کنید. مشاهده خواهید کرد که تا زمانی که کنترل اجرا روی Text1 قرار دارد،

اندازه مکان نمای متن در آن از حد معمول بزرگتر است.

شبیه سازی حالت Overtyping در یک Text Box

| نگارش ویندوز | نگارش ویژوال بیسیک | تاریخ | نام فایل |
|--------------|--------------------|----------------|-----------|
| All | All | ۲۱ ژانویه ۱۹۹۵ | 96210.TXT |

TextBox های ویژوال بیسیک دارای امکان OverType یا جایگزینی (که در آن هر کاراکتر

جدید روی یک کاراکتر قبلی نوشته می شود) نیست. برای انجام چنین کاری به مقداری برنامه

نویسی نیاز است. با افزودن چند سطر برنامه به واقعه KeyPress این ابزار می توانید عمل

OverType را شبیه سازی نمایید. قطعه برنامه زیر با این فرض نوشته شده است که شما یک

TextBox به نام Text1 دارید که قرار است در حالت OverType عمل کند.

```
' Form level code:
Const KEY_BACK = &H8

Sub Text1_KeyPress (KeyAscii As Integer)
  If KeyAscii <> KEY_BACK Then
    Text1.SelLength = 1
  End If
End Sub
```

با تغییر مقدار خصوصیت SelLength مربوط به این TextBox به مقدار ۱ باعث می شوید که کاراکتر بعد از مکان نما پررنگ شود (به حالت انتخاب شده). هنگامی که واقعه KeyPress اجرا شود و کاراکتر تایپ شده جدید به متن اضافه شود، بطور خودکار به جای کاراکتر پررنگ شده قرار خواهد گرفت.

اگر بخواهید به کاربر امکان دهید بتواند در موقع دلخواه بین حالت های جایگزینی و حالت معمولی تغییر حالت دهد، می توانید با افزودن مقداری کد به برنامه این کار را انجام دهید. علاوه بر این می توانید با استفاده از خصوصیت SelLength به کاربر نشان دهید که کدام کاراکتر قرار است با کاراکتر تایپ شده توسط وی جایگزین شود. مثال زیر نشان دهنده این قابلیت ها است:

۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید. Form1 بطور پیش فرض بوجود می آید.

۲- یک TextBox به نام Text1 به Form1 اضافه کنید.

۳- یک Label به نام Label1 به Form1 اضافه نمایید.

۴- کد زیر را در Form1 وارد نمایید:

```
' Key codes:
Const KEY_BACK = &H8
Const KEY_PRIOR = &H21
Const KEY_NEXT = &H22
Const KEY_END = &H23
Const KEY_HOME = &H24
Const KEY_LEFT = &H25
Const KEY_UP = &H26
Const KEY_RIGHT = &H27
Const KEY_DOWN = &H28
Const KEY_INSERT = &H2D

Const MODE_OVERTYPE = "overtime"
Const MODE_INSERT = "insert"

Sub Form_Load ()
  Text1.Tag = MODE_INSERT
  Label1.Caption = MODE_INSERT
```

```

End Sub

Sub Text1_Change ()
    ' You have taken some action that changed the text in the
    ' text box. Reset the SelLength if you are in overtype mode.
    If Text1.Tag = MODE_OVERTYPE And Text1.SelLength = 0 Then
        Text1.SelLength = 1
    End If
End Sub

Sub Text1_KeyPress (KeyAscii As Integer)
    ' If you press BACKSPACE and are in overtype mode,
    ' then set SelLength to 0 so the backspace will correctly
    ' delete the character to the left of the current caret
    ' position. SelLength will be reset when the Text1_Change
    ' event occurs following the backspace.
    If KeyAscii = KEY_BACK And Text1.Tag = MODE_OVERTYPE Then
        Text1.SelLength = 0
    End If
End Sub

Sub Text1_KeyUp (KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        ' Toggle between insert and overtype modes.
        Case KEY_INSERT
            If Text1.Tag = MODE_OVERTYPE Then
                Text1.Tag = MODE_INSERT
                Label1.Caption = MODE_INSERT
            Else
                Text1.SelLength = 1
                Text1.Tag = MODE_OVERTYPE
                Label1.Caption = MODE_OVERTYPE
            End If

            ' Handle keys that move the caret position and reset the
            ' SelLength if you are in overtype mode:
            '
            ' The following two lines of code should be all on one line:
            Case KEY_LEFT, KEY_RIGHT, KEY_UP, KEY_DOWN, KEY_HOME, KEY_END,
                KEY_PRIOR, KEY_NEXT
                If Text1.Tag = MODE_OVERTYPE Then
                    Text1.SelLength = 1
                End If
        End Select
    End Sub

Sub Text1_MouseUp (Button As Integer, Shift As Integer, _
    x As Single, Y As Single)
    ' You have clicked at a new location within the text box. Reset the
    ' SelLength if you are in overtype mode.
    If Text1.Tag = MODE_OVERTYPE And Text1.SelLength = 0 Then
        Text1.SelLength = 1
    End If
End Sub

```

۵- برای اجرای برنامه کلید F5 را فشار دهید. هنگامی که کلید INSERT را فشار می دهید، Label1 وضعیت جاری تایپ را در Text1 نشان می دهد. هنگامی که در حالت OverType قرار دارید کاراکتر جاری در محل ورود متن بصورت پررنگ در می آید.


| استفاده از کلید TAB در یک Text Box بدون انتقال کنترل اجرا | | | |
|---|-------------|--------------------|--------------|
| نام فایل | تاریخ | نگارش ویژوال بیسیک | نگارش ویندوز |
| 109261.TXT | ۲۱ جون ۱۹۹۵ | 3 | All |

در حالت عادی، کلید TAB باعث می شود کنترل اجرا از ابزار جاری به ابزار دیگری منتقل شود. مثال مطرح شده در این بخش نشان می دهد که چگونه می توان عملکرد عادی کلید TAB را در یک Text Box به گونه ای تغییر داد تا بتوان از این کلید همانند یک برنامه واژه پرداز برای ایجاد فواصل در متن استفاده نمود. برای انجام این کار باید زمانی که کنترل اجرا روی Text Box مورد نظر است، مقدار خصوصیت TabStop برای تمامی ابزارهای روی فرم را به False تغییر دهیم. در صورتیکه خصوصیت TabStop یکی یا بیشتر از ابزارهای روی یک فرم دارای مقدار True باشد، ویژوال بیسیک فشار داده شدن کلید TAB را به معنای جابجایی کنترل بین ابزارها فرض می کند. بنابراین با تغییر این خصوصیت به False باعث می شویم کاراکتر TAB برای ابزار ارسال شود.

در مثال زیر Text2 طوری تنظیم شده است که فشار کلید TAB را برای افزودن کاراکتر مربوطه به متن به کار برد. Text1 و Text3 چنین وضعیتی ندارند. هنگامی که کنترل اجرا روی Text1 و Text3 است، فشار کلید TAB کنترل اجرا را به ابزار بعدی در صف ابزارها منتقل می کند.


۱- یک پروژه جدید در ویژوال بیسیک ایجاد نمایید. Form1 بطور پیش فرض بوجود می آید.

۲- سه Text Box (به نامهای Text1 ، Text2 و Text3) به Form1 اضافه کنید. Text2 را انتخاب کرده و کلید F4 را فشار دهید تا پنجره خصوصیات ظاهر شود. خصوصیت MultiLine این ابزار را به مقدار True تغییر دهید.

 توجه: هنگامی که در یک ابزار Text Box تک سطر کلید TAB را فشار دهید، این ابزار فشار کلید TAB را قبول نکرده و صدای بوق سیستم شنیده می شود، اما Text Box های چند سطر فشار این کلید را قبول می کنند.

۳- روی ابزار Text2 دوبار کلیک کنید تا پنجره متن برنامه ظاهر شود. واقعه GotFocus را از لیست کشویی Proc انتخاب کنید و کد زیر را به آن بیافزایید:

```
Sub Text2_GotFocus ()
    ' When Text2 gets the focus, clear all TabStop properties on all
    ' controls on the form. Ignore all errors, in case a control does
    ' not have the TabStop property.
    On Error Resume Next
    For i = 0 To Controls.Count - 1 ' Use the Controls collection
        Controls(i).TabStop = False
    Next
End Sub
```

 توجه: برای توضیحات بیشتر در زمینه مجموعه ابزارها بخش مربوط به مجموعه ابزارها را در ادامه همین بخش مطالعه کنید.

۴- واقعه LostFocus را از لیست کشویی Proc انتخاب کنید و کد زیر را به آن بیافزایید:

```
Sub Text2_LostFocus ()
    ' When Text2 loses the focus, make the TabStop property True for all
    ' controls on the form. That restores the ability to tab between
    ' controls. Ignore all errors, in case a control does not have the
    ' TabStop property.
    On Error Resume Next
    For i = 0 To Controls.Count - 1 ' Use the Controls collection
        Controls(i).TabStop = True
    Next
End Sub
```

۵- برای اجرای برنامه کلید F5 را فشار دهید. برای اینکه کنترل اجرا به Text2 منتقل شود، کلید TAB را فشار دهید. در Text2 مقداری متن تایپ کنید و در صورت نیاز کلید TAB را فشار دهید. هنگامی که کنترل اجرای برنامه روی Text1 یا Text3 قرار دارد فشار دادن کلید TAB باعث می شود کنترل اجرا به کنترل بعدی منتقل شود. هنگامی که کنترل اجرا

روی Text2 قرار دارد، فشار کلید TAB روی متن موجود در این ابزار اثر می گذارد. برای خاتمه برنامه می توانید پنجره برنامه را ببندید.

ترتیب انتقال کنترل با TAB

ویژوال بیسیک به طور پیش فرض، ترتیب انتقال کنترل اجرا را به همان ترتیب که شما ابزارها را روی فرم قرار می دهید تنظیم می کند. هر ابزار جدیدی که روی فرم قرار می گیرد، در انتهای صف ترتیب مذکور قرار می گیرد. برای تغییر ترتیب انتقال کنترل اجرا بین ابزارها، باید یا در هنگام طراحی برنامه از خصوصیت ها استفاده کنید، یا در هنگام اجرای برنامه بوسیله بخشی از کد برنامه این کار را انجام دهید.

● برای تغییر دادن ترتیب انتقال کنترل در هنگام طراحی برنامه به روش زیر عمل کنید:

- ۱- روی ابزار مورد نظر کلیک کنید تا انتخاب شود.
- ۲- از پنجره Properties خصوصیت TabIndex را انتخاب کنید. در مستطیل مربوطه مقدار فعلی این خصوصیت نشان داده می شود.
- ۳- ترتیب قرارگیری این ابزار در صف انتقال کنترل را در این مستطیل وارد کنید.
- ۴- دکمه Enter را فشار دهید. در هنگام طراحی برنامه می توانید با انتخاب کنترل و فشار کلید TAB ترتیب جدید را کنترل کنید.

● برای فعال سازی یا از کار انداختن عملکرد دکمه TAB در هنگام طراحی برنامه به شرح زیر عمل کنید:

- ۱- روی ابزار مورد نظر خود کلیک کنید تا انتخاب شود.
- ۲- از پنجره Properties خصوصیت TabStop را انتخاب کنید. مقدار فعلی این خصوصیت در مستطیل مربوطه نشان داده می شود.
- ۳- اگر مقدار این خصوصیت را True قرار دهید باعث می شود در هنگام اجرا این ابزار در صف انتقال کنترل توسط کلید TAB قرار گیرد و اگر مقدار False را انتخاب کنید، فشار کلید TAB باعث انتقال کنترل اجرا به این ابزار نخواهد شد.
- ۴- کلید Enter را فشار دهید تا تغییرات اعمال شود.

هنگامی که ترتیب کنترل اجرا را برای یک ابزار تغییر می دهید، ویژگی `TabIndex` بطور خودکار ترتیب دیگر ابزارها را نیز تغییر می دهد تا با تغییر انجام شده توسط شما سازگار باشند.

ابزاری که خصوصیت `TabIndex` آن دارای مقدار `False` باشد، مکان خود را در ترتیب انتقال کنترل حفظ می کند، اما کنترل اجرا از ابزار قبل از آن به ابزار بعد از آن منتقل شده و بقولی از روی این ابزار می پرد. اگر خصوصیت `TabIndex` برای تمامی ابزارهای روی فرم دارای مقدار `False` باشد، می توانید کلید `TAB` را در `Text Box` های چند سطری بصورت یک کلید ویرایشی بکار ببرید.

مجموعه Controls

مجموعه `Controls` مجموعه ای است که اجزای تشکیل دهنده آن عبارتند از تمامی ابزارهای روی یک فرم، به همراه تمامی اجزای آرایه های ابزار. مجموعه `Controls` دارای یک خصوصیت به نام `Count` است که مشخص کننده تعداد اعضای تشکیل دهنده این مجموعه است. مجموعه `Controls` حاوی تمامی ابزارهای بار شده روی یک فرم است و بوسیله آن می توان تمامی ابزارهای روی یک فرم را مورد بررسی قرار داد. مقدار شمارشگر `index` از صفر تا `Count-1` تغییر می کند.

توجه: عبارت `Controls` یک کلمه کلیدی است نه یک کلمه رزرو شده. این عبارت مشخص کننده یک متغیر داخلی مربوط به فرم به نام `Controls` است. اگر هنگام استفاده از این مجموعه مشخص نکنید که کدام فرم مورد نظر شما است، باید عبارت `Controls` را ذکر کنید. اگر مشخص می کنید که کدام فرم مورد نظر شما است، می توانید عبارت `Controls` را حذف کنید. به عنوان مثال، هر دو سطر زیر یک مفهوم دارند:

```
MyForm.Controls(6).Top = MyForm.Controls(5).Top + increment
```

```
MyForm(6).Top = MyForm(5).Top + increment
```

| کار با محتویات Text Box | | | |
|-------------------------|---------------|--------------------|--------------|
| نام فایل | تاریخ | نگارش ویژوال بیسیک | نگارش ویندوز |
| 186271.TXT | ۱۷ آگوست ۱۹۹۸ | 5 , 6 | 32 Bit |

در این بخش نشان داده می شود که چگونه به کمک خصوصیات Text Box و فراخوانی توابع API ، محتویات Text Box را تحت کنترل کامل خود در آورید. با استفاده از تکنیک های گفته شده در این مبحث می توانید قابلیت های این ابزار را گسترش دهید. در ارائه این مطالب فرض بر این است که شما با نحوه استفاده از توابع API آشنایی دارید. تکنیک های مورد بحث عبارتند از :

- تعیین محل مکان نما در یک Text Box .
- تعیین شماره سطر محل مکان نما در یک Text Box چند سطری.
- تعیین طول یک سطر انتخاب شده.
- نمایش تمامی متن مربوط به یک سطر انتخاب شده.
- اضافه کردن متن در محل مکان نما.

در تکنیک اول از یک خصوصیت مربوط به Text Box استفاده می شود. در بقیه تکنیک ها از تابع SendMessage به همراه شناسه Text Box ، یک پیام و دیگر پارامترهای مورد نیاز آن پیام استفاده می شود.

برای تعیین محل مکان نما در یک Text Box از خصوصیت SelStart این ابزار استفاده می شود. این خصوصیت محل متن انتخاب شده را نسبت به اولین کاراکتر از اولین سطر مشخص می کند. اگر یک بلوک از متن را انتخاب کنید، خواهید توانست با اضافه کردن خصوصیت SelLength به خصوصیت SelStart موقعیت مکان نما را تعیین کنید.

برای سه تکنیک دیگر از تابع SendMessage به همراه خصوصیت Hwnd مربوط به text box ، پیام مناسب و پارامترهای مورد نیاز آن پیام استفاده می شود. تعریف تابع SendMessage به صورت زیر صورت می گیرد:

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
    (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long, _
    ByVal lParam As String) As Long
```

پیامهای زیر در این بخش مورد استفاده قرار می گیرند:

- EM_LINEFROMCHAR (معادل مقدار &HC9) - شماره سطر محل مکان نما را بر می گرداند. محل مکان نما را به عنوان پارامتر همراه این پیام ارسال کنید.
- EM_LINELENGTH (معادل مقدار &HC1) - طول سطری که مکان نما در آن قرار دارد را بر می گرداند. محل مکان نما را به عنوان پارامتر همراه این پیام ارسال کنید.
- EM_LINEINDEX (معادل مقدار &HBB) تعداد کاراکترهای تمامی سطرها قبل از محل مکان نما را مشخص می کند. نتایج پیام EM_LINEFROMCHAR را به عنوان پارامتر همراه این پیام ارسال نمایید. هر سطر حاوی یک کاراکتر Carriage Return و یک Line Feed می باشد.
- EM_GETLINECOUNT (معادل مقدار &HBA) - تعداد کل سطرها یک text box چند سطری را مشخص می کند.
- EM_GETLINE (معادل مقدار &HC4) - متن مربوط به سطری که مکان نما در آن قرار دارد را مشخص می کند. نتایج حاصل از پیام EM_LINEFROMCHAR را به همراه این پیام به عنوان پارامتر ارسال نمایید.

در مثال زیر نحوه استفاده از این تکنیک ها نشان داده می شود:

- ۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید. Form1 بطور پیش فرض بوجود می آید.
- ۲- یک CommandButton، دو عدد Label و دو عدد Text box روی Form1 قرار دهید و خصوصیت های گفته شده در زیر را برای آنها تنظیم کنید:

| نوع ابزار | نام پیش فرض | خصوصیت | مقدار |
|-----------|-------------|-----------|-------|
| Text Box | Text1 | Multiline | True |
| Label | Label1 | AutoSize | True |

۳- کد زیر را در Form1 وارد کنید:

```
Option Explicit
Private Declare Function SendMessage Lib "user32" _
    Alias "SendMessageA" _
    (ByVal hwnd As Long, _
    ByVal wParam As Long, _
    ByVal lParam As String) _
    As Long

Private Const EM_GETLINE = &HC4
Private Const EM_GETLINECOUNT = &HBA
Private Const EM_LINEFROMCHAR = &HC9
Private Const EM_LINEINDEX = &HBB
Private Const EM_LINELENGTH = &HC1
Private Const EM_REPLACESEL = &HC2

Private Sub Form_Load()
    With Command1
        .Caption = "Insert This Text"
        .Height = 375
        .Left = 120
        .Top = 3000
        .Width = 1455
    End With

    With Form1
        .Caption = "Enhanced Text Box Sample Project"
        .Height = 4485
        .Width = 6990
    End With

    With Label1
        .Height = 195
        .Left = 120
        .Top = 240
        .Width = 3015
        .WordWrap = True
    End With

    With Label2
        .Height = 255
        .Left = 1800
        .Top = 3360
        .Width = 3135
    End With

    With Text1
        .Height = 2655
        .Left = 3360
        .Top = 240
        .Width = 3375
    End With

    With Text2
        .Height = 285
        .Left = 1680
        .Top = 3000
        .Width = 5055
    End With
End Sub
```

```

End Sub

Private Sub Command1_Click()
    Call SendMessage(Text1.hwnd, EM_REPLACESEL, 0, Text2.Text)
    Label2.Caption = "Copied text is " + Text2.Text
    ShowInfo
End Sub

Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
    ShowInfo
End Sub

Private Sub Text1_MouseMove(Button As Integer, _
                             Shift As Integer, _
                             X As Single, _
                             Y As Single)
    ShowInfo
End Sub

Private Function ShowInfo()
'*****
' Purpose:  Runs all the functions to get the necessary information
'           in order to create the information text string. The text
'           string is then displayed in the label control.
'*****

    Dim strInfo As String
    Dim lCurPos As Long
    Dim lCurLineNum As Long, lTotLines As Long, lLineLength
    Dim lNumBChar As Long
    Dim sCurLine As String * 25

    'Determine Cursor Position
    If Text1.SelLength = 0 Then
        lCurPos = Text1.SelStart
    Else
        lCurPos = Text1.SelStart + Text1.SelLength
    End If

    'Determine Line Number
    lCurLineNum = SendMessage(Text1.hwnd, EM_LINEFROMCHAR, lCurPos, 0)

    'Determine the Line Length
    lLineLength = SendMessage(Text1.hwnd, EM_LINELENGTH, lCurPos, 0)

    'Determine the number of characters in lines before current
    'cursor position. Note that the number of characters includes a
    'carriage return and line feed characters at the end of each
    'line.
    lNumBChar = SendMessage(Text1.hwnd, EM_LINEINDEX, lCurLineNum, 0)

    'Determine Total number of lines
    lTotLines = SendMessage(Text1.hwnd, EM_GETLINECOUNT, 0, 0)

    'Determine Current Line
    sCurLine = Space(25)
    Call SendMessage(Text1.hwnd, EM_GETLINE, lCurLineNum, sCurLine)

```

```
'Display the information
strInfo = "Current Line: " + sCurLine + vbLf + _
         "Line Length: " + CStr(lLineLength) + _
         " Characters" + vbLf + _
         "Line Number: " + CStr(lCurLineNum + 1) + _
         " of " + CStr(lTotLines) + " Total Lines" + vbLf + _
         "Cursor Position: " + CStr(lCurPos) + vbLf + _
         "Total Characters in Previous Lines: " + _
         CStr(lNumBChar) + vbLf + _
         "Selected Length: " + CStr(Text1.SelLength) + vbLf + _
         "Selected Text: " + Text1.SelText

Label1.Caption = strInfo

End Function
```

۴- در منوی Run گزینه Start را انتخاب کنید یا کلید F5 را فشار دهید تا برنامه اجرا شود. در درون Text1 کلیک کنید تا اطلاعات مربوط به این Text Box در Label1 نمایش داده شود. برای وارد کردن متن در Text1، مکان نما را در محل مورد نظر خود در این ابزار قرار داده و متن مورد نظر را در Text2 وارد کنید و سپس CommandButton را کلیک نمایید.