

# Tutorial

## [Access]

*Michele de Nittis*

---

### **ACCESSO AL REGISTRO DI SISTEMA: LETTURA DEL NOME DEL COMPUTER.**

---

Versione: 1

Data Versione: domenica 12 dicembre 2004

## Introduzione

*Questo tutorial nasce per un'esigenza pratica. Mi capita spesso di studiare soluzioni e di acquisire nuove esperienze e nozioni su un particolare argomento per poi dimenticarle se non le applico frequentemente.*

*Dunque questo testo è una sorta di mio appunto mnemonico, scritto, per quanto possibile, in modo strutturato e corretto per essere riletto anche a distanza di tempo senza particolari sforzi.*

*Ho pensato, poi, che questo mio lavoro sarebbe potuto risultare utile anche a coloro che sono in cerca di materiale tecnico e documentale per la risoluzione delle problematiche che io ho già affrontato e descritte nel presente documento.*

*Nelle seguenti sezioni affronterò il tema dell'invocazione delle API Win32 per l'accesso alle informazioni immagazzinate nel registro di sistema mediante un esempio pratico: **la lettura dell'identificatore o nome del computer.***

## Dove leggere il Nome del Computer?

Il nome del computer è immagazzinato come valore di una chiave del registro di sistema. Per accedere al registro di sistema tramite VBA è necessario sfruttare le API di Windows. Le API Win32 sono librerie che forniscono insiemi di funzioni per l'interazione con il sistema operativo. Di tutti questi insiemi ci interessa solo quello costituito da tutte le funzioni di accesso al registro di windows. Non illustreremo le modalità d'impiego di tutte queste funzioni ma solo di due in quanto, una volta capito il procedimento, risulterà semplice invocare tutte le altre. Come si avrà modo di constatare, le funzioni delle API di Windows sono dichiarate in modo *C-Like*. Per essere fruibili in VBA devono essere **ridichiarate** all'interno di un modulo. In questo tutorial verrà ricordato, e non spiegato nei dettagli, come dichiarare una funzione VBA in modo che invochi una funzione di una libreria di sistema o, più in generale, di una DLL.

Per acquisire il nome identificativo del PC si deve leggere il valore della chiave di registro:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName  
\ActiveComputerName**

oppure

## HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\ComputerName

### \ComputerName

Per leggere il valore di una chiave di un registro è necessario richiamare le seguenti due funzioni di sistema:

- RegOpenKeyEx();
- RegQueryValueEx().

## RegOpenKeyEx[A|W]<sup>1</sup>

```
LONG RegOpenKeyEx(  
    HKEY hKey,           // handle to open key  
    LPCTSTR lpSubKey,   // address of name of subkey to open  
    DWORD ulOptions,    // reserved  
    REGSAM samDesired,  // security access mask  
    PHKEY phkResult     // address of handle to open key  
);
```

Questa funzione consente di **aprire** una chiave del registro. Il significato dei parametri è illustrato nella seguente tabella:

<b>hkey</b>	HKEY (Long)	Riferimento (lett. <i>Handle</i> ) alla chiave a cui si vuol accedere. Windows ha sette valori di chiavi riservati, corrispondenti alle chiavi di primo livello nella struttura ad albero del registro: <ul style="list-style-type: none"><li>- HKEY_CLASSES_ROOT (&amp;H80000000)</li><li>- HKEY_CURRENT_CONFIG</li><li>- HKEY_CURRENT_USER (&amp;H80000001)</li><li>- HKEY_DYN_DATA (solo win 95)</li><li>- <b>HKEY_LOCAL_MACHINE</b> (&amp;H80000002)</li><li>- HKEY_PERFORMANCE_DATA (solo NT)</li><li>- HKEY_USERS (&amp;H80000003)</li></ul>
<b>LpSubKey</b>	LPCSTR	Indirizzo di una stringa che specifica il nome della sottochiave dalla quale leggere il valore;
<b>UIOptions</b>	DWORD	Riservato
<b>SamDesired</b>	REGSAM	Maschera a bit che indica il tipo di accesso alla chiave-sottochiave. I possibili valori del parametro possono essere combinati tra loro con l'operatore OR. I principali valori per questo parametro sono i seguenti: <ul style="list-style-type: none"><li>- KEY_ALL_ACCESS: tutti i tipi di accesso (&amp;H2003F);</li><li>- KEY_CREATE_LINK: accesso con autorizzazione a creare un link ;</li><li>- KEY_CREATE_SUB_KEY: accesso con permesso di creazione di una sottochiave (&amp;H4);</li><li>- KEY_ENUMERATE_SUB_KEYS: accesso con autorizzazione ad enumerare le sottochiavi (&amp;H8);</li><li>- KEY_EXECUTE: accesso in lettura;</li><li>- KEY_NOTIFY: accesso con possibilità di cambiare la notifica (&amp;H10);</li><li>- <b>KEY_QUERY_VALUE</b>: accesso con possibilità di interrogare una sottochiave (&amp;H1);</li><li>- KEY_SET_VALUE: accesso per impostare i valori di una sottochiave</li></ul>
<b>PhkResult</b>	PHKEY	Puntatore all'Handle della chiave aperta.

<sup>1</sup> Alcune DLL a 32 bit includono funzioni con versioni leggermente diverse in modo che vengano riconosciute sia stringhe Unicode che ANSI. Se il nome di una funzione termina con una "A", si tratta della versione ANSI, se invece termina con una "W", si tratta della versione Unicode.

La funzione restituisce un valore intero: 0 (ERROR\_SUCCESS) se non si sono verificati problemi, altrimenti un valore positivo che descrive il problema riscontrato.

Si deve, dunque, **definire** la funzione VBA che apre la chiave del registro desiderata impiegando la funzione RegOpenKeyEx(). A tal proposito ricordiamo sinteticamente che per accedere ad una funzione di una DLL esterna è necessaria una particolare dichiarazione VBA che effettui una *mappatura* dei tipi di dato nativi di VBA con quelli:

- a) dei parametri formali attesi dalla funzione di sistema;
- b) del tipo di dato restituito;
- c) eventualmente del nome della funzione stessa.

La sintassi per dichiarare in VBA una funzione che accede ad una funzione di sistema o di una libreria DLL è la seguente:

**[Public | Private] Declare Function nome Lib "nomelibreria" [Alias "nomealias"]**  
**[[([elencoargomenti]])] [As tipo],**

dove *nome* è il nome della funzione VBA, *nomelibreria* è il nome della libreria che contiene la funzione da invocare, *nomeAlias* è il nome della funzione della Libreria di sistema/DLL da invocare ed *elencoargomenti* è, ovviamente, l'elenco degli argomenti da passera alla funzione.

Nell'area delle dichiarazioni delle variabili, dunque, dichiariamo la funzione della libreria *Win32* come segue:

```
Public Declare Function MyRegOpenKeyExA Lib "advapi32.dll" Alias  
"RegOpenKeyExA" (ByVal hkey As Long, ByVal lpSubkey As String, ByVal ulOptions  
As Long, ByVal samDesired As Long, ByRef phkresult As Long) As Long
```

Si osservi che il parametro *phkresult* viene passato alla funzione **ByRef** perché deve poter essere modificato dalla funzione stessa. Al termine dell'esecuzione della funzione, infatti, *phkresult* conterrà l'*handle* della chiave aperta.

Applichiamo, dunque, la funzione **RegOpenKeyEx** (ovvero MyRegOpenKeyExA) per aprire la chiave di registro necessaria per leggere il nome del computer.

```
Public Function ApriChiaveNomeComputer() As Long  
Dim MyName As String  
Dim RegPath As String  
Dim l_MyName As Long  
Dim risultato As Long  
Dim MyHandle As Long  
RegPath = "SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName"  
risultato = MyRegOpenKeyExA(HKEY_LOCAL_MACHINE, RegPath, 0, KEY_QUERY_VALUE, MyHandle)  
If risultato = 0 Then  
    ApriChiaveNomeComputer = MyHandle  
Else  
    MsgBox "Errore n°: " & risultato  
    ApriChiaveNomeComputer = 0  
End If  
End Function
```

Una volta aperta la chiave di registro specificata ed ottenuto il relativo handle, è possibile invocare la funzione che legge il valore di una sua sottochiave.

## RegQueryValueEx[A|W]<sup>2</sup>

```
LONG RegQueryValueEx(
    HKEY hKey,           // handle to key to query
    LPTSTR lpValueName, // address of name of value to query
    LPDWORD lpReserved, // reserved
    LPDWORD lpType,     // address of buffer for value type
    LPBYTE lpData,      // address of data buffer
    LPDWORD lpcbData    // address of data buffer size
);
```

Se la funzione viene eseguita con successo restituisce il valore ERROR\_SUCCESS (0), altrimenti un numero intero positivo che rappresenta il tipo di errore verificatosi.

Esaminiamo nel dettaglio i sei parametri:

<b>hkey</b>	HKEY	Riferimento alla chiave aperta.
<b>lpValueName</b>	LPTSTR	Stringa che specifica il nome della sottochiave dalla quale prelevare il valore;
<b>lpReserved</b>	LPDWORD	Riservato: deve essere NULL;
<b>lpType</b>	LPDWORD	Puntatore ad una variabile che riceve il <b>tipo</b> di valore della sottochiave. Alcuni tra i valori possibili sono: <ul style="list-style-type: none"> <li>- REG_BINARY : generico dato binario;</li> <li>- REG_DWORD (4): numero a 32 bit;</li> <li>- REG_SZ (1): Stringa terminata dal carattere NULL.</li> </ul>
<b>lpData</b>	LPBYTE	Puntatore ad un buffer di memoria di dimensione sufficiente a ricevere il valore della sottochiave;
<b>lpcbData</b>	LPDWORD	Puntatore ad una variabile che contiene la dimensione in byte del buffer puntato dal parametro <i>lpData</i> . Dopo l'esecuzione della funzione il valore del parametro viene aggiornato con l'effettiva lunghezza in byte del valore letto dalla sottochiave.

Dichiariamo, quindi, la funzione della libreria nell'area delle dichiarazioni di un modulo:

```
Public Declare Function MyRegQueryValueExA Lib "advapi32.dll" Alias
"RegQueryValueExA" (ByVal hkey As Long, ByVal lpValueName As String, ByVal
lpReserved as Long, ByVal lpType as Long, ByVal lpData As String, lpcbData As
Long) As Long
```

Il codice VBA della funzione che deve leggere il nome del computer, mediante l'invocazione della funzione di sistema **RegQueryValueEx** (ovvero MyRegQueryValueExA), è la seguente:

<sup>2</sup> Vedi la precedente nota 1.

```

Public Sub mostraNomeComputer()
Dim MyName As String
Dim RegPath As String
Dim l_MyName As Long
Dim KeyType As Long
Dim risultato As Long
l_MyName = 255
MyName = String(l_MyName, vbNullChar)
RegPath = "ComputerName"
risultato = ApriChiaveNomeComputer()
If risultato > 0 Then
    risultato = MyRegQueryValueExA(risultato, RegPath, 0, KeyType, MyName, l_MyName)
    If risultato = 0 Then
        MsgBox "Nome Computer: " & MyName
    Else
        MsgBox "Errore nella lettura del nome del computer"
    End If
Else
    MsgBox "Problemi nell'apertura della chiave del registro."
End If
End Sub

```

La stringa **MyName** viene modificata dalla funzione in modo da contenere il valore della sottochiave letta e quindi deve essere vista come *un buffer di un numero di byte sufficiente a contenere il risultato aspettato*.

Dunque **MyName** può essere **definita** con la funzione **String(numByte, vbNullChar)**

## Conclusione

L'esposizione del presente *tutorial*, sebbene interessi un caso molto particolare, evidenzia l'uso delle funzioni della libreria Win32 per accedere alle chiavi del registro di windows da VBA. Come già detto la metodologia illustrata può essere applicata a tutte le altre funzioni della libreria. Le fonti di questo *tutorial* sono state:

- Il sito [www.sitocomune.com](http://www.sitocomune.com);
- La documentazione MSDN Library a corredo di Visual Studio 6.0;
- Un tutorial sul registro di configurazione di Windows ([www.alground.com](http://www.alground.com));

Ringrazio in anticipo tutti coloro che vorranno contribuire a migliorare questi appunti con suggerimenti, correzioni e segnalazioni di imprecisione contattandomi dal mio sito personale ( [www.mdenittis.it](http://www.mdenittis.it) ).