



EBook Gratuito

APPENDIMENTO

Windows

Free unaffiliated eBook created from
Stack Overflow contributors.

#windows

Sommario

| | |
|---|-----------|
| Di..... | 1 |
| Capitolo 1: Iniziare con Windows | 2 |
| Osservazioni..... | 2 |
| Versioni..... | 2 |
| Versioni desktop | 2 |
| Versioni del server | 3 |
| Versioni mobili | 3 |
| Examples..... | 4 |
| Informazione..... | 4 |
| La storia | 4 |
| Capitolo 2: Porte seriali | 6 |
| introduzione..... | 6 |
| Examples..... | 6 |
| Elenco di tutte le porte seriali..... | 6 |
| Capitolo 3: Programmazione delle estensioni del menu di contesto File / Cartella desktop p | 9 |
| Examples..... | 9 |
| Utilizzando le voci del Registro di sistema..... | 9 |
| Utilizzo di ShellSharp..... | 9 |
| Titoli di coda | 11 |

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [windows](#)

It is an unofficial and free Windows ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Windows.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con Windows

Osservazioni

Windows è una serie di sistemi operativi sviluppati e pubblicati dalla società di software Microsoft.

puoi ottenere maggiori informazioni da [Wikipedia - Microsoft Windows](#)

Versioni

Versioni desktop

| Versione | Data |
|-----------------------|------------|
| DOS | 1981/08/01 |
| 1.01 | 1985/11/20 |
| 1.02 | 1986/05/01 |
| 1.03 | 1986/08/01 |
| 1.04 | 1986/04/01 |
| 2.03 | 1987/12/09 |
| 2.10 | 1988/05/27 |
| 2.11 | 1989/03/13 |
| 3.0 | 1990/05/22 |
| 3.1 | 1992/04/22 |
| NT 3.1 | 1993/06/27 |
| Gruppi di lavoro 3.11 | 1993/11/01 |
| 3.2 | 1993/11/22 |
| NT 3.5 | 1994/09/22 |
| NT 3.51 | 1995/05/30 |
| 95 | 1995/07/24 |
| NT 4.0 | 1996/07/24 |

| Versione | Data |
|--------------------------|------------|
| 98 | 1998/06/25 |
| 2000 | 2000/02/17 |
| ME | 2000-08-14 |
| XP | 2001/10/25 |
| XP Professional x64 | 2005-04-25 |
| Vista | 2007-01-30 |
| 7 | 2009-10-22 |
| 8 | 2012/10/26 |
| 8.1 | 2013/10/17 |
| 10 | 2015/07/29 |
| 10 anni di aggiornamento | 2016/08/01 |

Versioni del server

| Versione | Data |
|----------------|------------|
| NT 3.1 | 1993/08/01 |
| NT 3.5 | 1994/09/01 |
| NT 3.51 | 1996/06/14 |
| NT 4.0 | 1995/07/29 |
| 2000 | 2000/02/17 |
| Server 2003 | 2003-04-01 |
| Server 2003 R2 | 2006-12-01 |
| Server 2008 R2 | 2009-07-01 |
| Server 2012 | 2012-08-01 |
| Server 2012 R2 | 2013/10/01 |
| Server 2016 | 2016/09/01 |

Versioni mobili

| Versione | Data |
|----------------|------------|
| Pocket PC 2000 | 2000/04/19 |
| Pocket PC 2002 | 2001-10-01 |
| Mobile 2003 | 2003-06-23 |
| Mobile 2003 SE | 2003-03-24 |
| Mobile 5.0 | 2005-05-09 |
| Mobile 6.0 | 2007-01-03 |
| Mobile 6.1 | 2008-04-01 |
| Mobile 6.5 | 2009-05-11 |
| Telefono 7 | 2010-10-21 |
| Telefono 7.5 | 2011-09-27 |
| Telefono 7.8 | 2013/02/01 |
| Telefono 8 | 2012/10/29 |
| Telefono 8.1 | 2014/08/15 |
| 10 mobile | 2016/11/20 |

Examples

Informazione

La storia

Microsoft Windows è un sistema operativo disponibile in 137 lingue, scritto in C, C ++ e Assembly, può essere installato su piattaforme "ARM, IA-32, Itanium, x86-64, DEC Alpha, MIPS, PowerPC"

la prima versione era il 20 novembre 1985 come Windows 1.0

con circa 2 miliardi di PC (personal computer) windows è il sistema operativo più utilizzato

Leggi Iniziare con Windows online: <https://riptutorial.com/it/windows/topic/1710/iniziare-con->

windows

Capitolo 2: Porte seriali

introduzione

L'uso delle porte seriali su Windows può essere un po' complesso. Questa sezione della documentazione spiegherà nel tempo tutto ciò che riguarda l'uso di DCB, CreateFile (), eventi portuali e comunicazione seriale asincrona.

Examples

Elenco di tutte le porte seriali.

Spesso è necessario ottenere tutte le informazioni sulle porte seriali da Windows, si consiglia di dare all'utente una scelta di porte da aprire o controllare se il dispositivo è connesso.

Inoltre, alcune porte non possono essere aperte semplicemente usando una stringa "COMx" e devono essere aperte usando un nome di dispositivo. Alcune versioni precedenti di Windows non possono aprire le porte denominate "COMxx" quando il numero di porta è 10 o superiore. Quindi, utilizzare il numero di dispositivo è un modo sensato per identificare la porta con Windows.

Le informazioni sulla porta seriale sono conservate nel registro sotto la chiave

`HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\SERIALCOMM`

Questo esempio in C mostra come elencare tutte le porte seriali e come è possibile ottenere il nome del dispositivo da utilizzare per la chiamata a CreateFile.

Il metodo è abbastanza semplice:

- Aprire il registro in `HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\SERIALCOMM`
- Enumerare tutte le chiavi che troviamo lì. Il nome del dispositivo è il nome della chiave e il 'nome visualizzato' è il valore chiave.

Come spesso accade con la programmazione di Windows, gran parte dello sforzo è dedicato al controllo degli errori. Può sembrare un po' sciocco, ma non esiste un modo sicuro per evitarlo.

```
#define WIN32_LEAN_AND_MEAN // excludes stuff frokm windows.h that we won't need here.
#include <Windows.h>
#include <string.h>
#include <tchar.h>
#include <malloc.h>

void ShowErrorFromLStatus(LSTATUS lResult)
{
    LPTSTR psz;
    FormatMessage (FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        lResult,
        0,
        (LPTSTR) &psz,
```



```

        1024,
        NULL);

    _tprintf(_T("Windows reports error: (0x%08X): %s\n"), lResult, (psz) ? psz :
_T("(null)"));
    if (psz)
    {
        LocalFree(psz);
    }
}

int main()
{
    DWORD nValues, nMaxValueNameLen, nMaxValueLen;
    HKEY hKey = NULL;
    LPTSTR szDeviceName = NULL;
    LPTSTR szFriendlyName = NULL;
    DWORD dwType = 0;
    DWORD nValueNameLen = 0;
    DWORD nValueLen = 0;
    DWORD dwIndex = 0;

    LSTATUS lResult = RegOpenKeyEx(HKEY_LOCAL_MACHINE, L"HARDWARE\\\\DEVICEMAP\\\\SERIALCOMM", 0,
KEY_READ, &hKey);
    if (ERROR_SUCCESS != lResult)
    {
        printf("Failed to open key '\\HARDWARE\\\\DEVICEMAP\\\\SERIALCOMM\\' \n");
        ShowErrorFromLStatus(lResult);
        return 1;
    }

    lResult = RegQueryInfoKey(hKey, NULL, NULL, NULL, NULL, NULL, NULL,
        &nValues, &nMaxValueNameLen, &nMaxValueLen, NULL, NULL);

    if (ERROR_SUCCESS != lResult)
    {
        _tprintf(_T("Failed to RegQueryInfoKey()\n"));
        ShowErrorFromLStatus(lResult);
        RegCloseKey(hKey);
        return 2;
    }

    szDeviceName = (LPTSTR)malloc(nMaxValueNameLen + sizeof(TCHAR));
    if (!szDeviceName)
    {
        _tprintf(_T("malloc() fail\n"));
        RegCloseKey(hKey);
        return 3;
    }

    szFriendlyName = (LPTSTR)malloc(nMaxValueLen + sizeof(TCHAR));
    if (!szFriendlyName)
    {
        free(szDeviceName);
        _tprintf(_T("malloc() fail\n"));
        RegCloseKey(hKey);
        return 3;
    }

    _tprintf(_T("Found %d serial device(s) registered with PnP and active or available at the
moment.\n"), nValues);

```

```

for (DWORD dwIndex = 0; dwIndex < nValues; ++dwIndex)
{
    dwType = 0;
    nValueNameLen = nMaxValueNameLen + sizeof(TCHAR);
    nValueLen = nMaxValueLen + sizeof(TCHAR);

    lResult = RegEnumValueW(hKey, dwIndex,
        szDeviceName, &nValueNameLen,
        NULL, &dwType,
        (LPBYTE)szFriendlyName, &nValueLen);

    if (ERROR_SUCCESS != lResult || REG_SZ != dwType)
    {
        _tprintf(_T("SerialPortEnumerator::Init() : can't process registry value, index:
%d\n"), dwIndex);
        ShowErrorFromLStatus(lResult);
        continue;
    }
    _tprintf(_T("Found port \'%s\': Device name for CreateFile(): \'\\.\.%s\'\n"),
szFriendlyName, szDeviceName);
}

free(szDeviceName);
free(szFriendlyName);
RegCloseKey(hKey);
return 0;
}

```

Uscita del programma sul mio laptop:

```

Found 1 serial device(s) registered with PnP and active or available at the moment.
Found port 'COM23': Device name for CreateFile(): '\\.\Device\BthModem0'

```

Leggi Porte seriali online: <https://riptutorial.com/it/windows/topic/10700/porte-seriali>

Capitolo 3: Programmazione delle estensioni del menu di contesto File / Cartella desktop personalizzate

Examples

Utilizzando le voci del Registro di sistema

Funziona bene per il menu di scelta rapida con clic singolo a livello singolo. Tutto ciò che devi fare è creare una voce di registro sotto Classes Root HKEY_CLASSES_ROOT per un'estensione specifica. Se vuoi creare una voce per tutti i tipi di file scegli * else scegli estensione come .pdf ecc.

```
var regmenu = Registry.ClassesRoot.CreateSubKey("*\\shell\\hello");
if (regmenu != null)
    regmenu.SetValue("", "Hello World");
var regcmd = Registry.ClassesRoot.CreateSubKey("*\\shell\\hello\\command");
if (regcmd != null)
    regcmd.SetValue("", "Do something" );
```

Questo funziona bene per i menu di un livello

Un semplice esempio è su <http://www.codeproject.com/KB/cs/appendmenu.aspx?msg=2236729>

Utilizzo di ShellSharp

Quando sono necessari menu a più livelli, con più parametri, SharpShell viene in soccorso. <https://github.com/dwmkerr/sharpshell> ha un enorme numero di esempi e funziona perfettamente anche per i menu di contesto personalizzati a livello singolo o multi livello.

La cosa fondamentale è creare una classe con attributi [ComVisible (true)] e [COMServerAssociation (AssociationType.AllFiles)] e ereditare la classe from SharpContextMenu che implementa le funzioni CanShowMenu e CreateMenu ed è necessario registrare l'assembly tramite lo strumento regasm o ServerRegistrationManager che Sharkshell creator consiglia

```
[ComVisible(true)]
[COMServerAssociation(AssociationType.AllFiles)]
public class AdvancedContextMenu : SharpContextMenu
{
    protected override bool CanShowMenu()
    {
        // We can show the item only for a single selection.
    }
    protected override ContextMenuStrip CreateMenu()
    {
        // Create the menu strip.
    }
}
```

```
var menu = new ContextMenuStrip();  
... add any level of ToolStripMenuItems and add them to menu  
return menu  
}  
}
```

Maggiori dettagli possono essere ottenuti su <https://github.com/dwmkerr> e <http://www.codeproject.com/Articles/512956/NET-Shell-Extensions-Shell-Context-Menus>

Leggi Programmazione delle estensioni del menu di contesto File / Cartella desktop personalizzate online: <https://riptutorial.com/it/windows/topic/5423/programmazione-delle-estensioni-del-menu-di-contesto-file---cartella-desktop-personalizzate>

Titoli di coda

| S. No | Capitoli | Contributors |
|-------|---|---|
| 1 | Iniziare con Windows | Community , Derpcode , duskwuff , H. Pauwelyn , Mark Hurd , Pro-Fun , Pseudonym Patel |
| 2 | Porte seriali | Michaël Roy |
| 3 | Programmazione delle estensioni del menu di contesto File / Cartella desktop personalizzate | Pooran |