



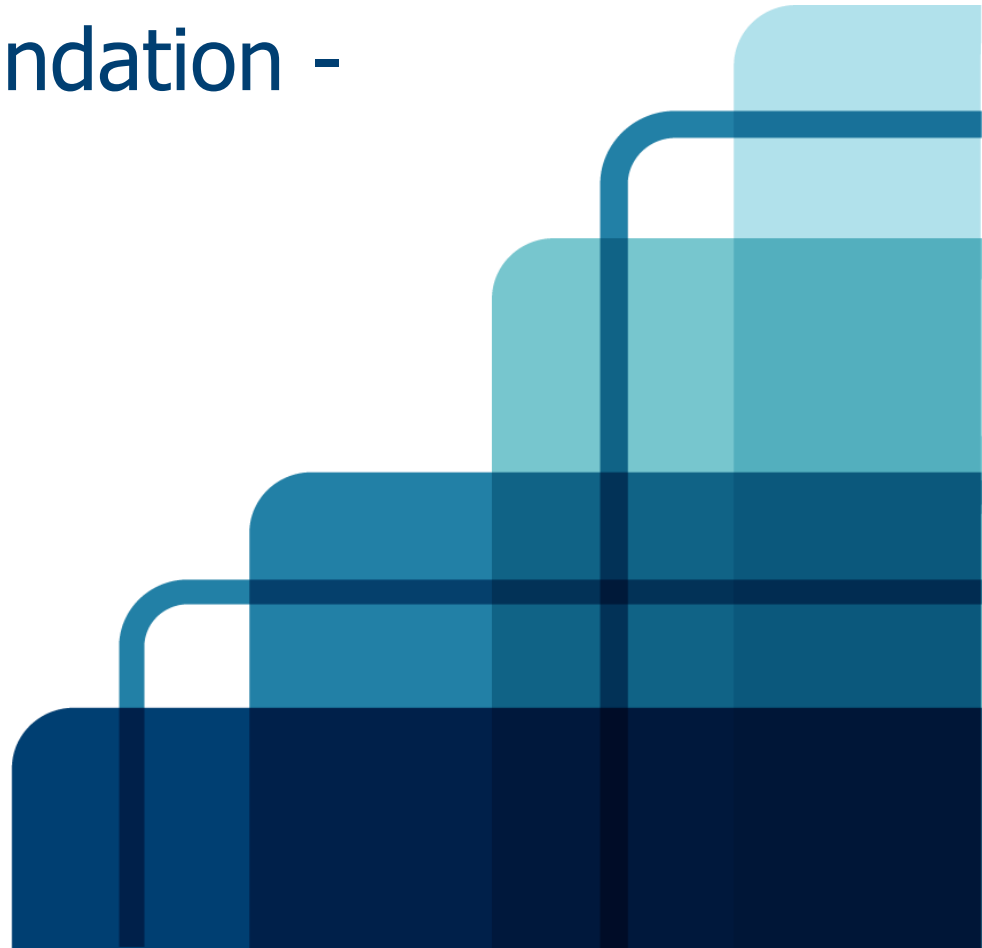
The Sector Specialists

Windows Presentation Foundation - advanced

Confidential

Prepared by: SJ
Submitted on: 2013-10-04
Version: 1.0

Confidential



Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

Converting Data – IValueConverter

Data Validation - IDataErrorInfo

Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

Converting Data – IValueConverter

Data Validation - IDataErrorInfo

WPF - reminder

- What are the main differences between WPF and Windows Forms? Select three differences.
- How can we change appearance of our application?
- Select three layout panels and indicate differences between them.

Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

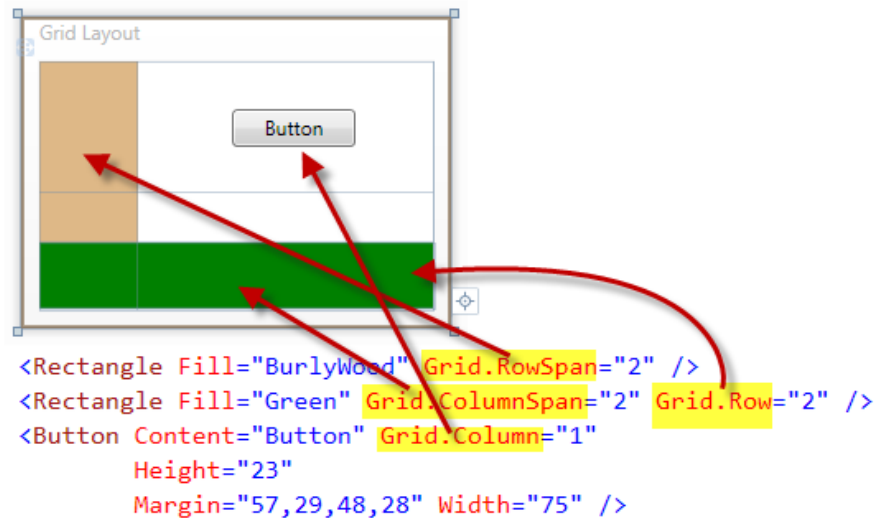
Converting Data – IValueConverter

Data Validation - IDataErrorInfo

Attached Properties

An attached property is intended to be used as a type of global property that is settable on any object.

Used to allow different child elements to specify unique values for a property that is actually defined in a parent element.



Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

Converting Data – IValueConverter

Data Validation - IDataErrorInfo

Binding - INotifyPropertyChanged

- The way to synchronize data between the model and the view
- INotifyPropertyChanged interface
- Four types of bindings mode: OneTime, OneWay, TwoWay, OneWayToSource
- Four types of timing of binding source updates: Default, LostFocus, PropertyChanged, Explicit

```
private string _name;

public string Name
{
    get { return _name; }
    set
    {
        _name = value;
        PropertyChanged("Name");
    }
}

private void PropertyChanged(string prop)
{
    if( PropertyChanged != null )
    {
        PropertyChanged(this, new PropertyChangedEventArgs(prop));
    }
}
```


Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

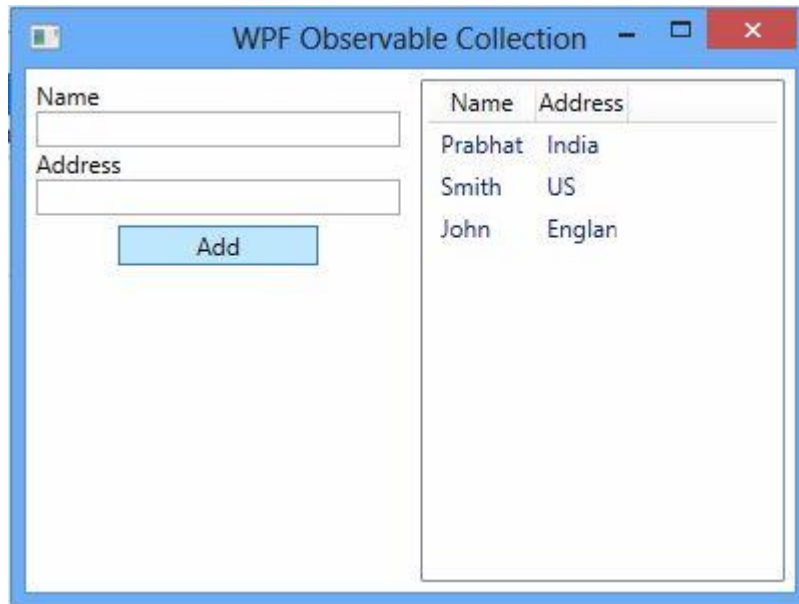
Binding - ObservableCollection

Converting Data – IValueConverter

Data Validation - IDataErrorInfo

Binding - ObservableCollection

- Represents a dynamic data collection that provides notifications when items get added, removed, or when the whole list is refreshed
- ObservableCollection class (System.Collections)
- Three types of bindings: OneWay, TwoWay, OneWayToSource, OneTime



```
public partial class MainWindow : Window
{
    private ObservableCollection<Person> person;
    public MainWindow()
    {
        InitializeComponent();
        person = new ObservableCollection<Person>()
        {
            new Person(){Name="Prabhat",Address="India"},
            new Person(){Name="Smith",Address="US"}
        };
        lstNames.ItemsSource = person;
    }
    private void btnNames_Click(object sender, RoutedEventArgs e)
    {
        person.Add(new Person() { Name = txtName.Text, Address = txtAddress.Text });
        txtName.Text = string.Empty;
        txtAddress.Text = string.Empty;
    }
}

public class Person
{
    public string Name { get; set; }
    public string Address { get; set; }
}
```

Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

Converting Data - IValueConverter

Data Validation - IDataErrorInfo

Converting Data - IValueConverter

- Used when a value needs to be converted to be synced between the binding source and destination
- IValueConverter interface

```
[ValueConversion(typeof(String), typeof(Visibility))]  
public class StringToVisibilityConverter : IValueConverter  
{  
    public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)  
    {  
        if (!String.IsNullOrEmpty(value as String))  
        {  
            return Visibility.Visible;  
        }  
        return Visibility.Collapsed;  
    }  
  
    public object ConvertBack(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)  
    {  
        throw new NotImplementedException();  
    }  
}
```

Agenda

WPF - reminder

Questions & Answers

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

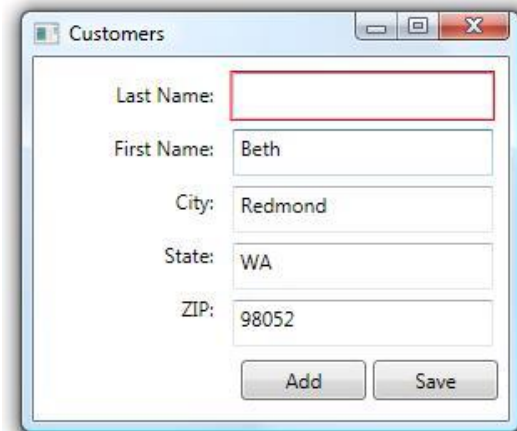
Converting Data – IValueConverter

Data Validation - IDataErrorInfo

Data Validation - IDataErrorInfo

Provides the functionality to offer custom error information that a user interface can bind to.

```
private string lastName;  
public string LastName  
{  
    get { return lastName; }  
    set  
    {  
        lastName = value;  
        OnPropertyChanged(" LastName ");  
    }  
}  
  
public string Error  
{  
    get { return String.Empty; }  
}  
  
public string this[string fieldName]  
{  
    get  
    {  
        string result = null;  
        if (fieldName == „LastName")  
        {  
            if (string.IsNullOrEmpty(FirstName))  
                result = „Nazwisko nie może być puste!";  
        }  
        return result;  
    }  
}
```



The screenshot shows a Windows-style dialog box titled "Customers". It contains five text input fields arranged vertically, each with a label to its left: "Last Name:", "First Name:", "City:", "State:", and "ZIP:". The "Last Name" field is currently empty and has a red border around it, indicating a validation error. The "First Name" field contains the text "Beth", "City" contains "Redmond", "State" contains "WA", and "ZIP" contains "98052". At the bottom right of the dialog, there are two buttons: "Add" and "Save".

Agenda

WPF - reminder

Attached Properties

Binding - INotifyPropertyChanged

Binding - ObservableCollection

Converting Data – IValueConverter

Data Validation - IDataErrorInfo

Questions & Answers

Literature

- <http://msdn.microsoft.com>
- http://blogs.msdn.com/blogfiles/wpfsl designer/WindowsLiveWriter/WPFSilverlightLayoutControls_8097/GridAttachedProperties_3.png