Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

PSI

Laboratory work # 2

Localization

Author:
Petru Negrei

Supervisor:
A. Railean

November 2014

1 Introduction

1.1 Objective

Develop a mechanism that enables a program to display its interface in multiple languages, depending on how it is configured.

1.2 Requirements

- The program must use Unicode for all string-related procedures
- The strings for each language must be loaded from a file
- The program must display the paths to the special folders in the system on which it is ran (for the currently logged on user):

2 Implementation

The below code represent a simple demonstration of how the localization could be applied in the real program.

```
#!/usr/bin/env ruby
require ’r18n-desktop’

class Internalization
  LANGS = ["en", "fr"]
  include R18n::Helpers

  def initialize lang = "en"
    R18n.default_places = ’./i18n/’
    @lang = LANGS.include?(lang) ? lang : "en"
    R18n.set(@lang)
  end

  def print_strings
    p t.user.edit    #=> "Edit user"
    p t.user.name(’John’) #=> "User name is John"
    p t.user.count(5) #=> "There are 5 users"
  end

  def print_time
    p l Time.now, :human #=> "now"
    p l Time.now, :full #=> "3rd of January, 2010 18:54"
  end
end

inter = Internalization.new(ARGV[0])
inter.print_strings
inter.print_time
p Dir.pwd
p File.expand_path(’˜’)
```

```
user:
  edit: Edit user
  name: User name is %1
  count: !!pl
    1: There is 1 user
    n: There are %1 users
```

```
user:
  edit: Modifier l’utilisateur
name: Nom d’utilisateur est %1
count: !!pl
 1: Il y a 1 utilisateur
   n: Il y a %1 utilisateur
```

3 Conclussion

After making this laboratory work I learn more about how to apply localization to your software application, found out what are the rules and requirements that need to be met in order to build a reliable and maintainable code.