

## Design a Utility Program to Manipulate the WINDOWS REGISTRY

**Haider M. Abdul\_Nabi**  
**Computer Department, College of Science, University of Basrah**

### Abstract:

The access to the system settings is done by a program provided by the operating system called *registry editor*, the access to this program and change its contents is very dangerous operation caused in most times failure in system if the changes are incorrect beside it doesn't have an undo feature. The suggested utility program designed to access to system settings that exists in the system registry, and change the settings without any defect in the system by the user. So the user can choose the correct options among many options that display on the screen with their description in a visual and safer way.

This program can be use from any computer user without any error that can be happened in system registry beside the easiness to the user and it doesn't require any experience in dealing with system registry because every registry's options will display as visual options and every option has comment on it's work and its effects on system and the ability to undo this options too. The code of this program are shown in the practical way part to show the practical application of this program.

Registry

"

Editor

### **Introduction**

An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner (Abraham & etal, 2003).

The windows operating system contain a set of data files used to help windows control hardware, software, the user environment, and the windows interface, this set of data files called *the registry* (Clyton, 1997) .

The registry has a subtle but important role in windows O.S. the registry is passive- its just a big collection of setting on hard disk, and probably don't think much about it while editing document, browsing the internet, or searching for a file. On the other hand it plays a key role in all those activities. This gives the registry great potential as a tool for power users enabling them to customize settings that aren't available in the user interface (Jerry, 2000).

### **The Registry**

The Registry is a database used to store settings and options for the 32 bit versions of Microsoft Windows including Windows 95, 98, ME, NT/2000 and XP. It contains information and settings for all the hardware, software, users, and preferences of the PC. Whenever a user makes changes to a Control Panel settings, or File Associations, System Policies, or installed software, the changes are reflected and stored in the Registry(Jerry, 2000).

### **Registry Warnings**

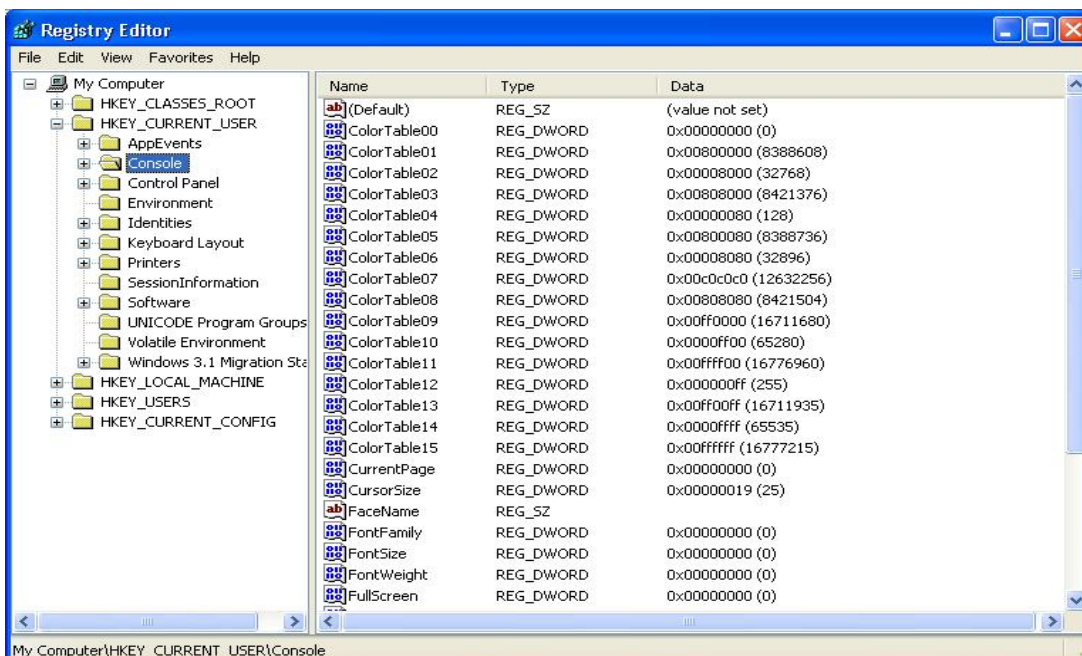
For all of its benefits, the registry is the central place for all of Windows configuration data. It's the keystone. On the other hand, the fact that the registry is so critical also makes it one of the operating system's weaknesses (Jerry, 2000). Changing the

registry is risky. The worst case scenario is that Windows will no longer start. The best case scenario is that user won't notice anything different, but on the whole, user will probably cause certain applications or devices to behave improperly (Dean & Etal, 2000).

The user can access and update the registry entries and values by using the *Registry Editor* (Fig.1.)provided by the windows operating system.

Using of registry editor is very dangerous because:

1. The *registry editor* doesn't validate user changes.
2. When a user make a change to one value, can easily miss related values in other parts of the registry.
3. The *registry editor* doesn't have an undo feature. Once a user make a change, it's a done.



**Fig. 1. Registry Editor**

### Registry Files

Windows stores the entire contents of the registry in two files: SYSTEM.DAT and USER.DAT. These are binary files that user can't view using a text editor. Windows also turns on the read-only, System, and hidden attributes of SYSTEM.DAT and USER.DAT so that user can't accidentally replace, change, or delete these files. SYSTEM.DAT contains configuration data specific to the computer on which user installed Windows. USER.DAT contains configuration data specific to the current user (Dean & etal, 2000) .

SYSTEM.DAT and USER.DAT can not edit directly, must use a tool commonly known as a "Registry Editor" to make any changes (Jerry, 2003).

## Registry Structure

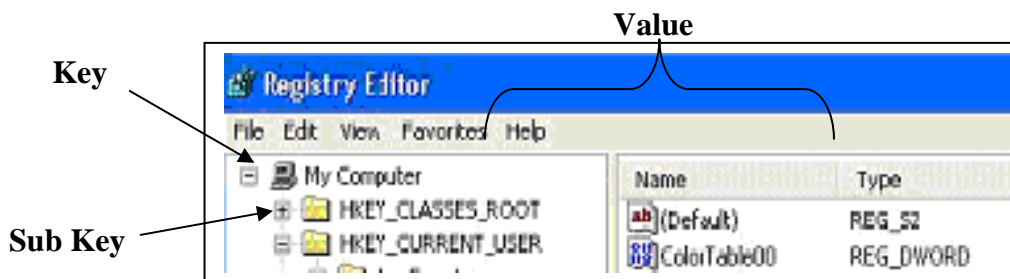
All of the information is stored in the hives, but the registry entries are displayed for review or editing in *handle keys* inside the registry editors (Fig.1). Handle keys are groupings of entries that are used to make finding and editing information easier (Clyton, 1997). There are six handle keys shown in (Table 1):

**Table 1: Handle Keys.**

Name	Abbreviation
HKEY_CLASSES_ROOT	HKCR
HKEY_CURRENT_USER	HKCU
HKEY_LOCAL-MACHINE	HKLM
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC
HKEY_DYN_DATA	HKDD

The keys is further divided into keys and sub keys creating a hierarchal structure. Each key has grouped information and is named based on the type of data in it.

Each key contains name and one or more values, each value has name, type, and data. Every key contains at least one value, and that's the default value. When user look at the registry through *registry editor*, the user see the default value as (Default) (Jerry, 2003) (fig. 2). Example of registry entry HKCU\Control Panel\Desktop\Wallpaper refers to a value called Wallpaper in the root key HKCU in a sub key of Control Panel called Desktop.



**Fig. 2. The Key Structures in the registry editor**

### Registry Utility Program

This utility program is dealing with the registry directly and it's ability to reach the values of the registry and updates these values safely that will help the user to avoid the risks of system failure when he want to change value. The registry of windows has too much values, the program takes the values which has no negative effects on the settings of windows system as much as possible, so the user of this program will feel safely when he changes some options in windows by using this program, also there is clarification with each value to explain the function of the chosen value before using it. Changing options and updating settings in the registry by this program will be easier than changing it by registry editor because the last needs continual exercise just to know the policy of the classification of the keys, but by this program the classification will be easier to understand and the user can recognize which key should be chosen to reach the required value. Below the structure of the classification policy of Registry utility program (Fig. 2).

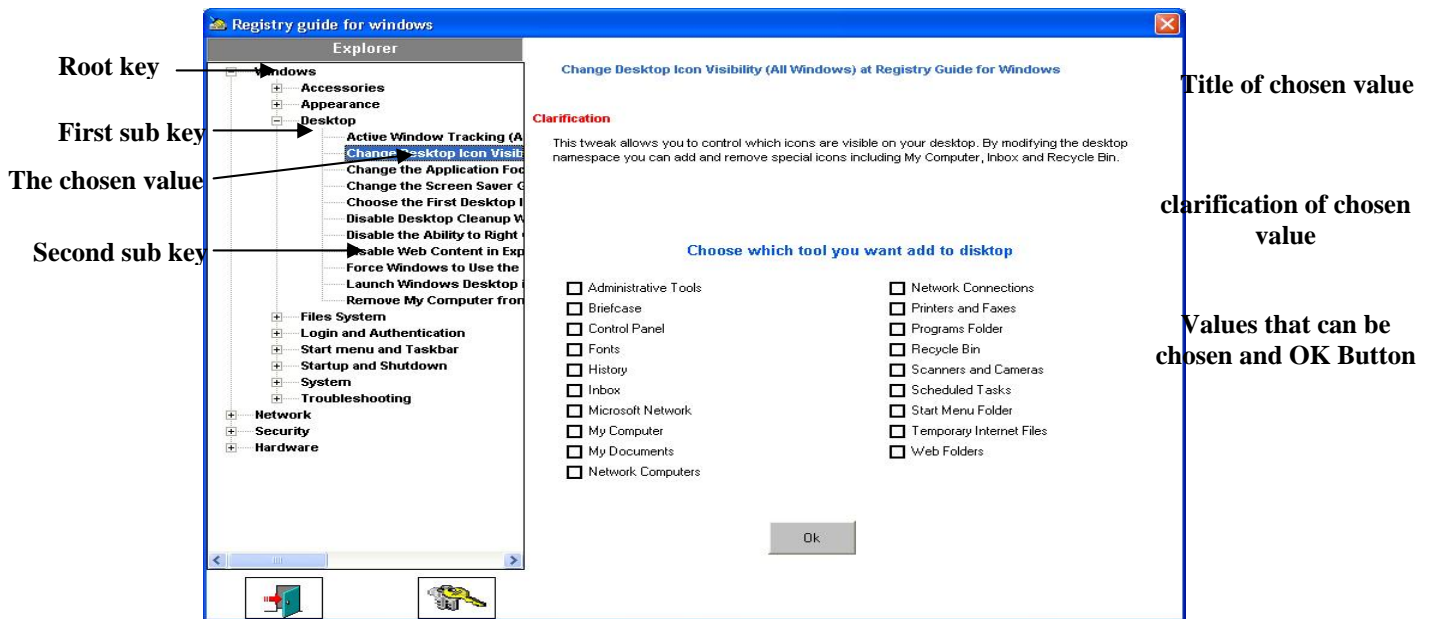


Fig. 3. The Registry Utility program Structure

- 1- Windows: Windows operating system settings enhancements.
- 2- Network: Network and connectivity settings enhancements.
- 3- Security: Security restrictions and system policies.
- 4- Hardware: Hardware and peripheral enhancements.

The above classification was for the root keys and each of them has sub keys and the last may has another sub keys and going on. The final key has many values which can change and update the values of windows registry, if the user selects the value from the sub-key then the title, the clarification and the options of the chosen value will be displayed on the board of the form, here the user can change or update the settings that explained by the below form (Fig. 3).

### **Steps of changing the values in windows registry**

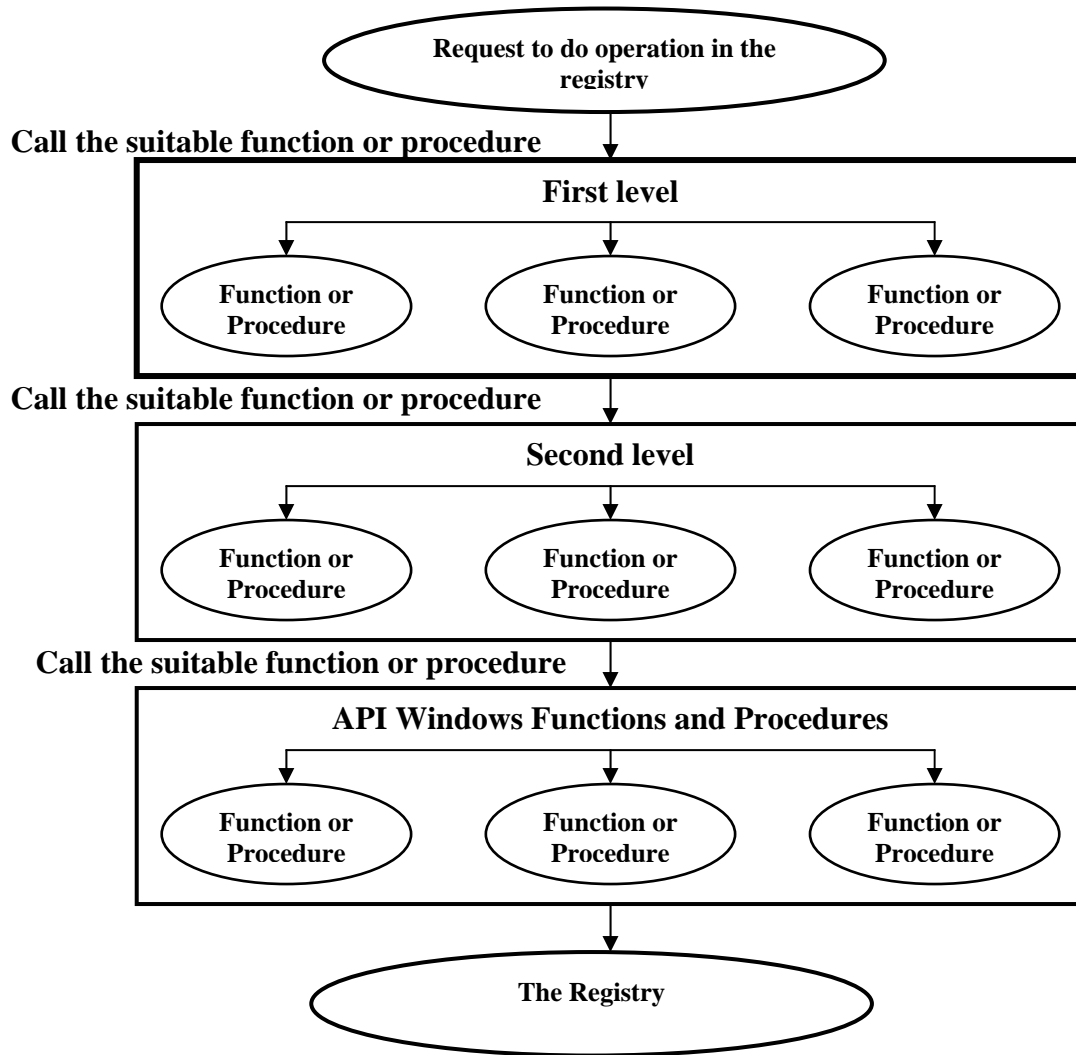
There are many steps in order to reach values of windows registry in this program.

Summary of these steps as following:-

- The definition of the basic Application Programming Interface (API) windows functions and procedures which associated with windows registry in visual basic 6 program.
- Using functions and procedures to call the defined API windows functions and procedures directly.
- Using second level of functions and procedures, their tasks are calling the functions and procedures which used to call the defined API windows functions. These functions will be called directly when user want to do any operation on the registry like write registry value and delete registry value....etc, therefore these functions and procedures have names refer to their jobs.

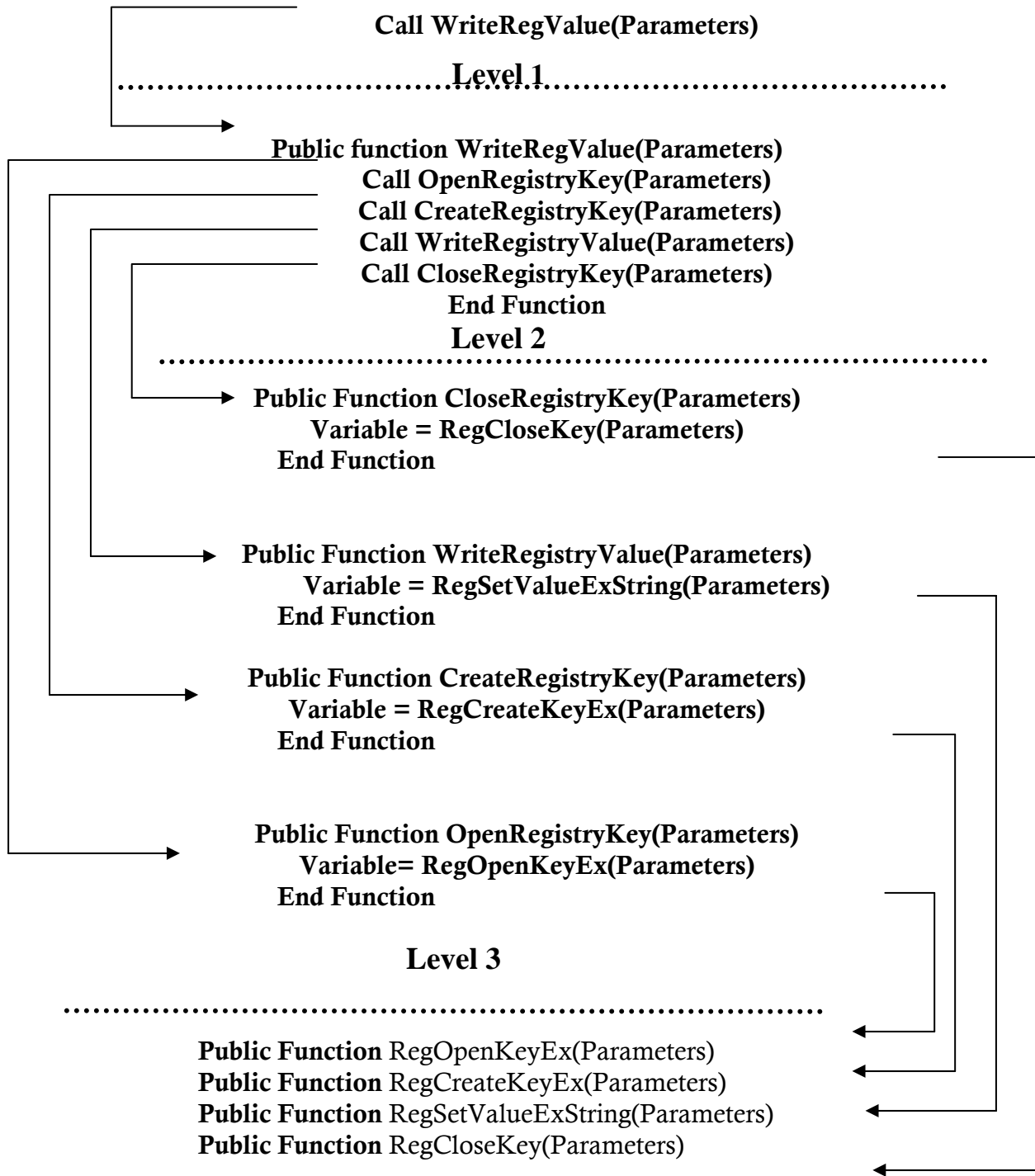
### **Easiness of program to call API Windows**

There are three levels of functions and procedures; the first level is the API windows functions and procedures which defined in module in the program to provide the ability to call them from any place in the program, the second level contains the procedures and functions which call the first level, but the third level is functions and procedures used for calling the second level and display a visual options to the users. Below is the structure of the program in order to reach and update the settings of the registry (Fig. 4):



**Fig. 4. Structure the work of utility program**

The above steps can be explained by the following example:



**The Practical Way of Accessing Registry**

The above steps can be explained by the following: if we want to change the registry settings this operation is done by the following steps:



1. open the registry program, then choose from the left panel which is called *Explorer* shown in ( fig. 3) and then choose from the tree structure the required option as follows (Windows\Accessories\Registry Editor\...) , then the options of required options are displayed, as shown in (fig. 5).

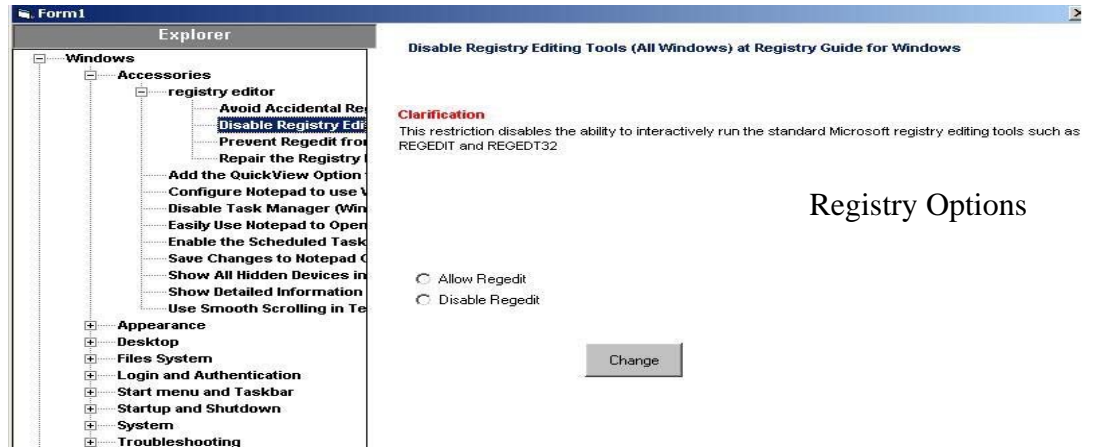


Fig. 5. Display Registry Options

Then the user can choose the appropriate option to disable or enable the required options.

2. The above operations are done by calling multiple procedures and modules as follows: when the user choose the node of tree, the node will be extended to show another nodes, reaching to the appropriate node and then the options of this node will be displayed, as shown in fig. 5. this operation is done by calling group of procedures to identify the root and its branch(es) the *TreeView1\_NodeClick* procedure as shown below:

```
Private Sub TreeView1_NodeClick(ByVal Node As MSComctlLib.Node)
    Dim path As String
    If Mid(Node.Key, 1, 5) = "value" Then
        Label2.Caption = Node
        Label3.Visible = True
        path = App.path & "\" & Node & ".txt"
        RichTextBox1.LoadFile path, 1
        Call Which_Root(Node.root, Node)
    End If
End Sub
```

Then the *TreeView1\_NodeClick( )* call *Which\_Root( )* procedure to identify the root & node and to call procedures that display the options of this selection as shown.

```
Private Sub Which_Root(ByVal root, ByVal value As MSComctlLib.Node)
    Select Case root
        Case "Windows"
            Call Windows_Root(value)
```

```

    Case "Security"
        Call Security_Root(value)
    Case "Network"
        Call Network_Root(value)
    Case "Hardware"
        Call Hrdware_Root(value)
    End Select
End Sub

```

After selecting the appropriate option from the tree the options of this selection are shown and as explained in the contents of `Windows_Root(value)` procedure as follows:

```
Private Sub Windows_Root(ByVal value As MSComctlLib.Node)
```

```
    Dim op_max() As String
```

```
    Dim options As Integer
```

```
    Dim x As String
```

```
    Dim i As Integer
```

```
    Dim v, f1 As Variant
```

```
    Select Case value
```

```
        Case "Avoid Accidental Registry Imports with Regedit (All Windows) at Registry Guide for Windows"
```

```
            ReDim op_max(1)
```

```
            op_max(0) = "Default"
```

```
            op_max(1) = "Just edit by notpad"
```

```
            options = 2
```

```
            Call Create_Options(options, op_max, "options")
```

```
        Case "Disable Registry Editing Tools (All Windows) at Registry Guide for Windows"
```

```
            ReDim op_max(1)
```

```
            op_max(0) = "Allow Regedit"
```

```
            op_max(1) = "Disable Regedit"
```

```
            options = 2
```

```
            Call Create_Options(options, op_max, "options")
```

```
        Case "Prevent Regedit from Saving the Last Location (Windows 2000-XP) at Registry Guide for Windows"
```

```
            ReDim op_max(1)
```

```
            op_max(0) = "Allow saving the last location"
```

```
            op_max(1) = "Don't Save the last location"
```

```
            options = 2
```

```
            Call Create_Options(options, op_max, "options")
```

```
        Case "Repair the Registry Editor (Windows 2000-Me-XP) at Registry Guide for Windows"
```

```
            ReDim op_max(1)
```

```
            op_max(0) = "Repair"
```

```
            op_max(1) = "Restor"
```

```
            options = 2
```

```
            Call Create_Options(options, op_max, "options")
```

```
        Case "Add the QuickView Option to All Files (All Windows) at Registry Guide for
```

```
            And so on for each select
```

```

.
.
.
End Select
End Sub

```

After displaying the options of a selection the user can choose from these options and then depending on this options calling Handle\_reg\_root1 procedure to manipulate the options and calling the functions that deal with registry as follows:

```

Private Sub Handle_reg_root1(ByVal Index As Integer)
    Dim x As String
    Dim v, f1 As Variant
    Dim create As Long
    Dim u As Long
    Select Case Index
    Case 1
        If Option1(0).value = True Then
            x = WriteRegValue(HKEY_CLASSES_ROOT, "regfile\shell", "", "")
        ElseIf Option1(1).value = True Then
            x = WriteRegValue(HKEY_CLASSES_ROOT, "regfile\shell", "", "edit")
        End If
        MsgBox x
    Case 2
        If Option1(0).value = True Then
            x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System",
"DisableRegistryTools", 0)
        ElseIf Option1(1).value = True Then
            x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System",
"DisableRegistryTools", 1)
        End If
        MsgBox x
    Case 3
        v=GetRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "LastKey")
        If Option1(0).value = True Then
            x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "LastKey", v)
        ElseIf Option1(1).value = True Then
            x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "LastKey", "")
        End If
        MsgBox x
    Case 4
        If Option1(0).value = True Then

```

```

    Call DeleteValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "LastKey")
    Call DeleteValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "FindFlags")
    v=GetRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "View")
    Call DeleteValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "View")
    ElseIf Option1(1).value = True Then
    x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "LastKey", "")
    x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "FindFlags", 2)
    x=WriteRegValue(HKEY_CURRENT_USER,
"SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Regedit", "View", v)
    End If
    .
    .
    .
    Case 44
    x = WriteRegValue(HKEY_CURRENT_USER, ".DEFAULT\Control Panel\Desktop",
"Wallpaper", Text1(0).Text)
    MsgBox x
    End Select
End Sub

```

After that a number of functions will be called to handle the registry like WriteRegValue, that calling the registry API functions like OpenRegistryKey, GetRegistryKey, WriteRegistryKey, DeleteRegistryKey, ... that exists in the module which is contains the declaration of API function and its variables to handles the right access to the registry and checking and changing the values (the registry entries, keys& new values) this module shown as follow:

```

' Declare Windows API functions...
Public Declare Function RegCloseKey Lib "advapi32.dll" (ByVal hKey As Long) As Long
Public Declare Function RegCreateKeyEx Lib "advapi32.dll" 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, _
    ByRef lpSecurityAttributes As SECURITY_ATTRIBUTES, ByRef
phkResult As Long, _
    ByRef lpdwDisposition As Long) As Long
Public Declare Function RegDeleteKey Lib "advapi32.dll" Alias "RegDeleteKeyA" (ByVal
hKey As Long, _
    ByVal lpSubKey As String) As Long
lpcbValueName As Long, _

```

```

        ByVal lpReserved As Long, ByVal lpType As Long, ByVal lpData As Any,
    -
        ByVal lpcbData As Long) As Long
Public Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA"
(ByVal hKey As Long, _
    ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired
As Long, _
    ByVal phkResult As Long) As Long
lpcbMaxClassLen As Long, _
    ByVal lpValues As Long, ByVal lpcbMaxValueNameLen As Long, ByVal
lpcbMaxValueLen As Long, _
    ByVal lpSecurityDescriptor As Long, ByVal lpftLastWriteTime As
FILE_TIME) As Long
Public Declare Function RegSetValueExString Lib "advapi32.dll" Alias
"RegSetValueExA" (ByVal hKey As Long, _
    ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType
As Long, _
    ByVal lpValue As String, ByVal cbData As Long) As Long
Public Declare Function RegSetValueExBoolean Lib "advapi32" Alias "RegSetValueExA"
(ByVal hKey As Long, _
    ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType
As Long, _
    ByVal lpData As Boolean, ByVal cbData As Long) As Long
Public Declare Function RegSetValueExLong Lib "advapi32.dll" Alias "RegSetValueExA"
(ByVal hKey As Long, _
    ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType
As Long, _
    ByVal lpValue As Long, ByVal cbData As Long) As Long
Public Declare Function RegDeleteValue Lib "advapi32.dll" Alias "RegDeleteValueA"
(ByVal hKey As Long, ByVal lpValueName As String) As Long

' Declare Windows API constants...
Public Const lngHKEY_CLASSES_ROOT = &H80000000
Public Const lngHKEY_CURRENT_USER = &H80000001
Public Const lngHKEY_LOCAL_MACHINE = &H80000002
Public Const lngHKEY_USERS = &H80000003

Public Const lngERROR_SUCCESS = 0&
Public Const lngERROR_FAILURE = 13&
Public Const lngUNREADABLE_NODE = 234&
Public Const lngNO_MORE_NODES = 259&
Public Const lngERROR_MORE_DATA = 234&

Public Const lngREG_OPTION_NON_VOLATILE = 0
Public Const lngSYNCHRONIZE = &H100000
Public Const lngSTANDARD_RIGHTS_READ = &H20000
Public Const lngKEY_QUERY_VALUE = &H1

```

```

Public Const lngKEY_ENUMERATE_SUB_KEYS = &H8
Public Const lngKEY_NOTIFY = &H10
Public Const lngKEY_SET_VALUE = &H2
Public Const lngKEY_CREATE_SUB_KEY = &H4
Public Const lngKEY_CREATE_LINK = &H20
Public Const lngSTANDARD_RIGHTS_ALL = &H1F0000
Public Const lngREG_SZ = 1
Public Const lngREG_BINARY = 3
Public Const lngREG_DWORD = 4

```

```

' Declare Windows API types...
Public Type FILE_TIME
    dwLowDateTime As Long
    dwHighDateTime As Long
End Type
Type SECURITY_ATTRIBUTES
    nLength As Long
    lpSecurityDescriptor As Long
    bInheritHandle As Boolean
End Type

```

### Conclusions

The design of the utility program shows that to access to the settings of windows registry safely and easily without any accidental errors. If there are any error happened when the user changed the setting then they can undo this change without any harm to the system. Provide good security for the user where the user can lock the registry then any user cannot reach the registry by using the original registry editor. Control on the most windows settings tools that provide more security and controls on these settings.

### References

- Abraham Silberschatz, Peter Galvin, Greg Gange, Applied Operating System Concepts, WindowsXP updates, First Edition, 2003.
- Clayton Johnson, Troubleshooting and Configuring the Windows NT/95 Registry, copyright Macmillan Computer publishing, 1997.
- Dean Andrews, Ty Belknap, Ed Bott , Special Edition Using Windows 98, Copyright Macmillan Computer Publishing.
- Jerry Honeycutt, Microsoft Windows XP Registry Guide, Copyright 2003 by Jerry Honeycutt, Library of Congress Cataloging-in-publication Data.
- <http://www.devarticles.com/c/a/Visual-Basic/Accessing-the-Windows-API-in-Visual-Basic/1>