

Re: GetDirectories Performance

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2006-04/msg02463.html>

- *From:* Frank Rizzo <nospam@xxxxxxxxxx>
 - *Date:* Sat, 22 Apr 2006 22:07:12 -0700
-

Tom Scales wrote:

"Herfried K. Wagner [MVP]" <hurf-spam-me-here@xxxxxx> wrote in message <news:OJXiDUMZGHA.1200@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

"Tom Scales" <tjscales@xxxxxxxxxx> schrieb:

I'm writing a VB.NET 2003 program that uses a Treeview to display the drive structure on the computer. I am having a major problem with performance. There are many files on one drive (over a million) and it is killing me. For example, one directory has a structure:

Instead of populating the whole treeview control on startup, only add the nodes on the first level and check if the folders contain subfolders. If the latter is the case, add a dummy subnode to the node representing the folder. Then catch the node expand event and replace the dummy node with nodes for the actual files and folders contained in the folder. This should lead to acceptable performance and memory usage would be much lower than populating the whole control. In addition, in many cases it's very unlikely that the user will expand every single node and thus much less memory will be occupied by your application in total.

--
M S Herfried K. Wagner
M V P <URL:<http://dotnet.mvps.org/>>
V B <URL:<http://classicvb.org/petition/>>

Unfortunately, that is essentially what I am doing. I am adding enough nodes to tell me if I need to add the + sign. That's where I get bitten, because the directory UNDER the one I am working with has 30,000+ files. GetDirectories must search every file to see if it is a directory. Very inefficient code on MS' part.

Re: GetDirectories Performance

Tom, it sounds like you would be better served by doing P/Invoke via FindFirst and FindNext API calls. The way they work is through one file at a time, thus, you'll have much more control over your operation.

In addition, another thing that maybe slowing you down is the actual treeview (which is also very inefficient). I'd advise to you to devise a quick test to load 30,000 nodes and see how fast it is loading. You'll be surprised at how slow it will be. To get around this problem I went with a 3rd party tree list control from <http://www.bennet-tec.com/> called TList. Their claim to fame is the speed and based on my usage it is not idle talk. It truly is pedal to the metal.

Regards