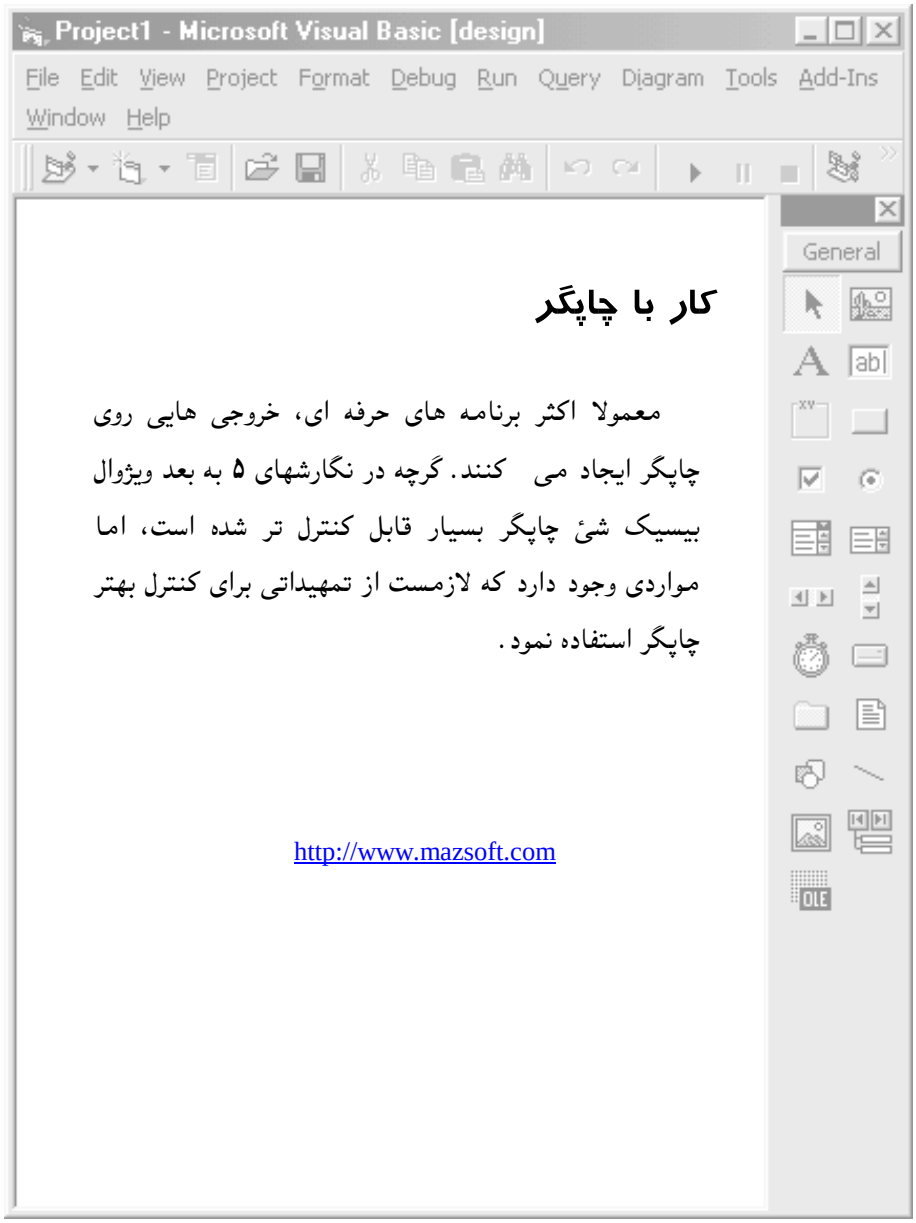

Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins
Window Help
General
کار با چاپگر
معمولا اکثر برنامه های حرفه ای، خروجی هایی روی چاپگر ایجاد می کنند. گرچه در نگارشهای ۵ به بعد ویژوال بیسیک شئ چاپگر بسیار قابل کنترل تر شده است، اما مواردی وجود دارد که لازمست از تمهیداتی برای کنترل بهتر چاپگر استفاده نمود.
http://www.mazsoft.com

استخراج نام چاپگر از رجیستری ویندوز ۹۵ / ۹۸ در برنامه ویژوال بیسیک

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
143274.TXT	۲۴ آگوست ۱۹۹۸	4 , 5 , 6	95 / 98

در ویندوز **۹۵** و ویندوز **۹۸** برای تعیین اینکه چه برنامه ها و چه سخت افزارهایی در کامپیوتر نصب شده اند، از رجیستری استفاده می شود. در این بخش روشی مورد بررسی قرار می گیرد که به کمک آن می توان نام چاپگر پیش فرض سیستم را توسط برنامه ویژوال بیسیک از رجیستری استخراج کرد.

کار با رجیستری در ویژوال بیسیک

رجیستری ویندوز **۹۵** ⁄ **۹۸** یک بانک داده است از اطلاعاتی شامل جزئیات پیکربندی سخت افزارها و نرم افزارهای نصب شده روی سیستم کامپیوتری شما. این اطلاعات در ویندوز ۱ ⁄۳ در درون فایلهای راه اندازی (INI) ذخیره می شدند.

رجیستری تشکیل شده است از یکسری "کلید"ها. هر کلید ممکنست حاوی یک مقدار مشخص یا زیر-کلیدهایی باشد که آنها نیز به نوبه خود ممکنست حاوی مقادیر یا زیر-کلیدهایی باشند. می توانید از درون یک برنامه ویژوال بیسیک به کمک توابع Win32 مربوطه یا از طریق ویرایشگر رجیستری (REGEDIT)، محتویات بانک اطلاعاتی رجیستری را بررسی نموده یا دستکاری نمایید.

برنامه زیر نشان می دهد که چگونه از توابع Win32 API برای تعیین نام چاپگر پیش فرض و استخراج آن از رجیستری استفاده نمایید.

۱- قدم اول در استخراج نام چاپگر عبارتست از فراخوانی تابع RegOpenKeyEx. این تابع کلید مشخص شده را در بانک اطلاعاتی رجیستری باز می کند. در مورد این مثال می خواهیم کلیدی که به چاپگر مربوط است را باز کنیم. این کلید در رجیستری بصورت زیر ذخیره می شود:

```
System
   Current Control Set
      Control
         Print
            Printers
               Default
```

تمامی آیتمهای فوق کلید و زیر-کلید هستند. کلید مورد نیاز ما در این مثال زیر-کلید Printers است.

به علاوه نیاز داریم به تابع RegOpenKeyEx بگوییم که هدف اصلی ما زیر-کلید Default است. بعد از فراخوانی این تابع یک مقدار برگردانده می شود که اگر تابع با موفقیت عمل کند این مقدار صفر خواهد بود.

۲- قدم بعدی عبارتست از بدست آوردن مقدار ذخیره شده در کلیدی که مورد نظر ما بوده است. از آنجا که قصد داریم نام نسبت داده شده به چاپگر پیش فرض را پیدا کنیم، باید تابع RegQueryValueEx را فرابخوانیم. باید به این تابع بگوییم که قصد داریم مقداری که در درون زیر-کلید subkey قرار دارد را بدست آوریم.

۳- مرحله آخر نباید فراموش شود. باید تابع RegCloseKey را فرابخوانید تا شناسه کلیدی که در بانک اطلاعاتی رجیستری مورد استفاده شما بوده است آزاد گردد. این کار باعث می شود دسترسی به بانک اطلاعاتی رجیستری خاتمه یافته و شناسه مورد استفاده آن برای کاربردهای آینده توسط سیستم آزاد شود.

چگونه برنامه آزمایشی را ایجاد کنیم

مثال زیر نشان می دهد چگونه نام چاپگر پیش فرض را از رجیستری ویندوز ۹۵ استخراج کنید.

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

۲- ثابت ها و دستورات Declare زیر را در بخش Declarations از From1 وارد کنید:

```
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias _
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, _
ByVal dwReserved As Long, ByVal samDesired As Long, phkResult _
As Long) As Long

Private Declare Function RegQueryValueEx Lib "advapi32" Alias _
```

```
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName$, ByVal _
lpdwReserved As Long, lpdwType As Long, lpData As Any, lpcbData As _
Long) As Long

Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As _
Long) As Long

Const HKEY_CURRENT_CONFIG As Long = &H80000005
```

۳- یک Text Box روی Form1 قرار دهید.

۴- یک Command Button روی Form1 قرار دهید.

۵- کد زیر را در واقعه کلیک از Command1 قرار دهید:

```
Private Sub Command1_Click()
    Dim PName As String
    PName = GetCurrPrinter()
    Text1.Text = PName
End Sub
```

۶- یک زیربرنامه به نام GetCurrPrinter ایجاد کرده و کد زیر را در آن وارد کنید:

```
Function GetCurrPrinter() As String
   GetCurrPrinter = RegGetString$(HKEY_CURRENT_CONFIG, _
   "System\CurrentControlSet\Control\Print\Printers", "Default")
End Function
```

۷- یک زیربرنامه دیگر به نام RegGetString ایجاد کرده و کد زیر را در آن وارد کنید:

```
Function RegGetString$(hInKey As Long, ByVal subkey$, ByVal valname$)
    Dim RetVal$, hSubKey As Long, dwType As Long, SZ As Long
    Dim R As Long

    RetVal$ = ""
    Const KEY_ALL_ACCESS As Long = &HF0063
    Const ERROR_SUCCESS As Long = 0
    Const REG_SZ As Long = 1

    R = RegOpenKeyEx(hInKey, subkey$, 0, KEY_ALL_ACCESS, hSubKey)
    If R <> ERROR_SUCCESS Then GoTo Quit_Now

    SZ = 256: v$ = String$(SZ, 0)
    R = RegQueryValueEx(hSubKey, valname$, 0, dwType, ByVal v$, SZ)
    If R = ERROR_SUCCESS And dwType = REG_SZ Then
        RetVal$ = Left$(v$, SZ)
    Else
        RetVal$ = "--Not String--"
    End If
    If hInKey = 0 Then R = RegCloseKey(hSubKey)
Quit_Now:
    RegGetString$ = RetVal$
End Function
```

۸- برنامه حاصل را با فشار کلید F5 اجرا کنید. هنگامی که روی Command Button کلیک می کنید، نام چاپگر پیش فرض شما در Text Box نمایش داده می شود.

چاپ نوشته های دوران یافته توسط Win32 API

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
154515.TXT	۹ فوریه ۱۹۹۹	4 , 6	95 / 98 / NT

به کمک توابع CreateFont و CreateFontIndirect این امکان وجود دارد که نوشته ها را روی یک دستگاه مانند Picture Box ، فرم یا چاپگر بصورت دوران یافته نمایش داد. در این بخش روش انجام این کار را به کمک توابع Win32 ارایه می دهیم.

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

۲- یک Command Button به Form1 اضافه کنید.

۳- یک Picture Box روی Form1 قرار دهید.

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

```
Option Explicit

Private Declare Function CreateFontIndirect Lib "gdi32" Alias _
  "CreateFontIndirectA" (lpLogFont As LOGFONT) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc _
  As Long, ByVal hObject As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal _
  hObject As Long) As Long
Private Const LF_FACESIZE = 32

Private Type LOGFONT
  lfHeight As Long
  lfWidth As Long
  lfEscapement As Long
  lfOrientation As Long
  lfWeight As Long
  lfItalic As Byte
  lfUnderline As Byte
  lfStrikeOut As Byte
  lfCharSet As Byte
```

```
  lfOutPrecision As Byte
  lfClipPrecision As Byte
  lfQuality As Byte
  lfPitchAndFamily As Byte
  lfFaceName as String * LF_FACESIZE
End Type

Sub Command1_Click()
  Dim font As LOGFONT
  Dim prevFont As Long, hFont As Long, ret As Long
  Const FONTSIZE = 10 ' Desired point size of font
  font.lfEscapement = 1800    ' 180-degree rotation
  font.lfFaceName = "Arial" & Chr$(0) 'Null character at end
  ' Windows expects the font size to be in pixels and to
  ' be negative if you are specifying the character height
  ' you want.
  font.lfHeight = (FONTSIZE * -20) / Screen.TwipsPerPixelY
  hFont = CreateFontIndirect(font)
  prevFont = SelectObject(Picture1.hdc, hFont)
  Picture1.CurrentX = Picture1.Left + Picture1.Width / 2
  Picture1.CurrentY = Picture1.ScaleHeight / 2
  Picture1.Print "Rotated Text"
  ' Clean up by restoring original font.
  ret = SelectObject(Picture1.hdc, prevFont)
  ret = DeleteObject(hFont)
  Picture1.CurrentY = Picture1.ScaleHeight / 2
  Picture1.Print "Normal Text"
End Sub
```

۵- کلید F5 را فشار دهید تا پروژه اجرا شود، سپس Command Button را فشار دهید. دو قطعه نوشته را خواهید دید که ظاهر می شوند، یکی معمولی و دیگری بصورت معکوس (دوران یافته).

شکل 1 -نتیجه اجرای مثال دوران متن

☜ توجه: اگر این دستور العمل ها را به جای Picture Box روی چاپگر بکار ببرید، در ویژوال بیسیک **۵** عمل نخواهد کرد(حتی با داشتن SP2 یا SP3). برای اینکه نوشته های دوران یافته را در ویژوال بیسیک **۵** روی چاپگر داشته باشید، به بخش شماره 175535 از بانک اطلاعاتی

مایکروسافت تحت عنوان "اشکالات چاپ نوشته های دوران یافته در ویژوال بیسیک ۵" مراجعه کنید.

اشکالات چاپ نوشته های دوران یافته در ویژوال بیسیک ۵

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
175535.TXT	۱۱ نوامبر ۱۹۹۸	4 , 5 , 6	95 / 98 / NT

اگر بخواهید در ویژوال بیسیک ۵ (مجهز به Sp2 یا Sp3) به کمک روش گفته شده در بخش شماره 154515 از بانک اطلاعاتی مایکروسافت تحت عنوان "چاپ نوشته های دوران یافته توسط Win32 API" متنی را روی چاپگر بصورت دوران یافته چاپ کنید با شکست مواجه خواهید شد.

سرویس پک های ۲ و ۳ ویژوال استودیو حاوی اصلاحاتی برای شئ چاپگر هستند که باعث می شوند نتوان فراخوانی SelectObject را برای انتخاب فونت برای Printer.hDC انجام داد. بنابراین ، هنگام رجوع به شئ چاپگر، فونتهای جدید باید از طریق خصوصیت های شئ چاپگر انتخاب شوند و الا مورد استفاده قرار نمی گیرند.

برای حل این مشکل یا باید فقط توسط API عمل چاپ را انجام دهید یا اینکه هنگام چاپ با توابع API ، اصلا به شئ چاپگر رجوع نشود. این مشکل مربوط به طراحی ویژوال بیسیک است.

روش مورد بحث در بخش شماره 154515 نوشته را روی یک Picture Box نمایش می دهد اما می توان آنرا با شئ چاپگر نیز بکار برد. در Sp2 نمایش نوشته روی Picture Box همچنان قابل انجام است ولی روی شئ چاپگر نتیجه نمی دهد. مثال زیر روی تمامی نگارش های ۴ به بعد ویژوال بیسیک قابل اجرا است.

نکته اصلی در انجام این کار استفاده از تابع TextOut به جای Printer.Print و عدم رجوع مستقیم به شئ چاپگر در حین تنظیم فونت و انجام چاپ با آن است. اگر از TextOut به همراه Printer.hDC استفاده کنید بازهم نتیجه نخواهید گرفت. باید از یک متغیر محلی به جای Printer.hDC استفاده نمایید. هر گونه رجوع به خصوصیت ها یا فرامین مربوط به شئ چاپگر به معنی استفاده از شئ چاپگر بوده و باعث می شود شئ چاپگر فونتها، قلم[۱] ها و قلم مو[۲] های خود را بحال اول بازگرداند. بنابراین، پیش از استفاده از SelectObject باید مقدار تمامی خصوصیت های شئ چاپگر که مورد نیاز شما خواهد بود را در متغیرهایی ذخیره نمایید و سپس فقط از این متغیرها استفاده نمایید.

۱- یک پروژه EXE در ویژوال بیسیک ایجاد کنید. Form1 بطور پیش فرض ایجاد می شود.

۲- دو Command Button به Form1 اضافه کنید.

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

```
Option Explicit

Private Const LF_FACESIZE = 32

Private Type LOGFONT
   lfHeight As Long
   lfWidth As Long
   lfEscapement As Long
   lfOrientation As Long
   lfWeight As Long
   lfItalic As Byte
   lfUnderline As Byte
   lfStrikeOut As Byte
   lfCharSet As Byte
   lfOutPrecision As Byte
   lfClipPrecision As Byte
   lfQuality As Byte
   lfPitchAndFamily As Byte
   lfFaceName As String * LF_FACESIZE
End Type

Private Type DOCINFO
   cbSize As Long
   lpszDocName As String
   lpszOutput As String
   lpszDatatype As String
```

---

[۱] Pen

[۲] Brush

```
       fwType As Long
    End Type

Private Declare Function CreateFontIndirect Lib "gdi32" Alias _
    "CreateFontIndirectA" (lpLogFont As LOGFONT) As Long

Private Declare Function SelectObject Lib "gdi32" _
    (ByVal hdc As Long, ByVal hObject As Long) As Long

Private Declare Function DeleteObject Lib "gdi32" _
    (ByVal hObject As Long) As Long

Private Declare Function CreateDC Lib "gdi32" Alias "CreateDCA" _
    (ByVal lpDriverName As String, ByVal lpDeviceName As String, _
    ByVal lpOutput As Long, ByVal lpInitData As Long) As Long

 Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) _
    As Long

 Private Declare Function TextOut Lib "gdi32" Alias "TextOutA" _
   (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, _
   ByVal lpString As String, ByVal nCount As Long) As Long ' or Boolean

 Private Declare Function StartDoc Lib "gdi32" Alias "StartDocA" _
    (ByVal hdc As Long, lpdi As DOCINFO) As Long

 Private Declare Function EndDoc Lib "gdi32" (ByVal hdc As Long) _
    As Long

 Private Declare Function StartPage Lib "gdi32" (ByVal hdc As Long) _
    As Long

 Private Declare Function EndPage Lib "gdi32" (ByVal hdc As Long) _
    As Long

 Const DESIREDFONTSIZE = 12     ' Could use variable, TextBox, etc.

 Private Sub Command1_Click()
    ' Combine API Calls with the Printer object
     Dim OutString As String
     Dim lf As LOGFONT
     Dim result As Long
     Dim hOldfont As Long
     Dim hPrintDc As Long
     Dim hFont As Long

     Printer.Print "Printer Object"
     hPrintDc = Printer.hdc
     OutString = "Hello World"

     lf.lfEscapement = 1800
     lf.lfHeight = (DESIREDFONTSIZE * -20) / Printer.TwipsPerPixelY
     hFont = CreateFontIndirect(lf)
     hOldfont = SelectObject(hPrintDc, hFont)
     result = TextOut(hPrintDc, 1000, 1000, OutString, Len(OutString))
     result = SelectObject(hPrintDc, hOldfont)
     result = DeleteObject(hFont)

     Printer.Print "xyz"
```

```
    Printer.EndDoc
End Sub

Private Sub Command2_Click()
   ' Print using API calls only
    Dim OutString As String  'String to be rotated
    Dim lf As LOGFONT        'Structure for setting up rotated font
    Dim temp As String       'Temp string var
    Dim result As Long       'Return value for calling API functions
    Dim hOldfont As Long     'Hold old font information
    Dim hPrintDc As Long     'Handle to printer dc
    Dim hFont As Long        'Handle to new Font
    Dim di As DOCINFO        'Structure for Print Document info

    OutString = "Hello World"   'Set string to be rotated

   ' Set rotation in tenths of a degree, i.e., 1800 = 180 degrees
    lf.lfEscapement = 1800
    lf.lfHeight = (DESIREDFONTSIZE * -20) / Printer.TwipsPerPixelY
    hFont = CreateFontIndirect(lf)  'Create the rotated font
    di.cbSize = 20                  ' Size of DOCINFO structure
    di.lpszDocName = "My Document" ' Set name of print job (Optional)

   ' Create a printer device context
    hPrintDc = CreateDC(Printer.DriverName, Printer.DeviceName, 0, 0)

    result = StartDoc(hPrintDc, di) 'Start a new print document
    result = StartPage(hPrintDc)    'Start a new page

   ' Select our rotated font structure and save previous font info
    hOldfont = SelectObject(hPrintDc, hFont)

   ' Send rotated text to printer, starting at location 1000, 1000
    result = TextOut(hPrintDc, 1000, 1000, OutString, Len(OutString))

   ' Reset font back to original, non-rotated
    result = SelectObject(hPrintDc, hOldfont)

   ' Send non-rotated text to printer at same page location
    result = TextOut(hPrintDc, 1000, 1000, OutString, Len(OutString))

    result = EndPage(hPrintDc)      'End the page
    result = EndDoc(hPrintDc)       'End the print job
    result = DeleteDC(hPrintDc)     'Delete the printer device context
    result = DeleteObject(hFont)    'Delete the font object
End Sub

Private Sub Form_Load()
    Command1.Caption = "API with Printer object"
    Command2.Caption = "Pure API"
End Sub
```

۴- برنامه را اجرا کنید و روی دکمه API with Printer object کلیک کنید. این کار باعث می شود یک صفحه روی چاپگر پیش فرض چاپ شود که روی آن عبارات Printer Object و xyz بصورت معمولی و عبارت Hello World با زاویه ۱۸۰ درجه وجود دارند.

۵- روی دکمه Pure API کلیک کنید. این کار باعث می شود یک صفحه روی چاپگر پیش فرض چاپ شود که روی آن عبارت Hello World بصورت معمولی و نیز با زاویه ۱۸۰ درجه به چشم می خورد.

استخراج تنظیمات از یک درایور چاپگر

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
190218.TXT	۷ آگوست ۱۹۹۸	4 , 5 , 6	95 / 98 / NT

شئ چاپگر ویژوال بیسیک امکان دسترسی به خیلی از خصوصیت های چاپگر را فراهم می آورد، اما تمامی خصوصیت ها را در دسترس قرار نمی دهد. مثال زیر روش استخراج اطلاعات از درایور چاپگر به کمک توابع Win32 API را نمایش می دهد.

تنظیمات یک چاپگر در یک ساختمان داده به نام DEVMODE ذخیره می شوند. مثال زیر DEVMODE را از چاپگر جاری استخراج کرده و تنظیماتش را در یک ListBox نمایش می دهد. توجه داشته باشید که این تنظیمات مقادیر عددی هستند که باید با ثابت های متناظر خود مقایسه شوند تا مفهوم آنها مشخص شود. این بخش فقط حاوی ثابت هایی است که در این مثال بکار می روند. بقیه ثابت ها را می توان در مراجع انتهای همین بخش یافت.

مثال زیر حاوی دو تابع StripNulls و ByteToString است. StripNulls یک رشته را می گیرد و تمام کاراکترهای بعد از اولین null را از آن حذف می کند. این روتین بجای تابع RTrim بکار می رود زیرا در ویژوال بیسیک ۵، فضای خالی در رشته های با طول ثابت به جای اینکه با کاراکتر فاصله پر شوند (مانند ویژوال بیسیک ۴) با کاراکتر null پر می شوند. تابع ByteToString یک آرایه از مقادیر از نوع Byte را به یک رشته تبدیل می کند. DEVMODE حاوی دو تا از این آرایه ها است.

هنگامی که مثال زیر را در محیط ویندوز NT اجرا می کنید باید به دو نکته توجه نمایید. اول، در خیلی از درایورهای تک رنگ خصوصیت dmColor معادل ۲ است، که به این معنی است که این چاپگر توان چاپ رنگی نیز دارد، اما در حقیقت چنین توانی ندارد. به علاوه، بعضی از تنظیمات برای درایور چاپگر Generic / Text Only ممکنست در محیط NT درست نباشد.

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

۲- یک CommandButton و یک ListBox به Form1 اضافه کنید.

۳- کد زیر را در Form1 وارد کنید:

```
Option Explicit

Private Const NULLPTR = 0&
' Constants for DEVMODE
Private Const CCHDEVICENAME = 32
Private Const CCHFORMNAME = 32
' Constants for DocumentProperties
Private Const DM_MODIFY = 8
Private Const DM_COPY = 2
Private Const DM_IN_BUFFER = DM_MODIFY
Private Const DM_OUT_BUFFER = DM_COPY
' Constants for dmOrientation
Private Const DMORIENT_PORTRAIT = 1
Private Const DMORIENT_LANDSCAPE = 2
' Constants for dmPrintQuality
Private Const DMRES_DRAFT = (-1)
Private Const DMRES_HIGH = (-4)
Private Const DMRES_LOW = (-2)
Private Const DMRES_MEDIUM = (-3)
' Constants for dmTTOption
Private Const DMTT_BITMAP = 1
Private Const DMTT_DOWNLOAD = 2
Private Const DMTT_DOWNLOAD_OUTLINE = 4
Private Const DMTT_SUBDEV = 3
' Constants for dmColor
Private Const DMCOLOR_COLOR = 2
Private Const DMCOLOR_MONOCHROME = 1

Private Type DEVMODE
    dmDeviceName(1 To CCHDEVICENAME) As Byte
    dmSpecVersion As Integer
    dmDriverVersion As Integer
    dmSize As Integer
    dmDriverExtra As Integer
    dmFields As Long
    dmOrientation As Integer
    dmPaperSize As Integer
    dmPaperLength As Integer
    dmPaperWidth As Integer
    dmScale As Integer
    dmCopies As Integer
    dmDefaultSource As Integer
```

```
        dmPrintQuality As Integer
        dmColor As Integer
        dmDuplex As Integer
        dmYResolution As Integer
        dmTTOption As Integer
        dmCollate As Integer
        dmFormName(1 To CCHFORMNAME) As Byte
        dmUnusedPadding As Integer
        dmBitsPerPel As Integer
        dmPelsWidth As Long
        dmPelsHeight As Long
        dmDisplayFlags As Long
        dmDisplayFrequency As Long
    End Type

Private Declare Function OpenPrinter Lib "winspool.drv" Alias _
    "OpenPrinterA" (ByVal pPrinterName As String, phPrinter As Long, _
    ByVal pDefault As Long) As Long

Private Declare Function DocumentProperties Lib "winspool.drv" _
    Alias "DocumentPropertiesA" (ByVal hwnd As Long, _
    ByVal hPrinter As Long, ByVal pDeviceName As String, _
    pDevModeOutput As Any, pDevModeInput As Any, ByVal fMode As Long) _
    As Long

Private Declare Function ClosePrinter Lib "winspool.drv" _
    (ByVal hPrinter As Long) As Long

Private Declare Sub CopyMemory Lib "KERNEL32" Alias "RtlMoveMemory" _
    (hpvDest As Any, hpvSource As Any, ByVal cbCopy As Long)

Function StripNulls(OriginalStr As String) As String
    If (InStr(OriginalStr, Chr(0)) > 0) Then
        OriginalStr = Left(OriginalStr, InStr(OriginalStr, Chr(0)) - 1)
    End If
    StripNulls = Trim(OriginalStr)
End Function

Function ByteToString(ByteArray() As Byte) As String
    Dim TempStr As String
    Dim I As Integer

    For I = 1 To CCHDEVICENAME
        TempStr = TempStr & Chr(ByteArray(I))
    Next I
    ByteToString = StripNulls(TempStr)
End Function

Function GetPrinterSettings(szPrinterName As String, hdc As Long) _
    As Boolean
    Dim hPrinter As Long
    Dim nSize As Long
    Dim pDevMode As DEVMODE
    Dim aDevMode() As Byte
    Dim TempStr As String

    If OpenPrinter(szPrinterName, hPrinter, NULLPTR) Then
        nSize = DocumentProperties(NULLPTR, hPrinter, szPrinterName, _
        NULLPTR, NULLPTR, 0)
```

```
ReDim aDevMode(1 To nSize)
nSize = DocumentProperties(NULLPTR, hPrinter, szPrinterName, _
aDevMode(1), NULLPTR, DM_OUT_BUFFER)
Call CopyMemory(pDevMode, aDevMode(1), Len(pDevMode))

List1.Clear   ' empty the ListBox
List1.AddItem "Printer Name: " & _
ByteToString(pDevMode.dmDeviceName)

If pDevMode.dmOrientation = DMORIENT_PORTRAIT Then
   TempStr = "PORTRAIT"
ElseIf pDevMode.dmOrientation = DMORIENT_LANDSCAPE Then
   TempStr = "LANDSCAPE"
Else
   TempStr = "UNDEFINED"
End If
List1.AddItem "Orientation: " & TempStr

Select Case pDevMode.dmPrintQuality
   Case DMRES_DRAFT
       TempStr = "DRAFT"
   Case DMRES_HIGH
       TempStr = "HIGH"
   Case DMRES_LOW
       TempStr = "LOW"
   Case DMRES_MEDIUM
       TempStr = "MEDIUM"
   Case Else   ' positive value
       TempStr = CStr(pDevMode.dmPrintQuality) & " dpi"
End Select
List1.AddItem "Print Quality: " & TempStr

Select Case pDevMode.dmTTOption
   Case DMTT_BITMAP    ' default for dot-matrix printers
       TempStr = "TrueType fonts as graphics"
   Case DMTT_DOWNLOAD  ' default for HP printers that use PCL
       TempStr = "Downloads TrueType fonts as soft fonts"
   Case DMTT_SUBDEV    ' default for PostScript printers
       TempStr = "Substitute device fonts for TrueType fonts"
   Case Else
       TempStr = "UNDEFINED"
End Select
List1.AddItem "TrueType Option: " & TempStr

' Windows NT drivers often return COLOR from Monochrome printers
If pDevMode.dmColor = DMCOLOR_MONOCHROME Then
   TempStr = "MONOCHROME"
ElseIf pDevMode.dmColor = DMCOLOR_COLOR Then
   TempStr = "COLOR"
Else
   TempStr = "UNDEFINED"
End If
List1.AddItem "Color or Monochrome: " & TempStr

If pDevMode.dmScale = 0 Then
   TempStr = "NONE"
Else
   TempStr = CStr(pDevMode.dmScale)
End If
```

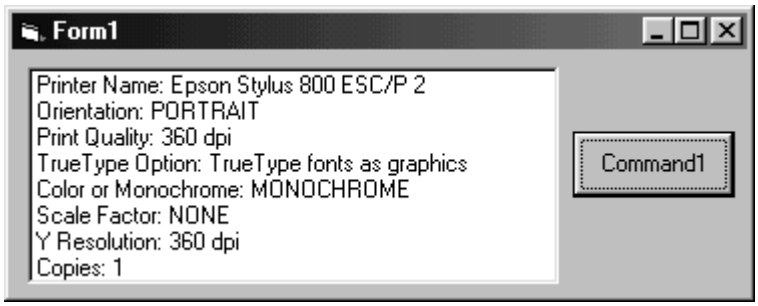
```
      List1.AddItem "Scale Factor: " & TempStr

      List1.AddItem "Y Resolution: " & pDevMode.dmYResolution & " dpi"
      List1.AddItem "Copies: " & CStr(pDevMode.dmCopies)
      ' Add any other items of interest ...

      Call ClosePrinter(hPrinter)
      GetPrinterSettings = True
   Else
      GetPrinterSettings = False
   End If
End Function

Private Sub Command1_Click()
   If GetPrinterSettings(Printer.DeviceName, Printer.hdc) = False Then
      List1.AddItem "No Settings Retrieved!"
      MsgBox "Unable to retrieve Printer settings.", , "Failure"
   End If
End Sub
```

پروژه را اجرا کنید و روی Command1 کلیک کنید. تنظیمات چاپگر جاری پیش فرض در ListBox نمایش داده خواهد شد.



شکل 2 - تنظیمات چاپگر

مراجع

برای کسب اطلاعات بیشتر در مورد بقیه ثابت ها، می توانید در مراجع زیر عبارت DEVMODE را جستجو کنید:

1- Win32 Programmer's Reference
2- Microsoft Developer Network (MSDN) Library CD-ROM

پیش نمایش چاپ در برنامه های ویژوال بیسیک

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
193379.TXT	۲۹ سپتامبر ۱۹۹۸	5 , 6	95 / 98 / NT

در این بخش چگونگی ایجاد یک برنامه را خواهید دید که امکان پیش نمایش چاپ را به برنامه های ویژوال بیسیک اضافه می کند. این برنامه با ایجاد یک شئ کمکی و استفاده از شئ چاپگر این امکان را برای تمامی برنامه های شما فراهم می نماید. در این بخش فرض بر این است که شما با اشیاء و ابزار دیالوگهای عمومی آشنایی دارید.

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

۲- با انجام مراحل زیر یک ارجاع به ابزار دیالوگهای عمومی مایکروسافت (نگارش ٦) به برنامه خود اضافه کنید:

- از منوی Project گزینه Components را انتخاب کنید تا پنجره Components نمایش داده شود.
- در تابلوی Controls آیتم Microsoft Common Dialog Control 6.0 را یافته روی آن کلیک کنید تا علامت چک مارک کنارش ظاهر شود.
- برای بستن پنجره Components روی OK کلیک کنید.

۳- سه CommandButton ، دو PictureBox و یک ابزار دیالوگهای عمومی (Common Dialog control) به Form1 اضافه کنید.

۴- کد زیر را در Form1 وارد کنید:

```
Option Explicit

Private Sub Form_Load()
   CommonDialog1.CancelError = True
   Command1.Caption = "Load Picture"
   Command2.Caption = "Print Preview"
   Command3.Caption = "Print"
End Sub

Private Sub Command1_Click()
   Dim sFileFilter As String
```

```
   On Error GoTo ErrHandler

   sFileFilter = "Bitmap Files (*.bmp)|*.bmp|"
   sFileFilter = sFileFilter & "GIF Files (*.gif)|*.gif|"
   sFileFilter = sFileFilter & "Icon Files (*.ico)|*.ico|"
   sFileFilter = sFileFilter & "JPEG Files (*.jpg)|*.jpg|"
   sFileFilter = sFileFilter & "Windows MetaFiles (*.wmf)|.wmf"
   With CommonDialog1
      .Filter = sFileFilter
      .ShowOpen
      If .FileName <> " " Then
         Picture2.Picture = LoadPicture(.FileName)
      End If
   End With

   ErrHandler:
      Exit Sub
End Sub

Private Sub Command2_Click()
   Dim dRatio As Double
   dRatio = ScalePicPreviewToPrinterInches(Picture1)
   PrintRoutine Picture1, dRatio
End Sub

Private Sub Command3_Click()
   Printer.ScaleMode = vbInches
   PrintRoutine Printer
   Printer.EndDoc
End Sub

Private Function ScalePicPreviewToPrinterInches _
   (picPreview As PictureBox) As Double

   Dim Ratio As Double ' Ratio between Printer and Picture
   Dim LRGap As Double, TBGap As Double
   Dim HeightRatio As Double, WidthRatio As Double
   Dim PgWidth As Double, PgHeight As Double
   Dim smtemp As Long

   ' Get the physical page size in Inches:
   PgWidth = Printer.Width / 1440
   PgHeight = Printer.Height / 1440

   ' Find the size of the non-printable area on the printer to
   ' use to offset coordinates. These formulas assume the
   ' printable area is centered on the page:
   smtemp = Printer.ScaleMode
   Printer.ScaleMode = vbInches
   LRGap = (PgWidth - Printer.ScaleWidth) / 2
   TBGap = (PgHeight - Printer.ScaleHeight) / 2
   Printer.ScaleMode = smtemp

   ' Scale PictureBox to Printer's printable area in Inches:
   picPreview.ScaleMode = vbInches

   ' Compare the height and with ratios to determine the
   ' Ratio to use and how to size the picture box:
```

```
   HeightRatio = picPreview.ScaleHeight / PgHeight
   WidthRatio = picPreview.ScaleWidth / PgWidth

   If HeightRatio < WidthRatio Then
      Ratio = HeightRatio
      smtemp = picPreview.Container.ScaleMode
      picPreview.Container.ScaleMode = vbInches
      picPreview.Width = PgWidth * Ratio
      picPreview.Container.ScaleMode = smtemp
   Else
      Ratio = WidthRatio
      smtemp = picPreview.Container.ScaleMode
      picPreview.Container.ScaleMode = vbInches
      picPreview.Height = PgHeight * Ratio
      picPreview.Container.ScaleMode = smtemp
   End If

   ' Set default properties of picture box to match printer
   ' There are many that you could add here:
   picPreview.Scale (0, 0)-(PgWidth, PgHeight)
   picPreview.Font.Name = Printer.Font.Name
   picPreview.FontSize = Printer.FontSize * Ratio
   picPreview.ForeColor = Printer.ForeColor
   picPreview.Cls

   ScalePicPreviewToPrinterInches = Ratio
End Function

Private Sub PrintRoutine(objPrint As Object, _
                         Optional Ratio As Double = 1)
   ' All dimensions in inches:

   ' Print some graphics to the control object
   objPrint.Line (1, 1)-(1 + 6.5, 1 + 9), , B
   objPrint.Line (1.1, 2)-(1.1, 2)
   objPrint.PaintPicture Picture2, 1.1, 1.1, 0.8, 0.8
   objPrint.Line (2.1, 1.2)-(2.1 + 5.2, 1.2 + 0.7), _
                 RGB(200, 200, 200), BF

   ' Print a title
   With objPrint
      .Font.Name = "Arial"
      .CurrentX = 2.3
      .CurrentY = 1.3
      .FontSize = 35 * Ratio
      objPrint.Print "Visual Basic Printing"
   End With

   ' Print some circles
   Dim x As Single
   For x = 3 To 5.5 Step 0.2
      objPrint.Circle (x, 3.5), 0.75
   Next

   ' Print some text
   With objPrint
      .Font.Name = "Courier New"
      .FontSize = 30 * Ratio
      .CurrentX = 1.5
```

```
        .CurrentY = 5
        objPrint.Print "It is possible to do"

        .FontSize = 24 * Ratio
        .CurrentX = 1.5
        .CurrentY = 6.5
        objPrint.Print "It is possible to do print"

        .FontSize = 18 * Ratio
        .CurrentX = 1.5
        .CurrentY = 8
        objPrint.Print "It is possible to do print preview"
    End With
End Sub
```

۵- از منوی Run گزینه Start را انتخاب کنید یا کلید F5 را فشار دهید تا برنامه اجرا شود. روی دکمه Load Picture کلیک کنید تا پنجره دیالوگهای عمومی ظاهر شود. یک فایل گرافیک مناسب را انتخاب کرده، روی OK کلیک کنید. تصویر گرافیکی مورد نظر در PictureBox نمایش داده می شود. روی دکمه Print Preview کلیک کنید تا تصویر گرافیکی و مقداری متن در PictureBox دوم نمایش داده شود. روی دکمه Print کلیک کنید تا محتویات PictureBox چاپ شود.

مراجع

برای اضافه کردن قابلیت پیش نمایش چاپ به برنامه های ۱۶ بیتی در ویژوال بیسیک، به آیتم زیر در بانک اطلاعاتی مایکروسافت رجوع کنید.

```
ARTICLE-ID: Q113236
TITLE     : HOWTO: Add Print Preview to Visual Basic Applications
```

استفاده از تابع GetDeviceCaps برای تعیین حاشیه های چاپ یک صفحه

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
193943.TXT	۱۸ دسامبر ۱۹۹۸	4 , 5 , 6	95 / 98 / NT

شئ چاپگر ویژوال بیسیک فقط سایز فیزیکی کاغذ و محدوده قابل چاپ آنرا در دسترس قرار می دهد و امکان دسترسی به حاشیه ها و محدوده غیر قابل چاپ را در اختیار نمی گذارد. برای به دست آوردن این اطلاعات باید از تابع GetDeviceCaps استفاده نمایید. این بخش روش دسترسی و استفاده از این اطلاعات را نشان می دهد.

GetDeviceCaps فقط به دو پارامتر ورودی نیاز دارد. اولی یک شناسه است که مشخص کننده دستگاهی است (DC) که قصد دارید مورد بررسی قرار دهید. در مثال زیر، دستگاه مورد نظر چاپگر جاری است. پارامتر دوم یک مقدار از نوع Long است که مشخص کننده آیتمی است که باید برگردانده شود. مثال زیر فقط حاوی ثابت هایی است که در همین برنامه مورد نیاز بوده اند. بطور معمول، واحد مقدار برگردانده شده متفاوت است اما در اینجا واحد اندازه گیری همیشه پیکسل است.

در مثال زیر Command1 حاوی کدی است که تابع GetDeviceCaps را چندین بار فرا می خواند تا بیت های مختلفی از اطلاعات چاپگر را تعیین کند، که این بیت ها بعداً همگی در یک زمان در انتهای روتین نمایش داده می شوند. اطلاعات به دست آمده حاوی تمامی حاشیه های پیش فرض است و محدوده قابل چاپ کاغذ را مشخص می کند.

کد مربوط به Command2 اطلاعاتی از چاپگر را مورد استفاده قرار می دهد تا پایین ترین محل چاپ روی کاغذ را تعیین کند. سپس از این اطلاعات برای چاپ یک پانویس شامل شماره صفحه در وسط عرض کاغذ استفاده می نماید. یک سربرگ حاوی تاریخ جاری که در سمت راست قرار دارد در بالای هر صفحه چاپ می شود. به علاوه، یک سربرگ گزارشی در بالای صفحه اول چاپ می کند. مثال زیر روش تنظیم مقادیر دلخواه حاشیه، درون محدوده قابل چاپ را نیز نمایش می دهد. برای این کار از تکنیک کنترل مقادیر X و Y جاری استفاده می شود.

می توانید با کمی تغییرات به کاربر امکان دهید این حاشیه ها را خودش تنظیم کند یا حتی متن سربرگ و پانویس را به دلخواه خود تغییر دهد.

روتین PrintHeader تعدادی متن بصورت ایتالیک و تعدادی بصورت معمولی را در یک سطر چاپ می کند. برای انجام این کار مقدار CurrentX و CurrentY را تغییر می دهد تا چاپ در یک سطر صورت بگیرد.

☜ توجه: برای مخلوط کردن صفات فونت می توانید دستور چاپ را با علامت " ; " خاتمه دهید. این کار باعث می شود آخرین موقعیت چاپ حفظ شود و به ابتدای سطر بعد یعنی CurrentX = 0 منتقل نشود. به این ترتیب، می توانید متونی را چاپ کنید، صفات (bold ، italic ، و غیره) فونت را تغییر دهید و سپس چاپ متون دیگری را درست از همان آخرین محل ادامه دهید.

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

۲- دو CommandButton روی Form1 اضافه نمایید و کد زیر را در Form1 وارد نمایید:

```
Option Explicit

Private Declare Function GetDeviceCaps Lib "gdi32" _
 (ByVal hdc As Long, ByVal nIndex As Long) As Long

' Constants for nIndex argument of GetDeviceCaps
Private Const HORZRES = 8
Private Const VERTRES = 10
Private Const LOGPIXELSX = 88
Private Const LOGPIXELSY = 90
Private Const PHYSICALWIDTH = 110
Private Const PHYSICALHEIGHT = 111
Private Const PHYSICALOFFSETX = 112
Private Const PHYSICALOFFSETY = 113

Private Sub PrintHeader(PrintTitle As Boolean, Margins As Long, _
   LMargin As Long)
   Dim BodyFontBold As Boolean, BodyFontSize As Integer
   Dim BodyFontItalic As Boolean, HeaderLine As String
   Dim PrtPositionX As Integer, PrtPositionY As Integer

   If PrintTitle Then ' Only Print Title on first Page
      PrtPositionX = Printer.Width / 2  ' center of page
      ' Save any attributes you change for the header...
      BodyFontBold = Printer.Font.Bold
      BodyFontSize = Printer.Font.Size
      ' Change the font for the header...
      Printer.Font.Size = 16
      Printer.Font.Bold = True
```

```
        ' Print header lines...
        HeaderLine = "Title of This Test Report"
        ' Center Report title
        Printer.CurrentX = PrtPositionX - (LMargin + _
          Printer.TextWidth(HeaderLine) / 2)
        Printer.Print HeaderLine
        ' Reset the font...
        Printer.Font.Size = BodyFontSize
        Printer.Font.Bold = BodyFontBold
     Else
        Printer.NewPage   ' Otherwise we are at the bottom of a page
     End If
     PrtPositionY = Printer.CurrentY  ' remember this line
     BodyFontItalic = Printer.Font.Italic
     Printer.Font.Italic = True
     Printer.Print "My very nice, Visual Basic report"
     Printer.Font.Italic = BodyFontItalic
     HeaderLine = Format(Date, "Long Date")
     ' Right justify today's date
     Printer.CurrentX = Printer.Width - (Margins + _
        Printer.TextWidth(HeaderLine))
     Printer.CurrentY = PrtPositionY  ' return to previous line
     Printer.Print HeaderLine
  End Sub

  Private Sub PrintFooter(LastLine As Integer, _
         LineHeight As Integer, LMargin As Long)
     Dim PrtPositionX As Integer, PrintText As String
     Static PageNum As Integer

     PageNum = PageNum + 1
     PrtPositionX = Printer.Width / 2   ' center of page
     ' Move to the end of the page to print the Page Footer:
     ' LastLine is where there is only room for one more line.
     ' Set CurrentY to LastLine minus LineHeight times the number of
     ' additional lines in the Footer.
     Printer.CurrentY = LastLine - LineHeight  ' for 2 footer lines
     PrintText = "Sample Page Footer"
     ' Print the Footer text centered
     Printer.CurrentX = PrtPositionX - (LMargin + _
        Printer.TextWidth(PrintText) / 2)
     Printer.Print PrintText
     PrintText = "This is Page " & CStr(PageNum) & " of Test Report"
     Printer.CurrentX = PrtPositionX - (LMargin + _
        Printer.TextWidth(PrintText) / 2)
     Printer.Print PrintText
  End Sub

Private Function GetLargeString() As String
   Dim BaseString As String, I As Integer

   BaseString = "This is just a fairly long string to demonstrate how "
   BaseString = BaseString & "to wrap text to impose your own right "
   BaseString = BaseString & "margin.  "
   For I = 1 To 3
       BaseString = BaseString & BaseString
   Next I
   GetLargeString = BaseString
End Function
```

```
Private Sub Command1_Click()
    Dim dpiX As Long, dpiY As Long
    Dim MarginLeft As Long, MarginRight As Long
    Dim MarginTop As Long, MarginBottom As Long
    Dim PrintAreaHorz As Long, PrintAreaVert As Long
    Dim PhysHeight As Long, PhysWidth As Long
    Dim Info As String

    dpiX = GetDeviceCaps(Printer.hdc, LOGPIXELSX)
    Info = "Pixels X: " & dpiX & " dpi"

    dpiY = GetDeviceCaps(Printer.hdc, LOGPIXELSY)
    Info = Info & vbCrLf & "Pixels Y: " & dpiY & " dpi"

    MarginLeft = GetDeviceCaps(Printer.hdc, PHYSICALOFFSETX)
    Info = Info & vbCrLf & "Unprintable space on left: " & _
    MarginLeft & " pixels = " & MarginLeft / dpiX & " inches"

    MarginTop = GetDeviceCaps(Printer.hdc, PHYSICALOFFSETY)
    Info = Info & vbCrLf & "Unprintable space on top: " & _
    MarginTop & " pixels = " & MarginTop / dpiY & " inches"

    PrintAreaHorz = GetDeviceCaps(Printer.hdc, HORZRES)
    Info = Info & vbCrLf & "Printable space (Horizontal): " & _
    PrintAreaHorz & " pixels = " & PrintAreaHorz / dpiX & " inches"

    PrintAreaVert = GetDeviceCaps(Printer.hdc, VERTRES)
    Info = Info & vbCrLf & "Printable space (Vertical): " & _
    PrintAreaVert & " pixels = " & PrintAreaVert / dpiY & " inches"

    PhysWidth = GetDeviceCaps(Printer.hdc, PHYSICALWIDTH)
    Info = Info & vbCrLf & "Total space (Horizontal): " & _
    PhysWidth & " pixels = " & PhysWidth / dpiX & " inches"

    MarginRight = PhysWidth - PrintAreaHorz - MarginLeft
    Info = Info & vbCrLf & "Unprintable space on right: " & _
    MarginRight & " pixels = " & MarginRight / dpiX & " inches"

    PhysHeight = GetDeviceCaps(Printer.hdc, PHYSICALHEIGHT)
    Info = Info & vbCrLf & "Total space (Vertical): " & _
    PhysHeight & " pixels = " & PhysHeight / dpiY & " inches"

    MarginBottom = PhysHeight - PrintAreaVert - MarginTop
    Info = Info & vbCrLf & "Unprintable space on bottom: " & _
    MarginBottom & " pixels = " & MarginBottom / dpiY & " inches"

    MsgBox Info, , "GetDeviceCaps Returned the Following:"
End Sub

Private Sub Command2_Click()
    Dim FooterLines As Integer, TextLine As String
    Dim LineHeight As Integer, LastLine As Integer
    Dim TotWidth As Long, TotHeight As Long
    Dim MarginTop As Long, MarginBottom As Long
    Dim TotalPrtAreaVert As Long, TotalPrtAreaHorz As Long
    Dim MarginLeft As Long, MarginRight As Long
    Dim LeftMargin As Long, HorzMargins As Long
    Dim Line As Integer, RptHeaderPrint As Boolean
```

```
Dim BreakPoint As Integer, PrintLength As Integer
Dim TempString As String, CharLength As Integer
Dim WrapPoint As Integer, EstWrapPoint As Integer

Printer.Font.Name = "Arial"
Printer.Font.Size = 12

' The next 8 values are in Pixels.
' TotalPrtAreaVert is eqivalent to Printer.ScaleHeight
TotalPrtAreaVert = GetDeviceCaps(Printer.hdc, VERTRES)
TotHeight = GetDeviceCaps(Printer.hdc, PHYSICALHEIGHT)
MarginTop = GetDeviceCaps(Printer.hdc, PHYSICALOFFSETY)
MarginBottom = TotHeight - TotalPrtAreaVert - MarginTop
' TotalPrtAreaHorz is eqivalent to Printer.ScaleWidth
TotalPrtAreaHorz = GetDeviceCaps(Printer.hdc, HORZRES)
TotWidth = GetDeviceCaps(Printer.hdc, PHYSICALWIDTH)
MarginLeft = GetDeviceCaps(Printer.hdc, PHYSICALOFFSETX)
MarginRight = TotWidth - TotalPrtAreaHorz - MarginLeft

HorzMargins = (MarginRight + MarginLeft) * Printer.TwipsPerPixelX
LeftMargin = MarginLeft * Printer.TwipsPerPixelX
' Covert to twips for ease of use
TotalPrtAreaHorz = TotalPrtAreaHorz * Printer.TwipsPerPixelX
LineHeight = Printer.TextHeight("Test")
FooterLines = 2   ' number of lines in footer
LastLine = TotalPrtAreaVert * Printer.TwipsPerPixelY - LineHeight
' Set the break point a line high so we can check it after printing
BreakPoint = LastLine - LineHeight * FooterLines
' reduce the right margin by 1 inch, which is 1440 twips
PrintLength = TotalPrtAreaHorz - 1440

PrintHeader True, HorzMargins, LeftMargin  ' Report Header
RptHeaderPrint = False
' Print the Footer when the current Y position >=
' LastLine - LineHeight * <Number Of Lines In Footer>
For Line = 1 To 120
   If RptHeaderPrint Then
      PrintHeader False, HorzMargins, LeftMargin  ' Page Header
      RptHeaderPrint = False
   End If
   TextLine = "This is line " & Line
   ' To indent or set a new left margin, reset CurrentX before each
   ' new line.
   ' To use a new right margin, wrap lines at appropriate points.
   If Line >= 10 And Line <= 14 Then
      TextLine = TextLine & " indented."
      Printer.CurrentX = 2 * 1440  ' indent 2 additional inches
   ElseIf Line >= 48 And Line <= 52 Then
      TextLine = GetLargeString()  ' Need String long for demo
   End If
   'Use TotalPrtAreaHorz to use the default right margin
   'or use your own value for PrintLength for a custom right margin
   'in this example we bring in the right margin by 1 inch
  If Printer.TextWidth(TextLine) > PrintLength Then  ' Wrap text
    ' This estimating technique is used for simplicity
    ' First, approximate character length
    CharLength = Printer.TextWidth("aeiou.") / 6
    ' Where to start looking for a line break (wrap)
    EstWrapPoint = PrintLength / CharLength
```

```
        WrapPoint = EstWrapPoint
        Line = Line - 1  ' Since current line has already been counted

        Do While Len(TextLine) > EstWrapPoint
           TextLine = Trim(TextLine)   ' remove spaces from either end
           ' break on a space near the estimated break point
           Do While InStr(WrapPoint, TextLine, " ") <> WrapPoint
                WrapPoint = WrapPoint - 1
           Loop
           TempString = Mid(TextLine, 1, WrapPoint)  ' what will fit
           Printer.Print Trim(TempString)  ' print what will fit
           TextLine = Mid(TextLine, WrapPoint + 1) ' get remaining text
           WrapPoint = EstWrapPoint  ' Reset the estimated wrap point
           Line = Line + 1  ' Increment line counter
           If Printer.CurrentY >= BreakPoint Then  ' another line?
              PrintFooter LastLine, LineHeight, LeftMargin
              If Len(TextLine) > 0 Then  ' Print header if text remains
                  PrintHeader False, HorzMargins, LeftMargin
              End If
           End If
        Loop
        If Len(TextLine) > 0 Then  ' leftover text?
             Printer.Print TextLine
             Line = Line + 1  ' Increment line counter
             If Printer.CurrentY >= BreakPoint Then  ' another line?
                PrintFooter LastLine, LineHeight, LeftMargin
                RptHeaderPrint = True  ' Print Header if more lines
             End If
        End If
     Else
          Printer.Print TextLine
          If Printer.CurrentY >= BreakPoint Then
             PrintFooter LastLine, LineHeight, LeftMargin
             RptHeaderPrint = True  ' Print Header if more lines
          End If
     End If
  Next Line
  If Printer.CurrentY > 0 Then
       PrintFooter LastLine, LineHeight, LeftMargin  ' For last page
  End If
    Printer.EndDoc   ' Finish
    MsgBox "Check Printer for results.", , "Done!"
End Sub
```

۳- پروژه را اجرا کنید.

۴- روی Command1 کلیک کنید. یک پنجره پیام ظاهر می شود که حاشیه ها و محدوده قابل چاپ کاغذ را نشان می دهد.

۵- روی Command2 کلیک کنید. چند صفحه چاپ می شود که حاوی موارد زیر است:

سربرگ گزارشی : فقط در صفحه اول چاپ می شود.

سربرگ صفحات : حاوی چند عبارت است که بصورت ایتالیک چاپ می شود، به علاوه تاریخ جاری که در سمت راست چاپ می شود. این سربرگ در تمامی صفحات چاپ می شود.

پـــانویس : حـــاوی چـــند عـــبارت و شماره صفحه است که در چند سطر چاپ می شوند. این پانویس در تمامی صفحات چاپ می شود.

علاوه بر این ها، سطرهای شماره ۱۰ تا ۱۴ بصورت دندانه دار هستند و ۵ سانتیمتر جلوتر چاپ می شوند و سطرهای شماره ۴۸ تا ٥٦ نمایش دهنده حاشیه راست تغییر یافته هستند.

تعیین انواع جایگاههای ذخیره کاغذ[۳] یک چاپگر

نام فایل	تاریخ	نگارش ویژوال بیسیک	نگارش ویندوز
194789.TXT	۲۷ اکتبر ۱۹۹۸	4 , 5 , 6	95 / 98 / NT

اگر عمل انتخاب جایگاه ذخیره کاغذ بوسیله تنظیم خصوصیت PaperBin شئ چاپگر با شکست مواجه شود، ممکنست به خاطر این باشد که درایور چاپگر مقدار جدید را تشخیص نداده است. برای مثال، مقدار ثابت مخصوص جایگاه بالایی کاغذ (vbPRBNUpper) را به این خصوصیت نسبت می دهید اما عمل نمی کند. بعد مشخص می شود که درایور چاپگر جایگاه بالایی خود را جایگاه کشویی فرض می کند. در این مورد مجبورید ثابت مربوط به جایگاه کشویی را برای انتخاب جایگاه بالایی به کار برید.

در این بخش خواهید دید که چگونه لیستی از تمامی جایگاههای کاغذ یک پرینتر بدست آورید.

۱- یک پروژه جدید در ویژوال بیسیک ایجاد کنید. Form1 بصورت پیش فرض ایجاد می شود.

---

[۳] Paper Bin

۲- یک CommandButton و یک TextBox به Form1 اضافه کنید.

۳- خصوصیت ScrollBar متعلق به Text1 را مقادل مقدار ۲ یا Vertical و خصوصیت MultiLine را معادل True تنظیم کنید.

۴- کد زیر را در Form1 وارد کنید:

```
Option Explicit

Private Declare Function OpenPrinter Lib "winspool.drv" Alias _
 "OpenPrinterA" (ByVal pPrinterName As String, phPrinter As Long, _
 ByVal pDefault As Long) As Long
Private Declare Function ClosePrinter Lib "winspool.drv" ( _
 ByVal hPrinter As Long) As Long
Private Declare Function DeviceCapabilities Lib "winspool.drv" _
 Alias "DeviceCapabilitiesA" (ByVal lpDeviceName As String, _
 ByVal lpPort As String, ByVal iIndex As Long, lpOutput As Any, _
 ByVal dev As Long) As Long

Private Const DC_BINS = 6
Private Const DC_BINNAMES = 12

Private Sub Command1_Click()
   Dim prn As Printer
   Dim hPrinter As Long
   Dim dwbins As Long
   Dim ct As Long
   Dim nameslist As String
   Dim nextString As String
   Dim numBin() As Integer

   Text1.Font.Name = "Courier New"
   Text1.Font.Size = 12
   Text1.Text = ""
   For Each prn In Printers
    If OpenPrinter(prn.DeviceName, hPrinter, 0) <> 0 Then
         dwbins = DeviceCapabilities(prn.DeviceName, prn.Port, _
           DC_BINS, ByVal vbNullString, 0)
         ReDim numBin(1 To dwbins)
         nameslist = String(24 * dwbins, 0)
         dwbins = DeviceCapabilities(prn.DeviceName, prn.Port, _
           DC_BINS, numBin(1), 0)
         dwbins = DeviceCapabilities(prn.DeviceName, prn.Port, _
           DC_BINNAMES, ByVal nameslist, 0)
         If Text1.Text <> "" Then
            Text1.Text = Text1.Text & vbCrLf & vbCrLf
         End If
         Text1.Text = Text1.Text & prn.DeviceName
         For ct = 1 To dwbins
            nextString = Mid(nameslist, 24 * (ct - 1) + 1, 24)
            nextString = Left(nextString, InStr(1, nextString, _
              Chr(0)) - 1)
            nextString = String(6 - Len(CStr(numBin(ct))), " ") & _
            numBin(ct) & "  " & nextString
            Text1.Text = Text1.Text & vbCrLf & nextString
         Next ct
```

```
            Call ClosePrinter(hPrinter)
        Else    ' OpenPrinter failed, so cannot retrieve information
           Text1.Text = Text1.Text & vbCrLf & vbCrLf & prn.DeviceName _
              & vbCrLf & "  <Unavailable>"
        End If
      Next prn
End Sub

Private Sub Form_Load()
   ' Size and position the Form and controls
   Me.Height = 7000
   Me.Width = 7000
   Text1.Top = 100
   Text1.Left = 100
   Text1.Height = 6450
   Text1.Width = 5000
   Text1.Text = ""    ' Clear the TextBox
   Command1.Left = 5300
   Command1.Top = 1000
   Command1.Width = 1500
   Command1.Caption = "List Bins"
End Sub
```

۵- پروژه را اجرا کنید و روی دکمه List Bins کلیک کنید. TextBox لیستی از تمامی چاپگرهای نصب شده و تنظیمات جایگاههای کاغذ مورد پشتیبانی هر یک را نمایش می دهد.

☜ توجه : این روش تمامی جایگاههای کاغذ شناخته شده برای درایور چاپگر را نمایش می دهد. هر جایگاهی که در این لیست ظاهر می شود لزوما بطور فیزیکی روی چاپگر نصب نشده است. برای مثال، بعضی چاپگرها دارای یک تغذیه کننده پاکت بصورت اختیاری هستند. این جایگاه کاغذ توسط تابع API گزارش داده می شود، حتی اگر عملا روی چاپگر نصب نشده باشد.