Any way to determine speed of a removable drive in windows?

Asked 16 years, 2 months ago Modified 16 years, 2 months ago

Viewed 1k times

4

Is there any way to determine a removable drive speed in Windows without actually reading in a file. And if I do have to read in a file, how much needs to be read to get a semi accurate speed (e.g. determine whether a device is USB2 or USB1)?

EDIT: Just to clarify, USB2 and USB1 were an example. These could be Compact Flash, could be SSD, could be a removable drive. And I am trying to determine this as fast as possible as it has a real effect on the responsiveness of the application.

EDIT: Should also clarify, this has to be done programatically. It will probably be done in C++.

EDIT: Boost answer is kind of what I was looking for (though I haven't written any WMI in C++). But I need to know what properties I have to check to determine relative speed. I don't need exact speed (like I said about the difference in speed between USB1 and USB2), but I need to know if it is going to be SLLOOOOWWW.

c++ windows removable-storage

Share

Improve this question

Follow

edited Oct 9, 2008 at 15:30

asked Oct 8, 2008 at 15:42

Kris Erickson

33.8k ● 26 ● 121 ● 179

WMI is usually slow as a method for fetching information. About a second/query you do. – Vincent Nov 19, 2008 at 2:37

7 Answers

Sorted by: Highest score (default)

▲

2

▼

✓

WMI - Physical Disks Properties is an article I found which would at least help you figure out what you have connected. I foresee things heading toward tables equating particular manufacturers and models to speeds, which is not as simple a solution as you may have hoped for.

Share Improve this answer

Follow

answered Oct 9, 2008 at 14:04

bugmagnet

7,749 ● 8 ● 72 ● 136

Wouldn't work too well I'm afraid. THe same USB2 disk may be a lot slower if connected via a USB1 hub. And that

happens to be exactly when you should warn the user.
– MSalters Oct 10, 2008 at 15:17

2

You may have better results querying the operating system for information about the hardware rather than trying to reverse engineer it from data transfer timing information.

For example, identical transfer speeds don't necessarily mean the same technology is being used by two devices, although other factors such as seek times would improve the accuracy, if such information is available to your application.

In order to keep the application responsive while this work is done, try doing the calls asynchronously and provide some sort of progress indicator to the user. As an example, take a look at how WinDirStat handles this progress indication (I love the pac-man animation as each directory is analyzed).

Share Improve this answer Follow

answered Oct 8, 2008 at 16:37

J c
6,413 ● 3 ● 31 ● 29

0

Several megabytes, I'd say. Transfer speeds can start out slow, and then speed up as the transfer progresses. There are also variations because of file sizes (a single 1GB file will transfer much faster than 1GB of smaller files).

Best way to do that would be to copy a file to/from the device, and time how long it takes with your code. USB1 speed is 11Mb/s (I think), and USB2 is 480Mb/s (note those are numbers for the whole bus, not each port, so multiple devices on the same bus will change the actual numbers).

Share Improve this answer Follow

answered Oct 8, 2008 at 16:09

TheSmurf

15.6k ● 3 ● 41 ● 48

0

Try TerraCopy and copy one large file ~400mb - 500mb from device and to the device and you'll see the speed.

Share Improve this answer Follow

answered Oct 8, 2008 at 16:23

daniels

19.2k ● 34 ● 109 ● 173

0

In Windows you can determine if a connected USB device is USB2 by selecting View -> "Devices by Connection" from the Device Manager and then checking to see if the device is under a USB2 controller (USB2 Enhanced Host Controller).

Note that this doesn't mean your device will actually perform at the higher speeds though, you would still need actual throughput tests for that. The Sisoft Sandra benchmarking software lists removable hard drives as supported in its feature list.

EDIT: Due to clarification in original question, I have submitted a new answer.

Share Improve this answer

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 ● 1

answered Oct 8, 2008 at 16:17

J c

6,413 ● 3 ● 31 ● 29

0

Consider the number of things that could affect data transfer speed:

- The speed of the bus used to connect the device to the system. This is unlikely to be your bounding factor unless it's connected via USB1.
- For hard drives, rotational speed and seek time matter. 7200 RPM drives will read and write blocks of data faster than 5400 RPM drives.
- Optical and magnetic drives usually spin down when not in use, so the first access will take orders of magnitude more than the second access.
- The filesystem used on the particular device.
- Caching of data and filesystem metadata. The less metadata is cached, the more a magnetic or optical drive has to seek to figure out where the data is.

- Data access pattern. Accessing a small number of large, contiguous files is almost always faster than accessing a large number of small files scattered around the disk.
- File system fragmentation

You might be able to work up some heuristics based on the various characteristics of the devices you expect to see, but in general there's no good way to figure out transfer speed for a particular combination of bus, media, filesystem, and data access pattern without actually measuring it. If you decide to measure, try to simulate your final access pattern as closely as possible.

Share Improve this answer Follow

answered Oct 8, 2008 at 17:23

Commodore Jaeger

33.3k ● 4 ● 57 ● 44

0

I'm going to borrow Raymond Chen's crystall ball and say that you really don't want this. You probably want to use asynchronous I/O. If you do not get the result of your I/O within a second, you want to check how much did happen. Take the inverse of that number, and you have a good estimate to quote to the user.

If nothing happened after a second, you may be in for a surprise. But even that can happen. For instance, a harddisk may need a second to spin up. Just poll every second until something has happened.

Share Improve this answer

Follow

answered Oct 10, 2008 at 15:21

MSalters

179k ● 11 ● 164 ● 368