

## نحوه فراخوانی نسخه ی Unicode توابع API در VB

Microsoft Help and Support  
Article ID : 145727  
Last Review : June 29,2004

تعریف و فراخوانی نسخه ی Unicode توابع ۳۲ بیتی API با آنچه در مورد نسخه های ANSI بکار گرفته می شود متفاوت است در این مقاله راهی ارائه شده است تا بتوان از نسخه ی Unicode یک تابع API بدون نیاز به ساختن Type Library استفاده کرد.

ساختار داخلی ویژوال بیسیک همانند ویندوز های NT و 2000 بر پایه استاندارد ۲ بایتی Unicode بنا شده است. با این وجود، ویبی فکر می کند که دنیای خارج از آن هنوز از مدل تک بایتی ANSI استفاده می کند. از این رو تمام رشته هایی که بصورت پارامتر برای یک تابع خارجی فرستاده می شوند را قبل از ارسال به ANSI تبدیل می کند. به همین ترتیب، تمام رشته هایی که یک تابع خارجی برمی گرداند را ANSI فرض کرده و به Unicode تبدیل می کند. این رفتار ویژوال بیسیک باعث شده است فراخوانی یک تابع خارجی که با رشته های Unicode کار می کند بصورت معمول غیرممکن شود.

یک راه برای فراخوانی یک تابع Unicode ساختن یک Type Library است که به آن تابع ارجاع کند. کفایت از درون برنامه به این Type Library رجوع کنید و تمام توابع Unicode را توسط آن فراخوانی کنید.

برای کسب اطلاعات بیشتر در این مورد می توانید نگاهی بر فایل متنی VB4DLL.txt یا VB5DLL.doc که در همان شاخه ی ویژوال بیسیک قرار دارند، بیاندازید.

راه دیگر برای استفاده از توابع Unicode استفاده از آرایه های از نوع Byte است. اگر رشته هایی که به توابع خارجی ارسال یا از آنها دریافت می شوند بصورت آرایه هایی از نوع Byte به ویژوال بیسیک معرفی شوند آنوقت ویژوال بیسیک هیچ تغییری در آنها نخواهد داد.

کد زیر نحوه ی تعریف و فراخوانی هر دو نسخه ی ANSI و Unicode تابع GetWindowsDirectory را نشان می دهد. همانطور که می بینید نسخه ی ANSI توابع API ویندوز با پسوند A و نسخه ی Unicode آنها با پسوند W (اول کلمه ی Wide) نامگذاری شده اند:

```
Declare Function GetWindowsDirectoryA Lib "Kernel32" _
    (ByVal lpBuffer As String, ByVal nSize As Long) As Long
Declare Function GetWindowsDirectoryW Lib "Kernel32" _
    (lpBuffer As Any, ByVal nSize As Long) As Long

Sub mainA()
    Dim sBuf As String
    Dim cSize As Long
    Dim retval As Long
    sBuf = String(255, 0)
    cSize = 255
    retval = GetWindowsDirectoryA(sBuf, cSize)
    sBuf = Left(sBuf, retval)
    Debug.Print sBuf
End Sub

Sub mainW()
    Dim Buf() As Byte, sBuf As String
    Dim cSize As Long
    Dim retval As Long
    ReDim Buf(254)
    cSize = 255
    retval = GetWindowsDirectoryW(Buf(0), cSize)
```

```
sBuf = Left(Buf, retval)
Debug.Print sBuf
End Sub
```

روال ( ) mainA نسخه ی ANSI و روال ( ) mainW نسخه ی Unicode تابع را فراخوانی می کند. توجه داشته باشید اگرچه عملکرد این دو روال بسیار متفاوت است اما خروجی هر دو یکسان است.

نحوه ی فراخوانی نسخه ANSI به همان روش معمولی صورت می گیرد که برای ارسال و دریافت رشته ها هنگام کار با یک فایل Dll خارجی استفاده می شود. در این موارد رشته را با یک طول مشخص محدود و آن را با کاراکتر space پر می کنند. سپس با ارسال این رشته و طولش بعنوان آرگومان تابع را فراخوانی می کنند. بعد از اتمام کار با تابع نیز، space های اضافی حذف و مقدار اصلی رشته خوانده می شود.

فراخوانی نسخه ی Unicode توابع اندکی متفاوت است. علاوه بر تعریف یک متغیر رشته ای، sBuf، یک آرایه ی پویا (Dynamic) از نوع Byte نیز تعریف کردیم. سپس با دستور ReDim اندازه ی آن را ۲۵۵ بایت قرار دادیم (آرایه از ۰ اندیس گذاری شده است). هنگام فراخوانی تابع، آدرس اولین عنصر (خانه) آرایه به همراه اندازه ی آن برای تابع ارسال می شود.

تابع، آرایه ی Buf را با کدهای Unicode مربوط به کاراکترهای مسیر پوشه ی ویندوز، پر می کند و تعداد بایت هایی را که نوشته است بعنوان خروجی برمی گرداند. به یاد داشته باشید تا وقتیکه تابع اطلاعات را به فرم Unicode برمی گرداند هر کاراکتر ۲ بایت از آرایه را اشغال می کند. برای مثال اگر پوشه ی ویندوز شما در مسیر 'C:\WINNT' قرار داشته باشد آنگاه Buf(0) برابر ۶۷ خواهد شد، Buf(1) برابر ۰، Buf(2) برای ۵۸، Buf(3) برابر ۰ و به همین ترتیب. اگر از کدهای غیر انگلیسی استفاده شود ممکن است کدها مقادیر متفاوت باشد.

در کد بالا در خطی که بلافاصله بعد از خط تعریف تابع آمده است ویژوال بیسیک آرایه ی بایتی را به رشته تبدیل می کند. در هنگام چنین عملیاتی ویژوال بیسیک فرض می کند اطلاعات موجود در آرایه به فرم Unicode ذخیره شده اند. بدین ترتیب متغیر sBuf یک رشته معمول ویژوال بیسیک خواهد بود و مسیر پوشه ی ویندوز در آن ذخیره خواهد شد. در مثال بالا یک آرایه ی خالی و دست نخورده به تابع ارسال شد، درحالیکه می توان یک آرایه از قبل مقداردهی شده را نیز به یک تابع خارجی ارسال کرد. برای مثال قطعه کد زیر، یک آرایه از نوع بایت را تعریف و مقداردهی و سپس آن را به تابعی به نام MyPassFunction ارسال می کند.

```
Dim bArray() As Byte
bArray = "Hello" & vbNullChar
MyPassFunction bArray(0)
```

حامد احمدی  
Tizfekr@yahoo.com  
۸۳/۱۲/۱۶