

ASSIGNMENT #6

You all have seen the advertising signs at McDonald's restaurants that say "Billions and billions of hamburgers sold."

Rubio's Restaurant Corporation is planning major expansions into your assigned country and into India, and is also negotiating a contract to site several Rubio's Baja Grill restaurants inside Disneyland theme parks. Rubio's would like to prepare a Visual Basic .NET application that it can display on wall-sized OLED displays inside these new restaurants, that will announce in a similar way the total number and total value of fish tacos that Rubio's has sold worldwide to date. The application must dynamically localize to English-US, to Hindi-India, to your assigned country's language/locale, and to a new language/locale designated "English-Disneyland" (en-DI), and should use our standard click-the-flag mechanism for on-the-fly language/locale selection. You should design an appropriate flag to represent Disneyland.

The application should display the current date and time in long-date-long-time format, together with the current number and value of fish tacos Rubio's has sold to date. These displays must be dynamic – the displayed current date and time, taco count and value must update continuously and correctly with advancing time.

For this assignment, assume that the cost of a fish taco is US\$4.95, and that as of noon on January 1, 2018, Rubio's had sold 63,208,417 fish tacos worldwide. Assume that thereafter, Rubio's total worldwide fish taco sales have been (and will continue to be) 600 fish tacos per second.

Number and date formatting for English-US and Hindi-India are available in existing, predefined Visual Studio .NET CultureInfo objects, but you may have to custom code number and date formatting for your assigned country's language, and you definitely will have to custom code the number and date formatting for English-Disneyland.

In English-Disneyland, positive integers are formatted as follows: ...ddd*ddd*ddd*ddd*ddd*ddd*dddd

(note that the least significant digit group is four digits wide, and all other digit groups are three digits wide),

The unit of currency in English-Disneyland is the "Mickey", whose symbol is the Tamil letter "Ja" = "ஜ" (U+0B9C) which is meant to suggest a stylized mouse with ears and a tail. Positive currency values are formatted the same as in en-US, except that the Mickey symbol occurs in the rightmost, not the leftmost, position of a currency string:

...ddd,ddd,ddd,ddd,ddd.dd ஜ

For this assignment, assume that the currency conversion rate between US\$ and Miceys is 2:1 (one Mickey has the same value as two US dollars). For converting between all other currency pairs, your application should use conversion rates contained in a text file named "currencyConversion.txt" which you should create manually (in whatever format you wish) and which should reside in the BIN subdirectory of your application's project directory. Your application's FormLoad procedure should contain code to read the file at startup.

In English-Disneyland, dates are displayed as YYYY:MM:DD and times are displayed in 24-hour format ("military time") as HHHH ranging from 0000 to 2359.

Your program must implement all date/time, number, and currency formatting localizations in a manner consistent with, and fully utilizing, the .NET architecture (i.e., by specifying formatting rules within appropriately constructed CultureInfo objects). If you use any other kludgy, ad hoc method for implementing the formatting localizations, you will receive ZERO points for the assignment.

On the due date and at any class meeting thereafter you must be prepared to demo your application in the lab for evaluation and grading.