

# How To WPF アプリケーション Part3 By 中博俊

## WPFとは

- いちおうWindows Vista向けに出荷された.NET3.0で、初搭載されたWindowsクライアントプログラムの一形態
- 現在の最新バージョンは3.5
- いちおうWindows Formsの置き換え
- クライアントの表現を豊かにしてくれるツールとしてこれから期待しています。
- XAMLというものでUIを記述することができます。
- SilverlightでもXAMLを使います。

## おさらい

- 第1回はButtonとTextBoxを使った簡単なアプリケーションの作り方でした。
- WPFアプリケーションはDocument-Viewとして完全にUIとドキュメントを分けて考えましょう。
- そしてINotifyPropertyChangedのインターフェイスを究めようという内容でした。
- NotifyPropertyChangedBaseは今回も出てきますのでおさらい

## NotifyPropertyChangedBase

```
public class NotifyPropertyChangedBase :  
    INotifyPropertyChanged {  
    public event PropertyChangedEventHandler  
        PropertyChanged;  
    protected void FirePropertyChanged(  
        string PropertyName) {  
        if (this.PropertyChanged != null) {  
            this.PropertyChanged(this,  
                new PropertyChangedEventArgs(PropertyName));  
        }  
    }  
}
```



## おさらい

- 第2回はコンバータと、ラジオボタンの制御についてでした。
- コンバータはIValueConverterを使い、ConvertとConvertBackを用意するところを解説しました。

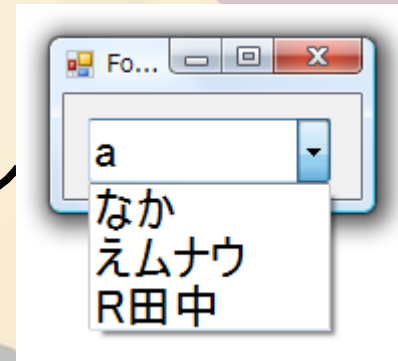


## 今回のアジェンダ

- コンボボックス
  - よく使うコントロールの代表
- Ancestor

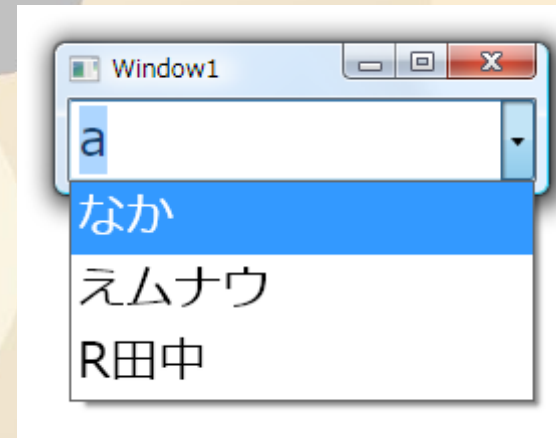
## コンボボックスとは

- テキストボックスとリストボックスを組み合わせたコントロール
- 入力の可/不可も設定可能



Windows Forms

# DEMO1

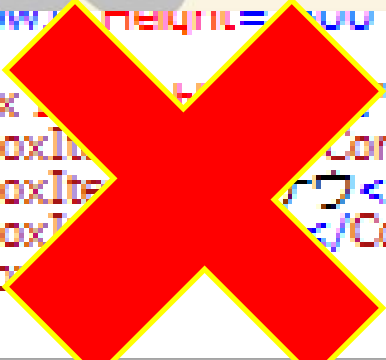


WPF

## バイディングしようよ

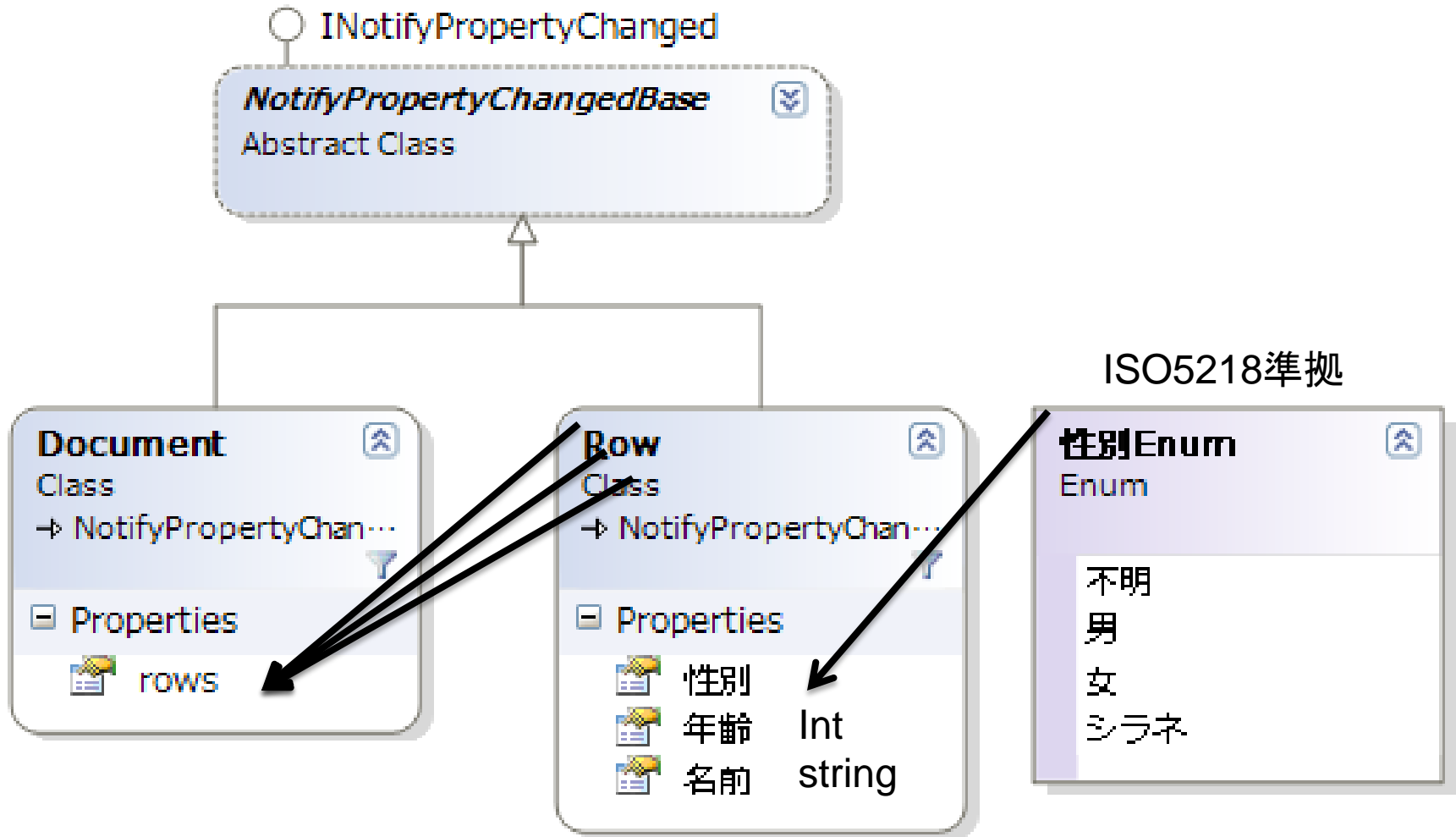
- このように固定値を利用して表現することは簡単にできます。
- しかしこの勉強会のシリーズでは、WPFのデータコンテキストを利用したバイディングですべてを解決していこうという趣旨です。
- ですので、このような利用の仕方はいけません。

```
NAME= WINDUW... HEIGHT= 300 WIDTH= 300 FONT...  
<Grid>  
  <ComboBox...>  
    <ComboBoxItem...>  
    <ComboBoxItem...>  
    <ComboBoxItem...>  
  </ComboBox...>  
</Grid>
```





# DEMO2のデータたち



ISO5218準拠

## DEMO2のデータたち

```
public class Document : NotifyPropertyChangedBase {  
    public Row[] rows {  
        get { return _rows; }  
        set { _rows = value; }  
    }  
  
    private Row[] _rows = new Row[]  
    {  
        new Row(){名前="なか",年齢 = 32, 性別=性別Enum.男},  
        new Row(){名前="えムナウ",年齢 = 21, 性別=性別Enum.男},  
        new Row(){名前="R田中",年齢 = 18, 性別=性別Enum.シラネ}  
    };  
}
```



# DEMO2

## ComboBoxのバインディングの方法

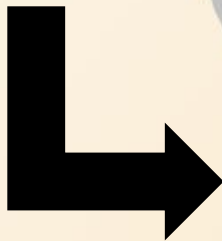
- SelectedItem
- Text
- SelectedValueと  
SelectedValuePath

それぞれのプロパティはどこで実装されているのか

## ComboBoxのバインディングの方法

- ComboBox.Text
- **Selector.SelectedItem**
- Selector.SelectedValue
- Selector.SelectedValuePath

最初の説明: テキストボックスとリストボックス  
を組み合わせたコントロール  
実態は兄弟



**System.Windows.Controls.Primitives.Selector**

System.Windows.Controls.ComboBox

System.Windows.Controls.ListBox

System.Windows.Controls.TabControl

System.Windows.Controls.ItemsControl

**System.Windows.Forms.ListControl**

System.Windows.Forms.ListBox

System.Windows.Forms.ComboBox

- WPFのコントロールのすごさを知ってもらいましょう。

# DEMO3

- ちょっと理由がわからないので、Snoopで覗いてみましょう。
- これが実際に出来上がったVisualTreeです。

ContentPresenterが  
ストレッチでないこと  
が想像できます。

**ActualWidth=282**

**ActualWidth=282**

**ActualWidth=101**

**ActualWidth=101**

**ActualWidth=47** (TextBlock) 0

**ActualWidth=30** (TextBlock) 0

**ActualWidth=24** (TextBlock) 0

(ColumnDefinition) 0

(ColumnDefinition) 0

(ColumnDefinition) 0

[-] (ComboBoxItem) 9

[-] Bd (Border) 8

[-] (ContentPresenter) 7

[-] (Grid) 6

[+] (ComboBoxItem) 9

[+] (ComboBoxItem) 9

HorizontalAlignment

Left

やっぱり



## 簡単な解決方法

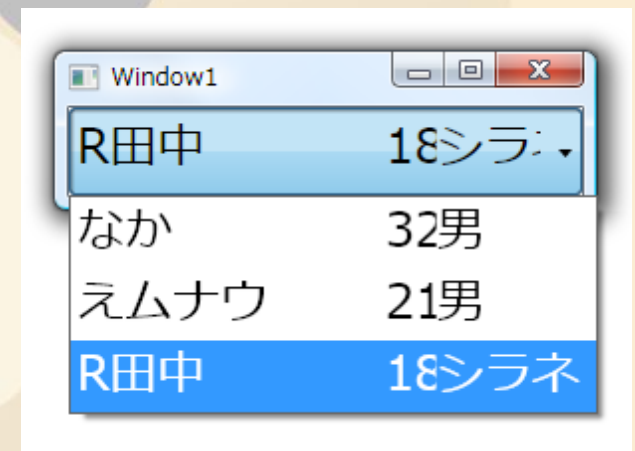
- Gridの幅をコンボボックスの幅などにしてしまえばいけるんじゃないか？

# DEMO4



## 簡単な解決方法

- ポップアップ側はシラネが全部表示されていますが、ボタン側には▼のエリアが上書きされちゃってます。
- 強引に右マージンを設定する場合に  
<ColumnDefinition  
Width="20"/>
- なんかを追加するといいかもしれません。



- ちょっとグダグダ感が……
- いえそんなことはありません。たぶん。
- 今回簡易的な解決方法を提供しましたが、本格的に対応する方法は結構厄介です。  
(アプローチは何種類か存在しますが……)
- スタイルとテンプレートについてはかなりややこしいので徐々に解説していきたいと思います。

Enjoy WPF