

sebastianfernandes.com

[Capturing the screen (Win32/VB)]

Because of the functions provided with the Windows API, capturing the screen to the Clipboard becomes a relatively simple task. This tutorial assumes that you have some knowledge of Visual Basic and an understanding of the Windows environment.

In the sample code that complements this tutorial, the following calls have been added directly to the Sub Main procedure. However, your application does much more, and you will have identified a more suitable place to add these.

1. Before we start making API calls in Sub Main, we must add the following function declarations and constant definitions:

User32

```
GetDesktopWindow  
GetDC  
GetActiveWindow  
OpenClipboard  
EmptyClipboard  
SetClipboardData  
CloseClipboard  
ReleaseDC
```

GDI 32

```
GetDeviceCaps  
CreateCompatibleDC  
CreateCompatibleBitmap  
SelectObject  
BitBlt  
DeleteDC
```

```
Private Const CF_BITMAP = 2  
Private Const HORZRES = 8  
Private Const VERTRES = 10
```

2. Onto our Sub Main procedure – Get the Desktop handle by calling GetDesktopWindow. Now, we can get the device context of the Desktop by calling the GetDC API.

```
HWND_DESKTOP = GetDesktopWindow  
scr_hDC = GetDC(HWND_DESKTOP)
```

3. We can now determine the screen dimensions.

```
scrWidth = GetDeviceCaps(scr_hDC, HORZRES)
scrHeight = GetDeviceCaps(scr_hDC, VERTRES)
```

4. Next, we create a compatible memory context and bitmap.

```
hdcMem = CreateCompatibleDC(scr_hDC)
hBitmap = CreateCompatibleBitmap(scr_hDC, scrWidth, scrHeight)
```

If this was successful, hBitmap should not be NULL, i.e. "if hBitmap" should evaluate to true.

5. Assuming step 4 was successful, we can select the bitmap into the memory context we just created.

```
SelectObject(hdcMem, hBitmap)
```

6. Next, we copy the Desktop to the memory device context.

```
BitBlt(hdcMem, 0, 0, scrWidth, scrHeight, scr_hDC, 0, 0, vbSrcCopy)
```

7. The clipboard – open, clear, copy, and close.

```
OpenClipboard GetActiveWindow
EmptyClipboard
SetClipboardData CF_BITMAP, hBitmap
CloseClipboard
```

OK, not so fast. We opened the clipboard providing the handle to our application window. When we emptied the Clipboard, we both cleared it's current contents and temporarily took ownership of the Clipboard.

We placed the bitmap data onto the Clipboard by specifying the data type with CF_BITMAP and providing a handle to our data. Then, we released ownership of the Clipboard.

8. Finally, we cleanup by deleting the memory context and releasing the device context.

```
DeleteDC hdcMem  
ReleaseDC HWND_DESKTOP, scr_hDC
```

The completed procedure is as follows:

```
' sfcapscr.bas : Capture screen to clipboard  
'  
' This application simply copies the screen to the system  
' clipboard - nothing more than you would get by pressing  
' Print Screen on your keyboard! It's only value is as a  
' demonstration of how you can do the same thing  
' programmatically....  
'  
' Copyright (c) 2002, sebastianfernandes.com  
  
Private Declare Function GetDesktopWindow _  
    Lib "User32" () As Long  
  
Private Declare Function GetDC _  
    Lib "User32" (ByVal hwnd As Long) As Long  
  
Private Declare Function GetDeviceCaps _  
    Lib "GDI32" (ByVal hdc As Long, _  
        ByVal iCapabilitiy As Long) As Long  
  
Private Declare Function GetActiveWindow _  
    Lib "User32" () As Integer  
  
Private Declare Function CreateCompatibleDC _  
    Lib "GDI32" (ByVal hdc As Long) As Long
```

```
Private Declare Function CreateCompatibleBitmap _
    Lib "GDI32" (ByVal hDC As Long, _
    ByVal nWidth As Long, ByVal nHeight As Long) As Long

Private Declare Function SelectObject _
    Lib "GDI32" (ByVal hDC As Long, _
    ByVal hObject As Long) As Long

Private Declare Function BitBlt _
    Lib "GDI32" (ByVal hDC As Long, _
    ByVal X As Long, ByVal Y As Long, _
    ByVal nWidth As Long, ByVal nHeight As Long, _
    ByVal hSrcDC As Long, ByVal xSrc As Long, _
    ByVal ySrc As Long, ByVal dwRop As Long) As Long

Private Declare Function OpenClipboard _
    Lib "User32" (ByVal hwnd As Long) As Long

Private Declare Function EmptyClipboard _
    Lib "User32" () As Long

Private Declare Function SetClipboardData _
    Lib "User32" (ByVal wFormat As Long, _
    ByVal hMem As Long) As Long

Private Declare Function CloseClipboard _
    Lib "User32" () As Long

Private Declare Function DeleteDC _
    Lib "GDI32" (ByVal hDC As Long) As Long

Private Declare Function ReleaseDC _
    Lib "User32" (ByVal hwnd As Long, _
    ByVal hDC As Long) As Long

Private Const CF_BITMAP = 2
Private Const HORZRES = 8
Private Const VERTRES = 10
```

```
Private Sub Main()  
  
    HWND_DESKTOP = GetDesktopWindow  
    scr_hDC = GetDC(HWND_DESKTOP)  
  
    scrWidth = GetDeviceCaps(scr_hDC, HORZRES)  
    scrHeight = GetDeviceCaps(scr_hDC, VERTRES)  
  
    hdcMem = CreateCompatibleDC(scr_hDC)  
    hBitmap = CreateCompatibleBitmap(scr_hDC, scrWidth, scrHeight)  
  
    If hBitmap Then  
        SelectObject hdcMem, hBitmap  
        BitBlt hdcMem, 0, 0, scrWidth, scrHeight, _  
            scr_hDC, 0, 0, vbSrcCopy  
        OpenClipboard GetActiveWindow  
        EmptyClipboard  
        SetClipboardData CF_BITMAP, hBitmap  
        CloseClipboard  
    End If  
  
    DeleteDC hdcMem  
    ReleaseDC HWND_DESKTOP, scr_hDC  
  
End  
  
End Sub
```