

# Возможности применения Windows Presentation Foundation и Silverlight 2.0

Андрей Складьевский

[andrew@oridea.org](mailto:andrew@oridea.org)

.NET Developer, Murano Software  
Microsoft Most Valuable Professional

# Главные проблемы Windows Forms

- Медленная основа в виде GDI+  предел возможностей платформы;
- Ограничения по позиционированию элементов;
- Существенные трудности при создании нестандартного дизайна;
- Отсутствие средств создания анимаций.

Сила Direc

Удобство  
.NET Framework

Windows Presentation Foundation

# **WPF: НОВАЯ ОСНОВА ПРЕДСТАВЛЕНИЯ WINDOWS**

# Windows Presentation Foundation

- **XAML:** декларативное описание объектов;
- **Элементы управления:** панели, кнопки, текстовые поля, меню, списки и т.д.;
- **Представление:** векторная графика, поддержка вывода видео- и аудио-содержимого;
- **Данные:** поддержка одно- и двухсторонней связи элементов окна с объектами;
- **Direct3D:** аппаратное ускорение вывода

# eXtensible Application Markup Language

- XML язык;
- Привязка элементов и атрибутов к .NET объектам, их свойствам и событиям;
- Дизайнер в Expression Blend работает напрямую с XAML;

```
<Window x:Class="FebruaryWpfApplication.Window1"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="February WPF" Height="200" Width="260" WindowStyle="SingleBorderWindow">
<Grid>
```

```
<Rectangle RadiusX="10" RadiusY="10" Stroke="Brown">
```

```
<Rectangle.Fill>
```

```
<LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
```

```
<GradientStop Color="White" Offset="0" />
```

```
<GradientStop Color="BlanchedAlmond" Offset="0.2" />
```

```
<GradientStop Color="White" Offset="1" />
```

```
</LinearGradientBrush>
```

```
</Rectangle.Fill>
```

```
</Rectangle>
```

```
</Grid>
</Window>
```



**Гибкие свой**

**Простая и тонкая  
настройка вида**

Windows Presentation Foundation

# **ЭЛЕМЕНТЫ УПРАВЛЕНИЯ**

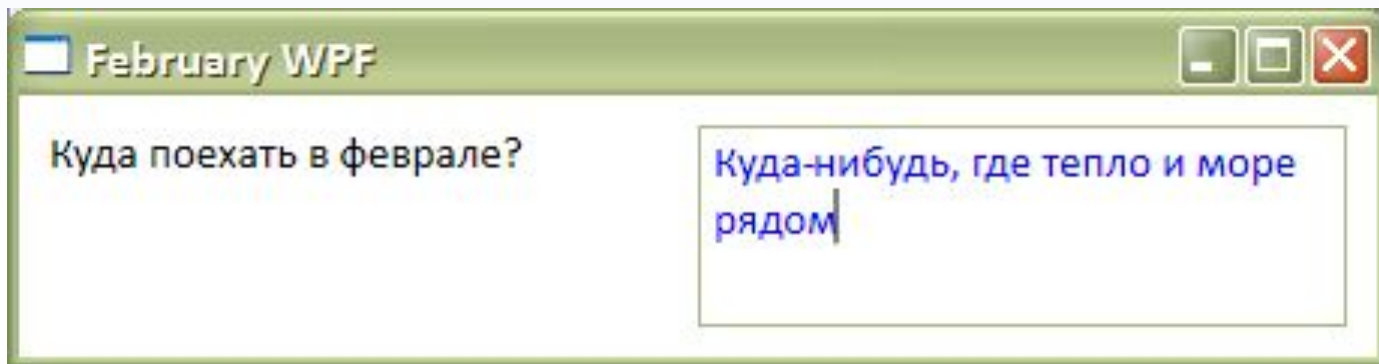
# Гибкие свойства

- Экземпляр наследуемого от `DependencyObject` класса является контейнером любого числа свойств;
- Большинство свойств такого класса имеют соответствующий статический объект типа `DependencyProperty`, являющийся определителем этого свойства;
- Объекту можно установить значение свойства, которого у него на самом деле нет, вызвав метод `SetValue`;

# Удобные панели

- StackPanel размещает элементы друг за другом вертикально или горизонтально;
- Grid размещает элементы в таблице:

```
<Grid Margin="10,10,10,10">  
  <Grid.ColumnDefinitions>  
    <ColumnDefinition/><ColumnDefinition/>  
  </Grid.ColumnDefinitions>  
  <Grid.RowDefinitions>  
    <RowDefinition/>  
  </Grid.RowDefinitions>  
  <TextBlock Text="Куда поехать в феврале?" Grid.Column="0" Grid.Row="0" />  
  <TextBox Grid.Column="1" Grid.Row="0" Foreground="Blue" TextWrapping="Wrap" />  
</Grid>
```





# Удобные панели

- **DockPanel** размещает элементы согласно их привязке по краям;
- **WrapPanel** размещает элементы друг за другом по мере возможности, при нехватке места переносит элемент на новую строку;
- **Canvas** абсолютно позиционирует элементы, согласно их X и Y координатам;
- Кроме этого есть: **VirtualizingPanel**, **TabPanel**, **ToolBarOverflowPanel**, **UniformGrid**

# Элементы управления

- Настраиваемый вид благодаря свойствам `OverrideDefaultStyle` и `Template`;
- Элементы являются контейнерами, например, в кнопку можно вставить и картинку и другие элементы;
- Нет аналогов для `DateTimePicker` и `DataGridView` из `Windows Forms`;
- Внутренние свойства ориентированы на данные, т.е. `ListView.SelectedItems` содержит в себе объекты, предоставленные через `ListView.ItemsSource`, а не какие-либо другие.

# Настройка дизайна на уровне приложения в App.xaml

```
<Application x:Class="FebruaryWpfApplication.App"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  StartupUri="Window1.xaml">
  <Application.Resources>
    <Style TargetType="{x:Type Button}">
      <Setter Property="OverridesDefaultStyle" Value="True" />
      <Setter Property="BorderBrush" Value="Brown" />
      <Setter Property="Margin" Value="2" />
      <Setter Property="Template">
        <Setter.Value>
          <ControlTemplate><Grid Width="{TemplateBinding ActualWidth}" Height="{TemplateBinding ActualHeight}">
            <Rectangle RadiusX="12" RadiusY="12" Width="{TemplateBinding ActualWidth}" Height="{TemplateBinding ActualHeight}" Stroke="{TemplateBinding
              Button.BorderBrush}" StrokeThickness="{TemplateBinding Button.BorderThickness}" x:Name="buttonRectangle"><Rectangle.Fill>
              <LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
                <GradientStop Color="White" Offset="0" />
                <GradientStop Color="BlanchedAlmond" Offset="0.2" x:Name="buttonColor" />
                <GradientStop Color="White" Offset="1" />
              </LinearGradientBrush>
            </Rectangle.Fill></Rectangle>
            <ContentPresenter Content="{TemplateBinding Button.Content}" VerticalAlignment="Center" TextBlock.TextAlignment="Center" />
          </Grid></ControlTemplate>
        </Setter.Value>
      </Setter>
    </Style>
  </Application.Resources>
</Application>
```



`<Ellipse />`

`<ColorAnimation />`

Windows Presentation Foundation

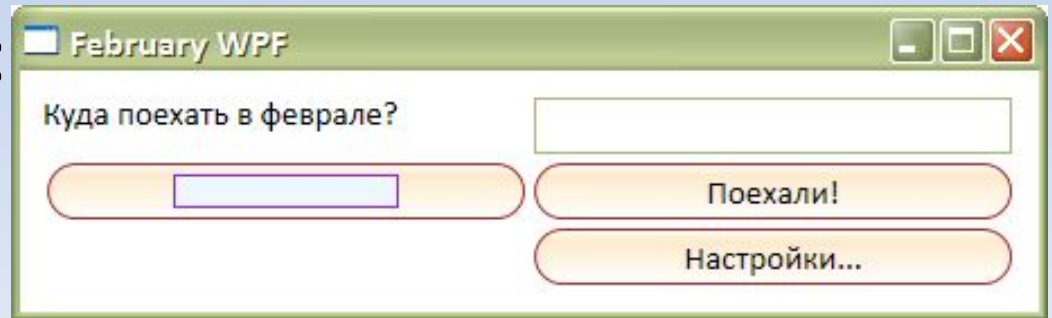
**ГРАФИКА, АНИМАЦИЯ И ЗВУК**

# Графика

- Множество объектов векторной графики: прямоугольники, эллипсы, линии и т.д. которые декларативно описываются в XAML;
- Рисовать можно в Expression Blend, есть так же и другие редакторы (в том числе и с открытым исходным кодом);
- Поддерживаются трансформации любых визуальных объектов, кроме элементов интеграции с Windows Forms;

# Трансформации

- Исходное окно:



- То же окно с применённой трансформацией размера (на 10%) и скашивания (-10°)



# Анимации

- Все анимации имеют универсальный характер, т.е. нет `OpacityAnimation` для анимирования степени прозрачности окна, но есть `DoubleAnimation`, позволяющий анимировать свойство `Opacity` (и другие свойства типа `Double`) любого объекта;
- Анимации записываются в объект `Storyboard`, которая занимается их представлением при вызове метода `Begin`;
- Есть возможность декларативно описать вызов анимации по событию

# MediaElement и Image

- MediaElement: проигрывает видео и аудио потоки, при этом, поддерживая и трансформации;
- Image: позволяет выводить изображения типов .bmp, .gif, .ico, .jpg, .png, .wdp, and .tiff, не поддерживает GIF-анимацию;



Data 😊

😊 Control

Windows Presentation Foundation

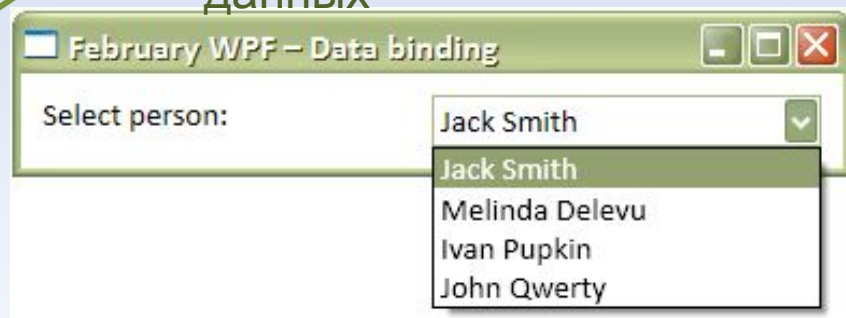
# ПРИВЯЗКА ДАННЫХ К WPF ЭЛЕМЕНТАМ

# Односторонняя связка

```
<Window x:Class="FebruaryWpfApplication.DataBinding"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:app="clr-namespace:FebruaryWpfApplication"
  Title="February WPF – Data binding" Height="300" Width="300">
  <Window.Resources>
    <app:SpecificDataSource x:Key="datasource" />
  </Window.Resources>
  <Grid Margin="10">
    <Grid.ColumnDefinitions>
      <ColumnDefinition /><ColumnDefinition />
    </Grid.ColumnDefinitions>
    <TextBlock Text="Select person:" Grid.Column="0" Grid.Row="0"/>
    <ComboBox ItemsSource="{Binding Source={StaticResource datasource}, Path=Persons}"
      DisplayMemberPath="FullName" SelectedValuePath="FullName"
      IsSynchronizedWithCurrentItem="True" Grid.Column="1" Grid.Row="0"/>
  </Grid>
</Window>
```

namespace  
приложения

ИСТОЧНИК  
ДАННЫХ



↑  
привязк  
а

# Двухсторонняя связка

```
<Window.Resources>
```

```
<app:SpecificDataSource x:Key="datasource" x:Name="datasource" />
```

```
</Window.Resources>
```

```
<Grid Margin="10">
```

```
<Grid.ColumnDefinitions><ColumnDefinition Width="*" /><ColumnDefinition Width="3*" /></Grid.ColumnDefinitions>
```

```
<Grid.RowDefinitions><RowDefinition /></Grid.RowDefinitions>
```

```
<TextBlock Text="Persons:" Grid.Column="0" Grid.Row="0"/>
```

```
<StackPanel Grid.Column="1" Grid.Row="0" Orientation="Vertical">
```

```
<ListView ItemsSource="{Binding Source={StaticResource datasource}, Path=Persons}"  
IsSynchronizedWithCurrentItem="True" SelectionMode="Single">
```

```
<ListView.ItemTemplate>
```

```
<DataTemplate><StackPanel Orientation="Horizontal">
```

```
<TextBox Text="{Binding Path=FirstName}" Width="100"/>
```

```
<TextBox Text="{Binding Path=LastName}" Width="100"/>
```

```
</StackPanel></DataTemplate>
```

```
</ListView.ItemTemplate>
```

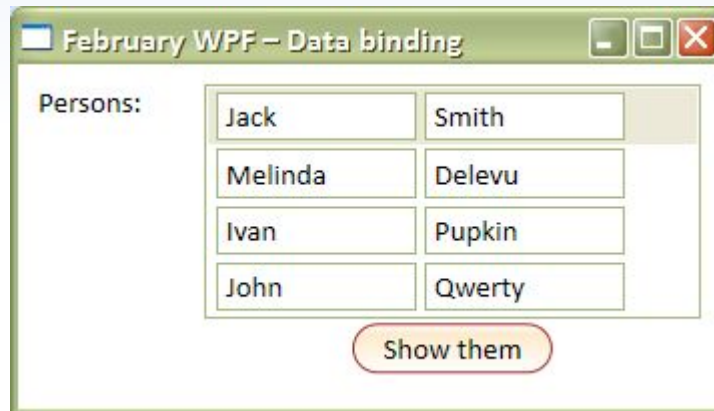
```
</ListView>
```

```
<Button x:Name="buttonShowThem" Click="buttonShowThem_Click" Width="100" Height="25">Show them</Button>
```

```
</StackPanel>
```

```
</Grid>
```

← привязка



**Демонстра**

**WPF**

Мощь WP

В браузере

Внедряемый в браузер компонент

**SILVERLIGHT 2.0**

# Проблемы HTML + DOM

- Существенные ограничения по отрисовке графики и производительности;
- Проблемы с кросс-браузерностью: каждый браузер имеет собственное представление о Веб-стандартах;
- Необходимость использовать JavaScript, почти без ООП и почти без простых путей;

# Решение: Silverlight 2

- Полноценная платформа для RIA – Rich Interactive Application («богатое интерактивное приложение»);
- Позволяет выводить загружаемый с сервера XAML прямо в браузере;
- Содержит в себе самые важные части .NET Framework 3.5 (LINQ, XML, communications, serialization, etc.) и множество элементов из WPF;

# “WPF Everywhere”

- **Silverlight 2.0** имеет практически идентичную WPF архитектуру вывода графики;
- Поддерживается вывод видео и аудио потоков из Сети;
- Позволяет сохранять данные в IsolatedStorage на компьютере клиента;
- Крайне безопасный – нет доступа ни к файлам, ни к GAC установленного .NET, и т.д.;
- Будут почти все элементы из WPF, а возможно и что-то новое (DataGrid);
- Есть ограниченный доступ к DOM модели странички с элементом, т.е. для взаимодействия с ней можно использовать код на C#;

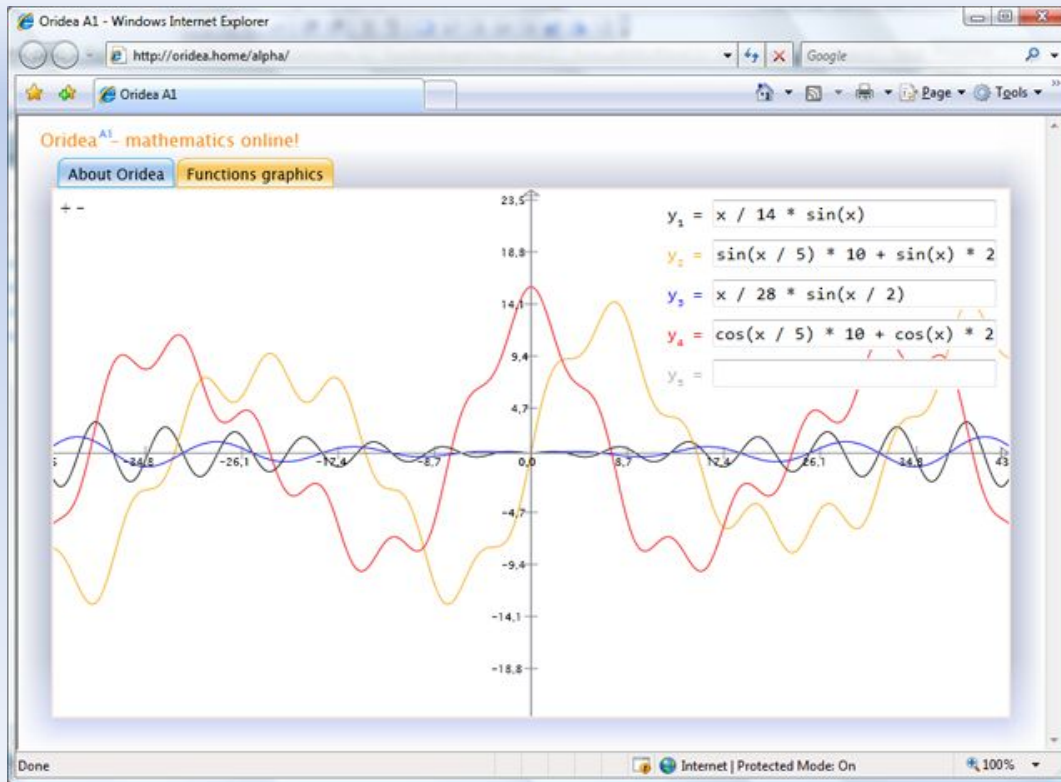


# Содержит Dynamic Language Runtime

- Среда для выполнения динамических языков вроде Python, Ruby и JScript.NET;
- Полная интеграция динамических языков в .NET среду – наивысшая производительность;
- Возможность комбинирования приложения: написания части кода на динамических языках, а части на C# или VB.NET;

# Кросс-платформенный

- Silverlight 2.0 разрабатывается для:
  - Windows (XP/Server 2003/Vista):
    - Firefox
    - Internet Explorer
    - Opera (запланировано в будущем)
  - Mac OS X:
    - Safari
    - Firefox
    - Opera (запланировано в будущем)
  - Linux (проект Moonlight при поддержке Microsoft):
    - Firefox
    - Konqueror



На рисунке: Oridea A1

# ДЕМОНСТРАЦИЯ ПРОЕКТА ДЛЯ SILVERLIGHT 1.1 ALPHA PREVIEW

**Спасибо за  
внимание**

**Время для  
вопросов**

Андрей Скляревский, 2008  
[andrew@oridea.org](mailto:andrew@oridea.org)  
<http://blogs.gotdotnet.ru/personal/snoralip/>  
.NET Developer, Murano Software  
Microsoft Most Valuable Professional