

Splitting Long Column Names to Allow Word Wrap in FpSpread

Submitted By: Gary Winey

Last Updated: 2/3/2006

Audience Level: Beginner

Product(s): Spread for Windows Forms 2

Language(s): Visual Basic.NET

Description: Here is a quick way to split long data table column names to allow FpSpread to word wrap the column headers.

Introduction

Sometimes in your application, table column names are very long and the data is very short. An example would be an engineering application with mostly numeric data (for example, MaximumIceLoad for a column of type Int). If you have several columns of such data, you end up with a very wide sheet and very little data. This requires the user to do an inordinate amount of scrolling to see their data. Fortunately, FpSpread allows ColumnHeader WordWrap if there are embedded spaces in the label. The example Visual Basic .NET code shown below demonstrates how to use regular expressions to split the column or row names (or both) when the capitalization changes. With word wrap, the column names are much shorter in width.

If you are new to regular expressions, the best tutorial I know of is at:

www.regular-expressions.info/dotnet.html

The regular expression pattern `(?<=[a-z])(?=[A-Z])|(?<=[A-Z])(?=[A-Z])` matches either capitalization changes or consecutive capital letters. As shown below, the name "MaximumIceLoad" would return "Maximum Ice Load" while "IWorkHard" would return "I Work Hard". Note that the code cannot differentiate intentional consecutive capital letters. "WeUseSQLServer" would return "We Use S Q L Server" and "WorldWarII" would return "World War I I". If you do not need the consecutive capital letter check, use this string instead `(?<=[a-z])(?=[A-Z])`, which only matches when changing from a lower-case letter to an upper-case letter.

Disclaimers

Here are some disclaimers for the example code shown below:

1. You must have the Imports statement.
2. You should not have to set any additional references.
3. The regular expression in the code matches either:
 - a. Capitalization changes from lower-case to upper-case or
 - b. Consecutive capital letters.
4. All consecutive capital letters are split.
For example, "IWorkInIT" produces "I Work In I T";
and "WorldWarII" produces "World War I I".

Code

Here is the first version, which handles all the alphabetic characters.

```
Imports System.Text.RegularExpressions
' Example
Private Function SplitLabel(ByVal text As String) As String
    Dim options As RegexOptions = RegexOptions.Compiled Or _
        RegexOptions.IgnoreCase Or RegexOptions.Multiline
    Return Regex.Replace(text, "(?<=[a-zA-Z])(?=[A-Z])", " ", options)
End Function 'SplitLabel
```

And here is a sample usage:

```
FarPoint.Win.Spread.Column.Label = SplitLabel(FarPoint.Win.Spread.Column.Label)
FarPoint.Win.Spread.Row.Label = SplitLabel(FarPoint.Win.Spread.Row.Label)
```

Here is a second version, which handles numeric characters.

```
For example, "MyColumn2LongName" becomes "My Column2 Long Name".
Private Function SplitLabel(ByVal text As String) As String
    Dim options As RegexOptions = RegexOptions.Compiled Or _
        RegexOptions.IgnoreCase Or RegexOptions.Multiline
    Return Regex.Replace(text, "(?<=\w)(?=[A-Z])", " ", options)
End Function 'SplitLabel
```

Here is a third version, which handles underscores (_) and hyphens (-).

```
Private Function SplitLabel(ByVal text As String, _
    Optional ByVal removeUnderscores As Boolean = False, _
    Optional ByVal removeDashes As Boolean = False) As String
    Dim options As RegexOptions = RegexOptions.Compiled Or _
        RegexOptions.IgnoreCase Or RegexOptions.Multiline
    If removeUnderscores Then text = text.Replace("_", " ")
    If removeDashes Then text = text.Replace("-", " ")
    Return Regex.Replace(text, "(?<=\w)(?=[A-Z])", " ", options)
End Function 'SplitLabel
```

You can only use one of these three in the same file or you get an overloads error because the optional parameters are ignored by the compile when determining uniqueness for a method signature. To the compiler all three functions are viewed as taking a string parameter and returning a string so it considers all three of them to be equivalent.