

Sign up for our free weekly **Mobile Newsletter**.



[articles](#) [Q&A](#) [forums](#) [lounge](#)

Search for articles, questions, tips



Another serial port enumerator



Joaquín M López Muñoz, 9 Sep 2005

Rate:

★★★★☆ 4.38 (26 votes)

A library for enumeration of serial ports that works on 9x, NT 4.0 and 2000, XP and CE platforms



Is your email address OK? You are signed up for our newsletters but your email address is either unconfirmed, or has not been reconfirmed in a long time. Please [click here](#) to have a confirmation email sent so we can confirm your email address and start sending you newsletters again. Alternatively, you can [update your subscriptions](#).

[Download source files - 6.47 Kb](#)

[Download demo project - 8.28 Kb](#)

Introduction

ListPorts is a function that lists all serial ports available on the system, along with some descriptive text suitable to be shown to the user in place of the somewhat terse "COM1", "COM2", etc. You can see an example of this on the "Port" combobox at the system Modem properties dialog box.

[PJ Naughter](#) has already wrote some code with this very purpose (see [EnumSerialPorts](#) at CodeProject, or click [here](#) for the latest version of his library). **ListPorts** has, IMHO, some advantages with respect to PJ Naughter's approach:

- it does not depend on MFC,
- it seems to handle non-standard ports (virtual, infrared) somewhat better,
- descriptive text is provided for each port,
- works for Windows CE.

Anyway, we wouldn't like to be blamed for proselytism: please compare both approaches and make your own choice.

Usage

listports.h header provides the following C language definitions:

[Hide](#) [Copy Code](#)

```
typedef struct
{
    LPCTSTR lpPortName;
```

```

    LPCTSTR lpFriendlyName;
    LPCTSTR lpTechnology;
}LISTPORTS_PORTINFO;

typedef BOOL (CALLBACK* LISTPORTS_CALLBACK)(LPVOID lpCallbackValue,
                                             LISTPORTS_PORTINFO* lpPortInfo);

BOOL ListPorts(LISTPORTS_CALLBACK lpCallback, LPVOID lpCallbackValue);

```

LISTPORTS_PORTINFO holds information about a particular serial port: **lpPortName** holds the typical "COMn" string with which one can get a handle to the port via **CreateFile()**, whereas **lpFriendlyName** holds a fuller description of the port (vg., "Infrared Communications Port (COM4)"). On those systems where it is supported, **lpTechnology** names the technology upon which the serial port operates: typical values are "BIOS", "INFRARED", "USB", etc.

LISTPORTS_CALLBACK defines a user-supplied callback routine that is provided the information on each serial port available on the system on successive calls from **ListPorts**. Do in your callback whatever that suits your needs: dump the information on to the console, plug it into a listbox or store it for later use. Please note, however, that the strings stored on the **LISTPORTS_PORTINFO** are not to be referenced after the callbacks returns: so, if you plan on storing the values you should make private copies of them instead.

If your callback returns **FALSE**, the enumeration is aborted.

ListPorts accepts an additional parameter named **lpCallbackValue**. This is treated opaquely by the library and passed to your callback, so that you can use it for your particular purposes (telling between different invocations to **ListPorts** or storing a pointer to some object responsible of using the results, for instance). This is a standard technique used in many callback-based APIs, anyway.

Please see the demo project for a particularly simple example of the use of the library.

Cross-platform issues

ListPorts works on the following OSs:

- Windows 95, 98, ME
- Windows NT 4.0
- Windows 2000, XP
- Windows CE

But there are several shortcomings depending on the operating system. On NT 4.0, we haven't been able to locate the description text for the serial ports (actually we suspect NT 4.0 does not have these): instead, the bare "COMn" strings are supplied. The **lpTechnology** field is not available for NT 4.0 and Windows CE platforms.

Windows CE devices have vendor-specific customizations of the operating system low-level layers which may not adjust to the algorithm used by **ListPorts**: our practical tests have shown, however, that the probability of **ListPorts** missing to locate some ports in Windows CE is very low.

Unicode

The code compiles and works just fine in Unicode, with macros **UNICODE** and **_UNICODE** defined.

Technical

If you don't have curiosity about the internals of the library, skip this section.

ListPorts finds the information on the serial ports available on the system by scanning the registry. On Windows 9x platforms, information on installed devices is stored under the **HKEY_LOCAL_MACHINE\Enum** key. The enumeration tree has three levels, and devices are described at the deepest one. For instance, a standard UART serial port built into the

motherboard and recognized by the BIOS could be stored like this:

[Hide](#) [Copy Code](#)

```
HKLM\ENUM
|-BIOS
|-*PNP0501
|-0D (or any other value, this is not important for us)
  . CLASS= "Ports"
  . PORTNAME= "COM1"
  . FRIENDLYNAME= "Communications Port (COM1)"
```

The value **CLASS** identifies the type of device and is used by **ListPorts** to pinpoint the communications ports.

On Windows 2000/XP, the situation is very similar, except that the enumeration tree is located at **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum**, and **CLASS** is deprecated in favor of a **CLASSGUID** identifier based on COM unique identifiers.

Windows NT 4.0 lacks a fully developed device enumeration tree. Information on serial ports can be found at **HKEY_LOCAL_MACHINE\Hardware\DEVICEMAP\SERIALCOMM**, but no **FRIENDLYNAME** is provided here.

Windows CE stores serial port entries along with other communications ports under **HKLM\Drivers\BuiltIn**.

The method used by the library is discussed in greater detail at the comments of the code in *listports.c* file.

New in version 2.0 (August 2005)

- New **lpTecnology** field.
- Support for Windows CE platforms.

License

This article has no explicit license attached to it but may contain usage terms in the article text or the download files themselves. If in doubt please contact the author via the discussion board below.

A list of licenses authors might use can be found [here](#)


Share

[EMAIL](#)[TWITTER](#)

About the Author



Joaquín M López Muñoz No Biography provided

Spain 

You may also be interested in...

[EnumSerialPorts v1.02 - Enumerating Serial Ports](#)

[How-To Intel® IoT Technology Code Samples: Robot arm in C++](#)

[Serial ports. Enumeration and FIFO control](#)

[How-To Intel IoT Code Samples: Watering System](#)

[Serial Port Enumeration on Windows Mobile](#)

[How-To Intel® IoT Technology Code Samples: Access control in C++](#)

Comments and Discussions

Add a Comment or Question



Search Comments

Go

First Prev Next

Will this work with Vista? new

PlusPlus 7-Sep-07 14:49

I was wondering if your serial port enumerator will work with Windows Vista. If not, do you have recommendations on where I should look for an equivalent solution?
Thanks!

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

1.00/5 (1 vote)

Re: Will this work with Vista? new

Joaquín M López Muñoz 9-Sep-07 8:10

I don't really know, and I don't have any Vista installation available to check it out. Maybe you can try yourself and report your results back?

Joaquín M López Muñoz

Telefónica, Investigación y Desarrollo

Want a Boost forum in Code Project? Vote [here](#) [^]!



[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)**Re: Will this work with Vista?** new**NigelQ** 13-Sep-07 4:18

This does work with Windows Vista (Business)

Nice work!

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)**Goto** new**bradley.jarvis** 10-Dec-06 16:47

I cant believe you use goto statements! but looks ok apart from that.

tigerarmy never die!

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)**Re: Goto** new**Joaquín M López Muñoz** 18-Dec-06 7:37

bradley.jarvis wrote:

I cant believe you use goto statements! but looks ok apart from that.

Well, the code is written in C, so I can't benefit from C++ RAI techniques. This idiomatic use of **goto** is a convenient emulation, in my humble opinion, of the **finally** construct of Java and C#. Consider the following challenge: try to rewrite in C the function **ScanEnumTree** without using **goto** so that the result is as compact and maintainable as in its current form. What do you get?

Thanks for your appreciation of the library.

Joaquín M López Muñoz
Telefónica, Investigación y Desarrollo
Want a Boost forum in Code Project? Vote [here](#)[^]!

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

5.00/5 (1 vote)

**Bug in QueryStringValue()** new**slkirtbzbqmyuk** 13-Dec-05 4:09


Your implementation of QueryStringValue() will loop forever if there's not enough buffer space allocated on the first run - cbStringValue is never incremented or anything.

Just one case where using more for(;;) loops than necessary will do harm sooner or later.

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

2.00/5 (1 vote)

Re: Bug in QueryStringValue() new**Joaquín M López Muñoz** 13-Dec-05 22:53

Your implementation of QueryStringValue() will loop forever if there's not enough buffer space allocated on the first run - cbStringValue is never incremented or anything. 

I don't think so. **cbStringValue** is automatically updated inside **RegQueryValueEx** when there's not enough space allocated --which is indicated by the error code **ERROR_MORE_DATA**.

Just to make sure, I've tried reducing the initial value of **cbStringValue** to 1 and the algorithm correctly loops twice to allocate the needed space. Have you observed otherwise? Have you encountered any specific situation where the algorithm fails? If so, I'll be happy to examine it and try to fix it.

Joaquín M López Muñoz
Telefónica, Investigación y Desarrollo
Want a Boost forum in Code Project? Vote [here](#)[^]!

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)**Enumerating registry** new**Vladimir Afanasyev** 9-Sep-05 21:50

Enumerating Registry for device enumerating is not good style.
I think right way to use SetupXXX and CM_XXX functions.
You can see my articles:


Serial ports. Enumeration and FIFO control
http://www.codeproject.com/system/serial_portsenum_fifo.asp

.NET - Diving into System Programming - Part 1
<http://www.codeproject.com/csharp/DivingSysProg1.asp>

.NET - Diving into System Programming - Part 2
<http://www.codeproject.com/csharp/DivingSysProg2.asp>

.NET - Diving into System Programming - Part 3
<http://www.codeproject.com/csharp/DivingSysProg2.asp>

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)**Re: Enumerating registry** new**Tony Kmoch** 9-Sep-05 22:01

Well, Vladimir, this is one of possible way. By the way, on Windows CE there is no other way. 

Tony

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)**Thanks,** new**Jon Lynes** 8-Feb-05 0:24

Does exactly what it says on the tin (and saved me a load of time).

Has anyone posted a, reliable and working, version that uses the Setup API (and no MFC)?

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

Re: Thanks, new

Joaquín M López Muñoz 8-Feb-05 5:26

Does exactly what it says on the tin (and saved me a load of time).

Glad to hear that 😊

Has anyone posted a, reliable and working, version that uses the Setup API (and no MFC)?

None that I know of, but PJ Naughter's library seems to be easily deMFCfiable. The only important dependence on MFC seems to be **CUIIntArray**.

Joaquín M López Muñoz

Telefónica, Investigación y Desarrollo

Want a Boost forum in Code Project? Vote [here](#) [^]!

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)



dialog based applicaton interaction new

moto muzphee 6-Jun-04 19:40

how can interact with your code with my dialog based application .i mean , if have to get the motor revolutions per minute in my dialog based application in an edit box ,where should i use your calsses (in my application code)

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

1.00/5 (1 vote)

Claim/release Serial port new

orfeas29 13-Apr-04 19:55

This is a general question regarding serial ports..

How do you actually release and claim a serial port e.g. COM5.

I have implemented a VirtualSerialPort and everytime I use a different port e.g. COM5, COM6

the ports stay registered and I cannot use them next time I reboot the system as they are registered in the registry.

Is there a way of releasing those ports without having to delete them from the registry?

Anyone have a clue?

Hint: I want to change the status of the COM port in the "Port Settings->Advanced->Com Port Number" from "(in use)"

into a clear field for each COM port

P.S. CreateFile does not give you a handle if e.g. COM5 port was used by a device which is not present. So this port must be released in some way first..

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

Re: Claim/release Serial port new**Anonymous 4-Oct-05 12:52**

guess you and I are the only ones with this problem!
I was wondering the same thing.
aGGraVaTOr



[Reply](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#) | [Edit](#) · [Delete](#)

Running on a laptop new**rrjimlad 5-Dec-03 2:39**

Hi There,
I've found this code extremely helpful and useful, but I did have one niggle.
I've tried running it on a couple of laptops (one Windows 2000 the other XP) and it seems to list every serial port that the registry has ever known, not just what's currently available.
(For example, this is the result from the Windows XP machine :
"COM5" "Communications Port (COM5)"
"COM1" "Qatech PCMCIA Serial Port (COM1)"
"COM7" "Qatech PCMCIA Serial Port (COM7)"
"COM1" "Qatech USB Serial Port (COM1)"
"COM1" "Prolific USB-to-Serial Comm Port (COM1)"

Here I have 3 devices claiming to be Com1.
Only one of these devices is presently Com1 - the Qatech USB Serial Port, although the others have been in the past.
But how can I tell which one is presently Com 1 ?
Any suggestions would be very welcome,

RR

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

Using registry works much better new**conrad Braam 20-Oct-03 21:50**

Your function does not correctly enumerate "virtual" devices like Ethernet extender or Ethernet serial servers more notably from Moxa (see : http://www.moxa.com/solution/serial_to_ethernet.htm). Doing so thru the registry works fine.

Just so U know, I have been using the registry for 7 years now in production code >6000 coppies, and it works just dandy on all proper 32-bit (not necesarily 95/98) machines.

Conrad - conradb@adroit.co.za
Always do badly to start off, that way when you get the hang of it suddenly, everyone is surprised.

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

5.00/5 (1 vote)

Re: Using registry works much better new**Joaquín M López Muñoz 20-Oct-03 22:34**

Hi Conrad,

Why not consider posting your code here to CodeProject and/or modifying my library to support those virtual devices? You'd be doing a great service to the CP community.

Joaquín M López Muñoz
Telefónica, Investigación y Desarrollo

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

Re: Using registry works much better new

conrad Braam 20-Oct-03 22:49

OK, well this is my blood, but it is good stuff.

```
// ----- EnumerateSerialPorts -----  
-----  
// PURPOSE: Retrieve hardware configuration from registry instead of  
// letting  
// the user guess what ports he has available.  
//  
LONG EnumerateSerialPorts (char *deviceName, DWORD maxLen, DWORD index)  
{  
    CHAR RegPath[MAX_PATH] = "HARDWARE\\\\DEVICEMAP\\\\SERIALCOMM";  
    HKEY hKey;  
    HKEY hKeyRoot = HKEY_LOCAL_MACHINE;  
    DWORD retCode;  
    CHAR ClassName[MAX_PATH] = ""; // Buffer for class name.  
    DWORD dwcClassLen = MAX_PATH; // Length of class string.  
    DWORD dwcSubKeys; // Number of sub keys.  
    DWORD dwcMaxSubKey; // Longest sub key size.  
    DWORD dwcMaxClass; // Longest class string.  
    DWORD dwcValues; // Number of values for this key.  
    CHAR valueName[MAX_VALUE_NAME] ;  
    DWORD dwcValueName = MAX_VALUE_NAME;  
    DWORD dwcMaxValueName; // Longest Value name.  
    DWORD dwcMaxValueData; // Longest Value data.  
    DWORD dwcSecDesc; // Security descriptor.  
    FILETIME ftLastWriteTime; // Last write time.  
    DWORD dwType;  
    DWORD retValue;  
    DWORD cbData;  
  
    // Use RegOpenKeyEx() with the new Registry path to get an open handle  
    // to the child key you want to enumerate.  
    retCode = RegOpenKeyEx (hKeyRoot,  
        RegPath,  
        0,  
        KEY_ENUMERATE_SUB_KEYS |  
        KEY_EXECUTE |  
        KEY_QUERY_VALUE,  
        &hKey);  
  
    if (retCode != ERROR_SUCCESS)  
        return(FAILED);
```

```

// Get Class name, Value count.
RegQueryInfoKey ( hKey, // Key handle.
ClassName, // Buffer for class name.
&dwcClassLen, // Length of class string.
NULL, // Reserved.
&dwcSubKeys, // Number of sub keys.
&dwcMaxSubKey, // Longest sub key size.
&dwcMaxClass, // Longest class string.
&dwcValues, // Number of values for this key.
&dwcMaxValueName, // Longest Value name.
&dwcMaxValueData, // Longest Value data.
&dwcSecDesc, // Security descriptor.
&ftLastWriteTime); // Last write time.

// Enumerate the Key Values
cbData = maxlen ;
dwcValueName = MAX_VALUE_NAME;
valueName[0] = '\\0';

retValue = RegEnumValue (hKey, index, valueName,
&dwcValueName,
NULL,
&dwType,
(BYTE *)&deviceName[0],
&cbData);

RegCloseKey (hKey); // Close the key handle.
if(dwType == REG_SZ && retValue == (DWORD)ERROR_SUCCESS)
return(SUCCESS);
else
return(FAILED);
} // EnumerateSerialports

```

As an example, to fill a list-box I provide the following function by way of example of use of the above function.

```

// ----- FillSerialCBox -----
-----
// the list box contains the strings that describe each serial comm
port available
// the port names themselves are retrieved when destroying the combo
box
void FillSerialCBox(CComboBox * cBox, LPCTSTR currentselection)
{
DWORD count;
LONG retCode;
CHAR portname[MAX_PORT_NAME];

ASSERT(cBox->m_hWnd!=0);
cBox->ResetContent();
count = 0;
while (TRUE)
{
retCode = EnumerateSerialPorts(portname, sizeof(portname), count);
if( retCode != SUCCESS)
break;
cBox->AddString(portname);
cBox->SetItemData(count, count);
if (strcmp(portname,currentselection)==0)

```

```
cBox->SetCurSel(count);  
count++;  
}  
} // FillSerialCBox
```

For a demo, go to codeguru, my project http://www.codeguru.com/network/mod_rssim.html and download and run it in a demo app, more work though.

Conrad - conradb@adroit.co.za


Always do badly to start off, that way when you get the hang of it suddenly, everyone is surprised.

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

5.00/5 (1 vote)

Re: Using registry works much better new

DavidCrow 13-Feb-09 22:57

Opening and closing the registry key that many times seems a bit inefficient. Of course, there shouldn't be more than a few serial ports anyhow. 

"Old age is like a bank account. You withdraw later in life what you have deposited along the way." - Unknown

"Fireproof doesn't mean the fire will never come. It means when the fire comes that you will be able to withstand it." - Michael Simmons

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

Detecting devices new

joelparker 18-Jun-03 3:07


How can you tell if something is actually connected to a serial or usb port? If I call create file, open, then write and nothing is connected to the device I have to wait for a timeout to find out, is there a better way of "autosensing" that my hardware is connected to a specified port?

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

2.00/5 (2 votes)

Re: Detecting devices new

Joaquín M López Muñoz 18-Jun-03 3:14

There's no autosensing API as far as I know. Detecting whether a device is connected really depends on what kind of device you expect to be dealing with. In the usual case of a modem, you can send an "AT\r" and wait for an answer. With **SetCommTimeouts** you can control how long you're waiting, so that the delay is not annoying to the user (a second or so should suffice). 

Joaquín M López Muñoz
Telefónica, Investigación y Desarrollo

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

5.00/5 (1 vote)

Great! new

Simon Hofverberg 8-Apr-03 1:18

Thanks a lot! It enumerates all my ports of different makes (PCI-extension ports) on different OSs (including XP). Other articles here on CP failed.

/Simon

This is not a signature.

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

it work on XP, but... new

Anonymous 21-Jun-02 0:50

i tested it and it works very well on XP.
but i'd like to know how to do an enumerator of port USB, can u help me?
thanks.

yli

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#) | [Edit](#) · [Delete](#)

2.00/5 (1 vote)

it work on XP, but... new

Anonymous 21-Jun-02 0:49

i tested it and it works very well on XP.
but i'd like to know how to do an enumerator of port USB, can u help me?
thanks.

yli

[Reply](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#) | [Edit](#) · [Delete](#)

2.00/5 (2 votes)

Not so good... new

Christian B 25-Jul-01 21:21

Enumerating the registry is not the way to get a list of devices...
You should use the SetupAPI's instead, check the ddk for samples.
Or I could send you a better one.

// christian

[Reply](#) · [Email](#) · [View Thread](#) · [Permalink](#) · [Bookmark](#)

1.43/5 (4 votes)

[Refresh](#)

[1](#) [2](#) [Next »](#)

[General](#) [News](#) [Suggestion](#) [Question](#) [Bug](#) [Answer](#) [Joke](#) [Praise](#) [Rant](#) [Admin](#)

