

Re: Icon transparency after Serialization

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.drawing/2005-10/msg00167.html>

- *From:* "Michael Phillips, Jr." <mphillips53@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 17 Oct 2005 09:36:06 -0400
-

There are a few problems with your approach.

> myIcon = System.Drawing.Icon.FromHandle(shinfo.hIcon)

The HICON returned from the SHGetFileInfo method should be disposed with DestroyIcon after you have finished with the handle.

The System.Drawing.Icon class is a wrapper around the gdiplus API. Gdiplus does not have an encoder for the Icon format.

I would not serialize a reference to a gdiplus icon. I would take the handle returned from SHGetFileInfo and use GetIconInfo to obtain the handles to the AND and XOR bitmap masks.

>From these handles you can use the Bitmap class to extract just the bits. You must use the Marshal class to read the bytes.
>From the bits you can construct a packed DIB which consists of a BITMAPINFOHEADER followed by the RGBQUAD color table followed by the XOR and AND bits.

You can then use the MemoryStream class to serialize this chunk of memory.

When you need to create the Icon (i.e., HICON) use the function CreateIconFromResourceEx function which requires a packed DIB to create the HICON.

"Mike_IntermediateVB" <MikeIntermediateVB@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:04A97959-037C-423D-A443-AAD18BA0B876@xxxxxxxxxxxxxxxxxxxx

> "Michael Phillips, Jr." wrote:
>> How are you serializing the HICON that is returned?
>
> Thanks for reply I will spend a bit of time trying to understand the API,
> below is a sample of the way I am capturing the Icons.

Re: Icon transparency after Serialization

```
> I create an Obj and add the ResourceIconCollection to it. Then add icons
> by
> Object.ResourceIconCollection.AddIcon(IconPath). Then serialise the whole
> obj
> using....
>
> Private Function SerilaiseObject() As Byte()
> Dim serializer As New BinaryFormatter
> Dim objData As New MemoryStream
> serializer.Serialize(objData, obj)
> objData.Seek(0, 0)
> Return objData.ToArray()
> End Function
>
> then I compress the byte array and save it to the database.
>
> Code.....
>
> '+++++
> '--List of ICONs
> '+++++
> <Serializable()> Public Class ResourceIconCollection
> Public IconList As New ArrayList
> Public Function AddIcon(ByVal strPath)
> If IconList.Count = 0 Then
> Dim newIcon As New SingleIcon(strPath)
> IconList.Add(newIcon)
> Else
> '--Check if Icon exists
> Dim Currenticon As SingleIcon
> Dim FileExt As String =
> LCase(System.IO.Path.GetExtension(strPath))
> For Each Currenticon In IconList
> If FileExt = Currenticon.Key Then
> '--Icon extension has been Found
> Exit Function
> End If
> Next
> '--No Existing Icons for this Extension found, so Add a new
> one
> Dim newIcon As New SingleIcon(strPath)
> IconList.Add(newIcon)
> End If
>
> End Function
> End Class
>
> '+++++
> '--Single ICON
> '+++++
>
```

Re: Icon transparency after Serialization

```
> <Serializable()> Public Class SingleIcon
>
> Public icon As System.Drawing.Icon
> Public Key As String
> Public Sub New()
>
> End Sub
> Public Sub New(ByVal strPath As String)
> icon = Icons.GetIcon(strPath)
> Key = LCase(System.IO.Path.GetExtension(strPath))
> End Sub
>
> End Class
>
>
> '+++++
> '---Get ICON
> '+++++
> Imports System.Runtime.InteropServices
> Public Class Icons
> Private Structure SHFILEINFO
> Public hIcon As IntPtr ' : icon
> Public iIcon As Integer ' : icondex
> Public dwAttributes As Integer ' : SFGAO_ flags
> <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=260)> _
> Public szDisplayName As String
> <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=80)> _
> Public szTypeName As String
> End Structure
>
> Private Declare Auto Function SHGetFileInfo Lib "shell32.dll" _
> (ByVal pszPath As String, _
> ByVal dwFileAttributes As Integer, _
> ByRef psfi As SHFILEINFO, _
> ByVal cbFileInfo As Integer, _
> ByVal uFlags As Integer) As IntPtr
>
> Private Const SHGFI_ICON = &H100
> Private Const SHGFI_SMALLICON = &H1
> Private Const SHGFI_LARGEICON = &H0 ' Large icon
> Private nIndex = 0
>
> Public Shared Function GetIcon(ByVal strPathName As String) As
> System.Drawing.Icon
> '---Get Icon
> Dim hImgSmall As IntPtr
>
> Dim shinfo As SHFILEINFO
> shinfo = New SHFILEINFO
>
> shinfo.szDisplayName = New String(Chr(0), 260)
```

Re: Icon transparency after Serialization

```
> shinfo.szTypeName = New String(Chr(0), 80)
>
> 'Use this to get the small icon.
> hImgSmall = SHGetFileInfo(strPathName, 0, shinfo, _
> Marshal.SizeOf(shinfo), _
> SHGFI_ICON Or SHGFI_SMALLICON)
>
> Dim myIcon As System.Drawing.Icon
> myIcon = System.Drawing.Icon.FromHandle(shinfo.hIcon)
>
> Return myIcon
> End Function
>
> End Class
```

• *Follow-Ups:*

- ◆ ***Re: Icon transparency after Serialization***
◇ *From: Mike_IntermediateVB*

• *References:*

- ◆ ***Re: Icon transparency after Serialization***
◇ *From: Michael Phillips, Jr.*
 - ◆ ***Re: Icon transparency after Serialization***
◇ *From: Mike_IntermediateVB*
 - ◆ ***Re: Icon transparency after Serialization***
◇ *From: Michael Phillips, Jr.*
 - ◆ ***Re: Icon transparency after Serialization***
◇ *From: Mike_IntermediateVB*
- Prev by Date: ***Re: StretchDIBits in managed***
 - Next by Date: ***tiff and header information***
 - Previous by thread: ***Re: Icon transparency after Serialization***
 - Next by thread: ***Re: Icon transparency after Serialization***
 - Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***