

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

# Windows Presentation Foundation 让多媒体开发更简单

徐栋

北京中达金桥技术服务有限公司

[xudong@itgoldenbridge.com](mailto:xudong@itgoldenbridge.com)



**MSDN Webcasts**

# 议题

- **WPF**是一个统一的多媒体平台
- 应用程序和控件都采用媒体来呈现
- 用户不需要过多考虑表现的逻辑
- 任何控件都能加动画

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- **2D Graphics**
  - **Shapes – a start**
  - Brushes, Drawings
  - Imaging
  - Media
  - Animation
- **3D Graphics**
  - WPF 3D

# Shapes

System.Windows.Shapes

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

- Shapes
  - Shape的属性都是Dependency Property
  - 所有的属性都能加动画和数据绑定
- Shapes 是 FrameworkElements
  - 2D图像基于Framework层
  - 大多数控件都采用Shape表现
  - 所有的Shapes参与Layout系统
- Shapes包括:
  - Ellipse, Line, Path, Polygon, PolyLine, Rectangle

# Shape Example

```
<Window xmlns="..."  
  Width="600" Height="460">  
  <StackPanel>  
    <Rectangle  
      Fill="Green"  
      Height="100"  
      Width="200" />  
  </StackPanel>  
</Window>
```





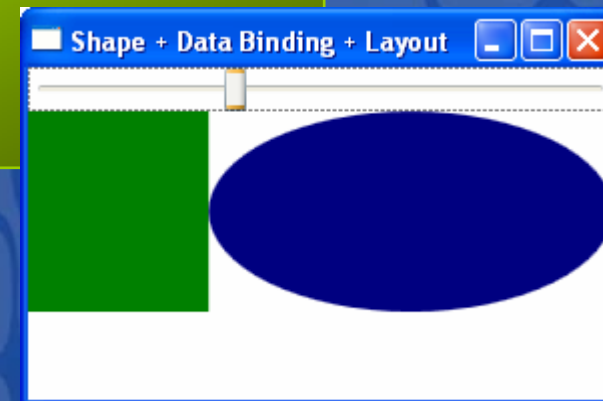
# Shape Example

## Property binding and layout

```
<Window ...>
  <DockPanel ...>
    <Slider Name="Mine"
      Maximum="240"
      Minimum="10"
      DockPanel.Dock="Top" />

    <Rectangle Fill="Green"
      Width="{Binding ElementName=Mine, Path=Value}" />

    <Ellipse Fill="Navy" />
  </DockPanel>
</Window>
```



# 议程

- **2D Graphics**
  - Shapes – a start
  - **Brushes, Drawings**
  - Imaging
  - Media
  - Animation
- **3D Graphics**
  - WPF 3D

# Brushes

- **Brushes** 用来填充几何图形
  - Solid colors, linear and radial gradients, images, visuals, and drawings都可以用作Brush
  - Images, Visuals, and Drawings都可以用作Brush的源
- **VisualBrush**
  - 界面内容的第二种表现
- **DrawingBrush**
  - 在控件上画图



# Drawings

## System.Windows.Media

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

- 轻量级的矢量图形资源
  - Drawings represent vector rendering instructions
  - Declare once, use many times
    - Vector clip-art, Icons, Vector Fills
  - 非交互的
- 属性是 `DependencyProperties`

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

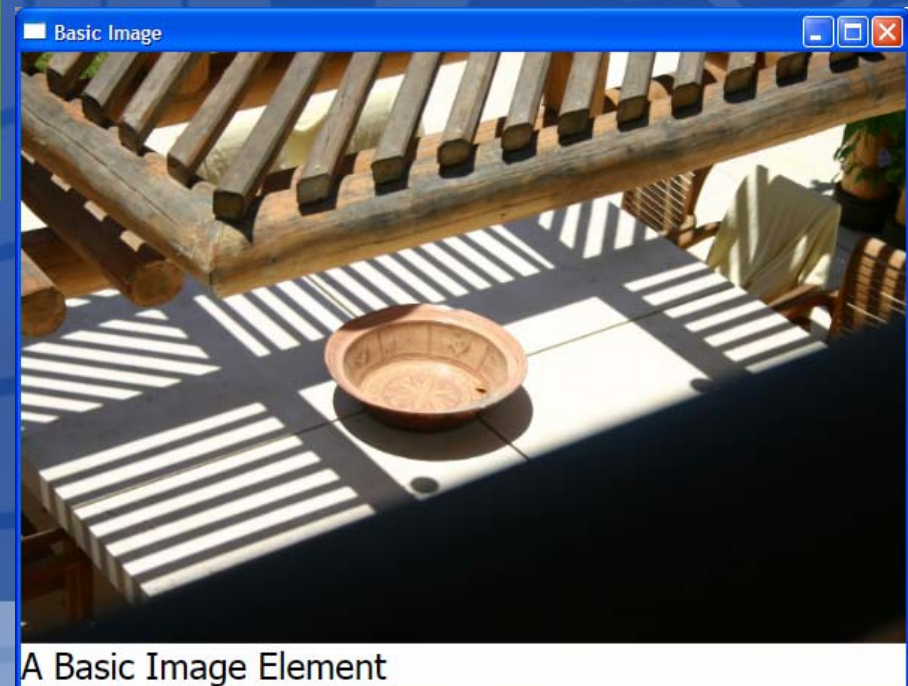
- **2D Graphics**
  - Shapes – a start
  - Brushes, Drawings
  - **Imaging**
  - Media
  - Animation
- **3D Graphics**
  - WPF 3D

# Imaging

- 图像是矩形的，有物理的尺寸
- 使用<Image>标签，通过引用ImageSource对象来访问图片
- ImageSource 也用在ImageBrush和ImageDrawing

# Basic Image Element Example

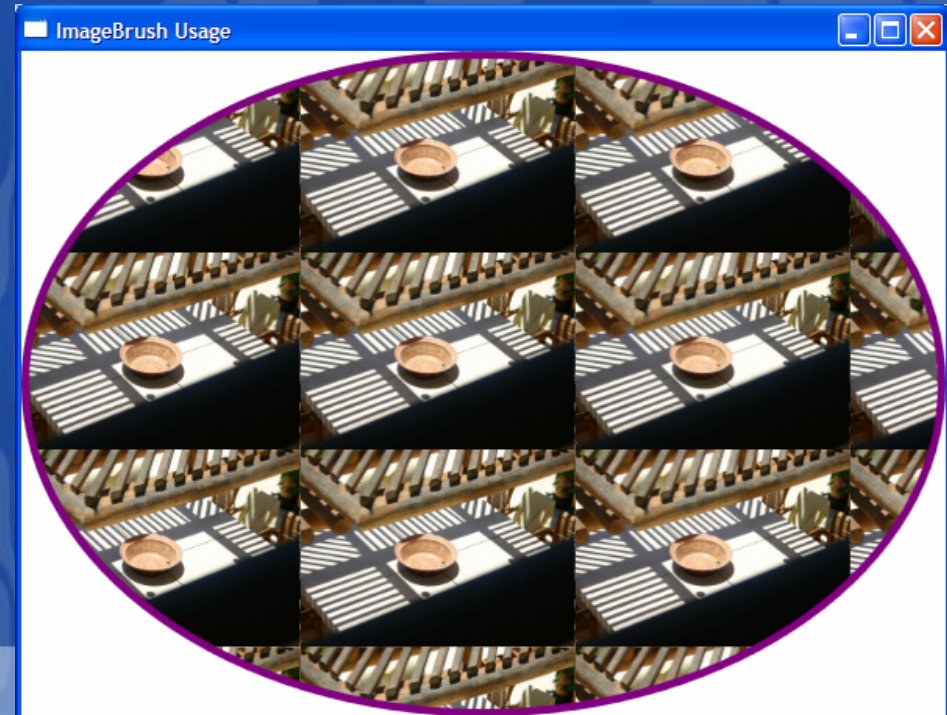
```
<Window xmlns="..."
  Width="600" Height="460"
  Title="Basic Image">
  <StackPanel>
    <Image Source="table-bowl.jpg" />
    <TextBlock FontSize="24">
      A Basic Image Element
    </TextBlock>
  </StackPanel>
</Window>
```





# Basic Image Brush Example

```
<Window xmlns="..."
  Width="600" Height="460"
  Title="ImageBrush Usage">
  <Ellipse Stroke="Purple"
    StrokeThickness="5.0">
    <Ellipse.Fill>
      <ImageBrush
        ImageSource="table-bowl.jpg"
        TileMode="Tile"
        Viewport="0,0,0.3,0.3" />
    </Ellipse.Fill>
  </Ellipse>
</Window>
```





# 拼图游戏

- TileBrush is our friend



- Will help us go from

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

to



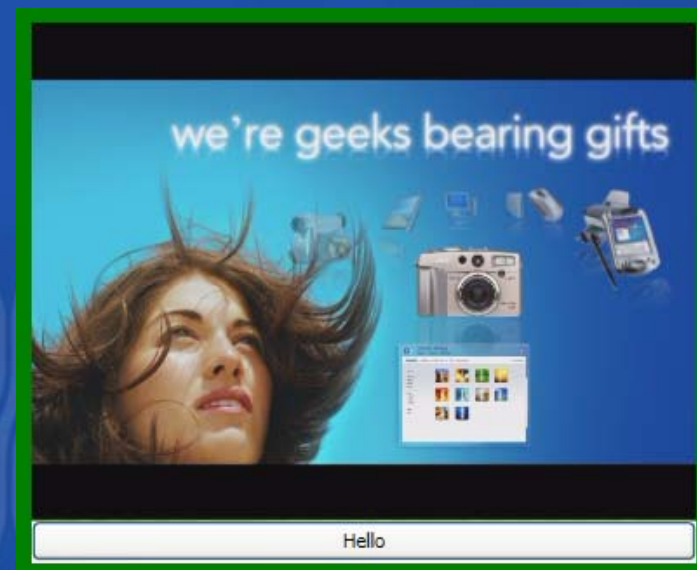
# 议程

- **2D Graphics**
  - Shapes – a start
  - Brushes, Drawings
  - Imaging
  - **Media**
  - Animation
- **3D Graphics**
  - WPF 3D

# Media XAML

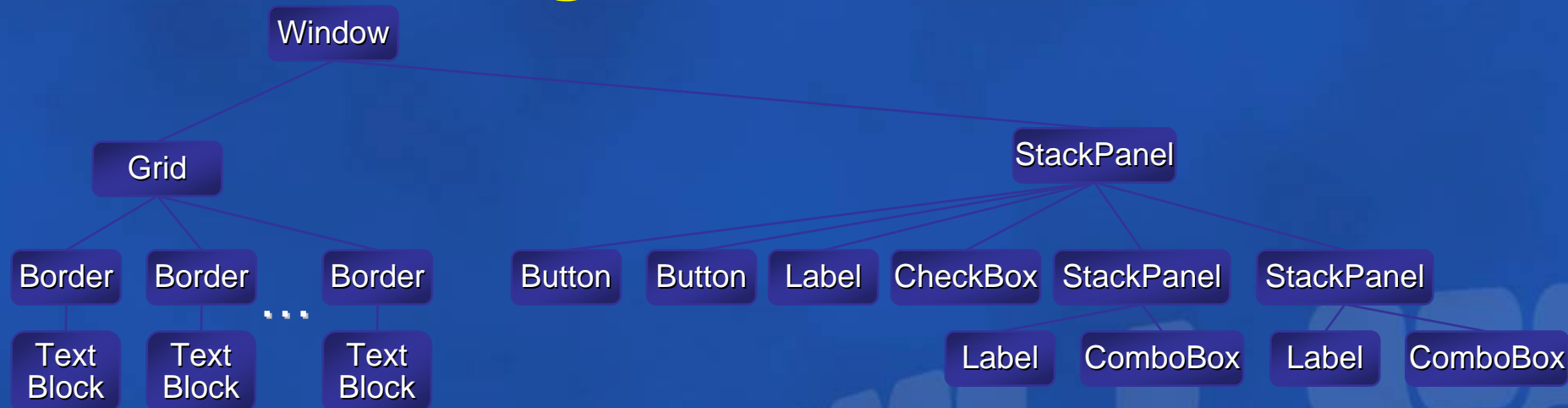
- Framework Level – MediaElement

```
<Border Width="400" Height="330"  
        BorderBrush="Green"  
        BorderThickness="9">  
    <StackPanel>  
        <MediaElement Source="aero.wmv" />  
        <Button>Hello</Button>  
    </StackPanel>  
</Border>
```



- Audio
- VideoDrawing

# What's Going On Here?



- 界面中WPF的元素采用树形关联，称为“**visual tree**”
- WPF中的**visual tree**
  - 概念上对树做一个遍历，所有的元素重新生成
  - 实际上有很多优化可以保证遍历最小化



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- **2D Graphics**
  - Shapes – a start
  - Brushes, Drawings
  - Imaging
  - Media
  - **Animation**
- **3D Graphics**
  - WPF 3D



# Animation

- 本质上，动画就是一个值在指定的时间内变化成为另外一个值
- **Animation**是 **Dependency Property**
- **Storyboard**协调多个动画
- 基于时间，不是基于帧“

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- **2D Graphics**
  - Shapes – a start
  - Brushes, Drawings
  - Imaging
  - Media
  - Animation
- **3D Graphics**
  - WPF 3D

# WPF 3D Graphics

- **WPF** 能够非常方便的实现**3D**效果
- 统一的 **2D** 和 **3D**
  - **2D** 和 **3D**采用统一的呈现管道
- 可以将**2D**的开发观念带入**3D**开发
  - 基于场景的编程方式
  - 传统的**3D**结构
    - Cameras, models, lights, materials, etc.
- 高质量的**3D**输出
  - **3D**中的**Text**和**media**
  - **3D**反锯齿

您的潜力. 我们的动力


**Microsoft**  
微软(中国)有限公司

# 机会

- 桌面应用程序更丰富体验UI
- **Web**更好的表现力
- 强大的生产效率
  - 很多工具工具的支持

## Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问

提问(A)

删除(D)

问题管理器(Q)



您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

**Microsoft®**

msdn  


**MSDN Webcasts**