---API---

Registry Functions

For Visual Basic

إعداد: Syntax Error 0x0

مع دعم الهاكر المعكوك

للمراسلة:

SyntaxErr0x0@yahoo.com

المقدمة

بسم الله والصلاة والسلام على رسول الله وعلى آله وصحبه ومن تبع هداه إلى يوم الدين.

أما بعد:

فقد قرأت في منتدى فيجوال بيسك للعرب www.vb4arab.com في قسم دوال الـــ API سؤال عن الدوال الخاصة بالـــ Registry ففكرت أن أجعل بعضاً من وقتي للإجابة عن هذا السؤال.

ملاحظة: لن أشرح ماهي الـــ Registry وكيفية التعامل معها يدوياً شرحاً مفصلاً لأنه يفترض من القاري أن يعرف هذه الأشياء مسبقاً.

ملاحظة أخرى:

لن يتم شرح كل الدوال مع أني أعدكم بشرحها في أقرب وقت وذلك لأسباب خاصة سأشرحها في الفصل الثالث ولأي إستفسارات يرجى مراسلتي على العنوان التالي:

SyntaxErr0x0@yahoo.com

الفصل الأول

أسماء الدوال والسجلات و الثوابت المستخدمة للتعامل مع الـــ Registry

لقد قمت بتضمين كل الدوال و السجلات و الثوابت في **Modules** باسم **RegistryDef.bas**
فقط قم بوضعه في الـــ **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 الــ **Registry** طبعاً من **Keys** و **Values**.

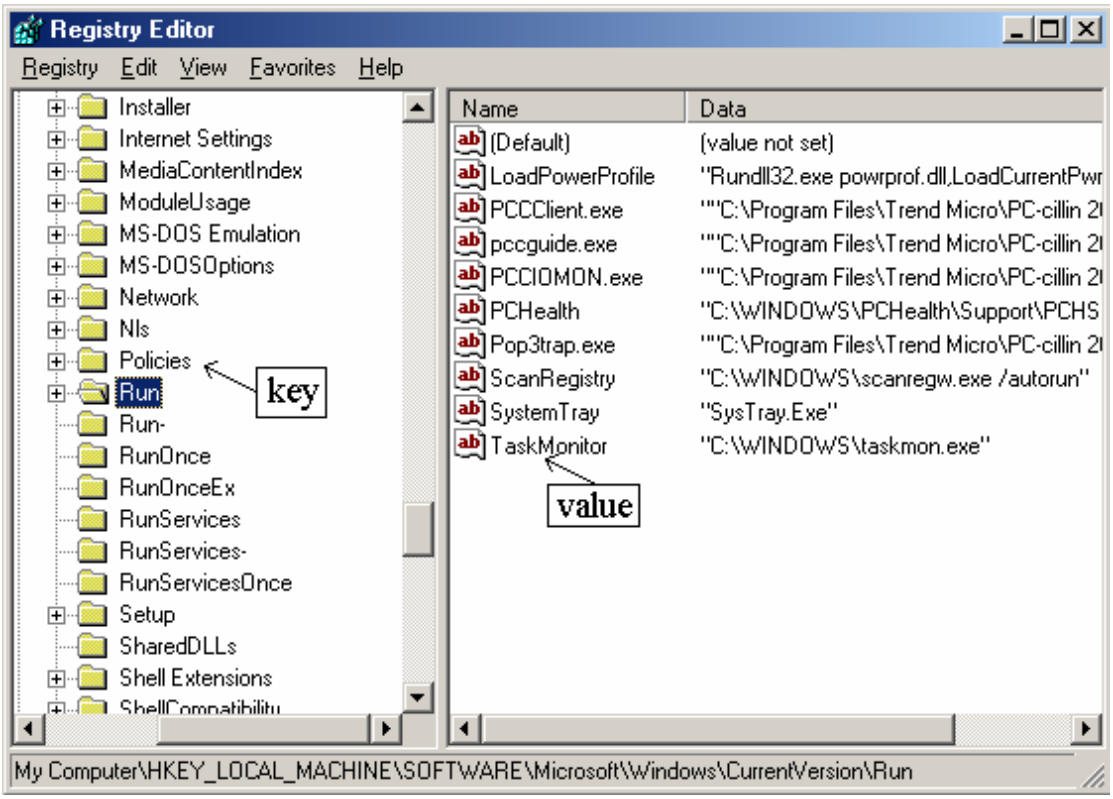
أما الــ **Keys**:

فهي تحتوي على **SubKeys** وهي عبارة عن **Keys**

وتحتوي على **Values**

و بفتح الــ **Registry** باستخدام البرنامج **regedit.exe** تظهر لنا الــ **keys** في الناحية اليسرى من الشاشة و الــ **Values** في الناحية اليمنى.

وهذه صورة توضح ذلك:



يمكن إنشاء أو تغييرالـــ keys والـــ Values يدوياً بسهولة ولكن كيف يمكن فعل كل ذلك بلغة Visual Basic ؟

قبل البدء يجب معرفة بعض الأساسيات للتعامل مع الـــ Registry.

1. يجب فتح الـــ Key المراد وضع أو تغيير الـــ values الموجودة بداخله

2. تغيير القيم الموجودة داخله.
3. إغلاق الـــ Key الذي تم فتحه. وهذه النقطة مهمة جداً لأنه من دونها لا يتم كتابة البيانات في الـــ Registry.

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

شرح النقاط السابقة:

1. فتح الـــ key.

يجب قبل فتح key معرفة handle لـــ key آخر يجب أن يكون مفتوح مسبقاً ويجب أن يكون الـــ key المراد فتحه عبارة عن subkey للـــ key الذي يوجد لدينا الـــ handle الخاص به.

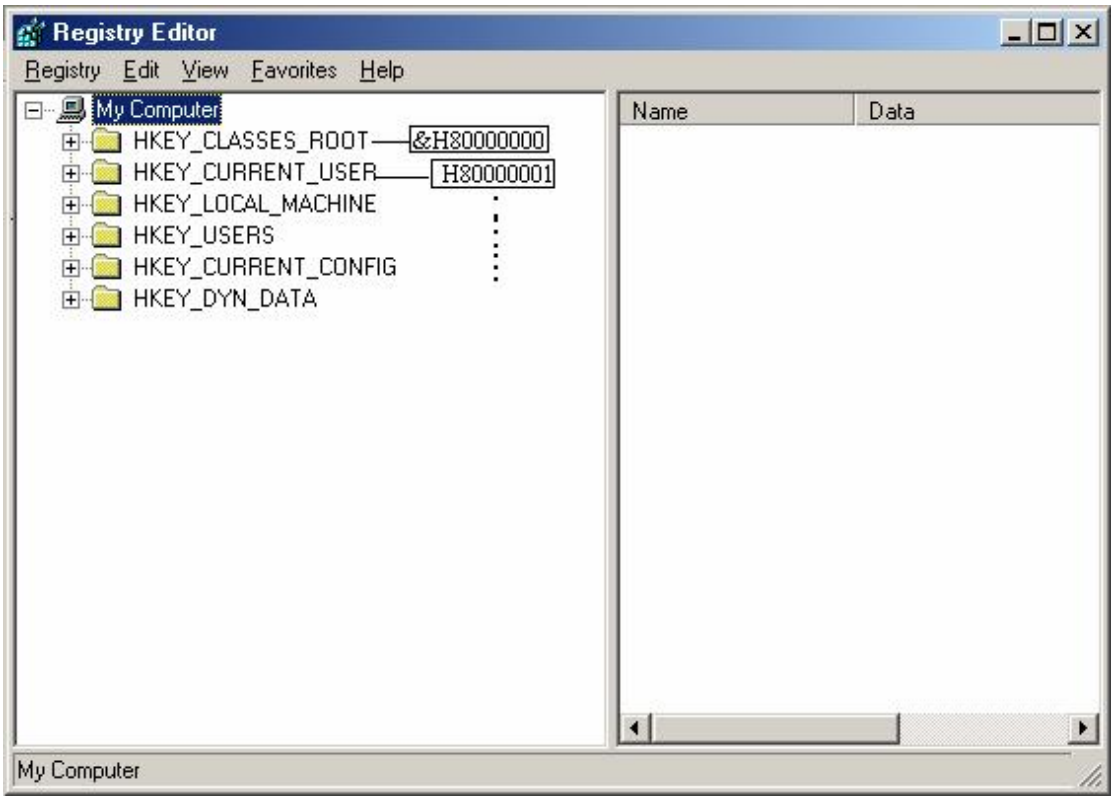
تتسآءل الآن كيف يمكنني معرف الـــ handle الخاص بالـــ key الأب؟

إجابة هذا السؤال عندي ولله الحمد وهي بسيطة.

يعطي نظام التشغيل Windows أرقام ثابتة للمفاتيح الرئيسية كلما تم تشغيل النظام وهذه الـــ keys مع الـــ 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 الــ keys السابقة بــ predefined keys

والدوال المستخدمة لفتح الــ keys هي :

RegOpenKey, RegOpenKeyEx
RegCreateKey,RegCreateKeyEx

2. كتابة وحذف البيانات:

يمكن استخدام الدوال التالية:

RegSetValueEx, RegSetValue وذلك لتخصيص بيانات (Data) لــ 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 الــ key المحذوف لايتم إزالته حتى يتم قفل آخر handle له.

ولتغيير أمان الــ keys نستخدم RegSetKeySecurity

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

لإرجاع بيانات مفصلة حول subkey معينة، البرنامج يمكن أن تستدعي الدالة RegQueryInfoKey

والدالة RegGetKeySecurity ترجع نسخة من الـــ SecurityDescription التي تحمي الـــ key

ولجلب الـــ values لـــ key تستخدم الدالة RegEnumValue

3. إغلاق الـــ key يتم باستخدام الدالة RegCloseKey

كل ماعليك فهمه من النقاط السابقة (إن لم تفهم شيئاً منه) هو التالي:

يجب فتح الـــ key ثم تغير 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

لم أشرح كل الدوال الخاصة بالتعامل مع الـ **Registry** لسبب ضيق الوقت الذي قمت بكتابة هذا الموضوع فيه إذ أنني كتبت ورتبت كل هذا في خلال يوم واحد ونصف وأحمد الله أنه يوجد لدي كود كتبته مسبقا في وقت فراغي وأرفقته مع هذا الملف.

مع أنني أعدكم بأنني سأشرح هذه الدوال والدوال الأخرى عندما أجد وقت فراغ إذ أنني مشغول بمشروع التخرج.

مع العلم بأنني اخترت أهم الدوال، ولا أعتقد أن الدوال الأخرى ستكون ذات أهمية كبيرة لكم على الأقل في الوقت الحالي حتى يتسنى لي الوقت لكي أكمل لكم البقية.

الباب الأول: التعامل مع الـــ 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 الـــ handle لـــ 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 لايمكن استعمال الـــ handle الذي تم قفله إلا إذا تم فتح الـــ key من جديد مع ملاحظة أن الـــ handle سيتغير.

وعليك أن لاتترك الـــ key مفتوح أكثر من الوقت الذي تستعمله فيه.

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

وعند كتابة بيانات كبيرة جداً في الـــ Registry وتنفيذ هذه الدالة قد تأخذ بضع ثواني في تخزين البيانات في الـــ 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 إستخدام الدالة RegCreateKeyEx والتي سأقوم بشرحها بعد هذه الدالة.

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

hKey : من نوع Long

يشير إلى handle لـــ key مفتوح حالياً، أو أي من الـــ Predefinedkeys

(راجع الفصل الثاني لمعرفة ماهي الـــ PredefinedKeys)

والـــ key المنشأ هو subkey لـــ key المعرف بواسطة hKey.

lpSubKey : من نوع String

يحدد إسم الـــ Key المراد إنشاءه أو فتحه وهذا الـــ Key يجب أن يكون مفتاح فرعي من الـــKey المحدد في hKey.

ويمكن أن يكون إسم واحد أو مسار بهذا الشكل:

"SOFTWARE"

أو

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

phkResult: من نوع Long

يؤشر إلى متغير يستقبل الـــ Handle الخاص بالـــ Key التي تم فتحها.

القيمة المرجعة:

ترجع هذه الدالة القيمة ERROR_SUCCESS إذا نجحت.

ملاحظة:

يمكن إنشاء مجموعة من الـــ 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 الـــ Handle لمفتاح مفتوح حالياً (راجع الدالة السابقة)

lpSubKey: من النوع String

يشير إلى متغير نصي يحمل اسم الـــ subkey الذي يجب أن يكون متفرع من المفتاح المشار إليه بـــ hKey.

مع ملاحظة أن النص يجب أن لا يبدأ بـــ "\" 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 الـــ Object لهذا الـــ key

أما إذا كان الـــ key موجود فإن هذا المعامل يتجاهل.

لن أشرح الـــ Classes الآن لما يحتاجه من وقت وجهد.

dwOption: من نوع Long

ملاحظة: هذه الخاصية موجودة في Windows NT و Windows 2000

أما في Windows 9x فإنه يتم تجاهلها.

هذا المعامل يحدد خيارات خاصة بالـــ 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

الـــ Key المنشأ باستخدام هذا الثابت تخزن في الذاكرة وغير محفوظة حفظ جيد

وهذا الثابت يتم تجاهله إذا كان الـــ key موجود.

REG_OPTION_BACKUP_RESTORE

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

samDesired من نوع Long

يبين كيفية أو نوع الأمن المراد استخدامه للـــ 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**

lpSecurityAttributes من نوع التركيبة : SECURITY_ATTRIBUTES

هذا المعامل يحدد هل الـــ handle المرجع يمكن أن يكون موروثـــاً بواسطة المعالجة الإبن (ولن أفصل هذه النقطة لما يتطلبه من وقت وجهد).
أما إذا كان هذا المعامل NULL فإن الـــ handle لا يمكن أن يورث لمعالجة إبن.
وسأشرح هذه التركيبة بعد قليل.

تحت نظام التشغيل Windows NT :

العنصر lpSecurityDescriptor في هذه التركيبة يوضح صفة الأمان للمفتاح الجديد وإن كان NULL فإن الأمن يأخد الـــ Defaults

تحت نظام التشغيل Windows 9x :

العنصر lpSecurityDescriptor يتم تجاهله.

phkResult : من النوع Long

تقوم هذه الدالة بوضع الـــ handle للـــ key المفتوح أو المنشأ ليتم إستخدامه من قبل المبرمج.

lpdwDesposition : من النوع: Long

تقوم الدالة بوضع إحدى القيم التالية:

REG_CREATED_NEW_KEY
REG_OPENED_EXISTING_KEY

أما الأول فيدل على أن المفتاح ليس موجود وتم إنشاءه، 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:

إذا نجحت الدالة فإنها ترجع القيمة ERROR_SUCCESS

يوجد مثال على هذه الدالة في المجلد المسمى Examp 03 ولكن يجب أن تدرس التركيبة التالية حتى تتمكن من فهم هذه الدالة جيداً

ملاحظة:

عند إنشاء أو فتح الـــ Keys استخدم

RegCreateKeyEx أو RegOpenKeyEx

ولاتسخدم

RegCreateKey و RegOpenKey

وذلك حتى تتمكن من استخدام بعض الدوال الأخرى.

التركيبة SECURITY_ATTRIBUTES

تتكون هذه التركيبة من :

nLength	من نوع Long
lpSecurityDescriptor	من نوع Long
bInheritHandle	من نوع Long

العنصر nLength :

يوضع به حجم هذه التركيبة بالبايت، ويجب عليك وضع حجم هذه التركيبة في هذا المتغير باستخدام الدالة Len .

العنصر lpSecurityDescriptor :

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

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

العنصر bInheritHandle من نوع Long

تحدد ما إذا كان الـــ Handle المرجع موروث عند إنشاء المعالجة الجديدة أم لا فإذا كان TRUE فإن المعالجة الجديدة تورث الـــ Handle.

راجع المثال الموجود في المجلد المسمى: Examp 03

الدالة الرابعة: RegDeleteKey

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

وتستقبل هذه الدالة معاملان هما:

hKey	من نوع	Long	يحوي الـــ 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 الـــ 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 الـــ Values

الدوال التي تتعامل مع الـــ values والتي سنشرحها هاهنا هي:

1. RegSetValue
2. RegSetValueEx
3. RegDeleteValue

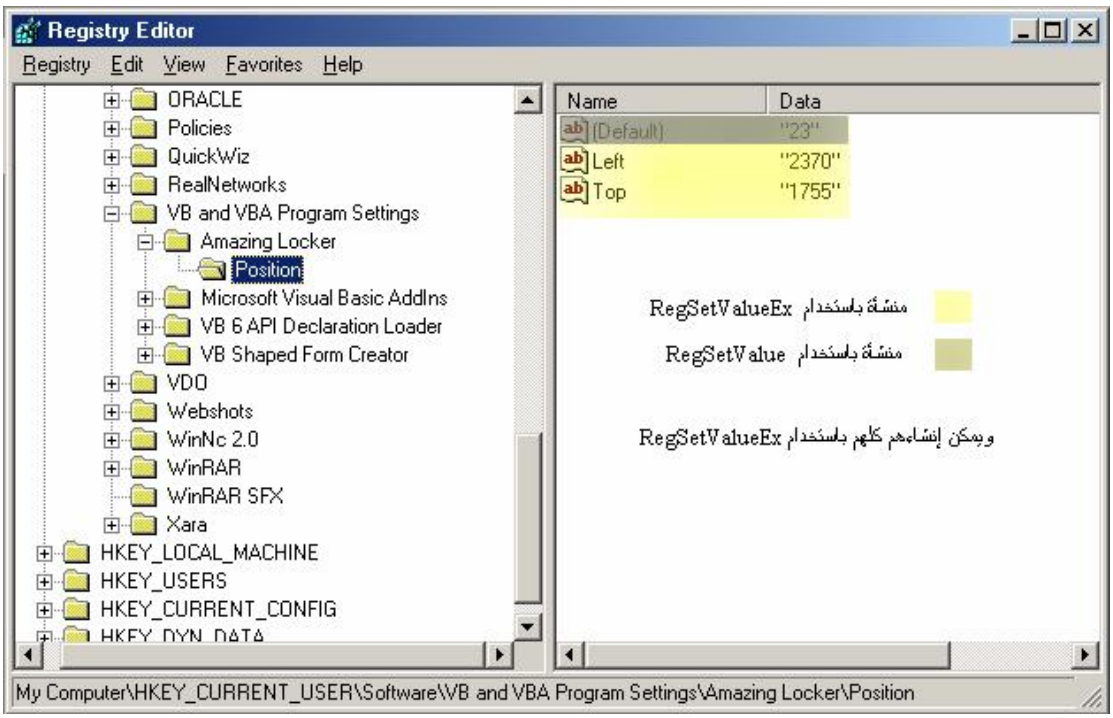
ملاحظات:

تستخدم هذه الدوال بعد فتح الـــ Key وقبل إغلاقه لأنك لو أقفلت الـــ 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 ولايمكن 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 الـــ **handle** لـــ **key** مفتوح حالياً.

lpSubKey عنوان المفتاح الفرعي المراد تخزين الـــ **value** بداخله.

وإذا كان هذا المعامل يساوي **vbNullString** فإن الـــ **value** تضاف إلى الـــ **key** المشار إليه بـــ **hKey** .

dwType هذا المعامل يجب أن يكون **REG_SZ**

ولتخزين أنواع أخرى استخدم الدالة **RegSetValueEx**

lpData يحتوي على النص المراد تخصيصه لـــ **value**

cbData يحتوي على طول **lpData** بالبايت مع مراعاة أنه لا يتم حساب حرف نهاية السلسلة الحرفية. 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 الـــ Value محدد على حسب الذاكرة.**

3. يجب عليك أن لا تخزن البيانات التي هي أطول من **2048** بايت في الـــ **Regsitrt** إذ أن ذلك يقلل من سرعة الجهاز والحل هو أن تقوم بحفظها في ملفات وتخزن أسماءها في الـــ **Registry**.

والمثال الموجود في المجلد المسمى **Examp 05** يوضح هذه الدالة.

الدالة السادسة: RegSetValueEx:

تخزن بيانات (Data) في حقل القيمة (Value Field) لـــ key مفتوح حالياً، وكذلك يمكنها تخصيص معلومات إضافية عن الـــ Values لـــ key محدد.

تستقبل هذه الدالة المعاملات التالية:

hKey: الـــ handle لـــkey مفتوح مسبقاً.

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 كـــ RegSetValue

Reserved: هو متغير محجوز ويجب أن يكون صفراً.

dwType: يحدد نوع المعلومات التي ستخزن في الـــ 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 ماحددنا له في المعامل dwType.

مع ملاحظة هامة جداً : يجب أن يبعت 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 المنتهي به السلسة الحرفية.

القيمة المرجعة:

إذا نجحت فترجع ERROR_SUCCESS و إلا فقيمة غيرها.

ملاحظة: لاتقوم بتخزين الملفات والأيقونات و الملفات التنفيذية في الـــ Registry حتى لايتم إنقاص سرعة الجهاز وقم بتخزينها في ملفات خارج الـــ Registry.

أما نحن الـــهاكر 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 الـــ Visual Basic ولكنني لا أنقص من هذه اللغة إذ أمضيت معها سنوات لا يفتح برنامج في جهازي إلا هذه اللغة Visual Basic وعملت 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 الـــ Registry

وتستقبل المعاملات التالية:

hKey :	يحمل handle لـــ key مفتوح حالياً.
lpValueName:	يحمل اسم الـــ Value المراد حذفه وإذا كانت vbNullString فإنها تزيل الـــ 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

انتهى الفصل الثالث وترقبوا الكتاب الذي يشرح كل هذه الدوال شرح وافياً جداً.

وسأبدأ في ذلك الكتاب عندما أكمل مشروع التخرج الحالي.

كتبه : ***Syntax Error 0x0***

والسلام عليكم ورحمة الله وبركاته