

Practical 10: Saving Data in Text Files

NOTE: Turn on Option Explicit and Option Strict compilation options for your project. Prepare your practical solution **AT HOME**. Practical sessions are for you to present and improve your solution.

Create a new windows form application project. Complete the following requirements:

1. Forms

- Create the following forms in the project. Name each relevant control by following the recommended naming convention:

FrmMain: The startup form that works as the switcher to other forms

FrmWrite: Allows user to add menu items and save them to a text file

FrmOpen: Allows user to load menu items from a text file and group them into relevant categories

- Configure the forms with appropriate usability features (access keys, tab order, accept button, cancel button, default focus, etc) before you continue to the programming part.

2. Form [FrmMain]

- Program the buttons accordingly:

Button	Task
Write File	Open the [FrmWrite] form as modal dialog
Read File	Open the [FrmRead] form as modal dialog
Exit	Exit the application

3. Form [FrmWrite]

- Add the following function to the form that detects duplicated item in the ListBox. It returns **True** if an duplicated item is found in the ListBox. Otherwise, it returns **False**.

```
Private Function IsMenuDuplicated(menu As String) As Boolean
    Dim rs = From item In lstMenu.Items
        Where CStr(item).ToUpper() = menu.ToUpper()
    Return rs.Any()
End Function
```

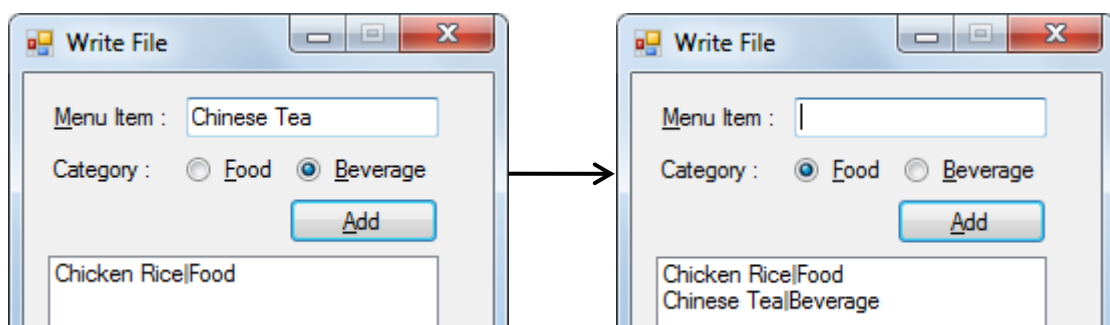
NOTE: The function is programmed by using **LINQ Query**. You may also use **FOR-EACH** loop and **IF** statement to achieve the same result.

- Add an **ErrorProvider** control to the form. In addition, set the following validation related properties for the respective controls:

Control	Property	Value
Form	AutoValidate	EnableAllowFocusChange
ListBox	CausesValidation	False
[Save] button		
[Close] button		

NOTE: Setting **CausesValidation** to **False** indicates that the relevant control should not trigger input validation. For example, it is pointless to perform input validation if user simply wants to close the form.

- Handle the **Shown** event of the form: Reset the form to its default state (e.g. clearing inputs, selecting the default RadioButton, clearing errors, setting default focus, etc).
- Handle the **Validating** event of the [Menu Item] TextBox: Ensure the menu item entered is not empty and not duplicated with existing items in the ListBox.
- Handle the **Click** event of the [Add] button: If inputs are valid, add the menu item entered to the ListBox together with its category (joined with a | character). Reset the inputs and focus after that.



FrmWrite: Add a menu item (together with its category) to the ListBox

- Handle the **DoubleClick** event of the ListBox: Remove the selected item from the ListBox. Ensure it will not cause run-time error if nothing is selected.
- Handle the **Click** event of the [Save] button: Show a **SaveFileDialog** to prompt user for the targeted file (i.e. path and filename). If the targeted file is selected, write the menu items in the ListBox to the file in the following format:

FrmWrite: Menu items are saved to the targeted file in the format as shown below

Chicken Rice|Food
Chinese Tea|Beverage
Curry Noodles|Food
Ice Coffee|Beverage

```
Private Sub btnSave_Click(sender As Object, e As EventArgs) _
    Handles btnSave.Click
    ' Initialize the SaveFileDialog
    With dlgSaveFile
        .InitialDirectory = Application.StartupPath
        .FileName = "Menu.txt"
        .Filter = "Text File|*.txt"
    End With

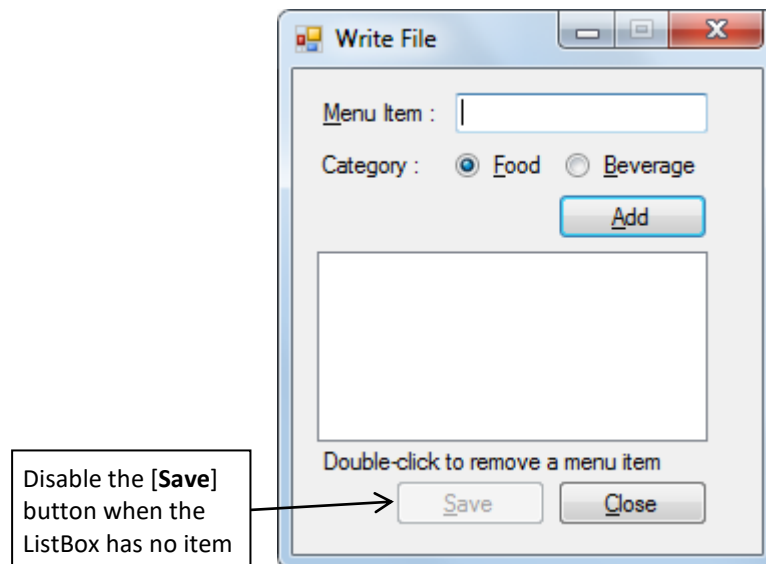
    ' Open the SaveFileDialog
    Dim result As DialogResult = dlgSaveFile.ShowDialog(Me)

    ' If a file is selected
    If result = DialogResult.OK Then
        Dim path As String = dlgSaveFile.FileName

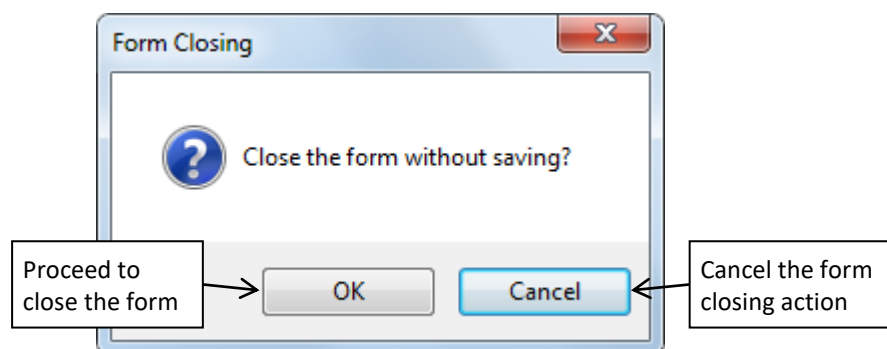
        ' TODO: Write menu items to the file
    End If
End Sub
```

Do it yourself

- Handle the **Click** event of the [Close] button: Close the form.
- **Enhancements:**
 - Disable the [Save] button when there is no item in the ListBox. Otherwise, enable it.



- When user is closing the form, display a confirmation message box if menu items in the ListBox have been modified (i.e. added or removed), but user has not save the menu items to file. **HINT:** Handle the **Closing** event of the form.

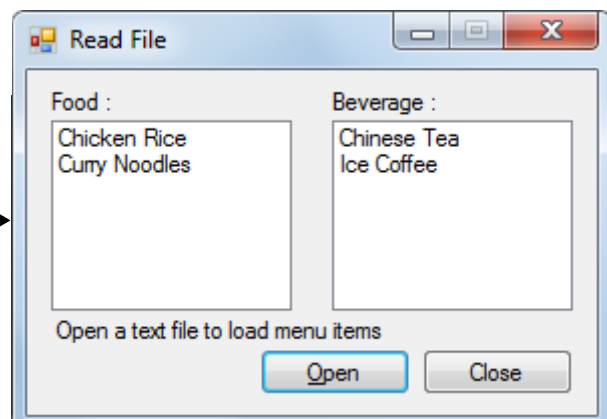


4. Form [FrmRead]

- Handle the **Shown** event of the form: Reset the form to its default state (e.g. clearing the ListBoxes, setting default focus, etc).
- Handle the **Click** event of the **[Open]** button: Show an **OpenFileDialog** to prompt user for the targeted file (i.e. path and filename). If the targeted file is selected, read the menu items from the file and add them to the respective ListBox:

FrmRead: Read menu items from the file and add them to the respective ListBox

Chicken Rice|Food
Chinese Tea|Beverage
Curry Noodles|Food
Ice Coffee|Beverage



```

Private Sub btnOpen_Click(sender As Object, e As EventArgs) _
Handles btnOpen.Click
    ' Initialize the OpenFileDialog
    With dlgOpenFile
        .InitialDirectory = Application.StartupPath
        .FileName = "Menu.txt"
        .Filter = "Text File|*.txt"
    End With

    ' Open the OpenFileDialog
    Dim result As DialogResult = dlgOpenFile.ShowDialog(Me)

    ' If a file is selected
    If result = DialogResult.OK Then
        ' Clear the ListBoxes before adding menu items
        lstFood.Items.Clear()
        lstBeverage.Items.Clear()

        Dim path As String = dlgOpenFile.FileName

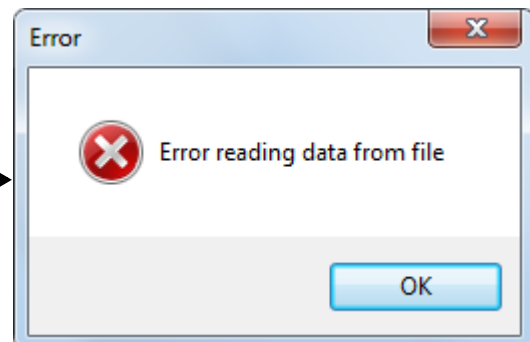
        ' TODO: Read menu items from the file and
        '       add them to the respective ListBox
    End If
End Sub

```

Do it yourself

- Handle the **Click** event of the [**Close**] button: Close the form.
- **Enhancements:**
 - Handle the runtime error that may occur when the content of the targeted file is not following the right format. Display a message box to inform user about the error.

Apple
Orange
Banana



FrmRead: If the content of the file to be read is not following the right format, runtime error may occur. Handle the error and display a message box to inform user about this