

ASSIGNMENT #8

The number of characters required to write a given word can differ significantly among different languages. For example, the four-character English word "next" translates as the eight-character "zunächst" in German and the seven-character "proxima" in Spanish. This poses a challenge to developers of world-ready software since it complicates the problem of designing the layout geometry for GUIs that include text-bearing controls. If all of the words that will appear in the GUI are known at design time, one can use the *localizable* property to create a different form design (and so, automatically, a different resX file) for each language/culture version, in which the sizes of text-bearing controls are matched to the lengths of their text strings. In some applications, however, the GUI may contain text-bearing controls whose associated text strings are determined at run time (or in fact may even vary, and therefore may vary in size, at various times during any given execution of the application). In the latter case, two possible approaches to on-the-fly language/culture-sensitive modification of the GUI are: (1) to maintain fixed font sizes for the text-bearing controls but resize the controls to match the length of their respective text strings, or (2) to maintain fixed sizes for text-bearing controls but resize their fonts to match the space available on their respective control.

For this assignment, write a complete Visual Basic .NET application which demonstrates the latter two methods for adapting GUI layout geometry automatically to accommodate the different string lengths of GUI text encountered during on-the-fly languages/culture switching. Your application must consist of a single Form containing the following controls: a label named "resizeControlLabel"; a label named "resizeFontLabel" ; and a button named "nextButton" with text property set to "NEXT".

Upon invocation, your program must read successive lines from a Unicode textfile "cs540hw8input.txt" which you provide in the BIN subdirectory of your project directory and which must contain at least ten, but no more than one hundred, words of various lengths from various languages/alphabets, one word per line, each word a lowercase string containing no embedded blanks or punctuation. Your program should read the first word during FormLoad. Each time the NEXT button is clicked thereafter, your program should read the next word from the file. After all words have been read from the file, clicking on the NEXT button should cause your program to terminate gracefully. Each time your program reads a new word from the file, it should display the word as the text property of both the resizeControlLabel control and the resizeFontLabel control. The resizeControlLabel control should always display its text in 14-point font size, but its horizontal size property should be adjusted automatically for each new word so that the entire word is visible on the control and so that the left and right margins within the control are each approximately two character-widths. The horizontal size of the resizeFontLabel control should always be exactly 4 inches, but the font size of the control should be adjusted automatically for each new word so that the entire word is visible on the control and so that the left and right margins within the control are each approximately two character-widths. Use either the "Tahoma" or the "Lucida Sans Unicode" font for all text display, and check to make sure your machine's Windows installation has Far Eastern Language Support enabled (see "Regional and Language Options" on the Control Panel).

On the due date and at any class meeting thereafter you must be prepared demo your application in class to the instructor or the grader/assistant for evaluation and grading.