

Forum:[Hack Forums](#)

Thread: [Visual Basics: Icon Changer Module](#)

Thresh - April 5, 2018, 9:26 am

This can be used for a crypter for example. ?

Source: [bit . ly/2EpMGgq](http://bit.ly/2EpMGgq)

Code Option Strict On

Option Explicit On

Option Infer On

Imports System.Runtime.InteropServices

Imports System.Security

Public Class IconChanger

<SuppressUnmanagedCodeSecurity()> _

Private Class NativeMethods

<DllImport("kernel32")> _

Public Shared Function BeginUpdateResource(_

ByVal fileName As String, _

<MarshalAs(UnmanagedType.Bool)> ByVal deleteExistingResources As Boolean) As IntPtr

End Function

<DllImport("kernel32")> _

Public Shared Function UpdateResource(_

ByVal hUpdate As IntPtr, _

ByVal type As IntPtr, _

ByVal name As IntPtr, _

ByVal language As Short, _

<MarshalAs(UnmanagedType.LPArray, SizeParamIndex:=5)> _

ByVal data() As Byte, _

ByVal dataSize As Integer) As <MarshalAs(UnmanagedType.Bool)> Boolean

End Function

<DllImport("kernel32")> _

Public Shared Function EndUpdateResource(_

ByVal hUpdate As IntPtr, _

<MarshalAs(UnmanagedType.Bool)> ByVal discard As Boolean) As

<MarshalAs(UnmanagedType.Bool)> Boolean

End Function

End Class

<StructLayout(LayoutKind.Sequential)> _

Private Structure ICONDIR

Public Reserved As UShort

Public Type As UShort

Public Count As UShort

End Structure

<StructLayout(LayoutKind.Sequential)> _

Private Structure ICONDIRENTRY

Public Width As Byte

Public Height As Byte

Public ColorCount As Byte

```

    Public Reserved As Byte
    Public Planes As UShort
    Public BitCount As UShort
    Public BytesInRes As Integer
    Public ImageOffset As Integer
End Structure
<StructLayout(LayoutKind.Sequential)> _
Private Structure BITMAPINFOHEADER
    Public Size As UInteger
    Public Width As Integer
    Public Height As Integer
    Public Planes As UShort
    Public BitCount As UShort
    Public Compression As UInteger
    Public SizeImage As UInteger
    Public XPelsPerMeter As Integer
    Public YPelsPerMeter As Integer
    Public ClrUsed As UInteger
    Public ClrImportant As UInteger
End Structure
<StructLayout(LayoutKind.Sequential, Pack:=2)> _
Private Structure GRPICONDIRENTRY
    Public Width As Byte
    Public Height As Byte
    Public ColorCount As Byte
    Public Reserved As Byte
    Public Planes As UShort
    Public BitCount As UShort
    Public BytesInRes As Integer
    Public ID As UShort
End Structure
Public Shared Sub InjectIcon(ByVal exeFileName As String, ByVal iconFileName As String)
    InjectIcon(exeFileName, iconFileName, 1, 1)
End Sub
Public Shared Sub InjectIcon(ByVal exeFileName As String, ByVal iconFileName As String,
ByVal iconGroupID As UInteger, ByVal iconBaseID As UInteger)
    Const RT_ICON As UInteger = 3UI
    Const RT_GROUP_ICON As UInteger = 14UI
    Dim iconFile As IconFile = iconFile.FromFile(iconFileName)
    Dim hUpdate = NativeMethods.BeginUpdateResource(exeFileName, False)
    Dim data = iconFile.CreateIconGroupData(iconBaseID)
    NativeMethods.UpdateResource(hUpdate, New IntPtr(RT_GROUP_ICON), New
IntPtr(iconGroupID), 0, data, data.Length)
    For i = 0 To iconFile.ImageCount - 1
        Dim image = iconFile.ImageData(i)
        NativeMethods.UpdateResource(hUpdate, New IntPtr(RT_ICON), New IntPtr(iconBaseID +
i), 0, image, image.Length)
    Next
    NativeMethods.EndUpdateResource(hUpdate, False)
End Sub
Private Class IconFile
    Private iconDir As New ICONDIR
    Private iconEntry() As ICONDIRENTRY

```

```

Private iconImage()() As Byte
Public ReadOnly Property ImageCount() As Integer
    Get
        Return iconDir.Count
    End Get
End Property
Public ReadOnly Property ImageData(ByVal index As Integer) As Byte()
    Get
        Return iconImage(index)
    End Get
End Property
Private Sub New()
End Sub
Public Shared Function FromFile(ByVal filename As String) As IconFile
    Dim instance As New IconFile
    Dim fileBytes() As Byte = IO.File.ReadAllBytes(filename)
    Dim pinnedBytes = GCHandle.Alloc(fileBytes, GCHandleType.Pinned)
    instance.iconDir = DirectCast(Marshal.PtrToStructure(pinnedBytes.AddrOfPinnedObject,
GetType(ICONDIR)), ICONDIR)
    instance.iconEntry = New ICONDIRENTRY(instance.iconDir.Count - 1) {}
    instance.iconImage = New Byte(instance.iconDir.Count - 1) {}
    Dim offset = Marshal.SizeOf(instance.iconDir)
    Dim iconDirEntryType = GetType(ICONDIRENTRY)
    Dim size = Marshal.SizeOf(iconDirEntryType)
    For i = 0 To instance.iconDir.Count - 1
        Dim entry = DirectCast(Marshal.PtrToStructure(New
IntPtr(pinnedBytes.AddrOfPinnedObject.ToInt64 + offset), iconDirEntryType), ICONDIRENTRY)
        instance.iconEntry(i) = entry
        instance.iconImage(i) = New Byte(entry.BytesInRes - 1) {}
        Buffer.BlockCopy(fileBytes, entry.ImageOffset, instance.iconImage(i), 0,
entry.BytesInRes)
        offset += size
    Next
    pinnedBytes.Free()
    Return instance
End Function
Public Function CreateIconGroupData(ByVal iconBaseID As UInteger) As Byte()
    Dim sizeOfIconGroupData As Integer = Marshal.SizeOf(GetType(ICONDIR)) +
Marshal.SizeOf(GetType(GRPICONDIRENTRY)) * ImageCount
    Dim data(sizeOfIconGroupData - 1) As Byte
    Dim pinnedData = GCHandle.Alloc(data, GCHandleType.Pinned)
    Marshal.StructureToPtr(iconDir, pinnedData.AddrOfPinnedObject, False)
    Dim offset = Marshal.SizeOf(iconDir)
    For i = 0 To ImageCount - 1
        Dim grpEntry As New GRPICONDIRENTRY
        Dim bitmapheader As New BITMAPINFOHEADER
        Dim pinnedBitmapInfoHeader = GCHandle.Alloc(bitmapheader, GCHandleType.Pinned)
        Marshal.Copy(ImageData(i), 0, pinnedBitmapInfoHeader.AddrOfPinnedObject,
Marshal.SizeOf(GetType(BITMAPINFOHEADER)))
        pinnedBitmapInfoHeader.Free()
        grpEntry.Width = iconEntry(i).Width
        grpEntry.Height = iconEntry(i).Height
        grpEntry.ColorCount = iconEntry(i).ColorCount
    Next
    Return data
End Function

```

```

        grpEntry.Reserved = iconEntry(i).Reserved
        grpEntry.Planes = bitmapheader.Planes
        grpEntry.BitCount = bitmapheader.BitCount
        grpEntry.BytesInRes = iconEntry(i).BytesInRes
        grpEntry.ID = CType(iconBaseID + i, UShort)
        Marshal.StructureToPtr(grpEntry, New IntPtr(pinnedData.AddrOfPinnedObject.ToInt64 +
offset), False)
        offset += Marshal.SizeOf(GetType(GRPICONDIRENTRY))
    Next
    pinnedData.Free()
    Return data
End Function
End Class
End Class

```

zeroexp - April 5, 2018, 1:56 pm

<https://github.com/malwares/Crypter/blob/master/%5BVB.Net%5D%20%2B%20%5BC%23%5D%20Crystal%20Crypter%20v2/IconChanger.vb>

Link source.

Maveraun.Nehr - September 23, 2018, 9:40 am

how to use ?? i try this code but i can't

```
InjectIcon(Textbox1.Text,Textbox2.Text)
```

Vyxn - September 24, 2018, 3:16 am

(Sep 23, 2018 09:40 AM)Maveraun.Nehr Wrote: [\[CITE\]](#) how to use ?? i try this code but i can't

```
InjectIcon(Textbox1.Text,Textbox2.Text)
```

So you are gonna call this method.

One parameter says, exeFileName the other one is iconFileName. Give the file path of your target assembly on the exeFileName parameter. Then get the file path of the icon your want to use in iconFileName.

```
Code Public Shared Sub InjectIcon(ByVal exeFileName As String, ByVal iconFileName As String)
```

```
    InjectIcon(exeFileName, iconFileName, 1, 1)
```

```
End Sub
```