

[Share](#)[About TheScarms](#)[Sample code](#)
[version info](#)

Windows Registry / Ini File Functions

Here are a set of functions that allow you to read and write to the [Window's Registry](#) or [Initialization \(.Ini\) files](#). These functions will help you perform most common tasks associated with storing and retrieving your application's settings. There are many situations where you may need to access the registry in a manner not supported by these functions. However, the use of most registry APIs is illustrated and it should not be difficult to use these functions as a starting point in developing your own routines.

These functions are provided as a [.Bas module](#) that you can include in your project. The module is self contained and the syntax for all functions is explained below. All functions return a non zero result when an error occurs. Although use of the registry is recommended, the need to access ini files still exists. For that reason, these function will operate on either.

In real life, I use these functions in various applications that are in production everywhere. Instead of implementing them as functions in a standard module, they are methods of a class module in a multi-threaded, out-of-process ActiveX code component. Also, instead of simply returning an error code they raise errors in the calling procedure. The intention here is to demonstrate registry access through API calls and not how to create ActiveX components. It is a simple matter to encapsulate these functions into a stand alone object.

[Download Source Code](#)

Registry /Ini File APIs Used

The registry APIs are from [Advapi32.dll](#) while the .Ini file APIs are from [Kernel32.dll](#).

RegCloseKey	RegOpenKeyEx	GetPrivateProfileInt
RegCreateKeyEx	RegQueryValueEx	GetPrivateProfileSection
RegDeleteKey	RegSetValueEx	GetPrivateProfileString
RegDeleteValue		WritePrivateProfileString
RegEnumValue		

Registry Module Description

This standard module provides functions to store and retrieve values from the *Windows registry* as well as from *.Ini* files. All functions return zero on success and non-zero otherwise. However, most of the examples provided below discard the return value.

Function Name	Description
fReadValue	Reads a <i>string or binary or integer</i> value from the <i>registry</i> or a <i>string or integer</i> value from an <i>.ini</i> file.
fWriteValue	Writes a <i>string or binary or integer</i> value to the <i>registry</i> or a <i>string</i> value to an <i>.ini</i> file. If the registry key or .ini file entry does not exist it is created.
fEnumValue	Enumerates all <i>string, dword and some binary</i> values

	under a <i>registry</i> subkey or <i>all</i> values in a particular section of an <i>.ini</i> file. The <i>.ini</i> section enumerated is limited to 8K characters.
fEnumKey	Enumerates all subkeys under a registry key.
fDeleteKey	Deletes a subkey from the registry. The key cannot have any subkeys.
fDeleteValue	Deletes a value from the registry or from an <i>.ini</i> file.
fReadIniFuzzy	Reads a string value from an <i>.ini</i> file when only a portion of the section heading is known. If the section heading contains multiple words, you can specify any complete word. The full section heading is also returned.

Common Parameters

sTopKeyOrIniFile - Can be one of the following *string* values:

A top level registry key *abbreviation* from this list:

Abbreviation Meaning

HKCU	HKey Current User
HKLM	HKey Local Machine
HKU	HKey User
HKDD	HKey Dynamic Data
HKCC	HKey Current Configuration
HKCR	HKey Classes Root

Or, the full path of an *.ini* file (ex. C:\Windows\MyFile.ini)

- registry or ini file operations or
- "B" for a 16-bit binary value or
- "D" for a 32-bit integer (*dword*) value.

sSubKeyOrSection - Can be one of the following *string* values:

- A registry subkey or
- An *.ini* file section name

sValueName - Can be one of the following *string* values:

- A registry entry or
- An *.ini* file entry

sValueType - Can be one of the following *string* values:

- "S" for a string value on

vDefault

The default *string*, *binary* or *integer* value to return.

vValue

A *string*, *binary* or *integer* value. When reading values, it is set to vDefault if unsuccessful (0 when reading an integer from an *.ini* file)

Sample Registry

```
HKey_Current_User
\Software
\CompanyName
```

```

\ProductName
\AppName
\Setting1 = "String1"
\Setting2 = "String2"
\Setting3 = True
\Setting4 = 12345
\AppVersion
\ (default) = "String1"
\Company2Name
\Company3Name

```

Sample Ini File

C:\Windows\MyFile.Ini

```

[Section1]
Setting1=String1
Setting2=String2
Setting3=5

```

```

[Section Name]
Setting4=String4
Setting5=String5
Setting6=6

```

Examples of Reading a Single Value

Function Syntax:

```

Function fReadValue (ByVal sTopKeyOrFile As String, _
    ByVal sSubKeyOrSection As String, ByVal sValueName As String, _
    ByVal sValueType As String, ByVal vDefault As Variant, _
    vValue As Variant) As Long

```

Example 1

Read the *Setting1* string value from the registry and store the result in variable sSetting. If the value does not exist, use "No Name" as a default.

```

Call fReadValue("HKCU", "Software\CompanyName\ProductName\AppName", _
    "Setting1", "S", "No Name", sSetting)

```

Example 2

Read a similar string value from an .ini file.

```

Call fReadValue("C:\Windows\Myfile.ini", "Section1", "Setting1", "S", _
    "No Name", sSetting)

```

Example 3

Read binary (True | False) value *Setting3* from the registry. Default is False.

```

Call fReadValue("HKCU", "Software\CompanyName\ProductName\AppName", _
    "Setting3", "B", False, bSetting)

```

Example 4

Read an integer (dword) value *Setting4* from the registry. Default is 100.

```

Call fReadValue("HKCU", "Software\CompanyName\ProductName\AppName", _
    "Setting4", "D", 100, iSetting)

```

Example 5

Read an integer value from an .ini file and check the return code.

```

lResult = fReadValue("C:\Windows\Myfile.ini", "Section1", "Setting3", "B", _
    "0", iSetting)

```

```

if lResult <> 0 then . . .

```

Example 6

Read the *Default* value from the ... \AppVersion registry entry. The only difference from Example 1 is the *sValueName* parameter is blank.

```

Call fReadValue("HKCU", "Software\CompanyName\ProductName\AppName", "", _
    "S", "", sSetting)

```

Examples of Writing a Single Value

Function Syntax:

```
Function fWriteValue(ByVal sTopKeyOrFile As String, ByVal sSubKeyOrSection As String, _  
    ByVal sValueName As String, _ByVal sValueType As String, ByVal _  
    vValue As Variant) As Long
```

Example 1

Write the string "String1" to the "Setting1" registry entry.

```
Call fWriteValue("HKCU", "Software\CompanyName\ProductName\AppName", _  
    "Setting1", "S", "String1")
```

Example 2

Write the string "String2" to the Setting2 entry in MyFile.ini.

```
IResult = fWriteValue("C:\Windows\Myfile.ini", "Section1", "Setting2",  
    "S", "String2")
```

Example 3

Write a binary (True | False) value to the "Setting3" registry entry.

```
bValue = True  
  
Call fWriteValue("HKCU", "Software\CompanyName\ProductName\AppName", _  
    "Setting3", "B", bValue)
```

Example 4

Write an integer (dword) value to the "Setting4" registry entry.

```
IValue = 1234  
  
Call fWriteValue("HKCU", "Software\CompanyName\ProductName\AppName", _  
    "Setting4", "D", IValue)
```

NOTE1: This function will create the registry key or ini entry if it does not exist. This function will create the registry key or ini entry if it does not exist.

NOTE2: This ini version of this function can only write string values to .ini files. This ini version of this function can only write string values to .ini files.

Examples of Enumerating Values

Function Syntax:

```
Public Function fEnumValue(ByVal sTopKeyOrIniFile As String, _  
    ByVal sSubKeyOrSection As String, sValues As String) As Long
```

Parameter:

sValues - A string of the form:

EntryName=Value|EntryName=Value|... EntryName=Value||

Where - Value can be a string or binary value. and "|" equals vbNullChar (chr(0)).

Example 1

Retrieve all values under the "AppName" registry key.

```
Call fEnumValue("HKCU", "Software\CompanyName\ProductName\AppName", sValues)
```

sValues will contain:

"Setting1=String1|Setting2=String2|Setting3=True||"

Example 2

Retrieve all values under the [Section1] section of "C:\Windows\Myfile.ini".

```
Call fEnumValue("C:\Windows\Myfile.ini", "Section1", sValues)
```

sValues will contain the same string as in Example 1.

NOTE1: The *.ini* section enumerated is limited to 8K characters. The *.ini* section enumerated is limited to 8K characters.

Examples of Enumerating Keys

Function Syntax:

```
Public Function fEnumKey(ByVal sTopKey As String, ByVal sSubKey As String, _  
    sValues As String) As Long
```

Parameter:

sValues - A string of the form:

```
SubKeyName|SubKeyName|.... SubKeyName|
```

Where - "|" equals vbNullChar (chr(0)).

Example 1

Retrieve all key names under HKCU\Software.

```
Call fEnumKey("HKLM", "Software", sValues)
```

sValues will contain:

```
"CompanyName|Company2Name|Company3Name|"
```

Examples of Deleting Keys

Function Syntax:

```
Public Function fDeleteKey(ByVal sTopKey As String, _  
    ByVal sSubKey As String, ByVal sKeyName As String) As Long
```

Parameters:

sSubKey - The path of the registry sub-key to delete.

sKeyName - The name of the registry sub-key to delete.

Example 1

Delete the "AppName" sub-key.

```
Call fDeleteKey("HKCU","Software\CompanyName\ProductName\AppName", "AppName")
```

NOTE: The key to be deleted cannot be a *top-level* key (HKCU, ...) and cannot have any sub-keys

Examples of Deleting Values

Function Syntax:

```
Public Function fDeleteValue(ByVal sTopKeyOrFile As String, _  
    ByVal sSubKeyOrSection As String, ByVal sValueName As String) As Long
```

Example 1

Delete the "Setting3" registry value.

```
Call fDeleteValue("HKCU","Software\CompanyName\ProductName\AppName", "Setting3")
```

Example 2

Delete the "Setting3" entry from MyFile.ini.

```
Call fDeleteValue("C:\Windows\Myfile.ini", "Section1", "Setting3")
```

Example of fReadIniFuzzy

Function Syntax:

```
Public Function fReadIniFuzzy(ByVal sIniFile As String, sSection As String, _  
    ByVal sIniEntry As String, ByVal sDefault As String, _  
    sValue As String) As Long
```

Parameters:

sIniFile - The path of the .ini file to read.

sSection - Any complete part of the .ini file section name.

Example: [Section Name]

sSection can be "Section" or "Name" but not a partial value such as "Sect" or "ame".

sIniEntry - The entry to read.

sDefault - The default string to return return.

sValue - The string read of sDefault on failure.

Example 1

Read the "Setting4" entry from the [Section Name] section of MyFile.ini.

```
IResult = fReadIniFuzzy("C:\Windows\Myfile.ini", "Name", Setting4, sValue)
```

Produces:

IResult = 0

sSection = "Section Name"

sValue = "String4"

Instructions

Download the Bas module and include it in your project.

If you use this code, please mention "www.TheScarms.com"

[Email this page](#) 

© Copyright 2017 TheScarms

[Top](#)