

Re: best design for parse

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2007-01/msg00699.html>

- *From:* "Stephany Young" <noone@localhost>
 - *Date:* Mon, 8 Jan 2007 15:13:09 +1300
-

Again you're missing the point.

I think the best thing you can do is post a relatively small sample of the text you are attempting to parse.

While you're doing that, execute the following and observe the results. It demonstrates what I am talking about:

```
Dim _source As String = "On 07/01/2007 the quick brown fox jumps over the  
lazy dog." & Environment.NewLine & _  
"On 08/01/2007 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On Jan/09/2007 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On 10/Jan/2007 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On 11/01/07 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On 01/12/07 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On Jan/13/07 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On 14/Jan/07 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"On 15/01 the quick brown fox again jumps over the lazy dog." &  
Environment.NewLine & _  
"The part number XYZ/72/84 is now discontinued."
```

```
Dim _regex As New  
Regex("\d{2}\d{2}\d{4}|[A-Za-z]{3}\d{2}\d{4}|\d{2}/[A-Za-z]{3}\d{4}|\d{2}\d{2}\d{2}|[A-Za-z]{3}\d{2}
```

```
Dim _candidates As Integer = 0  
Dim _matches As Integer = 0
```

```
Dim _match As Match = _regex.Match(_source)
```

```
While _match.Success  
_candidates += 1
```

Re: best design for parse

```
Console.WriteLine("{0} found at index {1}", _match.Value, _match.Index)
Try
Console.WriteLine("Converted value = {0:yyyy-MM-dd}",
DateTime.ParseExact(_match.Value, New String() {"dd/MM/yyyy", "MM/dd/yyyy",
"MMM/dd/yyyy", "dd/MMM/yyyy", "dd/MM/yy", "MM/dd/yy", "dd/MMM/yy",
"MMM/dd/yy", "dd/MM"}, Nothing, DateTimeStyles.None))
_matches += 1
Catch _ex As Exception
Console.WriteLine(_ex.Message)
End Try
_match = _match.NextMatch()
End While
```

```
Console.WriteLine("{0} candidates found", _candidates)
```

```
Console.WriteLine("{0} matches found", _matches)
```

"GS" <gsmsnews.microsoft.comGS@xxxxxxxxxxxxxxxxxxxx> wrote in message
[news:eFm\\$y5rMHHA.4376@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:eFm$y5rMHHA.4376@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

You are sort of on the same track as mine.

I must first apologize I did not tell you the complete story.

Although the application does not exactly know before hand what format the data may come in, however part of the application allow user to define and record favourite for a website

- to extract by text or html
- header content and format
- record format and date format (that is where the date format mask come in)
- optionally ordinal number for each column or re-ordering
- trailer content and format

For a given batch, at least for the body, date format are uniform

furthermore, the need to make the extract process generic and adaptable to the front end that takes the user definitions, I believe it would be easier to "normalize" date string to "yyyy-mm-dd".

Also the end target for of may not necessarily be SQL database but may be text, pasted to word report. or excel by user

Therefore, I can transform the date format mask to regex in the appropriate format and identifier I can use regex,replace to normalize the date. As a matter of fact the date separator does not have to / but can be space as

Re: best design for parse

long as there are identifiable delimiters around the date string.

I already have code for dealing with regex for dates from prior project.
all I have to do is adapt to the present need

who knows, maybe I taken on a totally offbeat tract

"GS" <gsmsnews.microsoft.comGS@xxxxxxxxxxxxxxxxxxxx> wrote in message
news:%23vnOBJiMHHA.1280@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

thanks for all pitched in so far.

let give it another shot.

looks like an easier way out would be

1. copy the date format string regex string holder and then derive the
relevant regex expression to be used for date normalization later in part

2:

replace the regex string the yyyy to regex year expression with year
identifier

look for yy and replace with 20yy and repeat the step above

replace mmm with the month regex expression associated with month
identifier

replace mm with the 2 digit month regex expression associated with

month

identifier

replace dd with the 2 digit day regex expression assoc. with day
identifier

2. use the resulting regex in regex replace to normalize to yyyy--mm-dd

any problem with the above approach?

"Cor Ligthert [MVP]" <notmyfirstname@xxxxxxxxxx> wrote in message
news:%23Qj7TbWMHHA.3944@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

GS,

Maybe can you avoid this in 2007 and all things like that as
DateTime.ParseExact, but have a look to the nicely by
Microsoft inbuilt
globalization and than the to that related ToString option.

Cor

"gs" <gs@xxxxxxxxxxxxxxxx> schreef in bericht

Re: best design for parse

news:OtrnsPTMHHA.4720@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

let say I have to deal with various date
format and I am give format
string from one of the following

dd/mm/yyyy
mm/dd/yyyy
dd/mmm/yyyy
mmm/dd/yyyy
dd/mm/yy
mm/dd/yy
dd/mmm/yy
mmm/dd/yy
dd/mm

what is the best way to come up a relevant
regex for the incoming

format

string

- a) use two array and statically match
- b) use regex to find the order