Getting Image File Information with GDI32

While it is possible to open and read any image file as a sequential file to find the image width and height, it may be easier to use the GDI32 `GetObjectA`. `GetObjectA` retrieves information for all bitmaps in memory. Load the image first with GDIPlus with a call to `GdipCreateBitmapFromFile`. Once the image is loaded, call `GetObjectA` to retrieve information. This code is based in part upon Alyce Watson's API Graphics Series. The code also uses Dan Teel's demo Load Any Image with GDIPlus.

Two structs are needed. The first is for the `GetObjectA` code. The struct holds the bitmap information.

```
    Struct ImageInfo, _
        imgType as Long, _
        imgWidth as Long, _
        imgHeight as Long, _
        imgWidthBytes as Long, _
        imgPlanes as Word, _
        imgBitsPixel as Word, _
        imgBits as Long
```

The second struct is for the `GdipCreateBitmapFromFile` code.

```
    Struct doubleWord, a as uLong
```

Once the image is loaded and `GetObjectA` is called, the ImageInfo struct will contain the bitmap. Copy the width and height information from the struct to numeric variables.

```
        imgWidth = ImageInfo.imgWidth.struct
        imgHeight = ImageInfo.imgHeight.struct
```

The advantages to loading an image with `GdipCreateBitmapFromFile` and then calling `GetObjectA` to find the width and height of the image are

- 1. No parsing of the image extension is needed to identify the image format
- 2. No complicated math formulas are needed to extract the width and height of the format
- 3. No error is given when trying to access a file path longer than 128 characters

When the image is no longer needed, delete the image from memory with `DeleteObject`.

```
    Call DeleteObject hImage

Sub DeleteObject hImage
```

```
    CallDLL #gdi32, "DeleteObject", _
        hImage as uLong, _
        result as Long
End Sub
```

Demo to Load an Image with GdipCreateBitmapFromFile and Find Dimensions with GetObjectA

```
' Extension variable for Filedialog
' Valid image formats are bmp, gif, ico, jpg, png, and tiff
    iExt$ = "*.bmp;*.gif;*.ico;*.jpg;*.png;*.tiff"

' Variables to hold the width and height of the image
    imageWidth = 0
    imageHeight = 0

' Struct necessary for loading an image with GDIPlus
    Struct doubleWord, a as uLong

' Struct necessary for returning image file information
    Struct ImageInfo, _
        imgType as Long, _
        imgWidth as Long, _
        imgHeight as Long, _
        imgWidthBytes as Long, _
        imgPlanes as Word, _
        imgBitsPixel as Word, _
        imgBits as Long

' Design a GUI to host the image
    WindowWidth = 800
    WindowHeight = 600
    UpperLeftX = Int((DisplayWidth - WindowWidth) / 2)
    UpperLeftY = Int((DisplayHeight - WindowHeight) / 2)

' Design a graphicbox to display the image
    Stylebits #main.g, 0, _WS_BORDER, 0, 0
    Graphicbox #main.g, 0, 0, 792, 580

    Open
"Getting the Image
 Width and Height with GetObjectA" for Window_nf as #main
    #main "Trapclose [closeApp]"
    #main "Font Verdana 10 Bold"
```

```
    hMain = hWnd(#main)
    hG = hWnd(#main.g)
    #main.g "Down"
' Load an image file
    Filedialog "Image to Load", iExt$, imagePathFile$
' Load the image using GDIPlus, hImage is the returned handle to the b
itmap in memory
    hImage = LoadImgFromFile(imagePathFile$)
' Get the bitmap information using GetObjectA
    imageDimen$ = ImageDimensions$(hImage)
    imageW = Val(Word$(imageDimen$, 1, Chr$(0)))
    imageH = Val(Word$(imageDimen$, 2, Chr$(0)))
' Load the bitmap in memory
    Loadbmp "pic", hImage
' Draw the image using native LB command Drawbmp
    #main.g "Drawbmp pic 10 10; Flush"
' Resize the graphicbox to the size of the image
    #main.g "Locate 0 0 ";imageW + 20;" ";imageH + 20
    #main "Refresh"
' Give the width and height of the image
    msg$ = "Image Dimensions";Chr$(10);Chr$(13); _
        "Width: ";imageW;" pixels";Chr$(10);Chr$(13); _
        "Height: ";imageH;" pixels";Chr$(10);Chr$(13)
    Notice msg$
Wait

[closeApp]
' Unload the native LB bitmap
    Unloadbmp "pic"
' Release memory from GDIPlus loaded image
    Call DeleteObject hImage
    Close #main
End

Function ImageDimensions$(hImage)
    lStruct = Len(ImageInfo.struct)
    CallDLL #gdi32, "GetObjectA", _
        hImage as uLong, _
        lStruct as Long, _
        ImageInfo as struct, _
        result as Long
        imgWidth = ImageInfo.imgWidth.struct
        imgHeight = ImageInfo.imgHeight.struct
        ImageDimensions$ = Str$(imgWidth);Chr$(0);Str$(imgHeight)
End Function
```

```
Function wChar$(string$)
    For i = 1 to Len(string$)
        wChar$ = wChar$ + Mid$(string$, i, 1) + chr$(0)
    Next i
    wChar$ = wChar$ + Chr$(0) + Chr$(0)
End Function

Function LoadImgFromFile(picFile$)
    Open "gdiplus.dll" for dll as #gdip
    gdistart$=Chr$(1)
    For i = 1 to 15
        gdistart$ = gdistart$ + Chr$(0)
    Next i
    Calldll #gdip,"GdiplusStartup", _
        doubleWord as struct, _
        gdistart$ as Ptr, _
        status as uLong
    token=doubleWord.a.struct
    If status <> 0 Then
        LoadImgFromFile = 0
    Else
        wFileLoc$=wChar$(picFile$)
        Calldll #gdip,"GdipCreateBitmapFromFile", _
            wFileLoc$ as Ptr, _
            doubleWord as struct, _
            status as uLong
        hPic = doubleWord.a.struct
        If status <>0 then
            LoadImgFromFile=0
        Else
            CallDLL #gdip,"GdipCreateHBITMAPFromBitmap", _
            hPic as uLong, _
            doubleWord as struct, _
            0 as uLong, _
            status as uLong
            hImage=doubleWord.a.struct
            If status <> 0 Then
                LoadImgFromFile = 0
            Else
                LoadImgFromFile = hImage
            End if
            CallDLL #gdip,"GdipDisposeImage", _
                hPic as uLong, _
                result as uLong
        End If
        CallDLL #gdip,"GdiplusShutdown", _
```

```
            token as uLong, _
            result as uLong
    End If
    Close #gdip
End Function

Sub DeleteObject hImage
    CallDLL #gdi32, "DeleteObject", _
        hImage as uLong, _
        result as Long
End Sub
```