WPF アプリケーションの多言語切替

2015 年 6 月 2 日

@twyujiro15

プロフィール

加藤 裕次郎

1982.03.03 生まれ（うお座）

左利き（お箸は右）

twitter : @twyujiro15

プログラミング経験

Excel VBA
MATLAB、MATX
C
VC++ (Windows SDK)
VC++ (MFC)
WPF + C#

本職は製造業の開発業務

- 2009 年 4 月に入社
  - ・組み込みソフトウェア開発で初めてまともに C 言語
  - ・デバッグソフトで Visual C++（MFC、Windows SDK）
  - ・Excel 大好きマンだったので VBA も使用

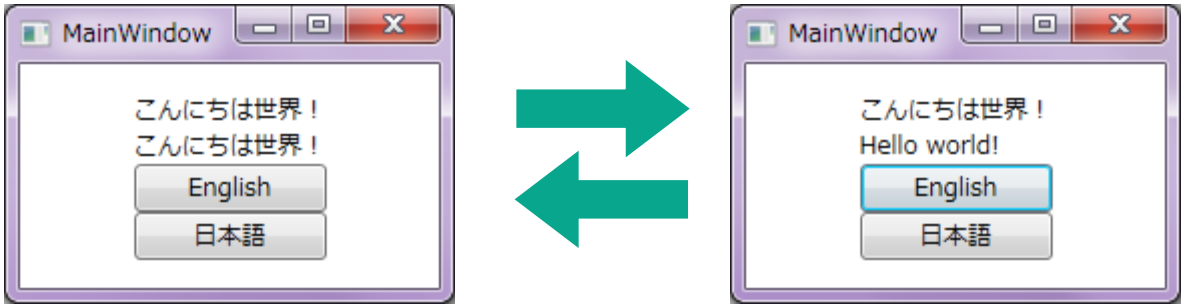
VBA でソケット通信したときは感動した

- 2013 年 10 月
  - ・あるサンプルが "WPF" なるものでできていることを知る

独自調査から @okazuki さんの MVVM 入門四則演算にたどり着く

今回のゴール

WPF デスクトップアプリケーションで動的に言語を切り替える



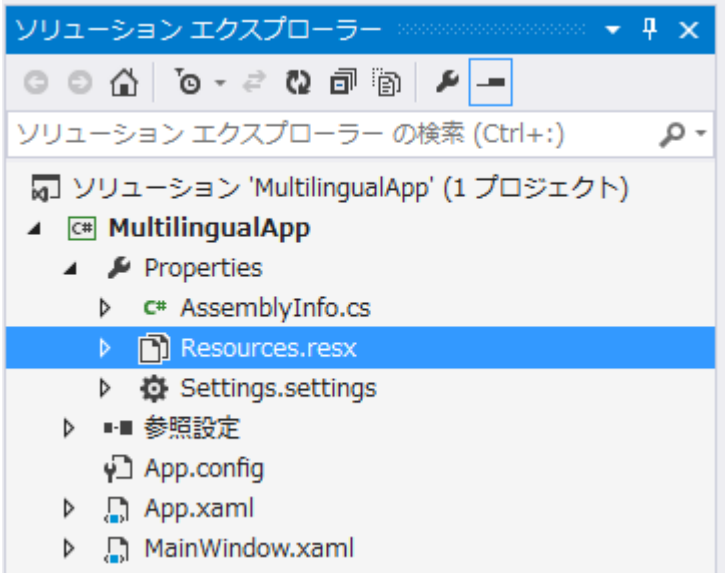
- 「English」ボタンを押すと英語表記になる
- 「日本語」ボタンを押すと日本語表記になる

リソースディクショナリを使う方法もあるようですが、
ここではアセンブリリソースを使う方法を紹介します。

目次

リソースによる文字列の表示 (1/2)

アセンブリリソースファイル Resources.resx に文字列リソースを定義する



ない or 削除した場合は追加しましょう

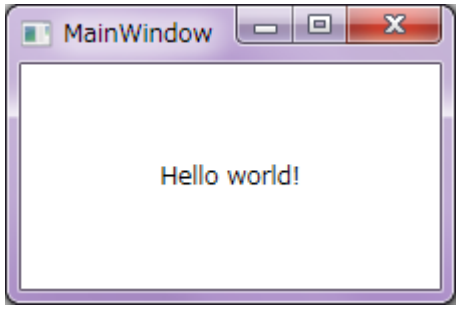
Public にします

リソースによる文字列の表示 (2/2)

XAML から参照する

```
<Window x:Class="MultilingualApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:properties="clr-namespace:MultilingualApp.Properties"
        Title="MainWindow" Height="150" Width="225">
    <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
        <TextBlock Text="{x:Static properties:Resources.HelloWorld}" />
    </StackPanel>
</Window>
```

Resources.resx の HelloWorld キーを指定



目次

- ✓ リソースによる文字列の表示
- ➤ 多言語リソースの追加

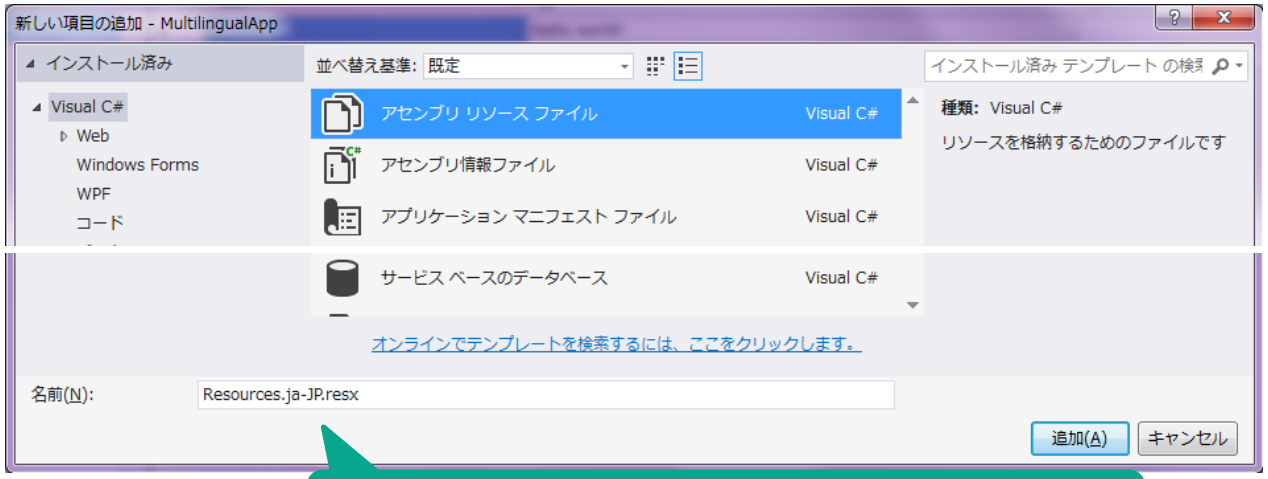
  動的に言語を切り替える

  - 準備
  - ボタンの追加

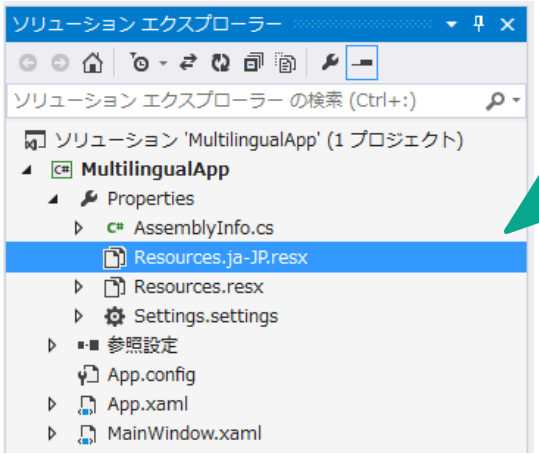
  おまけ

多言語リソースの追加 (1/2)

別の言語用のアセンブリリソースファイル Resources.<カルチャ>.resx を追加する



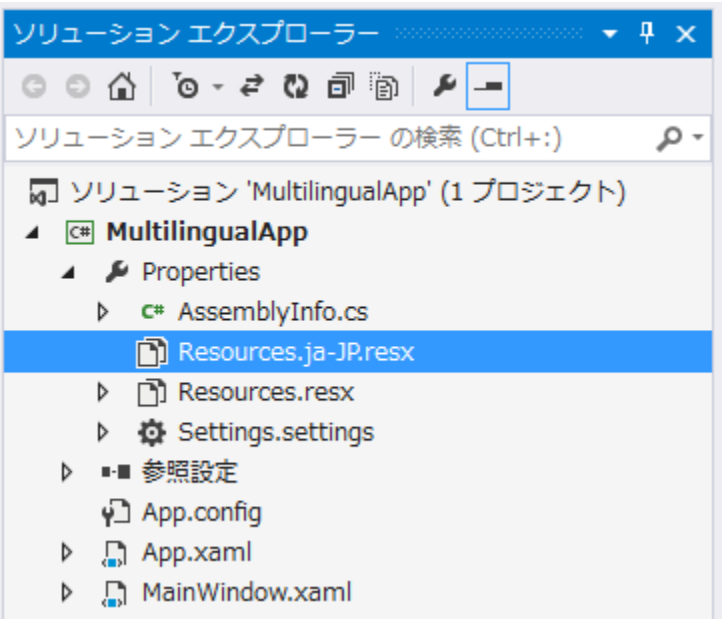
日本語は "ja-JP"、韓国語は "ko-KR" など



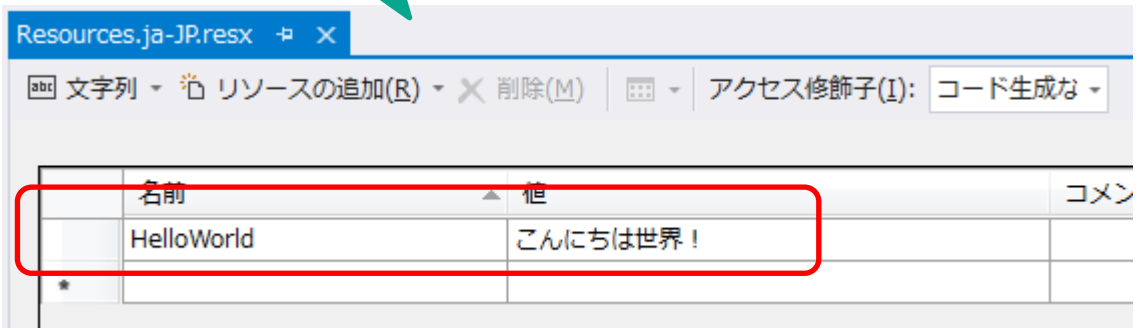
Properties 以下では直接追加できないので
いったんプロジェクト直下に追加して後から移動した

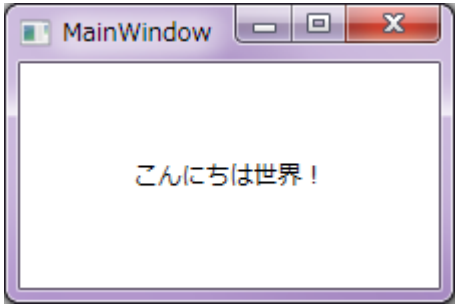
多言語リソースの追加 (2/2)

別の言語用の文字列リソースを設定する



同じ HelloWorld キーを定義する





日本語設定の Windows で実行すると日本語になる

- 設定が "ja-JP" なのでそのリソースを検索
- なければデフォルトの Resource.resx を参照する

つまり英語設定の Windows で実行すると英語になる

目次

- ✓ リソースによる文字列の表示
- ✓ 多言語リソースの追加
- ➢ 動的に言語を切り替える
  - 準備
  - ボタンの追加
- おまけ

動的に言語を切り替える準備 (1/3)

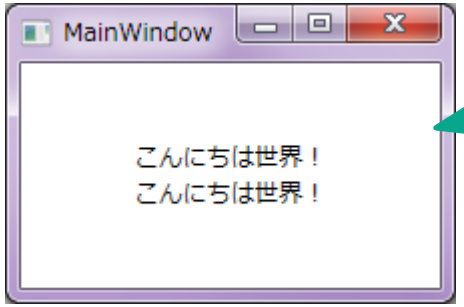
ResourceService クラスを定義

```
namespace MultilingualApp
{
    using MultilingualApp.Properties;

    /// <summary>
    /// 多言語化されたリソースと言語の切り替え機能を提供します。
    /// </summary>
    public class ResourceService
    {
        #region Singleton members
        private static readonly ResourceService _current = new ResourceService();
        public static ResourceService Current { get { return _current; } }
        #endregion Singleton members

        private readonly MultilingualApp.Properties.Resources _resource = new Resources();

        /// <summary>
        /// 多言語化されたリソースを取得します。
        /// </summary>
        public MultilingualApp.Properties.Resources Resources { get { return _resource; } }
    }
}
```

動的に言語を切り替える準備 (2/3)

```
<Window x:Class="MultilingualApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:properties="clr-namespace:MultilingualApp.Properties"
        xmlns:app="clr-namespace:MultilingualApp"
        Title="MainWindow" Height="150" Width="225">
    <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
        <TextBlock Text="{x:Static properties:Resources.HelloWorld}" />
        <TextBlock Text="{Binding Source={x:Static app:ResourceService.Current},
                                 Path=Resources.HelloWorld, Mode=OneWay}" />
    </StackPanel>
</Window>
```

Binding するように変更する



現時点では結果は同じ

動的に言語を切り替える準備 (3/3)

```
using MultilingualApp.Properties;
using System.ComponentModel;
using System.Globalization;
using System.Runtime.CompilerServices;

/// <summary>
/// 多言語化されたリソースと言語の切り替え機能を提供します。
/// </summary>
public class ResourceService : INotifyPropertyChanged
{
    ～～（中略）～～

    #region INotifyPropertyChanged members
    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void RaisePropertyChanged([CallerMemberName] string propertyName = null)
    {
        var h = this.PropertyChanged;
        if (h != null) h(this, new PropertyChangedEventArgs(propertyName));
    }
    #endregion INotifyPropertyChanged members

    /// <summary>
    /// 指定されたカルチャ名を使用してリソースのカルチャを変更します。
    /// </summary>
    /// <param name="name">カルチャ名を指定します。</param>
    public void ChangeCulture(string name)
    {
        Resources.Culture = CultureInfo.GetCultureInfo(name);
        this.RaisePropertyChanged("Resources");
    }
}
```

プロパティ変更通知を追加

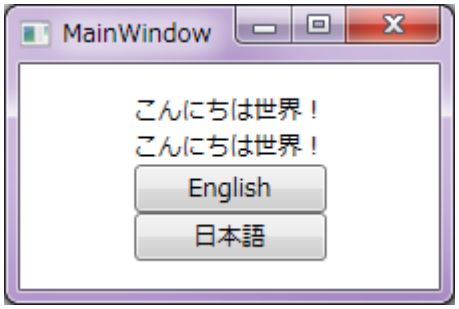
外部からカルチャを指定

Resources プロパティ変更通知

目次

言語切替ボタンの追加 (1/2)

```
<Window x:Class="MultilingualApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:properties="clr-namespace:MultilingualApp.Properties"
        xmlns:app="clr-namespace:MultilingualApp"
        Title="MainWindow" Height="150" Width="225">
    <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">
        <TextBlock Text="{x:Static properties:Resources.HelloWorld}" />
        <TextBlock Text="{Binding Source={x:Static app:ResourceService.Current},
                                 Path=Resources.HelloWorld, Mode=OneWay}" />
        <Button Content="English" Click="EnglishButton_Click" />
        <Button Content="日本語" Click="JapaneseButton_Click" />
    </StackPanel>
</Window>
```



後は Click イベントハンドラで
リソースを切り替えるだけ

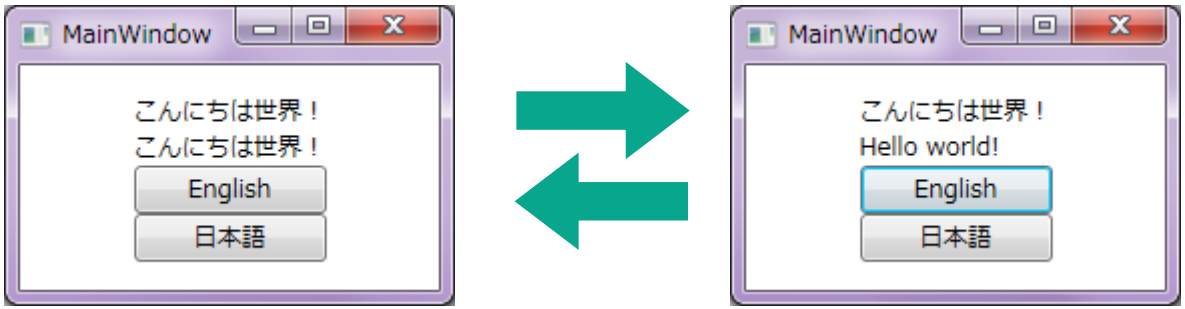
本来なら ViewModel でおこなうべき

言語切替ボタンの追加 (2/2)

```
private void EnglishButton_Click(object sender, RoutedEventArgs e)
{
    ResourceService.Current.ChangeCulture("en");
}

private void JapaneseButton_Click(object sender, RoutedEventArgs e)
{
    ResourceService.Current.ChangeCulture("ja-JP");
}
```

英語に切り替える

日本語に切り替える



完成！

本来なら ViewModel でおこなうべき

目次

- ✓ リソースによる文字列の表示
- ✓ 多言語リソースの追加
- ✓ 動的に言語を切り替える
  - ✓ - 準備
  - ✓ - ボタンの追加
- ➤ おまけ

おまけ

Resources.ja-JP.resx を追加してビルドすると以下のようなファイルができる

名前	更新日時	種類
ja-JP	2015/06/02 15:36	ファイル フォル...
MultilingualApp.pdb	2015/06/02 16:15	Program Debug...
MultilingualApp.exe.config	2015/06/02 14:53	XML Configurati...
MultilingualApp.vshost.exe.config	2015/06/02 14:53	XML Configurati...
MultilingualApp.exe	2015/06/02 16:15	アプリケーション
MultilingualApp.vshost.exe	2015/06/02 16:15	アプリケーション

ja-JP フォルダの中身

名前	更新日時	種類
MultilingualApp.resources.dll	2015/06/02 16:15	アプリケーショ...

サテライトアセンブリ（カルチャに固有のリソースのみを含む）

MultilingualApp.exe にはニュートラルのカルチャリソースが含まれる

Resources.resx

サテライトアセンブリを消すとそのカルチャの言語表示ができなくなる

ご清聴ありがとうございました