RegistryWorx Registry Access Component

This page describes the RegistryWorx ActiveX DLL that wraps up all the Windows API functions for working with the System Registry into nice, neat VB/VBA-friendly functions.  
ShortFadeBar

Introduction

Many applications need to store information between usage sessions. There are, broadly speaking, three ways to do this. The first option, which is the subject of this page, is to store the data in the System Registry. The other options are INI files and XML files. These last two options will be described elsewhere on the web site. The System Registry is where Windows stores all program and system configuration data. As such, care must be taken when modifying or deleting registry keys and values. If you change a key to an invalid value or you delete the wrong key, applications will not behave correctly or will not start. In the worst case, Windows itself will not be able to start. The System Registry is accessed and modified by a number of Windows API (Application Programmatic Interface) that are not particularly friendly to VBA.  The RegistryWorx component wraps up these API calls into nice VBA-friendly procedures. You can [download the Setup program here](http://www.cpearson.com/Zips/RegistryWorxSetup.zip).

The RegistryWorx component is written in Visual Basic 6, SP6. The VB6 code is based generally on the code presented on the [Function For Working With The Registry](http://www.cpearson.com/excel/Registry.htm) page. See [this page](http://www.cpearson.com/excel/registry.htm) for more information about the registry and VBA code that modifies the registry, and see [this page](http://www.cpearson.com/excel/modifyRegistry.htm) for information about modifying the registry.

There is nothing specific to Excel in the RegistryWorx component. It may be used in VBA in any application or in VB6. You can [download the VB6 Project source here](http://www.cpearson.com/Zips/RegWorx.zip).

SectionBreak

Getting Started With RegistryWorx

The RegistryWorx component is an ActiveX DLL that you reference via the References dialog in VBA. After you have installed the DLL via the Setup program, go to the *Tools* menu in the VBA Editor, choose *References*, and check the box next to RegistryWorx. Once a reference is established, you can use the RegistryWorx functions. Note that if you distribute a workbook that uses RegistryWorx, you must distribute the RegistryWorx DLL with the workbook and the other users must install the DLL on their machines. The easiest way to do this is to distribute the setup program to the other users.

RegistryWorx Functions

RegistryWorx provides the following functions:

Registry Functions

* RegistryKeyExists
* RegistryValueExists
* RegistryCreateKey
* RegistryCreateValue
* RegistryGetValue
* RegistrySetValue
* RegistryDeleteKey
* RegistryDeleteValue
* RegistryValueExists
* RegistrySubKeyNamesToArray
* RegistryValueNamesToArray
* RegistryNumberOfSubKeys
* RegistryNumberOfValues
* RegistrySubKeysOfKey
* RegistryValuesOfKey
* RegistrySubKeyCount
* RegistryValueCount

In addition, the following information and support functions are provided:

* RegistryWorxErrorNumber
* RegistryWorxErrorText
* SystemErrorNumber
* SystemErrorText
* About
* ShowAboutForm
* TrimToNull
* IsArrayAllocated

SectionBreak

RegistryWorx Functions

The functions are described below. The constants for the BaseKey value and error numbers are provided in the modRegistryInclude.bas file that is installed in the same directory as the RegistryWorx DLL file. You should include this file in your VBA project. In all functions, the valid values for BaseKey are HKEY\_CURRENT\_USER (or HKCU) or HKEY\_LOCAL\_MACHNE (or HKLM). For safety reasons, RegistryWorx will not update keys in HKEY\_CLASSES\_ROOT, HKEY\_USERS or HKEY\_CURRENT\_CONFIG.

In all functions that take KeyName as a parameter, KeyName must be the full key name under BaseKey. For example, Software\MyCompany\MyApplication\MyKeyName. Unqualified or partially qualified names are not supported. In the key name example, if the parent keys of MyKeyName do not exist, they will be created.

Public Function RegistrySubKeysOfKey(BaseKey As Long, KeyName As String) As String()  
This function replaces RegistrySubkeyNamesToArray and returns an array of strings, each element of which is a subkey of the specified key. Nested subkeys are not listed. Only direct child keys of the specified key are listed. If an error occurs or there are no subkeys, an unallocated array is returned. Your code should call IsArrayAllocated to determine if the array is allocated and contains data.  
  
Public Function RegistryValuesOfKey(BaseKey As Long, KeyName As String) As String()  
This function replaces RegistryValueNamesToArray and returns an array of strings, each element of which is a value of the specified key. Nested values are not listed. Only direct child values of the specified key are listed. If an error occurs or there are no values, an unallocated array is returned. Your code should call IsArrayAllocated to determine if the array is alloccated and contains data.  
  
Public Function RegistrySubKeyCount(BaseKey As Long, KeyName As String) As Long  
This function returns the number of subkeys that are direct child keys of the specified key. Nested subkeys are not counted. Only direct child keys are counted. If an error occurs, the function returns -1.  
  
Public Function RegistryValueCount(BaseKey As Long, KeyName As String) As Long  
This function returns the number of values that are direct child values of the specified key. Nested values are not counted. Only direct child values are counted. If an error occcurs, the function reutrns -1.  
  
Public Function RegistrySubKeyNamesToArray(BaseKey As Long, KeyName As String) As Variant  
This function returns an array of the names of all the subkeys under KeyName. It returns NULL if an error occurred or if KeyName does not exist.  
  
Public Function RegistryValueNamesToArray(BaseKey As Long, KeyName As String) As Variant  
This function returns an array of the names of all the values under KeyName. It returns NULL if an error occurred or if KeyName does not exist.  
  
Public Function RegistryNumberOfSubKeys(BaseKey As Long, KeyName As String) As Long  
This function returns the number of subkeys under KeyName. It returns -1 if an error occurred or if KeyName does not exist.  
  
Public Function RegistryNumberOfValues(BaseKey As Long, KeyName As String) As Long  
This function returns the number of Values under KeyName. It returns -1 if an error occurred or if KeyName does not exist.  
  
Public Function RegistryKeyExists(BaseKey As Long, KeyName As String) As Boolean  
This function returns True or False indicating whether KeyName exists.  
  
RegistryValueExists  
This function returns True or False indicating whether ValueName exists in the key KeyName.  
  
Public Function RegistryCreateKey(BaseKey As Long, KeyName As String) As Boolean  
This function creates the key named by KeyName. The function returns True if the key was successfully created or False if an error occurred. If the result is False, you can read the error properties to determine the cause of the error.  
  
Public Function RegistryCreateValue(BaseKey As Long, KeyName As String, \_  
ValueName As String, ValueValue As Variant) As Boolean  
This function creates a named value in KeyName. It returns True if the value was successfully created or False if an error occurred.   
  
Public Function RegistryGetValue(BaseKey As Long, KeyName As String, \_  
ValueName As String) As Variant  
This function returns the value of the named value ValueName. It returns Null if the value or key does not exist, or if an error occurred.  
  
Public Function RegistrySetValue(BaseKey As Long, KeyName As String, \_  
ValueName As String, NewValue As Variant) As Boolean  
This function sets the value of the named value ValueName to the value of NewValue. If ValueName does not exist, it is created. The function returns True if the value was successfully set or False if an error occurred.  
  
Public Function RegistryDeleteKey(BaseKey As Long, KeyName As String) As Boolean  
This function deletes KeyName. It return True if the key was successfully deleted or False if an error occurred. It will return True if the key does not exist.  
  
Public Function RegistryDeleteValue(BaseKey As Long, KeyName As String, ValueName As String) As Boolean  
This function deletes the registry value ValueName for the key KeyName. It return True if the value was deleted or the value does not exist, or False if an error occurrred.  
  
Public Property Get RegistryWorxErrorNumber() As Long  
This property returns the RegistryWorx error number. These numbers are defined in the modRegistryInclude.bas file.  
  
Public Property Get RegistryWorxErrorText() As String  
This property returns the text description of the RegistryWorxErrorNumber.  
  
Public Property Get SystemErrorNumber() As Long  
This property returns the Windows system error number.  
  
Public Property Get SystemErrorText() As String  
This property returns the text description of SystemErrorNumber.  
  
Public Property Get About() As String  
This property returns a string with information about the component.  
  
Public Sub ShowAboutForm()  
This function displays the About form.  
  
Public Property Get Version() As String  
This property returns the version number (major.minor.revision) of the component.

SectionBreak

Distribution Of RegistryWorx

You are free to use the RegistryWorx ActiveX DLL component in any way you see fit. You are free to distribute RegistryWorx with your workbooks and include it any any applications that you create, including commercial works and works for hire. This distribution is royalty-free and may be done without any license from me. RegistryWorx is explicitly granted to the Public Domain. As such, it carries no warranty of fitness, explicit or implied. If, however, you include RegistryWorx in a commercial product, I would appreciate a reference to this page, http://www.cpearson.com/Excel/RegistryWorx.aspx in the source code and in any printed documentation. This attribution, however, is optional. Do what you feel is the right thing to do.

SectionBreak

You can [download the Setup program here](http://www.cpearson.com/Zips/RegistryWorxSetup.zip). The downloadable file is a zip file containing an exe file. Downloade the file to the directory of your choice, unzip it, and then run the exe file. The exe file will prompt you for the location, typically C:\Program Files\RegistryWorx, where the actual DLL file is to be installed. The exe program will properly register the DLL with Windows so it will be ready to use in your applications.

You can [download the VB6 Project source here](http://www.cpearson.com/Zips/RegWorx.zip).

This page last updated: 30-July-2007