

# الريجستري

## Registry Functions



عالم الكمبيوتر

## الفصل الأول

# أسماء الدوال والسجلات و الثوابت المستخدمة للتعامل مع الـ Registry

لقد قمت بتضمين كل الدوال و السجلات و الثوابت في Modules باسم RegistryDef.bas فقط قم بوضعه في الـ Project ثم قم باستدعاء الدوال مباشرة.

### الدوال المستخدمة :

RegCloseKey	RegConnectRegistry
RegCreateKey	RegCreateKeyEx
RegDeleteKey	RegDeleteValue
RegEnumValue	RegEnumKey
RegEnumKeyEx	RegFlushKey
RegGetKeySecurity	RegLoadKey
RegNotifyChangKeyValue	RegOpenKey
RegQueryInfoKey	RegQueryMultipleValues
RegQueryValues	RegQueryValueEx
RegReplaceKey	RegRestoreKey
RegSaveKey	RegSetKeySecurity
RegSetValue	RegSetValueEx
RegUnloadKey	

### السجلات المستخدمة:

SECURITY_ATTRIBUTES	FILETIME
SECURITY_DESCRIPTOR	ACL

### الثوابت المستخدمة:

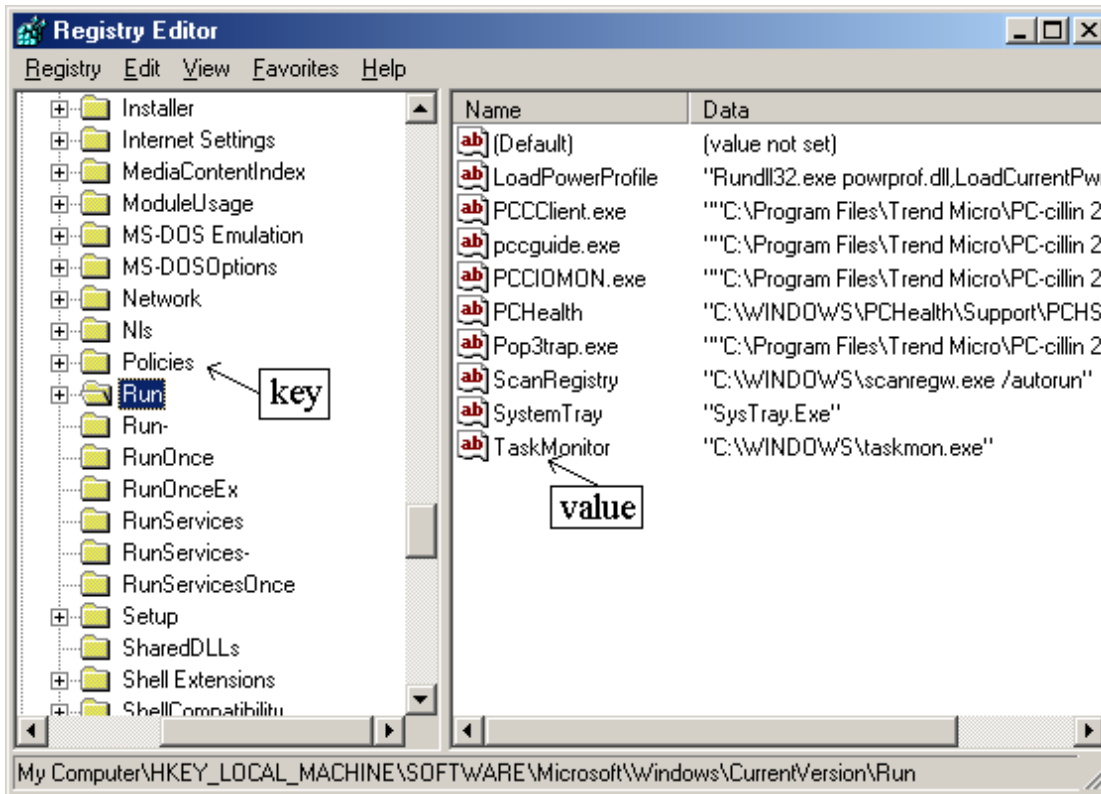
HKEY_CLASSES_ROOT	HKEY_CURRENT_CONFIG
HKEY_CURRENT_USER	HKEY_DYN_DATA
HKEY_LOCAL_MACHINE	HKEY_PERFORMANCE_DATA
HKEY_USERS	ERROR_SUCCESS
ERROR_INSUFFICIENT_BUFFER	MAX_PATH
READ_CONTROL	KEY_SET_VALUE
KEY_QUERY_VALUE	KEY_CREATE_SUB_KEY
KEY_CREATE_LINK	KEY_ENUMERATE_SUB_KEYS
KEY_EVENT	KEY_NOTIFY
SYNCHRONIZE	STANDARD_RIGHTS_ALL
STANDARD_RIGHTS_WRITE	STANDARD_RIGHTS_READ
KEY_READ	KEY_WRITE
KEY_ALL_ACCESS	REG_BINARY
REG_CREATED_NEW_KEY	REG_DWORD
REG_DWORD_BIG_ENDIAN	REG_DWORD_LITTLE_ENDIAN
REG_EXPAND_SZ	REG_NOTIFY_CHANGE_NAME
REG_LINK	REG_MULTI_SZ
REG_NONE	REG_NOTIFY_CHANGE_LAST_SET
REG_NOTIFY_CHANGE_SECURITY	REG_OPENED_EXISTING_KEY
REG_OPTION_BACKUP_RESTORE	REG_OPTION_CREATE_LINK
REG_OPTION_NON_VOLATILE	REG_OPTION_RESERVED
REG_OPTION_VOLATILE	REG_REFRESH_HIVE
REG_RESOURCE_LIST	REG_SZ
REG_WHOLE_HIVE_VOLATILE	REG_LEGAL_CHANGE_FILTER
REG_LEGAL_OPTION	REG_FULL_RESOURCE_DESCRIPTOR
REG_NOTIFY_CHANGE_ATTRIBUTES	REG_RESOURCE_REQUIREMENTS_LIST

## الفصل الثاني ( فكرة عامة عن كيفية استخدام هذه الدوال )

تتكون الـ Registry طبعاً من Keys و Values .  
أما الـ Keys:

فهي تحتوي على SubKeys وهي عبارة عن Keys  
وتحتوي على Values

و بفتح الـ Registry باستخدام البرنامج regedit.exe تظهر لنا الـ keys في الناحية اليسرى من الشاشة و الـ Values في الناحية اليمنى.  
وهذه صورة توضح ذلك:



يمكن إنشاء أو تغيير الـ keys و الـ Values يدوياً بسهولة ولكن كيف يمكن فعل كل ذلك بلغة Visual Basic ؟

قبل البدء يجب معرفة بعض الأساسيات للتعامل مع الـ Registry .

1. يجب فتح الـ Key المراد وضع أو تغيير الـ values الموجودة بداخله
2. تغيير القيم الموجودة داخله.
3. إغلاق الـ Key الذي تم فتحه. وهذه النقطة مهمة جداً لأنه من دونها لا يتم كتابة البيانات في الـ Registry .

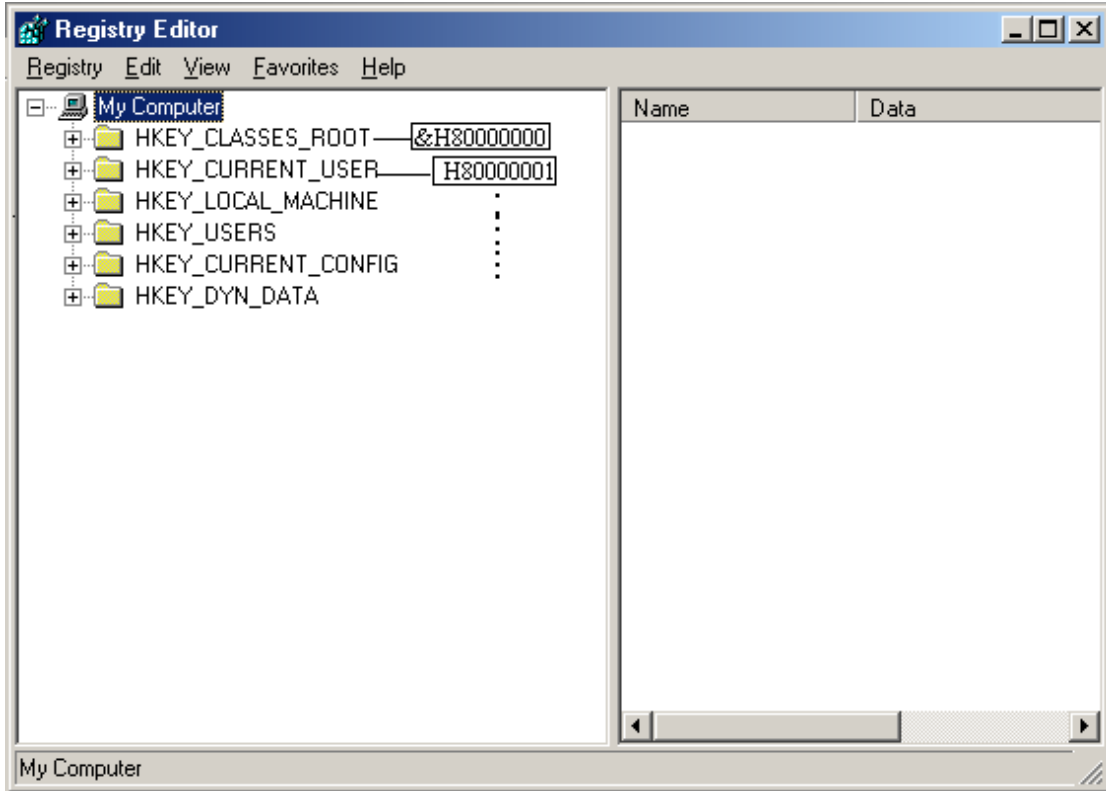
**ملاحظة:** يمكن كتابة البيانات في الـ Registry بدون إغلاق الـ Key وذلك باستعمال الدالة RegFlushKey ولكن هذه الدالة تستخدم مصادر النظام بشكل كبير حيث يراعى استخدامها إلا في حالات الضرورة القصوى.

## شرح النقاط السابقة:

١. فتح الـ key.  
يجب قبل فتح key معرفة handle لـ key آخر يجب أن يكون مفتوح مسبقاً ويجب أن يكون الـ key المراد فتحه عبارة عن subkey للـ key الذي يوجد لدينا الـ handle الخاص به.  
تتساءل الآن كيف يمكنني معرف الـ handle الخاص بالـ key الأب؟  
إجابة هذا السؤال عندي والله الحمد وهي بسيطة.  
يعطي نظام التشغيل Windows أرقام ثابتة للمفاتيح الرئيسية كلما تم تشغيل النظام وهذه الـ keys مع الـ handles الخاصة بها:

```
Const HKEY_CLASSES_ROOT = &H80000000
Const HKEY_CURRENT_USER = &H80000001
Const HKEY_LOCAL_MACHINE = &H80000002
Const HKEY_USERS = &H80000003
Const HKEY_CURRENT_CONFIG = &H80000005
Const HKEY_DYN_DATA = &H80000006
Const HKEY_PERFORMANCE_DATA = &H80000004
```

وهذه صورة توضح السابق:



وتسمى الـ keys السابقة بـ predefined keys  
والدوال المستخدمة لفتح الـ keys هي :

RegOpenKey, RegOpenKeyEx

RegCreateKey, RegCreateKeyEx

كتابة وحذف البيانات:

يمكن استخدام الدوال التالية:

RegSetValue, RegSetValueEx وذلك لتخصيص بيانات (Data) لـ key معين.



RegSetValue: تتعامل مع النصوص فقط، وسأتي شرحها في حينها.  
RegSetValueEx: يمكنها كتابة أي نوع من البيانات وهذه الدالة بإمكانها إنشاء key و value خاص به في نفس الوقت!!!

ولحذف Value من key تستخدم الدالة RegDeleteValue  
ولحذف key نستخدم الدالة RegDeleteKey مع ملاحظة أن الـ key المحذوف لا يتم إزالته حتى يتم قفل آخر handle له.

ولتغيير أمان الـ keys نستخدم RegSetKeySecurity

يمكن جلب الـ subkeys لـ key معين حتى يتم إيجاد key معين وأخذ البيانات منه وذلك باستخدام الدالة:  
RegEnumKey أو الدالة RegEnumKeyEx أما الأولى ترجع الـ subkeys فقط والثانية ترجع الـ subkeys مع الـ Classes

لإرجاع بيانات مفصلة حول subkey معينة، البرنامج يمكن أن تستدعي الدالة RegQueryInfoKey  
والدالة RegGetKeySecurity ترجع نسخة من الـ SecurityDescription التي تحمي الـ key ولجلب الـ values لـ key تستخدم الدالة RegEnumValue  
إغلاق الـ key يتم باستخدام الدالة RegCloseKey . ٣

كل ما عليك فهمه من النقاط السابقة ( إن لم تفهم شيئاً منه ) هو التالي:  
يجب فتح الـ key ثم تغيير فعل ماتريد عليه ثم بعد ذلك إغلاقه .

يوجد برنامج في المجلد المسمى Examp 01 فيه بعض الدوال السابقة.

لقد قمت بشرح عام على بعض الدوال وإليك بشرح مفصل.

## الفصل الثالث شرح بعض الدوال بالتفصيل

### \*\*الباب الأول: التعامل مع الـ Keys\*\*

#### الدالة الأولى: RegCloseKey

هي دالة تقوم بإزالة الحجز عن key تم فتحه وتعريفها بهذا الشكل:

تستقبل هذه الدالة معامل واحد فقط وهو:

hKey : هو الـ handle لـ key تم فتحه

إذا نجحت هذه الدالة فإنها سترجع القيمة ERROR\_SUCCESS والتي تساوي 0 ، أما إذا أخفقت فإنها ترجع قيمة غير الصفر.

مع ملاحظة أنه لا يمكن استعمال الـ handle الذي تم قفله إلا إذا تم فتح الـ key من جديد مع ملاحظة أن الـ handle سيتغير.

وعليك أن لا تترك الـ key مفتوح أكثر من الوقت الذي تستعمله فيه.

مع ملاحظة أنه لا يتم تخزين البيانات التي تم إضافتها إلى الـ Registry فعلياً إلا عندما يتم قفل الـ Key.

وعند كتابة بيانات كبيرة جداً في الـ Registry وتنفيذ هذه الدالة قد تأخذ بضع ثواني في تخزين البيانات في الـ Registry.

#### الدالة الثانية: RegCreateKey

هذه الدالة تقوم بإنشاء key محدد أما إذا كان موجوداً فإنها تقوم بفتحه.

هذه الدالة متوافقة مع نظام Windows 3.1. أما البرامج المعتمدة على win32 فمن الأفضل استخدام الدالة RegCreateKeyEx والتي سأقوم بشرحها بعد هذه الدالة.

هذه الدالة تقوم باستقبال ثلاث معاملات : hKey, lpSubKey, phkResult

hKey : من نوع Long

يشير إلى الـ handle لـ key مفتوح حالياً، أو أي من الـ Predefinedkeys (راجع الفصل الثاني لمعرفة

ماهي الـ PredefinedKeys )

والـ key المنشأ هو الـ subkey لـ key المعروف بواسطة الـ hKey.

lpSubKey : من نوع String

يحدد اسم الـ Key المراد إنشائه أو فتحه وهذا الـ Key يجب أن يكون مفتاح فرعي من الـ Key المحدد في الـ hKey.

ويمكن أن يكون اسم واحد أو مسار بهذا الشكل:

"SOFTWARE"

أو

"\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"



Long من نوع phkResult  
يؤشر إلى متغير يستقبل الـ Handle الخاص بالـ Key التي تم فتحها.

القيمة المرجعة:  
ترجع هذه الدالة القيمة  
ملاحظة:  
ERROR\_SUCCESS إذا نجحت.

يمكن إنشاء مجموعة من الـ Keys المتداخلة.  
ومثال على ذلك يوجد في المجلد المسمى Examp 02

## الدالة الثالثة: RegCreateKeyEx

هذه الدالة تشبه الدالة السابقة من حيث أنها تقوم بإنشاء Key أو تفتحه إن كان موجوداً وتستقبل هذه الدالة تسع معاملات :

hKey, lpSubKey, vReserved, lpClass, dwOption, samDesired, lpSecurityAttributes, phkResult, lpdwDisposition

hKey: من النوع Long  
قيمة الـ Handle لمفتاح مفتوح حالياً ( راجع الدالة السابقة )

lpSubKey: من النوع String  
يشير إلى متغير نصي يحمل اسم الـ subkey الذي يجب أن يكون متفرع من المفتاح المشار إليه بـ hKey.  
مع ملاحظة أن النص يجب أن لا يبدأ بـ "\" وهذا المعامل لا يمكن أن يكون فارغاً.

Reserved: من النوع Long  
هذا المعامل محجوز ويجب أن يكون صفر.

lpClass: من النوع String  
متغير نصي يحوي نوع الـ Object لهذا الـ key  
أما إذا كان الـ key موجود فإن هذا المعامل يتجاهل.  
لن أشرح الـ Classes الآن لما يحتاجه من وقت وجهد.

dwOption: من النوع Long  
ملاحظة: هذه الخاصية موجودة في Windows NT و Windows 2000  
أما في Windows 9x فإنه يتم تجاهلها.

هذا المعامل يحدد خيارات خاصة بالـ Key.  
وهذا يجب أن يكون واحداً من القيم التالية:  
REG\_OPTION\_NON\_VOLATILE or REG\_OPTION\_VOLATILE  
REG\_OPTION\_NON\_VOLATILE

المفتاح المنشأ باستخدام هذا الثابت ليس من السهولة ضياع البيانات الموجودة به وهذا هو الخيار الافتراضي. حيث أن المعلومات تخزن في ملف وتحمى أو تحفظ عندما يعاد تشغيل النظام.

### REG\_OPTION\_VOLATILE

الـ Key المنشأ باستخدام هذا الثابت تخزن في الذاكرة وغير محفوظة حفظ جيد وهذا الثابت يتم تجاهله إذا كان الـ key موجود.

### REG\_OPTION\_BACKUP\_RESTORE

إذا كانت هذا الثابت موجود فإن الدالة تتجاهل المعامل samDesired وتعطي صلاحية فتح الـ key على أساس Restore, Backup

Long من نوع samDesired

يبين كيفية أو نوع الأمن المراد استخدامه للـ key المراد فتحه.

وهذا المعامل يمكن أن يكون مزيجاً من القيم التالية:

KEY_ALL_ACCESS	يمكنك القيام بأي شيء ممكن
KEY_CREATE_LINK	صلاحية إنشاء ربط رمزي
KEY_CREATE_SUB_KEY	إمكانية إنشاء مفاتيح فرعية
KEY_ENUMERATE_SUB_KEYS	صلاحية عد ومعرفة أسماء المفاتيح الفرعية
KEY_EXECUTE	صلاحية قراءة وتنفيذ
KEY_NOTIFY	صلاحية الإعلام بالتغيير
KEY_READ	خليط من مجموعة من الصلاحيات
KEY_SET_VALUE	صلاحية تغيير بيانات المفاتيح الفرعية
KEY_WRITE	خليط من مجموعة من الصلاحيات

IpSecurityAttributes من نوع التركيبة : SECURITY\_ATTRIBUTES

هذا المعامل يحدد هل الـ handle المرجع يمكن أن يكون موروثاً بواسطة المعالجة الابن (ولن أفضل هذه النقطة لما يتطلبه من وقت وجهد).

أما إذا كان هذا المعامل NULL فإن الـ handle لا يمكن أن يورث لمعالجة ابن. وسأشرح هذه التركيبة بعد قليل.

تحت نظام التشغيل Windows NT :

العنصر IpSecurityDescriptor في هذه التركيبة يوضح صفة الأمان للمفتاح الجديد وإن كان NULL فإن الأمن يأخذ الـ Defaults

تحت نظام التشغيل Windows 9x :

العنصر IpSecurityDescriptor يتم تجاهله.

Long من النوع phkResult

تقوم هذه الدالة بوضع الـ handle للـ key المفتوح أو المنشأ ليتم إستخدامه من قبل المبرمج.

Long lpdwDesposition : من النوع:

تقوم الدالة بوضع إحدى القيم التالية:

REG\_CREATED\_NEW\_KEY

REG\_OPENED\_EXISTING\_KEY

أما الأول فيدل على أن المفتاح ليس موجود وتم إنشائه، أما الثاني فإنه يدل على أن المفتاح موجود ولكن تم فتحه فقط بدون أي تغيير.

يمكن الاستفادة من هذه القيمة لمعرفة هل هذه هي أول مرة تم فيها فتح البرنامج أم لا؟

القيمة المرجعة من الدالة:

ERROR\_SUCCESS

إذا نجحت الدالة فإنها ترجع القيمة

يوجد مثال على هذه الدالة في المجلد المسمى Examp 03 ولكن يجب أن تدرس التركيبية التالية حتى

تتمكن من فهم هذه الدالة جيداً

ملاحظة:

عند إنشاء أو فتح الـ Keys استخدم

RegOpenKeyEx أو RegCreateKeyEx

ولتستخدم

RegOpenKey و RegCreateKey

وذلك حتى تتمكن من استخدام بعض الدوال الأخرى.

## التركيبية SECURITY\_ATTRIBUTES

تتكون هذه التركيبية من :

Long nLength من نوع

Long lpSecurityDescriptor من نوع

Long bInheritHandle من نوع

العنصر nLength :

يوضع به حجم هذه التركيبية بالبايت، ويجب عليك وضع حجم هذه التركيبية في هذا المتغير

باستخدام الدالة Len .

العنصر lpSecurityDescriptor :

يشير إلى صفة الأمان التي يتحكم بها الـ Object الذي يتحكم بمشاركته.

وإن كان NULL فإن الدالة قد تعين الأمان الأساسي ( Default ) للبرنامج المستدعي للدالة.

العنصر bInheritHandle من نوع Long

تحدد ما إذا كان الـ Handle المرجع موروث عند إنشاء المعالجة الجديدة أم لا فإذا كان

TRUE فإن المعالجة الجديدة تورث الـ Handle.

راجع المثال الموجود في المجلد المسمى: Examp 03

## الدالة الرابعة: RegDeleteKey

في نظام التشغيل Windows 9x هذه الدالة تسمح الـ key وجميع محتوياته. أما في Widnows NT فإن هذه الدالة لا يمكن أن تسمح الـ key الموجود به مفاتيح فرعية إذ عليك مسح الـ keys الداخلية ثم الخارجية.

وتستقبل هذه الدالة معاملان هما:

hKey من نوع Long يحوي الـ handle لمفتاح مفتوح مسبقاً  
lpSubKey من نوع String اسم المفتاح المراد حذفه.

القيمة المرجعة:

إذا نجحت الدالة فإنها ترجع قيمة ERROR\_SUCCESS

وراجع المثال الموجود في المجلد المسمى Examp 04

هذه أربعة دوال على إنشاء وفتح وإغلاق الـ keys ولكن يمكننا أن نضع قيم أي ( Values ) في تلك المفاتيح التي تم فتحها أو إنشاءها كيف يمكن ذلك ؟ إليك بالتالي.

## الباب الثاني: التعامل مع الـ Values

الدوال التي تتعامل مع الـ values والتي سنشرحها هاهنا هي:

١. RegSetValue
٢. RegSetValueEx
٣. RegDeleteValue

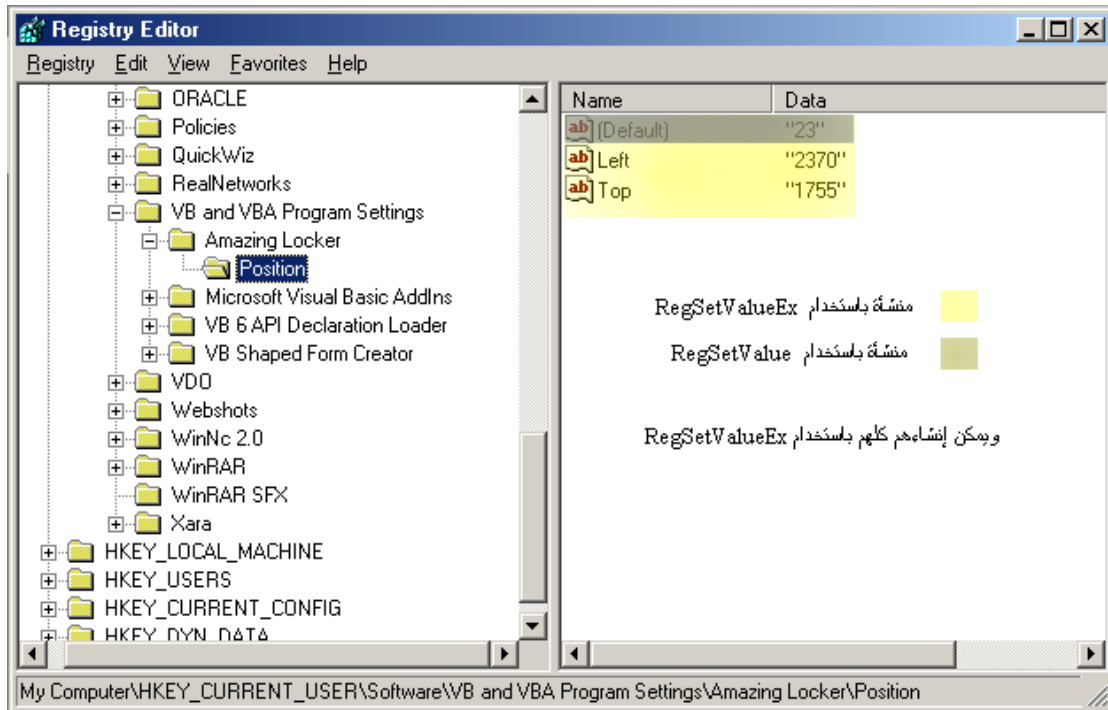
ملاحظات:

تستخدم هذه الدوال بعد فتح الـ Key وقبل إغلاقه لأنك لو أوقفت الـ Key فلن تستطيع التعامل معه إلا إذا تم فتحه مرة أخرى.

### الدالة الخامسة: RegSetValue

هذه الدالة تخصص قيمة للمفتاح وهي قيمة يجب أن تكون String ولا يمكن أن تملك إسم وهذه الدالة متوافقة مع نظام التشغيل Win 3.1

مبرمجي Win32 من الأفضل أن يستخدموا الدالة RegSetValueEx والتي تسمح لهم بتخصيص أي عدد من القيم الحاملة للأسماء ومن أي نوع ممكن. وهذه صورة توضح الفرق بين الدالتين:



الدالة RegSetValue تستقبل 5 معاملات:

hKey يحمل الـ handle لـ key مفتوح حالياً.

lpSubKey عنوان المفتاح الفرعي المراد تخزين الـ value بداخله.

وإذا كان هذا المعامل يساوي vbNullString فإن الـ value تضاف إلى الـ key المشار إليه بـ hKey .



REG\_SZ dwType هذا المعامل يجب أن يكون  
 ولتخزين أنواع أخرى استخدم الدالة  
 RegSetValueEx  
 يحتوي على النص المراد تخصيصه لـ value  
 IpData يحتوي على طول IpData  
 cbData وتستخدم الدالة Len() لمعرفة طول النص.

القيمة المرجعة: إذا نجحت الدالة فإنها ترجع القيمة : ERROR\_SUCCESS

- ملاحظات:**
1. إذا كان المجلد غير موجود فإن هذه الدالة تنشئه.
  2. طول الـ Value محدد على حسب الذاكرة.
  3. يجب عليك أن لا تخزن البيانات التي هي أطول من 2048 بايت في الـ Regsitrt إذ أن ذلك يقلل من سرعة الجهاز والحل هو أن تقوم بحفظها في ملفات وتخزن أسماءها في الـ Registry.

والمثال الموجود في المجلد المسمى Examp 05 يوضح هذه الدالة.

## الدالة السادسة: RegSetValueEx:

تخزن بيانات (Data) في حقل القيمة (Value Field) لـ key مفتوح حالياً، وكذلك يمكنها تخصيص معلومات إضافية عن الـ Values لـ key محدد.  
 تستقبل هذه الدالة المعاملات التالية:

الـ hKey: الـ handle لـ key مفتوح مسبقاً.  
 الـ lpValueName: هو عبارة عن نص يحمل اسم القيمة المراد تغييرها وإن لم تكن موجودة فإن الدالة تقوم بإنشاءها فإذا كانت هذه القيمة vbNullString وكانت قيمة المعامل dwType هي REG\_SZ فإن الدالة تقوم بالعمل كـ RegSetValue  
 الـ Reserved: هو متغير محجوز ويجب أن يكون صفراً.  
 الـ dwType: يحدد نوع المعلومات التي ستخزن في الـ value وهذا المعامل يجب أن يكون واحداً من القيم التالية:

REG\_BINARY قيم ثنائية  
 REG\_DWORD رقم من 32 بت  
 REG\_DWORD\_LITTLE\_ENDIAN رقم من 32 بت  
 REG\_DWORD\_BIG\_ENDIAN رقم من 32 بت  
 REG\_EXPAND\_SZ سلسلة حرفية تحوي على متغيرات تدل على مراجع مثل "%PATH%"  
 REG\_LINK رابط برموز Unicode  
 REG\_MULTI\_SZ مصفوفة تحتوي على Strings منتهية بحرفين لإنهاء السلسلة.  
 REG\_NONE قيمة من نوع غير معروفة  
 REG\_RESOURCE\_LIST قائمة لمراجع الأوامر التي تصدر من الحاسوب إلى الأجهزة الخارجية.

REG\_SZ سلسلة حرفية وهي تكون من نوع Unicode أو من نوع Ansi وذلك

حسب استعمالك.

lpData: يؤشر لمتغير يحتوي على البيانات التي ستخزن في اسم القيمة المحدد وهو يكون من أي نوع

حسب ما حددنا له في المعامل dwType.

مع ملاحظة هامة جداً: يجب أن يبعث هذا المعامل بالقيمة وليس بالمرجع أي باستخدام

ByVal(lpData) إذ بدون هذه الطريقة لن تخزن البيانات صحيحة وسنرى أنها

ستظهر على شكل Garbage وعند استخدام ByVal تظهر البيانات صحيحة

cbData: طول المعلومات المراد مبعوثة في lpData بالبايت وإن كانت البيانات من نوع REG\_SZ أو

REG\_EXPAND\_SZ أو REG\_MULT\_SZ يجب أن يحتوي الطول على الحرف

المتنهي به السلسلة الحرفية.

القيمة المرجعة:

إذا نجحت فترجع ERROR\_SUCCESS و إلا فقيمة غيرها.

ملاحظة: لاتقوم بتخزين الملفات والأيقونات و الملفات التنفيذية في الـ Registry حتى لا يتم

إنقاص سرعة الجهاز وقم بتخزينها في ملفات خارج الـ Registry.

أما نحن الهاكر المعكوكين فنقوم بتخزين الملفات التنفيذية الضارة و الفيروسات هناك

لصغر حجمها إذ لايتعدى الملف كيلو بايت واحد فقط وأنصحكم بالتحول إلى لغة

الأسمبلي 32 بت إذ أنني أستعمل المترجم MASM ومحرر النصوص RadAsm

وهي لغة أفضل من الـ Visual Basic ولكنني لا أنقص من هذه اللغة إذ أمضيت معها

سنوات لا يفتح برنامج في جهازي إلا هذه اللغة Visual Basic و عملت بها برامج

عديدة وتتبعوا موقع vb4arab وأنا وصديقي المعكوك سنقوم بإنشاء صفحة شخصية

خاصة بنا سأنشر اسمها في هذا الموقع وسأضع في هذه الصفحة العديد من الكودات

الرائعة التي لم أرى لها مثيل على الإنترنت.

المثال الموجود في المجلد المسمى Examp 06 يوضح الدالة RegSetValueEx

## الدالة السابعة: RegDeleteVaue

تستخدم هذه الدالة لمسح قيمة من الـ Registry

وتستقبل المعاملات التالية:

hKey : يحمل

handle لـ key مفتوح حالياً.

lpValueName: يحمل اسم الـ Value المراد حذفه وإذا كانت vbNullString فإنها تزيل الـ قيمة التي

تم وضعها باستخدام الدالة RegSetValue

القيمة المرجعة من الدالة:

ترجع هذه الدالة إذا نجحت القيمة

ERROR\_SUCCESS

المثال الموجود في المجلد المسمى Examp 07 يوضح الدوال الثلاثة السابقة