

استفاده از محضرخانه ویندوز

طریقه استفاده از محضرخانه ویندوز ([Registry](#)) در VB:

مقدمه:

اغلب برنامه های ویندوز برای ذخیره کردن طول و عرض و آخرین موقعیت فرم بر روی صفحه - کلمه رمز- آخرین فایل های گشوده شده و یا ایجاد ارتباط بین شکل ها و برنامه ای خاص و ... از محضرخانه ویندوز استفاده می کنند.

VB اطلاعات ذخیره شده توسط برنامه های خود را در محل استاندارد زیر ذخیره می کند:

***HKEY_CURRENT_USER\Software\VB and VBA Program
Settings\appname\section |key***

برای دیدن این قسمت بر روی دکمه **Start** ویندوز تکه بزنید و **Run** را انتخاب کنید. سپس کلمه **Regedit** را بنویسید و روی **Ok** تکه بزنید. به این ترتیب شما با استفاده از برنامه **Registry Editor** ویندوز می توانید قسمت فوق را ببینید. ابتدا **HKEY_CURRENT_USER** سپس **Software** و در نهایت **VB and VBA Program Settings** را در ویرایشگر محضرخانه ویندوز پیدا کنید.

(شکل: Icon)

توابع ویژوال بیسیک برای کار کردن با محضرخانه ویندوز:

VB چهار تابع برای خواندن و نوشتن در Registry دارد. این توابع عبارتند از:

- 1- `SaveSetting appname, section, key, value`
- 2- `GetSetting(appname, section, key[, default])`

- 3- `GetAllSettings`(appname, section)
- 4- `DeleteSetting`(appname, section, key)

SaveSetting-1

بوسیله این تابع می توان مقداری جدید را در قسمت **Key** برنامه نوشت.

appname|section|key

برای مثال برای ذخیره کردن تنظیمات برنامه خود در هنگام خروج از برنامه در روال رخداد **Form_Unload** از این تابع می توانید استفاده کنید.

مثال :

```
Sub SaveMainFormPosition_InRegistry(frmName As Form)
'put this sub in form_unload
SaveSetting App.EXENAME, "Startup", "Top", _
Trim(Str(frmName.Top))

SaveSetting App.EXENAME, "Startup", "Left", _
Trim(Str(frmName.Left))

SaveSetting App.EXENAME, "Startup", "FormWidth", _
Trim(Str(frmName.Width))

SaveSetting App.EXENAME, "Startup", "FormHeight", _
Trim(Str(frmName.Height))

End Sub
```

برای دیدن تنظیمات ذخیره شده به برنامه ویرایشگر محضرخانه ویندوز مراجعه کنید. (**regEdit**) پس از هر بار بستن برنامه برای دیدن تغییرات اعمال شده به محضرخانه باید دکمه **F5** را بفشارید. (**Refresh**)

روال فوق پهنا-ارتفاع و آخرین محل قرار گیری فرم را روی صفحه کار در محضر خانه ویندوز در محل زیر ذخیره می کند:

**HKEY_CURRENT_USER\Software\VB and VBA Program
Settings\Project1\Startup**

فرض شده است که نام فایل اجرایی برنامه **Project1** است.

از روال فوق به صورت زیر می توان استفاده کرد:

```
Private Sub Form_Unload(Cancel As Integer)
SaveMainFormPosition_InRegistry Me
End Sub
```

GetSetting-2

بوسیله این تابع یک مقدار ذخیره شده در محضرخانه بازگشت داده می شود. اگر مقداری وجود نداشته باشد مقدار آرگومان **Default** نتیجه اجرای تابع خواهد بود. مقدار پیش فرض برای این آرگومان پیش فرض رشته ای با طول صفر میباشد. ("")

مثال:

```
MsgBox "Project1->Startup->Left = " + GetSetting(App.EXENAME, _
    "Startup", "Left", "0"), vbInformation, ""
```

روال زیر مقدارهاي ذخیره شده توسط روال `SaveMainFormPosition_InRegistry` دریافت می کند و موقعیت جدید فرم را برابر آخرین مقدارهاي ذخیره شده قرار می دهد.

```
Sub LoadMainWindowPosition_FromRegistry(frmName As Form)
'put this sub in form_load
    frmName.Top = GetSetting(App.EXENAME, "Startup", "Top", 0)
    frmName.Left = GetSetting(App.EXENAME, "Startup", "Left", 0)
    frmName.Width = GetSetting(App.EXENAME, "Startup", "FormWidth", 8000)
    frmName.Height = GetSetting(App.EXENAME, "Startup", "FormHeight", 6000)
End Sub
```

بهترین مکان برای استفاده روال فوق هنگام بارگذاری فرم می باشد:

```
Private Sub Form_Load()
    LoadMainWindowPosition_FromRegistry Me
End Sub
```

GetAllSettings-3

برای بدست آوردن لیستی از Keyها و مقادیر آنها در محضرخانه از این تابع استفاده می شود. خروجی آن یک آرایه دوبعدی Variant میباشد.
مثال:

```
Dim avntSettings As Variant
Dim intX As Integer
avntSettings = GetAllSettings(App.EXENAME, "Startup")
For intX = 0 To UBound(avntSettings, 1)
    MsgBox avntSettings(intX, 0) + " = " + avntSettings(intX, 1)
Next intX
```

DeleteSetting-4

از این تابع در سه حالت برای حذف مقادیر ذخیره شده در محضرخانه استفاده می شود.

۱- `DeleteSetting appname`

در این حالت کل تنظیمات ذخیره شده پاک می شوند.

۲- `DeleteSetting appname, section`

در اینجا فقط کل Section حذف میشود.

۳- `DeleteSetting appname, section, key`

با این دستور کلید ذکر شده در قسمت Key در بخش Section حذف می شود.

چند مثال :

۱- نوشتن و خواندن کلمه رمز در محضرخانه:

```
Public Sub putPassWord_InRegistry(passW As String)
    SaveSetting App.EXENAME, "PW", "PW", passW
End Sub
```

.....

```

Public Sub readPassWord_FromRegistry()
    Dim passW As String

    Do

        passW = InputBox("Please Enter Password:", "Password", "?")
        If passW = "" Then End
        If passW = GetSetting(App.EXENAME, "PW", "PW", "") Then
            Exit Sub
        Else
            MsgBox "Wrong PassWord !" + vbCrLf + "You Are Using The Trial
Version Of This Program." _
                + vbCrLf + "Get Its Latest Version at -> 021-7878195 ", _
                vbExclamation, "Sorry !"
            End If

        Loop

    End Sub

```

طرز استفاده:

```

Private Sub Form_Load()
    putPassWord_InRegistry "Vahid"
    readPassWord_FromRegistry
End Sub

```

۲- مدت دار کردن برنامه ها (محدود کردن روزهاي استفاده از برنامه):

```

Public Sub putElapsedException_InRegistry(MaxAllowableDays As Long)

    Dim i As Double
    Dim j As String

    i = Val(GetSetting(App.EXENAME, "PW", "Day", "0"))

    If i >= MaxAllowableDays Then
        MsgBox "You Are Using The Trial Version Of This Program." _
            + vbCrLf + "Get Its Latest Version at -> 021-7878195 ", _
            vbExclamation, "Sorry !"
        End
    End If

    j = GetSetting(App.EXENAME, "PW", "Date", "")

    If j <> Date Then _
        SaveSetting App.EXENAME, "PW", "Day", i + 1

    SaveSetting App.EXENAME, "PW", "Date", Date

End Sub

```

طرز استفاده:

برنامه زیر چهارده روزه است!

```

Private Sub Form_Load()
    putElapsedException_InRegistry 14

```

End Sub

۳- ذخیره کردن آخرین فایل های مورد استفاده برنامه در محضرخانه و خواندن و نمایش دادن آنها:

```
Public Sub saveLastFileName_InRegistry(fileName As String, MaxCount _
As Long)

Dim i As Long
Dim fileNames(100) As Variant

For i = 1 To MaxCount
fileNames(i) = GetSetting(App.EXENAME, "LastFileNames", _
"lastFile " & i, "")
If fileName = fileNames(i) Then Exit Sub
' MsgBox fileNames(i)
Next i

For i = 1 To MaxCount - 1
If fileNames(i) = "" Then Exit For
SaveSetting App.EXENAME, "LastFileNames", "lastFile" + Str(i + 1), _
fileNames(i)
Next i

SaveSetting App.EXENAME, "LastFileNames", "lastFile 1", _
Trim(fileName)

End Sub
' .....
Public Sub loadLastFileName_FromRegistry(frmName As Form, MaxCount _
As Long)

' to use this sub place a menu on a form
' then make NNN sub menu and set its index to 0
' in this case menu must have at least one sub menu

Dim i As Long
Dim fileNames(100) As Variant

frmName.nnn(0).Visible = False

For i = 1 To MaxCount
fileNames(i) = GetSetting(App.EXENAME, "LastFileNames", _
"lastFile " & i, "")
If fileNames(i) = "" Then Exit For
Load frmName.nnn(i)
frmName.nnn(i).Caption = fileNames(i)
frmName.nnn(i).Visible = True
' MsgBox fileNames(i)
Next i

End Sub
```

استفاده از Win-API برای اعمال تغییرات در سایر قسمتهای Registry:

برای دسترسی به سایر قسمت های محضرخانه ویندوز و ویژوال بیسیک تابع دیگری ندارد. اما خود ویندوز توابع دیگری را برای این منظور دارا می باشد:

RegOpenKeyEx - ۱

دو تابع در API ویندوز برای باز کردن يك کلید از محضرخانه ویندوز وجود دارد: RegOpenKeyEx and RegOpenKey. بطور کلی قاعده ای را که باید در اینگونه موارد به خاطر داشت این است که اگر در API دو تابع با يك نام وجود داشت تابعی که به EX ختم می شود ۳۲ بیتی بوده و تابع دیگر فقط برای حفظ سازگاری با ویندوز ۳,۱ ارائه شده و استفاده از آن در ویندوز های ۳۲ بیتی توصیه نمی شود.

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

توضیح آرگومانها:

hKey یکی از چهار ثابت زیر می باشد و اگر برنامه RegEdit را اجرا کنید در ابتدا آنها را خواهید دید:

```
'Predefined Registry Keys used in hKey Argument
Public Enum RegistryKeys_in_hKey
HKEY_CLASSES_ROOT = &H80000000
HKEY_CURRENT_USER = &H80000001
HKEY_LOCAL_MACHINE = &H80000002
HKEY_USERS = &H80000003
HKEY_PERFORMANCE_DATA = &H80000004
End Enum
```

lpSubKey کلید برای گشودن می باشد برای مثال توابع استاندارد و ویژوال بیسیک برای کار با محضرخانه از کلید زیر استفاده می کنند:

```
HKEY_CURRENT_USER\Software\VB and VBA Program Settings\appname\section\
```

ulOptions معمولا بکار نمی رود و بجای آن 0 قرار می دهند. samDesired پوشش امنیتی برای دستیابی به محضرخانه است و یکی از ثوابت زیر می باشد:

```
'Security Mask constants
Public Const READ_CONTROL = &H20000
Public Const SYNCHRONIZE = &H100000
Public Const STANDARD_RIGHTS_ALL = &H1F0000
Public Const STANDARD_RIGHTS_READ = READ_CONTROL
Public Const STANDARD_RIGHTS_WRITE = READ_CONTROL
Public Const KEY_QUERY_VALUE = &H1
Public Const KEY_SET_VALUE = &H2
Public Const KEY_CREATE_SUB_KEY = &H4
Public Const KEY_ENUMERATE_SUB_KEYS = &H8
Public Const KEY_NOTIFY = &H10
Public Const KEY_CREATE_LINK = &H20
Public Const KEY_ALL_ACCESS = ((STANDARD_RIGHTS_ALL Or _
KEY_QUERY_VALUE Or KEY_SET_VALUE Or KEY_CREATE_SUB_KEY Or _
KEY_ENUMERATE_SUB_KEYS Or KEY_NOTIFY Or KEY_CREATE_LINK) And _
(Not SYNCHRONIZE))
Public Const KEY_READ = ((STANDARD_RIGHTS_READ Or KEY_QUERY_
VALUE Or KEY_ENUMERATE_SUB_KEYS Or KEY_NOTIFY) And _
```

```
(Not SYNCHRONIZE))
Public Const KEY_EXECUTE = ((KEY_READ) And (Not SYNCHRONIZE))
Public Const KEY_WRITE = ((STANDARD_RIGHTS_WRITE _
Or KEY_SET_VALUE _
Or KEY_CREATE_SUB_KEY) And (Not SYNCHRONIZE))
```

و بطور معمول از KEY_ALL_ACCESS استفاده می شود. برای مثال STANDARD_RIGHTS_READ به کاربر اجازه خواندن اطلاعات از محضر خانه را می دهد ولی به او اجازه اعمال تغییر در آن را نمی دهد.

phkResult متغیری است برای ذخیره handle کلید باز شده. از این متغیر در تابع RegQueryValueEx استفاده می شود.

RegQueryValueEx-۲

برای بدست آوردن اطلاعات ذخیره شده در کلیدی دلخواه که توسط تابع RegOpenKeyEx باز شده است بکار می رود.

```
Declare Function RegQueryValueEx Lib "advapi32" _
Alias "RegQueryValueExA" (ByVal hKey As Long, _
ByVal lpValueName As String, lpReserved As Long, _
lpType As Long, ByVal lpData As String, _
lpcbData As Long) As Long
```

توضیح آرگومانها:

hKey همان آخرین آرگومان تابع RegOpenKeyEx یعنی phkResult می باشد.

lpValueName مشخص می کند که شما به کدامیک از کلیدها برای بدست آوردن مقدار آن نیاز دارید. اگر فقط یک کلید وجود دارد یا مقدار کلید استاندارد Default را می خواهید آنرا خالی بگذارید.

lpReserved معمولاً بکار نمی رود و بجای آن 0& می گذارند.

lpType نوع داده ای است که شما مایل به دریافت اطلاعات آن هستید. و یکی از مقادیر زیر است:

```
'Data type Public Constants
Public Const REG_NONE = 0
Public Const REG_SZ = 1
Public Const REG_EXPAND_SZ = 2
Public Const REG_BINARY = 3
Public Const REG_DWORD = 4
Public Const REG_DWORD_LITTLE_ENDIAN = 4
Public Const REG_DWORD_BIG_ENDIAN = 5
Public Const REG_LINK = 6
Public Const REG_MULTI_SZ = 7
Public Const REG_RESOURCE_LIST = 8
Public Const REG_FULL_RESOURCE_DESCRIPTOR = 9
Public Const REG_RESOURCE_REQUIREMENTS_LIST = 10
```

نوع داده ای که غالباً بکار می رود رشته ای بوده و معادل REG_SZ می باشد.

lpData خروجي اين تابع بوده و آرگومان بعدي يعني lpcbData سايز آن مي باشد.
معمولا در تابع به صورت زير بكار مي روند:

```
Dim lpData As String
Dim lpcbData As Long
lpcbData = 255
lpData = Space(lpcbData)
```

اگر lpData را با کاراکترهاي خالي پر نکنيد اين تابع به درستي کار نخواهد کرد.

RegCloseKey-۳

بعد از گشودن يك كليد بايد آنرا بست. اگر اين کار صورت نگیرد سيستم با کمبود حافظه روبرو خواهد شد و پايداري اش کم مي گردد.

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

ثابت هاي خطاي محضر خانه :

```
' Return codes from Registration functions.
Public Const ERROR_SUCCESS = 0&
Public Const ERROR_BADDB = 1009&
Public Const ERROR_BADKEY = 1010&
Public Const ERROR_CANTOPEN = 1011&
Public Const ERROR_CANTREAD = 1012&
Public Const ERROR_CANTWRITE = 1013&
Public Const ERROR_OUTOFMEMORY = 14&
Public Const ERROR_INVALID_PARAMETER = 87&
Public Const ERROR_ACCESS_DENIED = 5&
```

اگر خروجي توابع کار کردن با محضر خانه غير از ERROR_SUCCESS باشد نشانه وجود خطايي در کار آن تابع مي باشد. مي توان از روال زير براي فهميدن اين موضوع استفاده کرد.

```
Sub APIDisplayError(Code As Long)

Select Case Code
Case ERROR_BADDB
MsgBox "Corrupt Registry Database!"
Case ERROR_BADKEY
MsgBox "Key name is bad"
Case ERROR_CANTOPEN
MsgBox "Cannot Open Key"
Case ERROR_CANTREAD
MsgBox "Cannot Read Key"
Case ERROR_CANTWRITE
MsgBox "Cannot Write Key"
Case ERROR_ACCESS_DENIED
MsgBox "Access to Registry Denied"
Case ERROR_OUTOFMEMORY
MsgBox "Out of memory"
Case ERROR_INVALID_PARAMETER
MsgBox "Invalid Parameter"
Case Else
MsgBox "Undefined key error code!"
```



```
End Select
```

```
End Sub
```

مثال زیر طرز استفاده صحیح از توابع فوق را بیان می کند:

```
Public Function myGetSetting( _  
hKey As RegistryKeys_in_hKey, _  
lpSubKey As String, _  
lpKeyValueName As String) As String  
  
'Usage:  
'MsgBox myGetSetting(HKEY_LOCAL_MACHINE, _  
"Software\Microsoft\Windows\CurrentVersion\App Paths\IEXPLORE.EXE", _  
""), vbInformation, "IE Path"  
' another Example:  
'SaveSetting App.Title, "Startup", "Left", "1000"  
'MsgBox myGetSetting(HKEY_CURRENT_USER, _  
"Software\VB and VBA Program Settings\" + App.Title + "\Startup", _  
"Left"), vbInformation, _  
"CURRENT_USER\Software\VB and VBA Program Settings\Project1\Startup -  
>Left ="  
  
Dim lReturn As Long  
Dim hWndSoftware As Long  
Dim lpData As String  
Dim lpcbData As Long  
  
lReturn = RegOpenKeyEx(hKey, _  
lpSubKey, 0&, KEY_ALL_ACCESS, hWndSoftware)  
  
lpcbData = 255  
lpData = Space(lpcbData)  
  
lReturn = RegQueryValueEx(hWndSoftware, lpKeyValueName, 0&, _  
REG_SZ, lpData, lpcbData)  
  
If lReturn <> ERROR_SUCCESS Then  
APIDisplayError lReturn  
Else  
myGetSetting = Left(lpData, lpcbData - 1)  
End If  
  
lReturn = RegCloseKey(hWndSoftware)  
End Function
```

[RegCreateKeyEx-۴](#)

یک کلید جدید در بانک اطلاعاتی محضرخانه ویندوز ایجاد می کند.

```
Declare Function RegCreateKeyEx Lib "advapi32" _  
Alias "RegCreateKeyExA" (ByVal hKey As Long, _  
ByVal lpSubKey As String, ByVal Reserved As Long, _  
ByVal lpClass As String, ByVal dwOptions As Long, _  
ByVal samDesired As Long, _  
lpSecurityAttributes As SECURITY_ATTRIBUTES, _  
phkResult As Long, lpdwDisposition As Long) As Long
```

```
Type SECURITY_ATTRIBUTES
nLength As Long
lpSecurityDescriptor As Variant
bInheritHandle As Long
End Type
```

توضیح آرگومانها:

hKey کلیدی را مشخص می کند که می خواهید برای آن lpSubKey بسازید.
Reserved طبق معمول صفر است.
lpClass نام کلاسی را مشخص می کند که مقدار جدید به آن متعلق خواهد بود.
dwOptions یکی از دو ثابت زیر است و مشخص می کند که کلیدی را که می خواهید ایجاد کنید آیا موقتی است یا دائمی .

```
'Options
Public Const REG_OPTION_VOLATILE = 0
Public Const REG_OPTION_NON_VOLATILE = 1
```

Desired همان پوشش امنیتی است که در تابع RegOpenKeyEx توضیح داده شد.
lpSecurityAttributes : برای استفاده آن در تابع یک متغیر از نوع ساختار معرفی شده
تعریف کنید و سپس بدون مقدار دهی اولیه آنرا در تابع بکار برید.
phkResult یک handle از کلید تازه ایجاد شده را برمی گرداند.
lpdwDisposition را هم زیاد جدی نگیرید و فقط یک متغیر از نوع Long تعریف کنید
و بدون مقدار دهی اولیه آنرا در تابع استفاده نمایید.

RegSetValueEx

یک مقدار به کلید ایجاد شده یا موجود نسبت میدهد.

```
Declare Function RegSetValueEx Lib "advapi32" _
Alias "RegSetValueExA" (ByVal hKey As Long, _
ByVal lpValueName As String, ByVal Reserved As Long, _
ByVal lpType As Long, ByVal lpData As String, _
ByVal cbData As Long) As Long
```

توضیح آرگومانها:

تمام آرگومانهای این تابع مانند توابع فوق بوده و توضیحات کاملی برای آنها ارائه شد.

مثال زیر طرز استفاده از دو تابع اخیر را بیان می کند:

```
Sub my_setRegValue( _
RegistryKey As RegistryKeys_in_hKey, _
lpSubKey As String, _
sectionKey As String, _
sectionKeyValue As String, _
sectionNewValue As String _
)

' VB Savesetting function Professional simulation!

'Usage:
'my_setRegValue HKEY_CURRENT_USER, _
"Software\VB and VBA Program Settings\Project1", _
"Startup", "Left", "0"
```

```

Dim lReturn As Long
Dim hWndQue As Long
Dim hWnd As Long
Dim sClass As String
Dim Security As SECURITY_ATTRIBUTES
Dim lDisposition As Long
sClass = Space(255)
lReturn = RegCreateKeyEx(RegistryKey, lpSubKey, 0&, _
    sClass, REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, Security, _
    hWndQue, lDisposition)
If lReturn <> ERROR_SUCCESS Then
gAPIDisplayError lReturn
End If
sClass = Space(255)
lReturn = RegCreateKeyEx(hWndQue, sectionKey, 0&, sClass, _
    REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, Security, hWnd, _
    lDisposition)
If lReturn <> ERROR_SUCCESS Then
gAPIDisplayError lReturn
End If
'Save value in the Registry
lReturn = RegSetValueEx(hWnd, sectionKeyValue, 0&, REG_SZ, _
    sectionNewValue, Len(sectionNewValue))
If lReturn <> ERROR_SUCCESS Then
gAPIDisplayError lReturn
End If
'Close open keys
lReturn = RegCloseKey(hWndQue)
lReturn = RegCloseKey(hWnd)

End Sub

```

[RegDeleteKey-٦](#)

```

Declare Function RegDeleteKey Lib "advapi32" _
Alias "RegDeleteKeyA" (ByVal hKey As Long, _
ByVal lpSubKey As String) As Long

```

قابلیت حذف کلید ایجاد شده را بوجود می آورد. این تابع کلیدی را با زیر شاخه هایش حذف نمی کند. ابتدا باید تمام زیر شاخه ها را حذف کنید تا بتوانید خود کلید را حذف نمایید.

[RegDeleteValue -٧](#)

```

Declare Function RegDeleteValue Lib "advapi32" _
Alias "RegDeleteValueA" (ByVal hKey As Long, _
ByVal lpValueName As String) As Long

```

ValueName را از Key حذف می کند.

[مثال :](#)

برنامه های زیادی در ویندوز وجود دارند که با تقه زدن روی شکلک آنها برنامه مربوطه اجرا شده و فایل مورد نظر را باز می کنند. در ویژوال بیسیک فقط با استفاده از توابع فوق و کمی

اطلاعات در مورد ساختار محضرخانه ویندوز می توان این کار را انجام داد. قسمت بعد به چگونگی انجام این کار (ایجاد ارتباط بین نوع فایلها و برنامه ها) می پردازد:

ابتدا برنامه RegEdit را اجرا کنید. نوع فایل های شناخته شده در سیستم شما تحت کلید **HKEY_CLASSES_ROOT** قابل مشاهده اند. برای مثال فایل های وابسته به برنامه NotePad ویندوز فایل های به صورت *.TXT هستند. تحت ریشه فوق اگر در برنامه ویرایشگر محضرخانه txt را پیدا کنید کلید پیش فرض آن با txtfile مقدار دهی شده است. اگر در همین ریشه کمی پایین بیایید کلید txtfile را می توانید پیدا کنید. کلید پیش فرض آن با Text Document مقدار دهی گشته است. این مقدار اخیر همان Tip کوچکی است که با نگه داشتن اشاره گر ماوس روی شکلک فایل متنی ظاهر می شود. اگر روی علامت بعلاوه تکه بزنید تحت شاخه فرعی shell\open\command می توانید نام و مسیر فایل اجرایی NotePad را ببینید. Open در شاخه فرعی فوق همان نامی است که با تکه راست روی شکلک فایل متنی در منوی جهنده ظاهر می شود. با مراجعه به HKEY_CLASSES_ROOT\txtfile\DefaultIcon مسیر شکلک وابسته کلید

به فایل های برنامه مشاهده می شود. که یا می تواند مسیر یک فایل شکلک (Icon) روی کامپیوتر شما باشد و یا معمولاً شماره فایل شکلکی است که در فایل اجرایی یا یک Dll وابسته به برنامه وجود دارد. فایل های اجرایی ویژه ال بیسیک یک شکلک بیشتر ندارند و آن نیز با اندیس صفر نمایش داده میشود در کلید DefaultIcon یا مسیر یک فایل شکلک ذکر می شود و یا نام فایل اجرایی به همراه اندیس شکلک مورد نظر در آن. به سادگی با ایجاد کلید های فوق و مقدار دهی کلید های پیش فرض می توان بین برنامه ها و فایل های آنها وابستگی ایجاد نمود:

```
Public Sub CreateEXEAssociation( _
    ByVal sExePath As String, _
    ByVal sectionClassName As String, _
    ByVal sAssociationExtention As String, _
    ByVal strInfoTip As String, _
    Optional ByVal lDefaultIconIndex As Long = -1 _
)
'Usage:
'CreateEXEAssociation "Notepad.exe", "myNewFileType", _
    "vn1", "", 0
' Association between *.vn3 and your VB project
' Visual Basic applications seem to support displaying
' only the first icon (index 0).
'CreateEXEAssociation App.Path + "\" + App.EXEName + ".exe", _
    "myNewFileType3", "vn3", "My project1 file. cbcjhdh!!", 0
' to use this file Association write this code in form_load .
Private Sub Form_Load()
' MsgBox Command, vbInformation, "my EXE Association"
End Sub

setRegValue HKEY_CLASSES_ROOT, _
    "." & sAssociationExtention, "", _
    "", sectionClassName
```

```

If Trim(strInfoTip) <> "" Then
    setRegValue HKEY_CLASSES_ROOT, _
        "." & sAssociationExtention, "", _
        "InfoTip", strInfoTip
End If

setRegValue HKEY_CLASSES_ROOT, _
    "." & sAssociationExtention & "\shell\open", "command", _
    "", sExePath & " "%1""

setRegValue HKEY_CLASSES_ROOT, _
    sectionClassName, _
    "", "", sectionClassName

setRegValue HKEY_CLASSES_ROOT, _
    sectionClassName & "\shell\open", _
    "command", "", sExePath & " "%1""

If lDefaultIconIndex > -1 Then
    setRegValue HKEY_CLASSES_ROOT, _
        sectionClassName, "DefaultIcon", "", _
        sExePath & ", " & CStr(lDefaultIconIndex)
End If

End Sub

```

بعد از اجرای تابع فوق و يك بار Restart کردن کامپیوتر فایل دلخواه شما به سیستم معرفی می شود و اگر روی شکلک فایل وابسته کلیک کنید برنامه شما خود به اجرا می شود تا فایل را باز کند. برای این منظور در Form_Load برنامه از تابع Command در VB باید استفاده نمود. در این حالت این تابع نام و مسیر فایلی را که به برنامه وابسته است و روی آن تکه زده شده را بر می گرداند.

[طریقه استفاده از فایلهاي *.ini](#)

استفاده از اینگونه فایلها در ویندوزهاي ۱۶ بیتی که محضرخانه ندارند مرسوم است (بود!). ولی چون توابع ۳۲ بیتی آنها نیز ارائه شده و در ضمن ویندوزهاي ۳۲ بیتی هم فایلی به نام *win.ini* دارند به توضیح آنها پرداخته می شود.

[WritePrivateProfileString-۱](#)

برای ذخیره کردن تنظیمات برنامه در يك فایل متنی بکار می رود.

```

'The Declare statement for WritePrivateProfileString in the
'WIN32API.TXT file has the lpString parameter as Any.
'It should be declared as ByVal lpString as String.

```

```
Public Declare Function WritePrivateProfileString Lib "kernel32" _

```

```
Alias "WritePrivateProfileStringA" _
(ByVal lpApplicationName As String, ByVal lpKeyName As String, _
ByVal lpString As String, ByVal lpFileName As String) As Long
```

توضیح آرگومانها:

lpApplicationName متنی است که در بین دو براکت "[]" در فایل lpFileName قرار می گیرد.
lpKeyName کلیدی است که مقدار lpString به آن نسبت داده میشود.

مثال:

```
Public Sub saveSetting_In_my_INI_file( _
ByVal ApplicationName As String, _
ByVal KeyName As String, _
ByVal StringToStore As String, _
ByVal iniFileName As String _
)
'Usage:
'saveSetting_In_my_INI_file "Section 1", "StartUp", 1100, _
App.Path + "\" + "mySetting.ini"
'saveSetting_In_my_INI_file "Section 2", "Left", "500", _
App.Path + "\" + "mySetting.ini"

Dim lReturn As Long

lReturn = WritePrivateProfileString(ApplicationName, KeyName, _
StringToStore, iniFileName)

If lReturn <> ERROR_SUCCESS Then
APIDisplayError lReturn
End If

End Sub
```

در اثر اجرای کد زیر:

```
saveSetting_In_my_INI_file "Section 2", "Left", "500", _
App.Path + "\" + "mySetting.ini"
```

در مسیر جاری فایل اجرایی برنامه فایلی با نام mySetting.ini ایجاد می شود که ساختار آن به صورت زیر است:

```
[Section 2]
Left=500
```

[GetPrivateProfileString-۲](#)

```
'The Declare statement for GetPrivateProfileString in the
'WIN32API.TXT file has the lpKeyName parameter as Any.
'It should be declared as ByVal lpKeyName as String.
```

```
Public Declare Function GetPrivateProfileString Lib "kernel32" _
Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, _
ByVal lpKeyName As String, ByVal lpDefault As String, _
ByVal lpReturnedString As String, ByVal nSize As Long, _
ByVal lpFileName As String) As Long
```

برای دریافت مقدار موجود در lpKeyName در قسمت lpApplicationName در فایل

lpFileName بکار می رود.
تابع زیر طرز استفاده از آنرا نشان می دهد:

```
Public Function getSetting_From_my_INI_file( _
ByVal lpAppName As String, _
lpKeyName As String, _
ByVal lpDefault As String, _
ByVal lpINIFileName) As String

'Usage:
'MsgBox getSetting_From_my_INI_file("Section 1", "StartUp", _
    "10", App.Path + "\" + "mySetting.ini")

'MsgBox getSetting_From_my_INI_file("Section 2", "Left", _
    "10", App.Path + "\" + "mySetting.ini")

Dim lReturn As Long

Dim snSize As String
snSize = Space(50)

lReturn = GetPrivateProfileString(lpAppName, lpKeyName, lpDefault, _
    snSize, Len(snSize), lpINIFileName)
If lReturn <> ERROR_SUCCESS Then
APIDisplayError lReturn
End If

getSetting_From_my_INI_file = snSize

End Function
```

GetPrivateProfileInt -۳

```
Declare Function GetPrivateProfileInt Lib "kernel32" _
Alias "GetPrivateProfileIntA" (ByVal lpApplicationName As _
String, ByVal lpKeyName As String, ByVal nDefault As Long, _
ByVal lpFileName As String) As Integer
```

عملکرد آن مانند تابع GetPrivateProfileString می باشد فقط با این تفاوت که خروجی آن یک عدد صحیح است.

سه تابع دیگر هم در این زمینه وجود دارند که بطور اختصاصی با فایل **WIN.INI** کار میکنند.

```
Public Declare Function WriteProfileString Lib "kernel32" _
Alias "WriteProfileStringA" (ByVal lpszSection As String, _
ByVal lpszKeyName As String, ByVal lpszString As String) As Long

Public Declare Function GetProfileString Lib "kernel32" _
Alias "GetProfileStringA" (ByVal lpAppName As String, _
ByVal lpKeyName As String, ByVal lpDefault As String, _
ByVal lpReturnedString As String, ByVal nSize As Long) As Long

Public Declare Function GetProfileInt Lib "kernel32" _
Alias "GetProfileIntA" (ByVal lpAppName As String, _
ByVal lpKeyName As String, ByVal nDefault As Long) As Long
```

و طرز کار آنها مانند توابعی است که در بالا توضیح داده شد.

تذکر:

هیچگاه از فایل **WIN.INI** برای ذخیره کردن تنظیمات برنامه خود استفاده نکنید. زیرا حجم این فایل محدود است و اگر زیاد شود ویندوز را دچار مشکل خواهد کرد.

توسط : وحید نصیری

Tuesday, 19 September, 2000