

## Re: Get regular expression

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-06/msg03540.html>

---

- *From:* "Kevin Spencer" <[uce@xxxxxxx](mailto:uce@xxxxxxx)>
  - *Date:* Thu, 22 Jun 2006 07:26:11 -0400
- 

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 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 of 2 strings, returning the number of chars that match from the beginning of the first string. If you can be sure that your nodes will break on spaces, you could optimize this by using a word-by-word comparison.

string[] items;

```
// Compare each char of a string in an array with
// Each char of the next string in the array, and
// return the length of the matched string.
int Match(int index, int length)
{
    int maxLength = (length < 0 ? items[i].Length : length);
    if (index == items.Length - 1) return 0;
    for (int i = 0; i < maxLength; i++)
    {
        if (items[index][i] != items[index + 1][i]) return i - 1;
        if (i == items[index + 1].Length - 1) return i;
    }
}
```

## Re: Get regular expression

Now, what I would do with this is, since you want to create a hierarchical tree, use the System.Xml Namespace, and an XmlDocument class to create your in-memory structure. You could certainly create your own lightweight hierarchical node type, but this way, if the need ever arises (and it probably will) that you want to transform your data to another format, you have the ability to use the XmlDocument class as an XML Document, and transform it any way you like (including as pure XML text), one of the beauties of XML.

Once you've created your root node, you loop through the "array", calling the Match method for each item in the "array" until it returns 0. You initialize it by passing -1 to it, which indicates that it compares the entire length of the first string. After that, you pass the return value from the first comparison, which gives you the length of the first child node. At this point you have your first sub-grouping, and your first child node, which is the substring of the first string having the length returned by the first comparison.

If the number of iterations is less than the length of the "array" you start again with the next item in the array, in the same manner as the first. Each pass of this routine returns a "node" and the length of the node value.

You recursively repeat this process for each subset of each node, starting with the length of the node value, and using the substring starting from that point for each element in the subset. This adds a list of nodes to each node, and recursively does the same for each child node of each node, and so on. When you have reached the end of all the strings, you're done.

This is about as elegant a solution as I can come up with. I'll be interested to hear about your final solution.

--

HTH,

Kevin Spencer  
Microsoft MVP  
Professional Chicken Salad Alchemist

I recycle.  
I send everything back to the planet it came from.

"Mike" <msggrinnell@xxxxxxxxxxxx> wrote in message  
[news:1150921410.056758.274750@xx](mailto:news:1150921410.056758.274750@xx)

Hi Kevin,

Since you appear to be rational about the actual objective of groups (communication), I'll try to respond effectively.

## Re: Get regular expression

1. You have a set of data that is pure text, and is either stored in an actual database, or in the text equivalent of a database as a multi-line text document. I can't be exactly sure.

Yes, the text is now stored in a database with webservice front-end.

2. In any case, this data consists multiple single-line entries of text.

More-or-less, yes. I can get the data as such after jumping through some hoops (user enters the first word of the index entry they want which retrieves all the codes that have an index entry that has an Index Entry property starting with that word. Then, before each code can have multiple index entries -- not necessarily all ones they want -- I have to discard the ones that do not start with 'Ablation' for example).

3. The data is stored in such a way that the text represents a hierarchical structure of nodes.

If I understand your comment, yes. Actually, sorted alphabetically, ALL the index entries are in the order they would appear in a tree except for "WITH, BY" and a couple other terms that can become nodes themselves. The issues is that the surgical indexers are used to viewing the data in a certain way and want to continue to view it without all the duplicate text in surrounding index entries.

4. This is achieved by a top-level classification that is repeated in each "record" (line) for every record that falls under it.

Yes, I believe so.

5. Sub-nodes are indicated in the same way by the first text that follows the top-level node text.

Yes. Sub nodes are just the text that repeats across lines after the repeated substring in a larger set of lines has been removed.

6. The node identifier text in the sub-nodes can be identified by comparing

## Re: Get regular expression

it with other records that are under the top-level node. There is no other way to distinguish this text from any other text in the record, other than by comparing it with other records.

Yes, because the actual text has no meta data to identify which is a parent and which is a child.

7. Therefore, the structure of the hierarchy can be inferred by using a recursive procedure that identifies increasingly "deep" sub-nodes within the set of records.

Yes, I believe I can do this. On the old OS390 it was done kind of like this with heavy parsing.

8. (Now here's where I'm a bit fuzzy). Your task is to put all of this into some form of data structure that can be used as an index, probably a hierarchical structure such as a tree.

Yes, exactly, I believe I can setup a recursive algorithm that will populate a tree view control that will represent the codes in a way the surgical coders are used to seeing. They will, for example, be able to explode "ABLATION" and see subnodes of "ENDOMETRIAL (HYSTEROscopic) 68.23"  
"Heart (Conduction Defect) 27.33/2"

Then, upon exploding those nodes any child nodes would be displayed, etc.

Question: Will these records be ordered in any way? IOW, for example, will they be ordered alphabetically? If they are ordered alphabetically, the structure is already present, by virtue of the rules as stated above. Otherwise, it will be necessary to do some form of re-scanning of the data.

In my pre-processing I can sort these alphabetically. As always, the question is really how to eliminate the duplicated text from surrounding lines and correctly place children/parents in relation to each other.

## Re: Get regular expression

Question: Can you tell me what sort of format the end result is supposed to be in? Is it simply a data structure in memory? Or what?

Simply a data structure in memory as the end users want to be able to pull this up on demand as they are coding surgical cases.

Kevin Spencer wrote:

Hi Mike,

As far as Top-Posting is concerned, AFAIK it's still a matter of debate, and as we're talking about Netiquette, not ISO or W3C standards, my personal feeling is that anyone who scolds one about top- or bottom-posting has poor sense of priority. After all, the purpose of groups such as this is communication. I find it far more difficult to deal with poor communication than with the format of a post, but that's just me! ;-)

In your case, you have done a pretty darned good job of communication, and I appreciate that, so I will certainly do all I can to help out! I did have to do a little research into ICD9, but that wasn't hard with Google.

It took me a few minutes of study to figure out (for the most part) what your requirements are. Let me see if I can repeat them back to you in my own words, and ask a couple of questions:

1. You have a set of data that is pure text, and is either stored in an actual database, or in the text equivalent of a database as a multi-line text document. I can't be exactly sure.
2. In any case, this data consists multiple single-line entries of text.
3. The data is stored in such a way that the text represents a hierarchical structure of nodes.
4. This is achieved by a top-level classification that is repeated in each "record" (line) for every record that falls under it.
5. Sub-nodes are indicated in the same way by the first text that follows the top-level node text.
6. The node identifier text in the sub-nodes can be identified by comparing

Re: Get regular expression

it with other records that are under the top-level node. There is no other way to distinguish this text from any other text in the record, other than by comparing it with other records.

7. Therefore, the structure of the hierarchy can be inferred by using a recursive procedure that identifies increasingly "deep" sub-nodes within the set of records.

8. (Now here's where I'm a bit fuzzy). Your task is to put all of this into some form of data structure that can be used as an index, probably a hierarchical structure such as a tree.

Question: Will these records be ordered in any way? IOW, for example, will they be ordered alphabetically? If they are ordered alphabetically, the structure is already present, by virtue of the rules as stated above. Otherwise, it will be necessary to do some form of re-scanning of the data.

Question: Can you tell me what sort of format the end result is supposed to be in? Is it simply a data structure in memory? Or what?

--  
HTH,

Kevin Spencer  
Microsoft MVP  
Professional Chicken Salad Alchemist

I recycle.  
I send everything back to the planet it came from.

"Mike" <msgrinnell@xxxxxxxxxxxx> wrote in message  
[news:1150899561.715476.17480@xx](mailto:news:1150899561.715476.17480@xx)

Must say I get burned in six different ways. Some groups I top post and get scolded. On other groups others people top post and nobody appears to have a problem. I'll top post here.

Given I've been asked for details I'll provide them, but typically nobody wants to wade through them.

## Re: Get regular expression

In the dark ages I had 24,000 lines of ICD9 index entries which got appended with ICD9 codes and were processed one time per year into a big paper report with a tree-like structure by an assembler program on an OS390. An abbreviated example of the report is below for the Ablation entry.

Ablation  
Endometrial (Hysteroscopic) 68.23  
Heart (Conduction Defect) 27.33/2  
With Catheter 37.34/2  
Inner Ear (Cryosurgery) (Ultrasound) 20.79/4  
By Injection 20.72  
Lesion Heart  
By Peripherally Inserted Catheter 37.34

Across my institution in the past there have been multiple "master" copies of ICD9 codes and index entries. The order came down that long-term we will work towards a single copy of ICD9 codes with index entries that will be accessed via webservices. The structure of the data in our old database was as follows (no line breaks -- each entry was one line):

ABLATION ENDOMETRIAL (HYSTEROSCOPIC) 68.23  
ABLATION HEART (CONDUCTION DEFECT) 37.33/2  
ABLATION HEART (CONDUCTION DEFECT) WITH  
CATHETER 37.34/2  
ABLATION INNER EAR (CRYOSURGERY)  
(ULTRASOUND) 20.79/4  
ABLATION INNER EAR (CRYOSURGERY)  
(ULTRASOUND) BY INJECTION 20.72  
ABLATION LESION HEART BY PERIPHERALLY  
INSERTED CATHETER 37.34  
ABLATION LESION HEART ENDOVASCULAR  
APPROACH 37.34  
ABLATION LESION HEART MAZE PROCEDURE  
(COX-MAZE) ENDOVASCULAR  
APPROACH 37.34  
ABLATION LESION HEART MAZE PROCEDURE  
(COX-MAZE) OPEN (TRANS-THORACIC)  
APPROACH 37.33  
ABLATION LESION HEART MAZE PROCEDURE  
(COX-MAZE) TRANS-THORACIC

Re: Get regular expression

APPROACH 37.33  
ABLATION PITUITARY 7.69  
ABLATION PITUITARY BY COBALT-60 92.32  
ABLATION PITUITARY BY IMPLANTATION  
(STRONTIUM-YTTRIUM) (Y) NEC 92.39  
ABLATION PITUITARY BY PROTON BEAM (BRAGG  
PEAK) 92.33  
ABLATION PROSTATE (ANAT = 59.02) BY LASER,  
TRANSURETHRAL 60.21  
ABLATION PROSTATE (ANAT = 59.02) BY  
RADIOFREQUENCY THERMOTHERAPY  
60.97  
ABLATION PROSTATE (ANAT = 59.02) BY  
TRANSURETHRAL NEEDLE ABLATION  
(TUNA) 60.97  
ABLATION PROSTATE (ANAT = 59.02) PERINEAL BY  
CRYOABLATION 60.62  
ABLATION PROSTATE (ANAT = 59.02) PERINEAL BY  
RADICAL CRYOSURGICAL  
ABLATION (RCSA) 60.62  
ABLATION PROSTATE (ANAT = 59.02)  
TRANSURETHRAL BY LASER 60.21  
ABLATION PROSTATE (ANAT = 59.02)  
TRANSURETHRAL CRYOABLATION 60.29  
ABLATION PROSTATE (ANAT = 59.02)  
TRANSURETHRAL RADICAL CRYOSURGICAL  
ABLATION (RCSA) 60.29  
ABLATION TISSUE HEART - SEE ABLATION,  
LESION, HEART 0  
ABLATION VESICLE NECK (ANAT = 60.02) 57.91

The new webservices still have this same index structure except now, for example, "Ablation Vesicle Neck (ANAT = 60.02)" is just a property of code 57.91. The surgical coders still want to view the index entries in a tree structure on demand. Without getting into mind-numbing details, I can jump through some hoops and get back a set of index entries that look like above for ABLATION but they are not formatted in the way the surgical coders desire. I believe I have a recursive algorithm that will work to format these into a tree structure but this algorithm is predicated on being able to find the nodes.

If you look carefully, the root node for entire set of index

## Re: Get regular expression

entries  
above is "ABLATION" (as that is what begins each entry  
and repeats  
across all of them). Subsequently, Endometrial  
(Hysteroscopic) + code  
is a child of ABLATION with no children of its own because  
it is not  
repeated. Next, Heart (Conduction Defect) + code is a node  
with "With  
Catheter + code" as a child of that node because "Heart  
(Conduction  
Defect)" repeats across both those lines.

I have begged the group that now owns the webservice to  
allow me to  
restructure the data but no go (they say that would be  
bastardizing the  
concept of everything being 'code-centric'). I am stuck with  
this and  
also with the demand by the coders that they get the  
formatted tree  
structure to look at when they code.

In general, I think if I do the following I can figure out the  
nodes  
and children:

1. Read index entries until the first word changes.
2. Get the substring that begins the string and is repeated  
elsewhere  
in the string (this is the node).
3. Remove that node and keep processing until the base case  
is hit etc.

If anyone has any better ideas of how to deal with this I  
would be  
thrilled to no end to hear them.

Thanks,

Mike

Kevin Spencer wrote:

I want to access the  
expression "HEART  
(CONDUCTION  
DEFECT)" I'll  
try

## Re: Get regular expression

your suggestion first off in the morning.

First, "HEART (CONDUCTION DEFECT)" is not an expression. That is a substring of the original string. The regular expression is the string `"^(.+)(?=\s*).*\1"` that you are using to get your match. Assuming that "HEART (CONDUCTION DEFECT)" is your match (which it is not), you could call it a match for the regular expression (which may match more than once in a string). But it is a substring of the original string. It may seem picky, but in order to communicate effectively, one must use the right terms. As an example, if I told you that I ate a car for breakfast, would you know that I ate an apple?

Second, the string you posted contains 2 instances of the substring "HEART (CONDUCTION DEFECT)". Do you want to get both of them? If so, what exactly are your pattern-matching rules? A regular expression matches a pattern. Obviously, not all of the strings you will be working with will be:

```
" HEART (CONDUCTION  
DEFECT) 37.33/2 HEART  
(CONDUCTION DEFECT) WITH  
CATHETER 37.34/2 "
```

In fact, probably due to this being a newsgroup, and my using a newsreader, I would doubt that the line breaks in the

## Re: Get regular expression

string are where they are,  
if  
they  
are. And I have to wonder whether the string  
actually begins and ends  
with a  
space.

In other words, you're going to be using a  
regular expression to  
isolate  
substrings of various strings (most  
probably). A regular expression is  
shorthand for a set of rules that defines a  
pattern you're looking  
for.

Whether the strings contain line breaks, for  
example, is important.

Your  
regular expression begins with the caret '^'  
character. This character  
can  
indicate the beginning of a string, or the  
beginning of a line \*or\* a  
string, depending upon what options you  
use. You didn't specify the  
option(s) you're using, so we have no way to  
know.

In addition, your pattern is not likely to work  
in the way you expect.  
for  
example, the following would match:

THIS IS NOT WHAT THIS IS SUPPOSED  
TO BE. (Matches the phrase "THIS IS  
")

And in addition, if there are line breaks, like  
your example (as split  
by  
the newsreader), the matching substring  
would be:

DEFECT) 37.33/2 HEART  
(CONDUCTION DEFECT)

So, can you explain what your rules are, and  
what you are trying to  
match  
here? I'm just guessing that you're parsing



Re: Get regular expression

```
apply
^(.+)(?=\s*).*\1
to
"
HEART
(CONDUCTION
DEFECT)
37.33/2
HEART
(CONDUCTION
DEFECT)
WITH
CATHETER
37.34/2
"
the
expression
is
"HEART
(CONDUCTION
DEFECT)".
How
do
I
gain
access
to
the
expression
(not
the
matches)
at
runtime?
```

```
you want to
access the
expression
"HEART
(CONDUCTION
DEFECT)"
or
the
regex
"^(.+)(?=\s*).*\1"
at
run-time??
dont think
you can get
exactly
```

Re: Get regular expression

the latter  
one though,  
for the  
previous  
one, you  
can use  
named  
capture,  
like

```
^(?<expr>.+) (?=\s*).*\k<expr>
```

and access  
the variable  
"expr" at  
run time?

Xicheng

I want to access the  
expression "HEART  
(CONDUCTION  
DEFECT)" I'll  
try  
your suggestion first off in  
the morning.

Thanks,