

Part 3: A Little Introduction on Programming with Registry

(for Microsoft Windows 32-bit)

Seksyen 1: Istilah – istilah Penting

Beberapa istilah penting di dalam programming Windows yang perlu diketahui terlebih dahulu sebelum memasuki alam programming dengan Registry.

- **Win32 API**
Windows 32 bit, Application Programming Interface. API disediakan untuk membekalkan routine routine yang membolehkan dan memudahkan kawalan dan interaction antara Aplikasi, Komponen dan juga OS. Juga untuk membolehkan kekonstanan aplikasi antara version version Windows 32 bit yang berbeza beza. Aplikasi dan device driver pada user mode (ring-3) menggunakan routine yang terdapat dalam Win32 API.
- **Handle**
Merujuk kepada jenis ‘pegangan’ kepada sesuatu objek. Merupakan 32-bit unsign integer yang digunakan oleh Windows sendiri dalam merujuk kepada setiap komponen (seperti cursor, start menu, fail dan lain lain). Bernilai DWORD
- **SID**
Season ID, iaitu struktur yang akan menyimpan nama komputer, nama user dan nama kumpulan. SID yang berbeza untuk komputer, user, kumpulan atau kombinasi antaranya. SID mestilah unik (uniqueness threory). Biasanya proses Windows akan menggunakan SID untuk mendapatkan atau merujuk kepada user dan kumpulan yang tertentu.

Seksyen 2: Library dan Method

2.1: Header

Header/include yang penting melibatkan operasi registry adalah seperti berikut:

Menggunakan Win32 API.

C/C++

```
#include <Windows.h>
```

VB

Public Declare Function xxx Lib "advapi32.dll" Alias "xxxW" (parameter) As Long
‘ dimana xxx adalah nama API call yang diperlukan, (xxxW adalah untuk Unicode manakala xxxA adalah untuk ansi).

Menggunakan .NET Framework

C#.NET

using Microsoft.Win32;

VB.NET

Import Microsoft.Win32

Untuk Scripting Language seperti Windows Scripting Host (WSH), Visual Basic Scripting (VBS) atau JavaScript/JScript (JS), penggunaan Objek WScript.Shell dapat membantu dalam pengolahan registry tetapi tidak akan diterangkan disini kerana diluar skop kita.

2.2: Win32 Registry API untuk C, C++ dan VB

Beberapa API yang berkaitan dengan operasi registry serta penerangan ringkasnya adalah seperti berikut dibawah. Setiap function akan memulangkan nilai ERROR_SUCCESS sekiranya function ini berjaya dilaksanakan.

- **RegCloseKey(HKEY hkey)**
Digunakan untuk melepaskan handle hkey yang telah digunakan. Function ini sangat berguna kerana setiap key yang berjaya dibuka, perlulah ditutup (good manner).
- **RegConnectRegistry(LPCTSTR lpMachineName, HKEY hKey, PHKEY phkResult)**
Mengakses registry pada remote komputer (lpMachineName dalam format \\namakomputer). Sekiranya parameter ini = NULL, maka local computer akan digunakan. hKey pula merujuk kepada handle top-level-nodes (iaitu HKCR, HKCU, HKLM, HKPD atau HKU, serta HKCC atau HKDD untuk Windows 9x sahaja.) dengan beberapa restriction Windows yang berdasarkan kepada remote computer itu sendiri. Manakala phkResult pula merupakan pointer yang menerima handle dari remote komputer tersebut. Function ini tidak boleh digunakan untuk Windows XP Home Edition.
- **RegCreateKeyEx(HKEY hKey, LPCTSTR lpSubKey, DWORD Reserved, LPTSTR lpClass, DWORD dwOptions, REGSAM samDesired, LPSECURITY_ATTRIBUTES lpSecurityAttributes, PHKEY phkResult, LPDWORD lpdwDisposition)**
Mencipta key baru (atau membuka key yang sedia ada sekiranya key tersebut telah sedia wujud) dan memulangkan opened key handle kepada phkResult. hKey merujuk handle kepada top level nodes. Key tersebut (hKey) mestilah mempunyai KEY_CREATE_SUB_KEY akses. lpSubKey pula merupakan pointer ke subkey (tidak boleh dimulai dengan character “\” dan tidak boleh diwakili dengan NULL sahaja). Manakala Reserved pula mestilah 0. lpClass pula adalah class kepada key ini, atau NULL sekiranya tidak digunakan. dwOptions pula digunakan untuk meletakkan opsyen didalam pembinaan key baru (lihat konstant di bawah). samDesired pula merupakan setting mask untuk pengaksesan key tersebut (lihat konstant di bawah). lpSecurityAttributes pula adalah pointer ke structure untuk

attrib sekuriti sama ada handle ini boleh dikongsi oleh child yang dicipta oleh current program (inheritable?). Sekiranya NULL, maka child program tidak boleh menginherit handle ini dan key yang dicipta akan menerima default security attrib. phkResult akan menerima handle tersebut. Handle ini perlulah ditutup dengan RegCloseKey selepas digunakan. lpwDisposition akan memulangkan salah satu daripada nilai ini: REG_OPENED_EXISTING_KEY sekiranya key tersebut telah sedia wujud atau REG_CREATED_NEW_KEY sekiranya key tersebut dicipta oleh function ini.

- **RegDeleteKey(HKEY hKey, LPCTSTR lpSubKey)**
Memadam subkey serta value dan data yang ditetapkan didalam lpSubKey. hKey pula merujuk kepada top-level-node key (rujuk Part1). lpSubKey tidak boleh NULL (iaitu tidak boleh padam top-level-node). Key tidak akan dipadam secara terus kecuali handle kepada subkey tersebut telah ditutup. Subkey yang dinyatakan di dalam lpSubKey juga tidak boleh mengandungi subkey lain. Untuk memadam subkey serta childnya, developer perlu memadam satu persatu atau menggunakan function SHDeleteKey (dideclare dalam shlwapi.h).
- **RegDeleteValue(HKEY hKey, LPCTSTR lpValueName)**
Memadam value yang dispesifikasikan di dalam lpValueName berdasarkan kepada handle top-level node hKey.
- **RegDisablePredefinedCache()**
Sekiranya function ini dipanggil, maka sebarang akses selepas panggilan function ini akan terus dilakukan ke HU\“sid”. Lihat Part 1/Seksyen 2 mengenai HKEY_USER.
- **RegEnumKeyEx(HKEY hKey, DWORD dwIndex, LPTSTR lpName, LPDWORD lpcName, LPDWORD lpReserved, LPTSTR lpClass, LPDWORD lpcClass, PFILETIME lpftLastWriteTime)**
Digunakan untuk memulangkan key ke dwIndex setiap kali ianya dipanggil. dwIndex = 0 untuk mendapatkan subkey yang pertama. Disebabkan subkey tidak disusun mengikut alpabet, maka dwIndex yang berikutnya akan memulangkan subkey yang kedua dan tidak semestinya mengikut turutan. lpName pula merupakan buffer yang akan menerima subkey mengikut dwIndex dengan saiz oleh lpcName (di dalam bentuk TCHAR). lpcName akan merujuk kepada pemboleh ubah DWORD yang akan memberikan saiz dalam TCHARs setelah function ini berjaya dipanggil (tidak termasuk NULL character). lpReserved mestilah NULL. lpClass pula adalah pointer kepada buffer yang akan menyimpan class kepada subkey yang diterima dengan saiz lpcClass (juga di dalam bentuk TCHAR) tanpa mengira NULL character. Sekiranya lpClass adalah NULL, maka lpcClass juga adalah NULL. lpftLastWriteTime pula adalah pointer kepada struct FILETIME merujuk kepada masa terakhir subkey tersebut ditulis.

- **RegEnumValue** (HKEY hKey, DWORD dwIndex, LPTSTR lpValueName, LPDWORD lpcValueName, LPDWORD lpReserved, LPDWORD lpType, LPBYTE lpData, LPDWORD lpcbData)
Memulangkan pointer pointer untuk value, data dan jenis data serta saiz value dan data ke lpcValueName, lpData, lpType serta lpcValueName dan lpcbData (dalam bytes) respectively, berdasarkan kepada index dwIndex. Untuk mengetahui jenis data, sila rujuk kepada jadual constant dan/atau jenis data (Part 1/Seksyen 3). lpData, lpType dan lpcbData dibenarkan NULL sekiranya tidak digunakan tetapi lpReserved mestilah NULL. Sekiranya lpcbData lebih besar daripada saiz yang ditetapkan oleh pembolehkan yang dipointerkan, maka function ini akan gagal dan memulangkan ERROR_MORE_DATA serta memberikan saiz data yang betul melalui lpcbData.
- **RegFlushKey** (HKEY hKey)
Memaksa setiap perubahan pada hKey ditulis pada registry. hKey merujuk kepada top-level-node atau handle yang dipulangkan oleh RegCreateKeyEx atau RegOpenKeyEx. Handle hKey mestilah mempunyai akses KEY_QUERY_VALUE. Function ini hanya akan memulangkan nilai sekiranya setiap perubahan telah dilakukan. Biasanya, aplikasi jarang memanggil function ini kerana registry sendiri akan automatik flush (lazy flushing system) untuk setiap key yang berubah. Microsoft sendiri tidak menggalakkan para developer menggunakan function ini.
- **RegGetKeySecurity** (HKEY hKey, SECURITY_INFORMATION SecurityInformation, PSECURITY_DESCRIPTOR pSecurityDescriptor, LPDWORD lpcbSecurityDescriptor)
Menyalin dan mendapatkan security descriptor bagi opened key (yang diwakili dengan handle hKey). lpcbSecurityDescriptor merupakan pointer kepada pemboleh ubah yang memegang saiz pSecurityDescriptor yang dihasilkan. Sekiranya nilai tersebut disetkan lebih kecil daripada nilai yang dihasilkan, maka function ini akan memulangkan error ERROR_INSUFFICIENT_BUFFER.
- **RegLoadKey** (HKEY hKey, LPCTSTR lpSubKey, LPCTSTR lpFile)
Digunakan untuk membuat subkey baru (lpSubKey) di bawah nod HU atau HKLM mengikut info registry melalui file lpFile. lpFile mestilah dicipta oleh RegSaveKey manakala hKey merujuk kepada HU atau HKLM atau handle yang dipulangkan oleh RegConnectRegistry (pada phkResult). Sekiranya lpFile tidak wujud, maka file baru akan dicipta. Windows 95, 98 dan ME mestilah menggunakan shortfilename (format 8.3).
- **RegNotifyChangeKeyValue**(HKEY hKey, BOOL bWatchSubtree, DWORD dwNotifyFilter, HANDLE hEvent, BOOL fAsynchronous)
Memerhatikan perubahan pada hKey. Sekiranya bWatchSubtree adalah true, maka subkey bagi hKey juga akan diperhatikan. dwNotifyFilter menentukan jenis perubahan yang ingin dipantau. hEvent pula adalah handle kepada event yang memerhatikan perubahan tersebut sekiranya fAsynchronous adalah true.

- **RegOpenCurrentUser** (REGSAM samDesired, PHKEY phkResult)
Membuka HKCU bagi user semasa berdasarkan samDesired yang dibekalkan dan akan memulangkan handle sekiranya berjaya kepada phkResult.
- **RegOpenKeyEx** (HKEY hKey, LPCTSTR lpSubKey, DWORD ulOptions, REGSAM samDesired, PHKEY phkResult)
Membuka subkey lpSubKey dari top-level-node handle hKey dengan set sekuriti samDesired dan memulangkan handle subkey tersebut kepada phkResult. Sebarang perubahan yang ingin dilakukan kepada subkey ini mestilah dibenarkan melalui sekuriti samDesired. ulOptions mestilah 0.
- **RegOpenUserClassesRoot** (HANDLE hToken, DWORD dwOptions, REGSAM samDesired, PHKEY phkResult)
Membuka HKCR melalui user yang didapati dari handle token hToken. hToken mestilah mempunyai akses TOKEN_QUERY untuk menggunakan function ini. samDesired adalah sekuriti terhadap cara nodes ini dilayan. dwOptions mestilah 0.
- **RegOverridePredefKey**(HKEY hKey, HKEY hNewHKey)
Digunakan untuk membuat pemetaan dari hKey ke hNewHKey, dengan hKey adalah salah satu daripada berikut: HKLM, HKCU, HKPD, HKCR, HKCC atau HU. Sekiranya hNeyHKey adalah NULL, maka function ini akan menyimpan segala perubahan yang dilakukan di dalam pemetaan kepada hKey. hNewHKey mestilah dibuka dengan RegCreateKeyEx atau RegOpenKeyEx.
- **RegQueryInfoKey** (HKEY hKey, LPTSTR lpClass, LPDWORD lpcClass, LPDWORD lpReserved, LPDWORD lpcSubKeys, LPDWORD lpcMaxSubKeyLen, LPDWORD lpcMaxClassLen, LPDWORD lpcValues, LPDWORD lpcMaxValueNameLen, LPDWORD lpcMaxValueLen, LPDWORD lpcbSecurityDescriptor, PFILETIME lpftLastWriteTime)
Mendapatkan informasi mengenai hKey. hKey merupakan salah satu daripada berikut: HKLM, HKCU, HKPD, HKCR, HKCC atau HU dan mungkin HKDD (untuk Windows 95, 98 dan ME) atau key yang telah dibuka oleh RegCreateKeyEx atau RegOpenKeyEx. lpClass pula memberikan kelas kepada string key tersebut dan saiz string kelas terbesar adalah lpcMaxSubKeyLen serta saiznya pada lpcbClass atau boleh meletakkan NULL (sekiranya lpcbClass juga adalah NULL). lpcSubKeys merujuk pointer ke subkeys yang berada dibawah hKey dengan saiz subkey yang terbesar terletak pada lpcMaxSubKeyLen. lpcValues memberikan value-value yang terdapat di dalam subkey tersebut dengan lpcMaxValueNameLen merupakan saiz terpanjang value dan lpcMaxValueLen saiz maksimum data yang terpanjang (dalam byte). lpcbSecurityDescriptor merupakan pointer ke struct SecurityDescriptor dan lpftLastWriteTime pula adalah pointer ke struct FILETIME mengenai key tersebut. lpcClass, lpcMaxSubKeyLen, lpcMaxClassLen dan lpcMaxValueNameLen adalah dalam TCHAR (tidak termasuk NULL character).

- **RegQueryMultipleValues** (HKEY hKey, PVALENT val_list, DWORD num_vals, LPCTSTR lpValueBuf, LPDWORD ldwTotsize)
Mendapatkan informasi mengenai value-value pada hKey. hKey merupakan salah satu daripada berikut: HKLM, HKCU, HKPD, HKCR, HKCC atau HU dan mungkin HKDD (untuk Windows 95, 98 dan ME) atau key yang telah dibuka oleh RegCreateKeyEx atau RegOpenKeyEx. val_list merujuk kepada pointer bagi struct VALENT dengan bilangan elemen dalamnya disetkan pada num_vals. lpValueBuf merupakan pointer buffer yang menerima buffer untuk setiap value dan ldwTotsize pula adalah saiz bytes yang dicopy ke lpValueBuffer.
- **RegQueryValueEx** (HKEY hKey, LPCTSTR lpValueName, LPDWORD lpReserved, LPDWORD lpType, LPBYTE lpData, LPDWORD lpcbData)
Mendapatkan jenis data dan data kepada value yang ditetapkan oleh lpValueName. Manakala hKey merujuk kepada HKCU, HKCC, HKCR, HKLM, HKU, HKPD, HKPN atau HKPT serta HKDD (untuk Windows 95, 98 dan ME sahaja) atau key yang telah dibuka oleh RegCreateKeyEx atau RegOpenKeyEx. lpReserved mestilah 0 (reserved by Windows). Manakala lpType pula adalah jenis data yang diperolehi dengan saiz data (dalam byte) adalah pada lpcbData, dimana data itu sendiri adalah lpData.
- **RegReplaceKey** (HKEY hKey, LPCTSTR lpSubKey, LPCTSTR lpNewFile, LPCTSTR lpOldFile)
Digunakan untuk mereplace key hKey pada subkey dan/atau value lpSubKey. hKey merujuk kepada HKLM, HKCU, HKPD, HKCR, HKCC atau key yang telah dibuka oleh RegCreateKeyEx atau RegOpenKeyEx. lpNewFile akan menggantikan lpOldFile.
- **RegRestoreKey** (HKEY hKey, LPCTSTR lpFile, DWORD dwFlags)
Menyimpan semula key hKey dengan file lpFiles berdasarkan salah satu flag yang ditetapkan pada dwFlags.
- **RegSaveKey** (HKEY hKey, LPCTSTR lpFile, LPSECURITY_ATTRIBUTES lpSecurityAttributes)
Menyimpan key hKey (HKCR, HKCU, HLCC atau handle dari RegCreateKeyEx atau RegOpenKeyEx) kedalam file lpFile dengan menggunakan atribut lpSecurityAttributes. Sekiranya lpFile telah wujud (already exist), maka function ini akan gagal dengan memulangkan error ERROR_REGISTRY_IO_FAILED (untuk windows 95, 98 dan ME) atau ERROR_ALREADY_EXISTS (untuk Windows NT, 2000, XP dan 2003).
- **RegSetKeySecurity** (HKEY hKey, SECURITY_INFORMATION SecurityInformation, PSECURITY_DESCRIPTOR pSecurityDescriptor)
Meletakkan opsyen sekuriti SecurityInformation dan struct pSecurityDescriptor pada hKey yang tertentu. Juga boleh meletakkan top-level-nodes dengan syarat hKey tersebut mestilah ditutup dengan RegCloseKey.

- **RegSetValueEx** (HKEY hKey, LPCTSTR lpValueName, DWORD Reserved, DWORD dwType, const BYTE* lpData, DWORD cbData)
Set value dengan nama lpValueName pada subkey hKey dengan jenis value dwType dan data (buffered) lpData dalam bentuk bytes atau string (untuk jenis data string iaitu REG_SZ, REG_EXPAND_SZ dan REG_MULTI_SZ). hKey mestilah salah satu dari berikut: HKCC, HKCR, HKCU, HKLM, HKPD, HKU, HKDD (untuk Windows 9x) atau handle yang diterima dari RegCreateKeyEx atau RegOpenKeyEx.
- **RegUnLoadKey** (HKEY hKey, LPCTSTR lpSubKey)
Meng'unload'kan key hKey dan subkey lpSubkey dari registry. hKey mestilah handle dari RegConnectRegistry atau dari HKLM atau HKU.

Terdapat beberapa lagi API yang dibekalkan oleh windows untuk mengakses selain dari advapi32.dll, iaitu dengan menggunakan library shlwapi.dll (SHxxx) atau NT registry function (ntdll.dll), tetapi ianya tidak akan diterangkan di sini.

Declaration konstant dalam VB untuk top-level nodes adalah seperti berikut:

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

Manakala untuk possible return value pula adalah seperti berikut:

```
Private Const ERROR_SUCCESS = 0&
Private Const ERROR_MORE_DATA = 234 ' dderror
Private Const ERROR_CANTREAD = 1012&
Private Const ERROR_ALREADY_EXISTS = 183&
Private Const ERROR_REGISTRY_IO_FAILED = 1016&
```

Konstant (dengan nilai taranya) adalah seperti dibawah. Konstan ini digunakan bersama dengan RegCreateKeyEx.

- REG_OPTION_NON_VOLATILE (0): Key yang ditulis akan disimpan di dalam file registry dan akan wujud sekiranya sistem akan direboot.
- REG_OPTION_VOLATILE (1): Key yang ditulis akan disimpan di dalam memori tetapi tidak akan ditulis di dalam file registry. Setelah registry hive yang menggunakan opsyen ini ditutup, maka key yang dicipta tidak akan wujud lagi. Sekiranya ditulis pada HKLM, maka nilainya tidak akan ditutup sehinggalah sistem shutdown. Key ini tidak diendahkan untuk Windows 9x.
- REG_OPTION_BACKUP_RESTORE (4): Digunakan untuk membuat backup operation. Sekiranya opsyen ini dipilih, maka parameter samDesired akan tidak

diendahkan. Bergantung juga kepada sekuriti attribut yang diperlukan semasa thread yang memanggil function ini.

Konstan ini pula untuk RegNotifyChangeKeyValue serta nilai taranya.

- REG_NOTIFY_CHANGE_NAME (1): Notify pemanggil sekiranya subkey tertentu dipadam atau ditambah.
- REG_NOTIFY_CHANGE_ATTRIBUTES (2): Notify sekiranya attribut berubah.
- REG_NOTIFY_CHANGE_LAST_SET (4): Notify sekiranya value ditambah, diubah atau dipadam.
- REG_NOTIFY_CHANGE_SECURITY (8): Notify sekiranya terdapat perubahan sekuriti pada sekuriti key tersebut.

Konstant ini pula adalah untuk RegRestoreKey.

- REG_WHOLE_HIVE_VOLATILE (1): Memastikan segala perubahan pada hive HKU atau HKLM berlaku dalam memory (Lihat opsyen REG_OPTION_VOLATILE untuk mendapatkan penerangan mengenai volatile).
- REG_REFRESH_HIVE (2): Akan memaksa registry untuk kembali meng'load' hKey ke asal sebelum diflush dengan lazy flush. Hive tersebut mestilah tidak digunakan dengan menggunakan parameter REG_NO_LAZY_FLUSH (dipanggil dari RegRestoreKey).
- REG_NO_LAZY_FLUSH (4): Tidak menggunakan lazy flush (automatic).
- REG_FORCE_RESTORE (8): Memaksa restore walaupun handle hKey masih belum lagi ditutup.

Struct (structure) yang boleh dikaitkan dengan registry programming adalah seperti berikut:

- SECURITY_ATTRIBUTES
Struct ini menyimpan field yang berkaitan dengan penerangan sekuriti (security descriptor). nLength merujuk kepada saiz bytes struct tersebut. lpSecurityDescriptor pula merujuk kepada (pointer) security descriptor bagi mengendalikan sesuatu objek. Sekiranya parameter ini adalah NULL, maka default security descriptor akan digunakan (iaitu dengan menggunakan security descriptor yang sama dengan thread pemanggil). Parameter lpSecurityDescriptor tidak digunakan untuk Windows 9x. bInheritHandle pula memastikan sama ada proses yang baru (dipanggil dengan thread yang sama) berkeupayaan meng'inherit' attribut sekuriti yang sama.
Declaration dalam C++:

```
typedef struct _SECURITY_ATTRIBUTES {
    DWORD nLength;
    LPVOID lpSecurityDescriptor;
    BOOL bInheritHandle;
} SECURITY_ATTRIBUTES, *PSECURITY_ATTRIBUTES;
```

Manakala untuk VB pula:

```
Private Type SECURITY_ATTRIBUTES
    nLength As Long
    lpSecurityDescriptor As Long
    bInheritHandle As Long
End Type
```

- FILETIME

Merupakan 64-bit masa yang dikira dari 1 Januari 1601 dengan selang masa 100 nanosaat.

Declaration dengan menggunakan C++:

```
typedef struct _FILETIME {
    DWORD dwLowDateTime;
    DWORD dwHighDateTime;
} FILETIME, *PFILETIME;
```

Declaration dengan menggunakan VB:

```
Private Type FILETIME
    dwLowDateTime As Long
    dwHighDateTime As Long
End Type
```

- VALENT

Digunakan untuk mendapatkan informasi mengenai registry value. ve_valuename digunakan untuk menspesifikasikan value yang terlibat. ve_valueptr merupakan pointer ke data dengan saiz ve_valuelen (dalam bytes). ve_type pula adalah jenis data.

Declaration dengan menggunakan C++:

```
typedef struct value_ent {
    LPTSTR ve_valuename;
    DWORD ve_valuelen;
    DWORD_PTR ve_valueptr;
    DWORD ve_type;
} VALENT, *PVALENT;
```

Declaration dengan menggunakan VB:

```
Private Type VALENT
    ve_valuename As String
    ve_valuelen As Long
    ve_valueptr As Long
    ve_type As Long
```

End Type

2.2: Registry Function dengan menggunakan .NET Framework

Class-class yang berkaitan dengan registry melalui .NET Framework adalah seperti berikut:

- Class Registry
- Class RegistryKey

Class Registry: Menyediakan field - field yang mengandungi top-level-nodes. Setiap field akan memulangkan instance bagi class RegistryKey. Field-field tersebut adalah seperti berikut:

- CurrentUser
Akan memulangkan RegistryKey pada HKEY_CURRENT_USER.
- LocalMachine
Akan memulangkan RegistryKey pada HKEY_LOCAL_MACHINE.
- ClassesRoot
Akan memulangkan RegistryKey pada HKEY_CLASSES_ROOT.
- Users
Akan memulangkan RegistryKey pada HKEY_USERS.
- PerformanceData
Akan memulangkan RegistryKey pada HKEY_PERFORMANCE_DATA.
- CurrentConfig
Akan memulangkan RegistryKey pada HKEY_CURRENT_CONFIG.
- DynData
Akan memulangkan RegistryKey pada HKEY_DYN_DATA. Hanya boleh digunakan pada Windows 98 dan ME.

Class RegistryKey pula menyediakan informasi dan editing registry. Setiap properties bagi class ini adalah read-only (get accessor) dan penerangan ringkasnya adalah seperti berikut:

- Name
Akan memulangkan string yang mengandungi qualified registry key bermula dari top-level-nodes (menggunakan penandaan Windows path system - '\').
- SubKeyCount
Memulangkan bilangan subkey kepada currentkey yang sedang diakses di dalam bentuk integer. Exception yang mungkin berlaku adalah: SecurityException dan ObjectDisposedException. Untuk mendapatkan (request) sekuriti, gunakan class RegistryPermission atau attribut RegistryPermissionAttribute.
- ValueCount
Memulangkan bilangan value yang terdapat didalam key yang diakses menggunakan RegistryKey. Seperti SubKeyCount, exception yang mungkin berlaku adalah seperti berikut: SecurityException dan ObjectDisposedException.

Untuk mendapatkan (request) sekuriti, gunakan class RegistryPermission atau attribut RegistryPermissionAttribute.

Method yang biasa digunakan dalam pengaksesan dan pengeditan registry pula adalah seperti berikut:

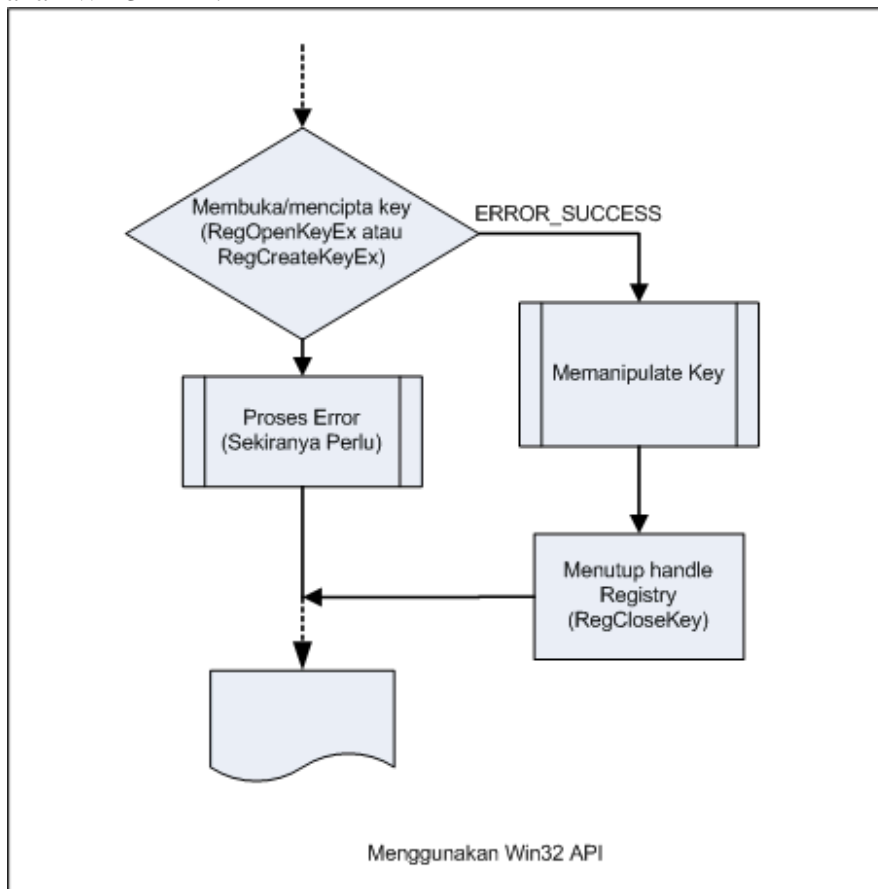
- void **Close()**
Menutup RegistryKey dan flush sekiranya terdapat perubahan pada registry.
- RegistryKey **CreateSubKey(string subkey)**
Mencipta subkey baru. Sekiranya berjaya, maka subkey tersebut akan dipulangkan. Sekiranya tidak, null (nothing) akan dipulangkan.
- void **DeleteSubKey(string subkey)**
Memadam subkey dari current key. Sekiranya tidak berjaya, maka exception ini akan dithrow: InvalidOperationException (sekiranya subkey terdapat child), ArgumentException (subkey tersebut tidak sah), ArgumentNullException (parameter subkey tersebut adalah null/nothing) atau SecurityException (sekiranya tiada permission)
- void **DeleteSubKey(string subkey, bool throwOnMissingSubkey)**
Memadam subkey daripada current-key. Sekiranya throwOnMissing, maka exception akan dithrow sekiranya subkey tersebut tidak wujud. Sekiranya false dan subkey tersebut tidak wujud, maka tiada tindakan akan berlaku. Exception yang mungkin berlaku adalah sama seperti DeleteSubKey(string).
- void **DeleteSubKeyTree(string subkey)**
Memadam subkey, termasuk child yang terdapat dibawah subkey. Exception yang mungkin berlaku: ArgumentNullException (sekiranya parameter subkey adalah null/nothing), ArgumentException (sekiranya hive cuba dipadam atau parameter subkey tidak wujud) atau SecurityException (sekiranya aplikasi tiada akses sekuriti).
- void **DeleteValue(string value)**
Memadam value dari current subkey. Exception: ArgumentException (sekiranya value tidak wujud), SecurityException (sekiranya tiada akses sekuriti), ObjectDisposedException (sekiranya RegistryKey telah ditutup) atau UnauthorizedAccessException (sekiranya RegistryKey adalah read-only).
- void **DeleteValue(string value, bool throwOnMissingValue)**
Memadam value dari current subkey. Sekiranya throwOnMissingValue adalah true, maka exception akan dithrow sekiranya value tersebut tidak wujud. Exception: SecurityException, ObjectDisposedException dan UnauthorizedAccessException adalah seperti DeleteValue(string) manakala ArgumentException boleh berlaku hanya apabila throwOnMissing adalah true.

- **void Flush()**
Akan menyebabkan segala perubahan pada registry ditulis pada file. Tidak digalakkan memanggilnya secara terus, kerana dikhuatiri akan menyebabkan segala perubahan ditulis segera. Hanya akan dipulangkan setelah setiap perubahan ditulis pada registry.
- **string[] GetSubKeyNames()**
Mendapatkan nama subkey daripada current key (rk). Akan dipulangkan dalam bentuk string array. Exception yang mungkin berlaku adalah seperti berikut: SecurityException (tiada registry security permission untuk membaca) atau ObjectDisposedException (RegistryKey tersebut telah ditutup).
- **object GetValue(string valuenam)**
Memulangkan data dari valuenam tersebut. Sekiranya valuenam adalah null/nothing atau empty string, maka akan memulangkan default value bagi current registrykey. Sekiranya value tersebut tidak wujud, maka null/nothing akan dipulangkan. Exception yang mungkin berlaku: SecurityException (tiada registry security permission untuk membaca), ObjectDisposedException (RegistryKey tersebut telah ditutup) atau ArgumentException sekiranya RegistryKey tersebut tidak wujud.
- **object GetValue(string valuenam, object defaultvalue)**
Seperti overloadednya, kecuali apabila valuenam tidak wujud, maka (object) defaultvalue akan dipulangkan.
- **string[] GetValueNames()**
Memulangkan array of string yang mengandungi setiap nama value yang terdapat atau memulangkan empty array sekiranya tiada value yang terdapat. Exception yang mungkin berlaku adalah seperti berikut: SecurityException (tiada registry security permission untuk membaca) atau ObjectDisposedException (RegistryKey tersebut telah ditutup).
- **RegistryKey OpenSubKey(string name)**
Membuka Subkey name dan akan dipulangkan dalam bentuk RegistryKey atau null sekiranya tidak berjaya. Exception: ArgumentNullException (name = null/nothing), ArgumentException (panjang subkey melebihi daripada 255 character), SecurityException (tiada registry security permission) atau ObjectDisposedException (RegistryKey tersebut telah ditutup).
- **RegistryKey OpenSubKey(string name, bool writeable)**
Seperti overloadednya, kecuali menentukan cara pengendalian subkey tersebut dengan parameter writeable. Sekiranya writeable adalah true, maka subkey tersebut akan direquest untuk dibuka dengan flag membaca, mengedit dan memadam dan akan memulangkan null (and exception rised) sekiranya tidak mendapat permisi.

- void **SetValue**(string name, object data)
Set value name dengan data '(object) data'. Exception: ArgumentNullException (parameter adalah null/nothing), ArgumentException (panjang 'name' melebihi 255 character), ObjectDisposedException, UnauthorizedAccessException atau SecurityException.
- string **ToString**()
Override. Akan memulangkan subkey dalam bentuk (string) path bermula dari top-level-nodes atau null sekiranya RegistryKey tidak merfer kepada subkey yang wujud.

2.3: Mengakses dan Mengedit Registry.

Berikut merupakan algoritma mudah dalam mengakses dan mengedit registry menggunakan Win32 API.



Contoh membuka key HKLM\SOFTWARE\Putera.com\Fafa dan menghasilkan Message Box sekiranya berjaya dengan menggunakan Visual Basic 6.

' API untuk buka dan tutup key

```
Private Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal
ulOptions As Long, ByVal samDesired As Long, phkResult As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32.dll" (ByVal hKey As
Long) As Long

' konstant untuk top-level-nodes
Private Const HKEY_LOCAL_MACHINE = &H80000002

' registry akses
Private Const READ_CONTROL = &H20000
Private Const STANDARD_RIGHTS_READ = (READ_CONTROL)
Private Const KEY_QUERY_VALUE = &H1
Private Const KEY_ENUMERATE_SUB_KEYS = &H8
Private Const KEY_NOTIFY = &H10
Private Const SYNCHRONIZE = &H100000
Private Const KEY_READ = ((STANDARD_RIGHTS_READ Or KEY_QUERY_VALUE Or
KEY_ENUMERATE_SUB_KEYS Or KEY_NOTIFY) And (Not SYNCHRONIZE))

' error
Private Const ERROR_SUCCESS = 0&

Sub Main()
Dim hKey As Long

If (RegOpenKeyEx(HKEY_LOCAL_MACHINE, "Software\Putera.com\Fafa", 0&,
STANDARD_RIGHTS_READ, hKey) = ERROR_SUCCESS) Then
    MsgBox "HKLM\Software\Putera.com\Fafa telah berjaya dibuka."
    ' Manipulate key kat sini
    ' handle untuk HKLM\Software\Putera.com\Fafa terletak kat hKey

    ' ...

    ' kalo dah sudah, jangan lupe tutup key
    Call RegCloseKey(hKey)
Else
    MsgBox "Parse error kat sini"
End If

End Sub
```

Ini pula kod dengan menggunakan C++ (bukan .NET).

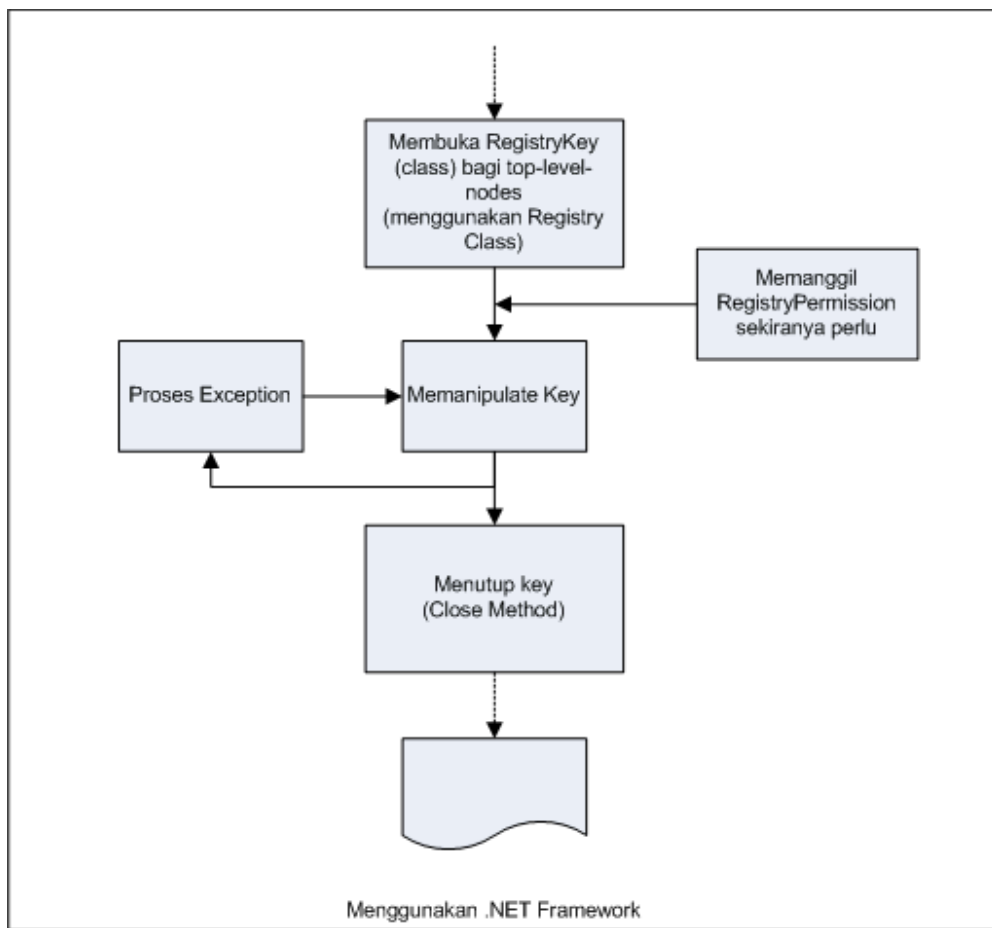
```
#include <windows.h>
#include <iostream.h>

void main()
{
    HKEY hKey;

    if (RegOpenKeyEx(HKEY_LOCAL_MACHINE, "Software\\Putera.com\\Fafa",
        0, KEY_READ, &hKey) == ERROR_SUCCESS)
    {
        // Manipulate Key kat sini
        cout << "HKLM\\Software\\Putera.com\\Fafa telah berjaya
dibuka\n";
    }
}
```

```
// ...  
  
// Tutup hKey  
RegCloseKey(hKey);  
}  
else  
// RegOpenKeyEx != ERROR_SUCCESS  
// Proses Error Kat Sini  
cout << "Maaf, tak berjaya\n";  
}
```

Berikut pula adalah algoritma mudah dalam mengakses dan mengedit registry menggunakan .NET Framework.



Dengan menggunakan C#, kod berikut akan membuka key HKLM\SOFTWARE\Putera.com\Fafa dan akan memberi mesej di console sekiranya berjaya.

```
using System;  
using Microsoft.Win32;
```

```
namespace putera01
{
class Class1
{
    /// <summary>
    /// STAThread Main -> Single Threading Apartment
    /// </summary>
    [STAThread]
    static void Main(string[] args)
    {
        RegistryKey rk = Registry.LocalMachine;

        // atau boleh gunakan
        Registry.LocalMachine.OpenSubKey(..);

        if (rk.OpenSubKey(@"SOFTWARE\Putera.com\Fafa",
false) != null)
        {
            // Manipulate keys

Console.WriteLine(@"HKLM\SOFTWARE\Putera.com\Fafa telah berjaya
dibuka");

            // Tutup opened keys
            rk.Close();
        }
        else
            Console.WriteLine("Tak Berjaya
dibuka");
    }
}
}
```

Appendix A: Exported HKLM\SOFTWARE\Putera.com\Fafa dari Registry.Exe

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Putera.com\Fafa]
; REG_DWORD
"TestDWord"=dword:00000400

; REG_DWORD_LITTLE_ENDIAN
"TestDWordLE"=dword:00000400

; REG_DWORD_BIG_ENDIAN
"TestDWordBE"=hex(5):00,04,00,00

; REG_SZ
"TestSZ"="@fa"

; REG_MULTI_SZ
"TestMSZ"=hex(7):66,00,61,00,00,00,66,00,61,00,66,00,61,00,00,00,41,06,
00,00,\
00,00

; REG_EXPAND_SZ
"TestESZ"=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,00,
4e,00,\
```

```
6f,00,74,00,65,00,70,00,61,00,64,00,2e,00,65,00,78,00,65,00,00,00
; REG_NONE
"TestNone"=hex(0):00,04,00,00

; REG_BINARY (null)
"TestBinari"=hex:

[HKEY_LOCAL_MACHINE\SOFTWARE\Putera.com\Fafa\MyChild.01]
[HKEY_LOCAL_MACHINE\SOFTWARE\Putera.com\Fafa\MyChild.02]
[HKEY_LOCAL_MACHINE\SOFTWARE\Putera.com\Fafa\MyChild.03]

; eof

[Regards, Fa]
[Greetz: Fafa, Putera.com]
```