

## Re: Get regular expression

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-06/msg03634.html>

---

- *From:* "Mike" <[msgrinnell@xxxxxxxxxxx](mailto:msgrinnell@xxxxxxxxxxx)>
  - *Date:* 22 Jun 2006 13:09:45 -0700
- 

Hi Kevin,

I have been working on this problem on and off for maybe 9 months. New ICD9 codes are released in October so I have a few months to get a solution in place. I am going to see if there is some other way to get the owners of the webservices to offer something different.. I'd like to get the data more structure (i.e do something similar to this: <http://www.developerfusion.co.uk/show/4633/>) Any idea if retrieving data structured like in the article would fit well with the C# tree structure?

I will also look at your suggestion, but I had previously tried to implement something similar with a two-dimensional array and the problem was with performance. Very slow for those entries with several hundred rows such as "Biopsy"...

Mike

Kevin Spencer wrote:

Hi Mike,

I fiddled with this problem using regular expressions for entirely too long last night, and finally came to the conclusion that regular expressions aren't going to provide what you need in this case. As you discovered, your regular expression solution, which was about as close as one could get to something that works with regular expressions, can't identify an unknown pattern and then match that, which is essentially what you tried valiantly to do. I have to give you credit for creativity!

Of course, this doesn't bring you any closer to a solution, so I gave that some thought as well. It seems to me that you're looking for some sort of recursive nested looping function. Once the data is sorted alphabetically, it's basically a matter of comparing each line with the line that follows. If you can be sure that the pattern will break on a word break (space), the

## Re: Get regular expression

task becomes easier. I'll try to sketch something out along the lines of what I'm thinking, and you can see what you think and perhaps flesh it out:

This is the comparison method. It does a char-by-char comparison