SimpleRegistry Control 2.1

The SimpleRegistry Control allows you to access the Windows Registry in much more intuitive and simple way then the Win32 Registry API's. No more messing with handles, allocated buffers etc. The SimpleRegistry control exposes the building blocks of the registry (Keys, Sub Keys and Values) in a object oriented fashion. Keys, Sub keys and values are all treated as objects and each have their own properties and methods. Programming the registry from Visual Basic is now as simple as VB gets.

The control requires the presence of the Visual Basic run-time.

SimpleRegistry API

The SimpleRegistry API is based on an object oriented approach. The control exposes four objects:

CRegistry

Allows access to top level registry keys.

CRegistryKey

Allows access to registry keys.

CRegistryValue

Allows access to registry values.

CRemoteRegistry

Allows access to a registry of a remote machine.

In order to access keys and values of the registry the control exposes six properties corresponding to the top level hives of the registry. These are:
HKEY_CURRENT_CONFIG, HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_DYN_DATA, HKEY_LOCAL_MACHINE & HKEY_USERS.

Each one of these properties returns a CRegistryKey object which allows the developer to add, remove, edit or perform any other registry operation (see examples section for the various operations).

SimpleRegistry Examples

CRegistry Control

In order to access keys and values of the registry the control exposes six properties corresponding to the top level hives of the registry. These are:
HKEY_CURRENT_CONFIG, HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,
HKEY_DYN_DATA, HKEY_LOCAL_MACHINE & HKEY_USERS.

Each one of these properties returns a CRegistryKey object which allows the developer to add, remove, edit or perform any other registry operation (see examples section for the various operations).

Properties:
HKEY_CURRENT_CONFIG, HKEY_CLASSES_ROOT, HKEY_CURRENT_USER,
HKEY_DYN_DATA, HKEY_LOCAL_MACHINE, HKEY_USERS As CRegistryKey -
Read only. Returns a registry key object that represents a root key. This **CRegistryKey** object allows you to add, delete sub keys and values. (See **CRegistryKey** for a complete list of this objects properties and methods)

RemoteRegistry(RemoteMachine As String) As CRemoteRegistry - Read Only. Returns a remote
registry object based on a remote machine name or address.
(See **CRemoteRegistry** for a complete list of this objects' properties).

CRegistryKey

Properties:
DefaultValue As CRegistryValue - Read Only. Allows access the default value of a registry key. The property returns a **CRegistryValue** object allowing us to access the values name and data.

hKey As Long - Read Only. Returns the Win32 handle of a registry key. This handle can be used if we need to call one of the registry Win32 API's.

Key(KeyName As String) As CRegistryKey - Read Only. Returns a registry key object based on its name.

KeyByIndex(Index As Long) As CRegistryKey - Read Only. Returns a registry key object based on an index. All indexes are 1 based (Zero is not a valid index)

Name As String - Read Only. Returns the name of the registry key.

NoOfKeys As Long - Read Only. Returns the number of sub keys belonging to this registry key.

NoOfValues As Long - Read Only. Returns the number of values belonging to this registry key.

Options As EOptions - Set/Gets the mode of key creation. This property determines how sub keys will be created. If they will be volatile or non-volatile.

Value(ValueName As String) As CRegistryValue - Read Only. Returns a registry value object
based on a value name.

ValueByIndex(Index As Long) As CRegistryValue - Returns a registry value object based on a
value index. All indexes are 1 based (Zero is not a valid index)

Methods:
Function AddKey(KeyName As String) As CRegistryKey - Add a new key to the registry. A **CRegistryKey** object is returned if we need to further access this key.

AddValue(ValueName As String) As CRegistryValue - Add a new value to the registry key. **CRegistryValue** object is returned if we need to further access this value.

DeleteKey(KeyName As String) - Delete an existing registry key.

DeleteValue(ValueName As String) - Delete an existing registry value..

Sub Flush() - Flush the contents of a key to the registry. When keys are closed (The object goes out of scope the keys are flushed automatically).

Load(FileName As String, KeyName) As CRegistryKey - Load a registry backup file into a specific registry key.

SaveToFile(FileName As String) - Saves a registry key into a registry backup file.

CRegistryValue

Properties:

LongValue As Long - Gets or sets the value of a registry value. The type of the value should be
Long (REG_DWORD).

StringValue As String - Default Property. Gets or sets the value of a registry value.
The type of the value should be String (REG_SZ or REG_MULTI_SZ).

BinaryValue As Variant - Gets or sets the value of a registry value.
The type of the value should be binary (REG_BINARY). In order to set a binary value you need to pass a variant with an array of bytes.

e.g.:

```
With Registry1.HKEY_CLASSES_ROOT.AddKey("SimpleBinary")
Dim Data As Variant
Data = Array(CByte(1), CByte(2), CByte(3))

Dim InData As Variant

' Add New Binary Value
.AddValue("BinValue").BinaryValue = Data

' Get Binary Value
InData = .Value("BinValue").BinaryValue

End With
```

Name As String - Read only. Returns the name of the registry value.

ValueType As EValueType - Read only. Returns the type of the registry value. (REG_SZ, REG_DWORD etc.)

Exists As Boolean - Returns TRUE or FALSE indicating if the current value actually exists.

CRemoteRegistry

Properties:

HKEY_LOCAL_MACHINE, HKEY_USERS As CRegistryKey -
Returns a registry key object that
represents a root key on a remote machine. This **CRegistryKey** object allows you to add, delete
sub keys and values. (See **CRegistryKey** for a complete list of this objects properties and methods).
Access to the remote registry can be restricted according to the users access rights.

RemoteMachine - Read Only. Returns the remote machine name or address that we are connected to.

SimpleRegistry Examples

Take a look how easy it is to access the registry using the SimpleRegistry control. No need to mess with the Win32 API and wonder why VB is crashing on you. You don't need to remember to allocate buffers, close handles and figure out if ByVal should be used or not. The examples below demonstrate how simple and intuitive registry access can be if you use the SimpleRegistry Control.

How to Add a key to the registry ?

```
With Registry1.HKEY_CLASSES_ROOT
  .AddKey ("SimpleKey")
End With
```

Or

```
With Registry1.HKEY_CLASSES_ROOT
  Dim Key as CRegistryKey
  Set Key = .AddKey ("SimpleKey")
  Key.AddKey("SimpleSubKey")
End With
```

How to Delete a key from the registry ?

```
With Registry1.HKEY_CLASSES_ROOT
  .AddKey ("SimpleKey")
  .DeleteKey ("SimpleKey")
End With
```

How to Add a Value to an existing registry key ?

```
With Registry1.HKEY_CLASSES_ROOT
  Dim Key as CRegistryKey
  Set Key = .AddKey ("SimpleKey")

  Key.AddValue("SimpleStringValue").StringValue = "SimpleValue"
  Key.AddValue("SimpleLongValue").LongValue = 1998
End With
```

How to Set/Get a binary value ?

```
With Registry1.HKEY_CLASSES_ROOT.AddKey("SimpleBinary")
  Dim Data As Variant
  Data = Array(CByte(1), CByte(2), CByte(3))

  Dim InData As Variant

  ' Add New Binary Value
  .AddValue("BinValue").BinaryValue = Data

  ' Get Binary Value
  InData = .Value("BinValue").BinaryValue
End With
```

How to Delete a Value to from a registry key ?

```
With Registry1.HKEY_CLASSES_ROOT
  .AddValue("SimpleValue").StringValue = "SimpleValue"
  .DeleteValue ("SimpleValue")

End With
```

How to access the registry on a remote machine ?

```
' Machine1 is the name of the remote workstation
With Registry1.RemoteRegistry("Machine1").HKEY_LOCAL_MACHINE
  ' Access the default value of a remote key.
  MsgBox "Remote Value Is:" & .Key("RemoteKey").DefaultValue.StringValue
End With
```

Questions: support@4developers.com

SimpleRegistry Registration

You can register SimpleRegistry online at:
http://www.4developers.com/simreg/register.htm

Questions: support@4developers.com

For more information:
Take a look at the Order.txt file that comes with the software.