



XAML 基础教程

郑贵锋

课程目标



通过这一节课，对XAML概念，XAML基础语法（如：命名空间、对象、属性和事件等）有更深入的理解，掌握如何通过XAML代码生成常用控件。



大纲

1

XAML概览

2

基础语法

3

常用控件

XAML概览

The background is a solid green rectangle. It is decorated with several white circles of varying sizes. Some circles are complete, while others are partially cut off by the edges of the frame. The circles are scattered across the green field, with a higher concentration in the upper right and lower right areas.

XML简介

- XML 指可扩展标记语言 (Extensible Markup Language)
- XML 是一种标记语言，很类似 HTML
- XML 的设计宗旨是传输数据，而非显示数据
- XML 标签没有被预定义，需要自行定义标签

```
<note>  
<to>George</to>  
<from>John</from>  
<heading>Reminder</heading>  
<body>Don't forget the meeting!</body>  
</note>
```

XAML简介

XAML = Extensible Application Markup Language



- 控件层次结构一目了然
- UI代码与业务逻辑代码相隔离
- 声明式语言，更便于利用工具自动生成代码
- XAML 标记通常使用与 XML 中所定义的字符实体和转义序列相同的字符实体和转义序列
- 主要用于Silverlight，WPF,Windows Phone和Windows 8应用开发

```
<Rectangle  
  Fill="Red"  
  Width="150"  
  Height="100"/>
```

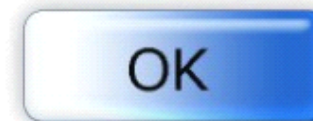


C# VS XAML

C#

```
Button b1 = new Button();  
b1.Content = "OK";  
b1.Background = new  
SolidColorBrush(Colors.LightBlue);  
b1.Width = 100;
```

Compile and Run

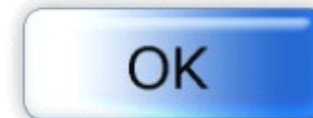


XAML

```
<Button Width="100">  
OK  
<Button.Background>  
LightBlue  
</Button.Background>  
</Button>
```



Load, Parse, Display



为什么学习XAML



- XAML语言应用于多个开发领域
- 具有Silverlight, WPF, Windows Phone以及Windows 8开发经验的程序员,可以将开发技能再利用,加快学习速度,提高开发效率
- 语言性能强大,学习曲线平缓,有强大的开发工具支持
- XAML是Windows 8 Style风格应用主要开发语言之一, XAML与C#,VB,C++配合开发Windows 8 Style应用。

XAML作用

- XAML用于声明UI及UI元素的一种格式
- XAML用于声明样式和模板的格式，这些样式和模板应用于控件和 UI 的逻辑基础
- XAML 定义 UI 的可视外观，关联的代码隐藏文件定义逻辑
- XAML 可在 Expression Blend 与 Visual Studio 之间互换

基础语法

命名空间

- XAML命名空间的概念和C#代码中的Using，VB.Net代码中的Import类似
- 为对象元素的实例化提供引用类库声明
- 编程框架能够区分用户声明的标记和框架声明的标记，并通过命名空间限定来消除可能的标记冲突

Windows 8应用默认命名空间

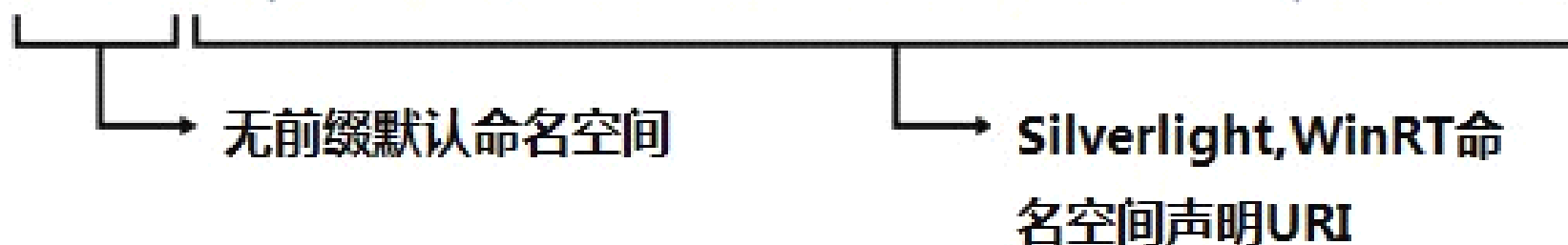
```
<Page
  x:Class="App1.MainPage"
  IsTabStop="false"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="using:App1"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
  mc:Ignorable="d">
```

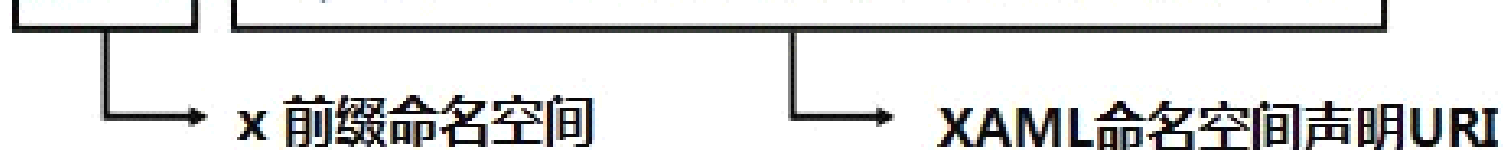
命名空间格式

- 语法结构为 “xmlns:” + “命名空间前缀名”，默认命名空间无需定义命名空间前缀名

`xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"`



`xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"`



常见 “X:前缀” 类型命名空间

x:Key	为ResourceDictionary中的每个资源设置一个唯一键
x:Class	为 XAML 页提供代码隐藏的类的命名空间和类名称，并命名由标记编译器的生成操作在应用程序模型中创建或链接的类
x>Name	处理 XAML 中定义的对象元素后，为运行时代码中存在的实例指定运行时对象名称

对象声明方式

- 使用开始标记和结束标记将对象实例化为XML格式的元素

<Canvas>

</Canvas>

- 对象中可以包含其他对象

<Canvas>

 <Rectangle></Rectangle>

</Canvas>

- 若一个对象中不包含其他对象，可以使用一个自结束标记来声明对象

<Rectangle/>

属性设置

- 属性是对XAML元素特征进行描述的方法
- 属性不允许在XAML中重复设置多次
- 允许在托管代码中改变元素的属性值

属性设置几种方式

- 使用属性语法
- 使用属性元素语法
- 使用内容元素语法
- 使用集合语法

属性语法

- 只有实例化对象才可以设置实例属性，格式如下：
 <objectName propertyName= "propertyValue" />
或者
 <objectName propertyName= "propertyValue" >
 <objectName>
- 每个属性对应一个属性值，属性值类型必须与属性匹配
- 一个标记中可以设置对象的多个属性

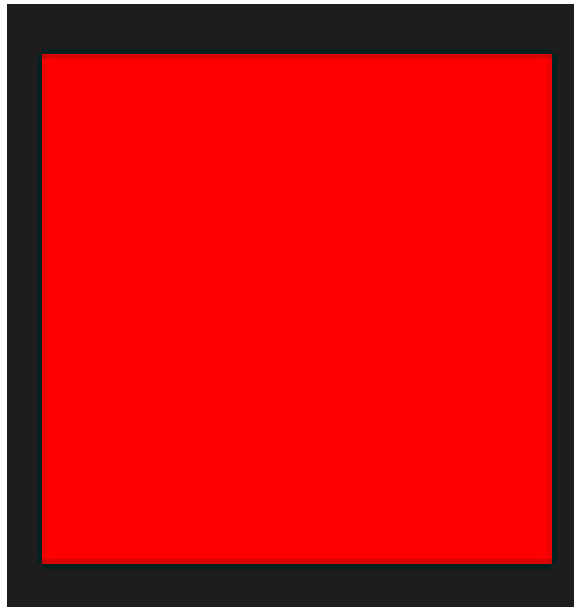
属性语法

`<Canvas Width="150" Height="150" Background="Red"/>`

或者

`<Canvas Width="150" Height="150" Background="Red">`

`</Canvas>`



属性元素语法

- 某些属性可以使用属性元素语法来设置，格式为：

```
<object>
```

```
  <object.property>
```

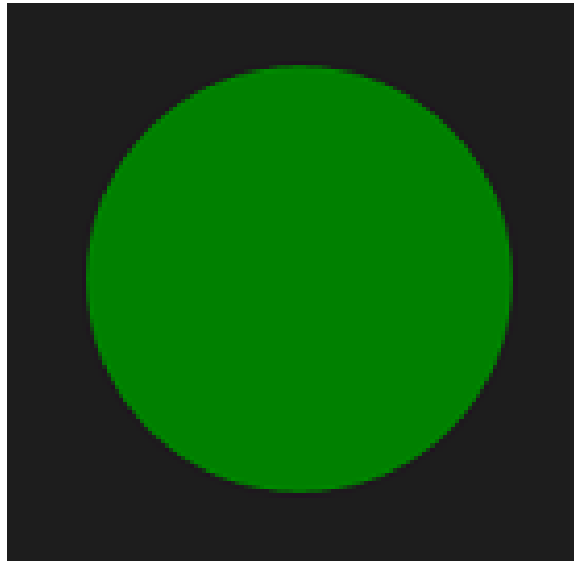
```
    <!--元素属性值-->
```

```
  </object.property>
```

```
</object>
```

属性元素语法

```
<Ellipse Width="150" Height="150">  
  <Ellipse.Fill>  
    <SolidColorBrush Color="Green"/>  
  </Ellipse.Fill>  
</Ellipse>
```



内容元素语法

- 某些元素的属性支持内容元素语法，允许忽略元素的名称
- 实例对象会根据XAML元素中的第一个标记值来设置属性
- 对于大量的格式化文本，使用内容元素语法更加灵活
- 属性标记之间可以插入大量的文本内容

<TextBlock>Windows 8</TextBlock>

内容元素语法

```
<TextBlock Width="200" TextWrapping="Wrap">
```

Windows 8 是微软即将推出的最新 Windows 系列系统。Windows 8 支持个人电脑（Intel 平台系统）及平板电脑（Intel 平台系统 或 ARM 平台系统）。

Windows 8 大幅改变以往的操作逻辑，提供最佳的屏幕触控支持。

```
</TextBlock>
```

Windows 8 是微软即将推出的最新 Windows 系列系统。Windows 8 支持个人电脑（Intel 平台系统）及平板电脑（Intel 平台系统 或 ARM 平台系统）。Windows 8 大幅改变以往的操作逻辑，提供最佳的屏幕触控支持。

集合语法

- 元素支持一个属性元素的集合，才使用集合语法进行设置属性
- 使用托管代码的Add方法来增加更多的集合元素
- 本质是向对象的集合中添加属性项

集合语法

```
<Rectangle Width="200" Height="150">  
  <Rectangle.Fill>  
    <LinearGradientBrush>  
      <GradientStopCollection>  
        <GradientStop Offset="0.0"  
Color="Gold"/>  
        <GradientStop Offset="1.0"  
Color="Green"/>  
      </GradientStopCollection>  
    </LinearGradientBrush>  
  </Rectangle.Fill>  
</Rectangle>
```

简化



集合语法

```
<Rectangle Width="200" Height="150">  
  <Rectangle.Fill>  
    <LinearGradientBrush>  
      <LinearGradientBrush.GradientStops>  
        <GradientStop Offset="0.0" Color="Gold"/>  
        <GradientStop Offset="1.0" Color="Green"/>  
      </LinearGradientBrush.GradientStops>  
    </LinearGradientBrush>  
  </Rectangle.Fill>  
</Rectangle>
```

省略GradientStopCollection
隐式的属性设置方法



附加属性

- 附加属性作用于支持附加属性的元素
- 附加属性是由支持附加属性的父元素产生作用，支持附加属性的元素会继承所在的父元素的属性
- 附加属性的格式：
`AttachedPropertyProvider.PropertyName`

附加属性

<Canvas>

<Rectangle Canvas.Left="50" Canvas.Top="50" Width="200"
Height="150" RadiusX="10" RadiusY="10" Fill="Gold"/>

</Canvas>



依赖属性

- 英文名称：Dependency Properties
- 依赖属性和CLR属性类似，提供一个实例级私有字段的访问封装，通过GetValue和SetValue访问器实现属性的读写操作
- 最重要一个特点是属性值依赖于一个或者多个数据源，提供这些数据源的方式也可以不同
- 由于依赖多数据源的缘故，故称之为依赖属性

依赖属性

可以通过多种不同类型的数据源进行赋值，不同赋值顺序影响属性值的改变



依赖属性

```
<Page.Resources>
<Style x:Key="ButtonStyle" TargetType="Button">
    <Setter Property="Foreground" Value="Red"/>
    <Setter Property="FontSize" Value="24"/>
</Style>
</Page.Resources>

<Button Content="依赖属性测试" Style="{StaticResource ButtonStyle}"
Width="240"/>
```



依赖属性测试

依赖属性

```
<Button Content="依赖属性测试"  
Style="{StaticResource ButtonStyle}" Width="240"  
Foreground="Yellow" FontFamily="14"/>
```



依赖属性测试

事件(Event)

- Windows消息机制中重要概念之一，最常见的人机交互手段之一
- XAML帮助应用管理用户输入，执行不同的行为
- 引入增强型事件处理系统-Routed Event(路由事件)
- Windows 8应用开发中，事件常常被用于控制更改通知操作

事件基础语法

基础语法：<ObjectName EventName= "EventHandle" >

XAML:

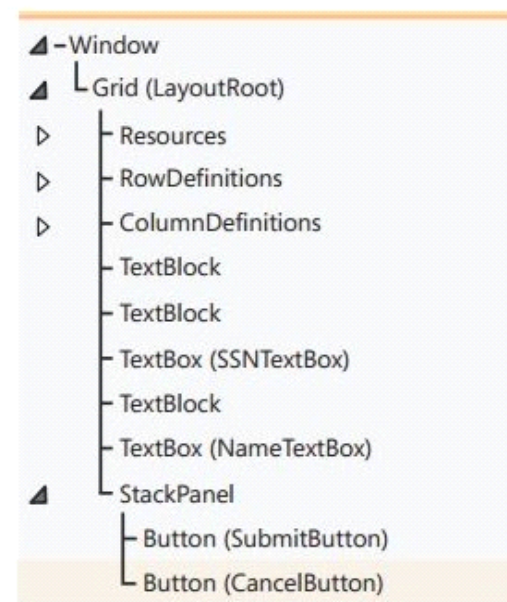
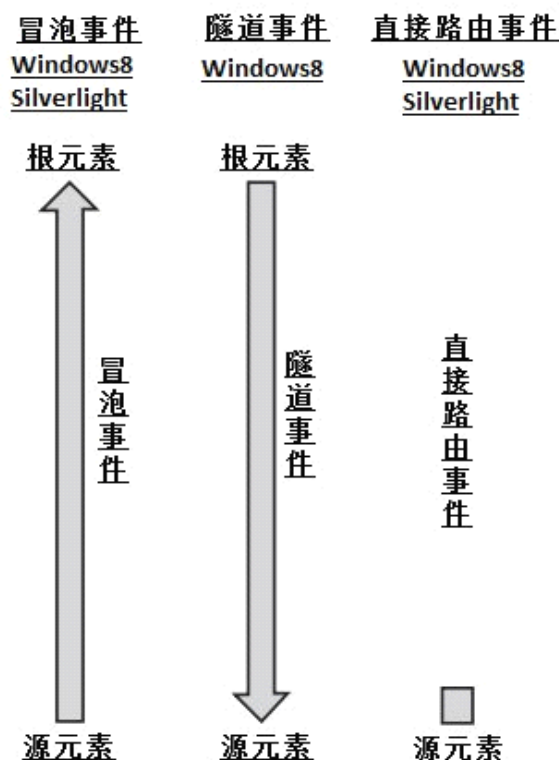
```
<Button Click="Button_Click_1"/>
```

C#:

```
private void Button_Click_1(object sender, RoutedEventArgs e)
{
}
}
```

路由事件处理方式

- Bubbling Event
- Tunneling Event
- Direct Routing Event



标记扩展(Markup Extensions)

- 实际项目中为XAML控件属性赋值经常遇到：
 - 设计时属性值处于未知状态
 - 运行时才能获取到
- 轻松实现XAML页面属性赋值，资源引用，类型转换等操作

常用标记扩展

Binding

XAML载入时，将数据绑定到XAML对象

StaticResource

引用数据字典中定义的静态资源

RelativeSource

对特定数据源绑定

TemplateBinding

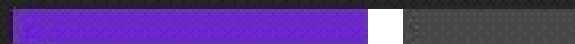
XAML页面中对象模板绑定调用

Binding

```
<TextBox Text="{Binding Path=UserName}"/>
```

```
<TextBox Text="{Binding ElementName=sliderOneTime  
DataSource, Path=Value, Mode=OneTime}" .../>
```

TwoWay(双向绑定)



66

StaticResource

```
<ListBox Height="200" ItemsSource="{Binding Source
e={StaticResource teamsCVS}}">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <Border Background="{Binding Color}" Width="200"
      CornerRadius="10" HorizontalAlignment="Left">
        <TextBlock Text="{Binding Name}" Style="{StaticResource BasicTextStyle}" HorizontalAlignment="Center" FontWeight="Bold"/>
      </Border>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```

TemplateBinding

```
<Style x:Key="ButtonStyle" TargetType="Button">
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="Button">
        <Grid Background="Transparent">
          <TextBlock x:Name="text"
Text="{TemplateBinding Content}" Style="{StaticResource
SubheaderTextStyle}"/>
        </Grid>
      </ControlTemplate>
    </Setter.Value>
  </Setter>
</Style>
```


RelativeSource两种应用模式

两种应用模式：Self Mode和TemplatedParent Mode

语法结构：

```
<Binding RelativeSource="{RelativeSource Self}" .../>
```

-或-

```
<object property="{Binding RelativeSource={RelativeSource Self} ...}" .../>
```

```
<Binding RelativeSource="{RelativeSource TemplatedParent}" .../>
```

-或-

```
<object property="{Binding RelativeSource={RelativeSource Tem
```

使用Self模式

- 目标对象将作为源对象绑定到自身
- 可以实现同一对象元素不同属性之间的绑定操作

```
<TextBox Text="{Binding Path=UserName}" ToolTipService.ToolTip="{Binding Text, RelativeSource={RelativeSource Self}}" />
```

使用TemplatedParent模式

- 仅在ControlTemplate或者DataTemplate下有效
- 不同的模板，将返回不同类型的绑定结果

```
<Style TargetType="local:Calendar">
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="local:Calendar">
        <Grid>
          <TextBlock
            Text="{Binding Path=Namer,RelativeSource={RelativeSource TemplatedParent},Mode
            =TwoWay}" />
          <TextBlock Text="{TemplateBinding Namer}" />
        </Grid>
      </ControlTemplate>
    </Setter.Value>
  </Setter>
</Style>
```

大小写区分

- XAML 是一种区分大小写的语言
- XAML 元素和属性的名称区分大小写
- 属性值可能区分大小写，这取决于如何针对特定属性处理属性值

空白规范化

- 移除东亚字符之间的换行符
- 将所有空白字符（空格、换行符和制表符）转换为空格
- 删除所有连续的空格，将其替换为一个空格
- 删除紧跟在开始标记后面的一个空格
- 删除紧靠在结束标记前面的一个空格

The background is a solid orange color. It is decorated with several white outline shapes that resemble stylized leaves or droplets. These shapes are scattered across the slide, with some overlapping the text. There are approximately 15-20 such shapes in total, varying in size and orientation.

XAML语法

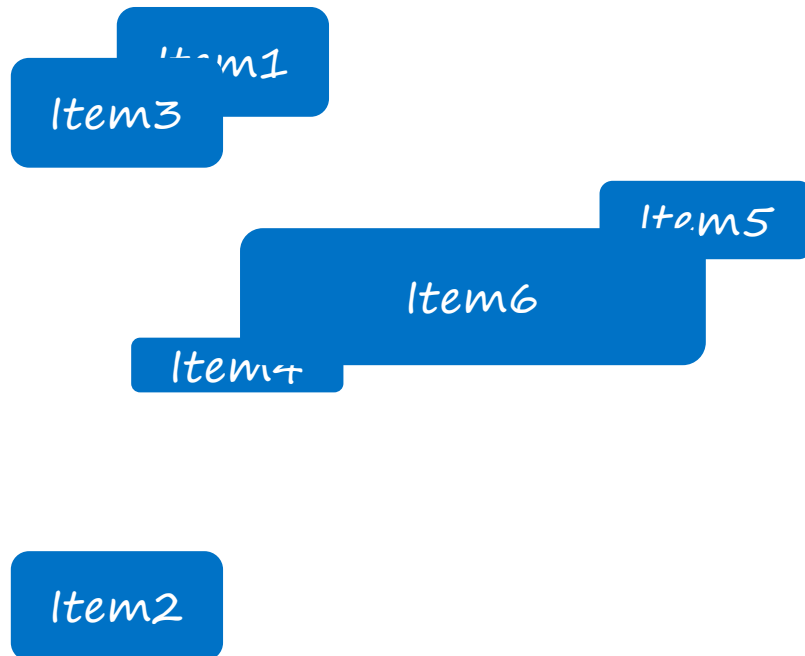
演示

常用控件

布局控件

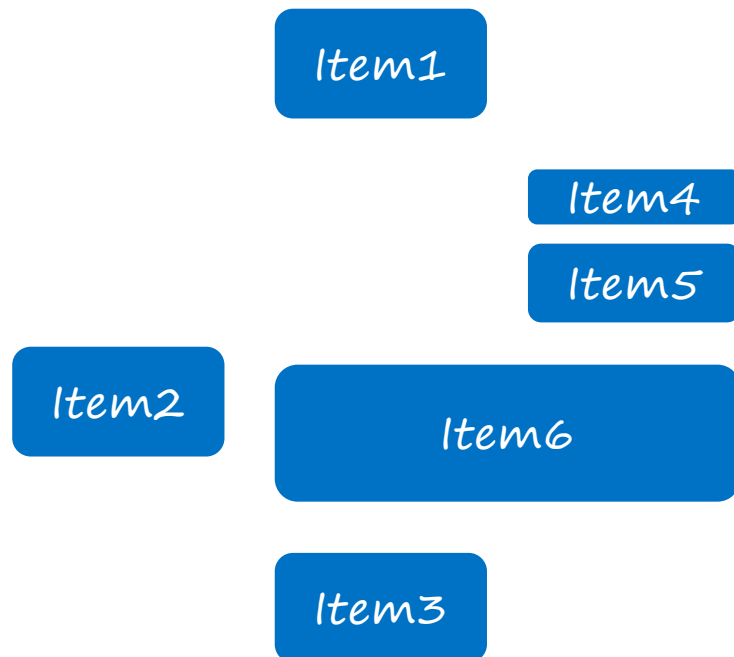
- Canvas
- Grid
- StackPanel

Canvas



```
<Canvas ... >  
  <Button  
    Canvas.Left=200  
    Canvas.Top =50  
    Content="Item1"  
    Background="Green"  
    Width="150"  
    Height="50" />  
    .....  
    .....  
</Canvas>
```

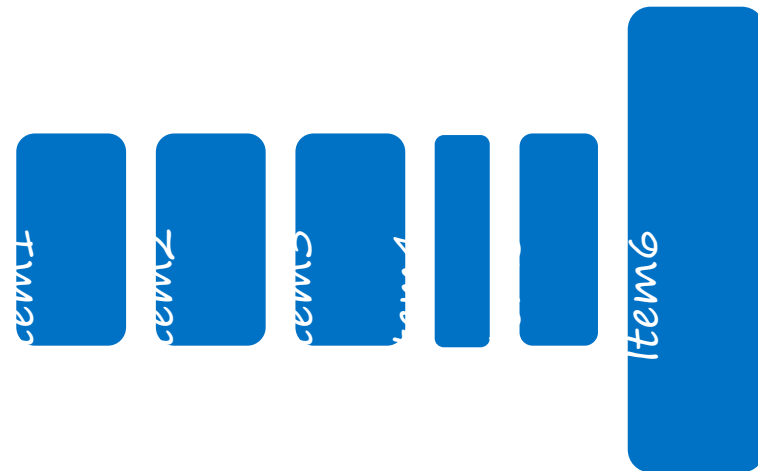
Grid



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

  <Button
    Grid.Row="2"
    Grid.Column="0"
    Content="Item2"
    Background="Green"
    Width="200"
    Height="100" />
  ...
</Grid>
```

StackPanel



```
<StackPanel Orientation="Horizontal">  
  <Button Content="Item1" Background="Green"  
    Width="200" Height="100" />  
  <Button Content="Item2" ... />  
  ...  
</ StackPanel >
```

布局控件 演示

基础控件

按钮



Submit Query

OK

复选框



组合框

California



- Alaska
- Arizona
- Arkansas
- California
- Colorado
- Connecticut
- Delaware
- Florida
- Georgia
- Hawaii

日期选取器

September



13



2011



超链

<http://www.buildwindows.com>

列表框

- Apple
- Banana
- Grape
- Orange
- Watermelon

进度条



进度环



单选按钮



评级



滑块



时间选取器

3



15



PM



切换开关

On



Off



Button

```
<Button Margin="0,0,0,0"  
Content="Button" Click="Button_Click" HorizontalAlignment="Left" VerticalAlignment="Top"/>
```

```
private void Button_Click(object sender, RoutedEventArgs e)  
{  
  
}
```



RadioButton

```
<RadioButton x:Name="RadioButton1" Content="{Binding Text,  
ElementName=RadioButtonContent}  
" Grid.Row="4" Margin="0,0,0,0" HorizontalAlignment="Left" VerticalAlignment="Top" />
```

☐ RadioButton Content

Content:

☒ IsChecked

HyperlinkButton

```
<HyperlinkButton Content="{Binding  
Text,ElementName=HyperlinkButtonContent}  
" Grid.Row="3" Margin="5,0,0,0" Click="HyperlinkButton_Click" VerticalAlignment="Top"/>
```

```
void HyperlinkButton_Click(object sender, RoutedEventArgs e)  
{  
    ClickMessage2.Text = "You clicked the HyperlinkButton";  
}
```

HyperlinkButton Content

Content:

Click Message: You clicked the HyperlinkButton

CheckBox

```
<CheckBox x:Name="CheckBox1" Content="{Binding  
Text,ElementName=CheckBoxContent}" Grid.Row="2" Margin="0,0,0,0" HorizontalAlignment="Left"  
VerticalAlignment="Top"/>
```



CheckBox Content

Content:

CheckBox Content



IsChecked

ToggleSwitch

```
<ToggleSwitch Grid.Row="1" Width="130" Margin="0,0,0,0" OffContent="{Binding  
Text,ElementName=OffContent}" OnContent="{Binding Text, ElementName=OnContent}" Header="{Binding Text,  
ElementName=HeaderText}" VerticalAlignment="Center"/>
```

Having fun yet?

Nope



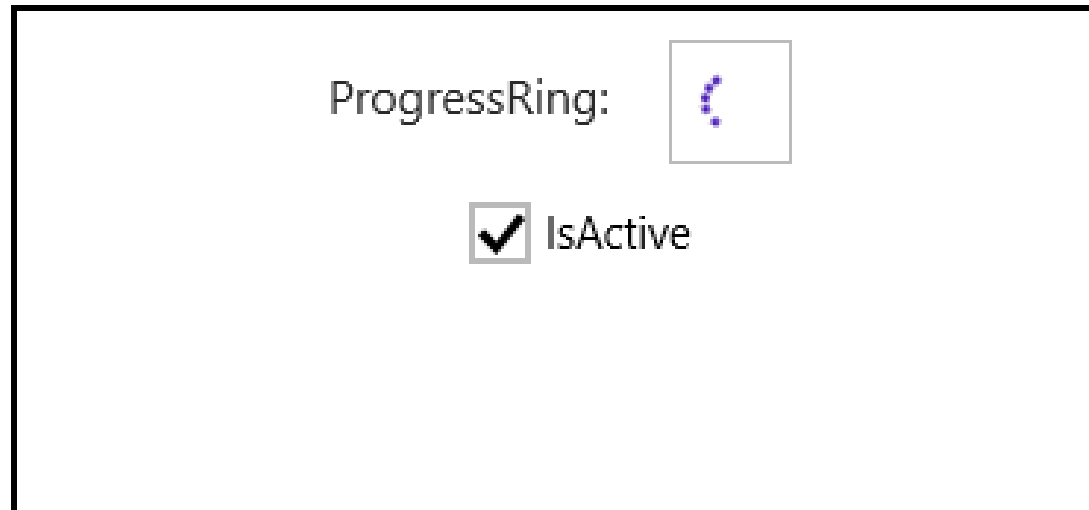
Header: Having fun yet?

OffContent: Nope

OnContent: Sure

ProgressRing

```
<ProgressRing x:Name="ProgressRing1" IsActive="{Binding IsChecked, ElementName=ActiveCB}"/>
```



ProgressBar

```
<ProgressBar x:Name="ProgressBar1" IsIndeterminate="{Binding IsChecked, ElementName=ActiveCB2}"
Width="100" Value="{Binding Text, ElementName=Value, Converter={StaticResource
StringToDoubleConverter}}" ShowPaused="{Binding IsChecked,ElementName=PausedCB}" ShowError="{Binding
IsChecked, ElementName=ErrorCB}"/>
```

ProgressBar:

.....

☒ IsIndeterminate: Value:

(max value is 100)

☐ ShowPaused

☐ ShowError

ComboBox

```
<ComboBox x:Name="ComboBox1" Width="150" Margin="20,0,0,0"
SelectionChanged="ComboBox1_SelectionChanged">
  <TextBlock Text="Item 1"/>
  <TextBlock Text="Item 2"/>
  <TextBlock Text="Item 3"/>
  <TextBlock Text="Item 4"/>
  <TextBlock Text="Item 5"/>
</ComboBox>
```

Output

ComboBox:

Item 1
Item 2
Item 3

SelectedItem:

Slider

```
<Slider x:Name="Slider1" Width="130" Margin="60,0,0,0" VerticalAlignment="Center"/>
```

Slider:



Value: 59

The background is a solid orange rectangle. Scattered across it are several white, stylized geometric shapes. These shapes are composed of thin white outlines and include circles, semi-circles, and teardrop-like forms. Some shapes are partially cut off by the edges of the frame. The overall aesthetic is modern and minimalist.

基础控件

演示

文本编辑控件

单行文本框

http://www.microsoft.com



清除按钮

多行文本框

Using built-in controls to create great Metro style apps

Using built-in controls to create great Metro style apps

Presenter:

Paul Gusmorino
Senior Program Manager Lead
Microsoft Corporation

Key Takeaways:

- New and improved set of built-in controls for making great Metro style apps.
- Built-in controls designed for touch, mouse, and keyboard.
- Style controls to take on your own app's look, while preserving behaviors and capabilities.
- Available to both HTML and XAML in a way that is 'native' to each language.

Notes:

Great talk! Super excited to see how I can apply my HTML and XAML knowledge to make great Metro apps for Windows 8!

密码

.....



显示按钮

格式文本框

This is a *Rich Text Control* from the app
NoteSpace!

文本编辑控件 - 行为

文本选择：鼠标和键盘

BUILD is a new event that shows modern hardware and software developers how to take advantage of the future of Windows. Learn how to work with the all new touch-centric user experience to create fast, fluid, and dynamic applications that leverage the

触摸

Tuesday, Sept 13

7:00am - 9:00am

Copy

9:00am - 11:00am

Keynote

11:30am - 6:00pm

Sessions

剪切、复制和粘贴

Cut

Copy

Paste

<http://www.microsoft.com>

拼写检查

attendees

attendee's

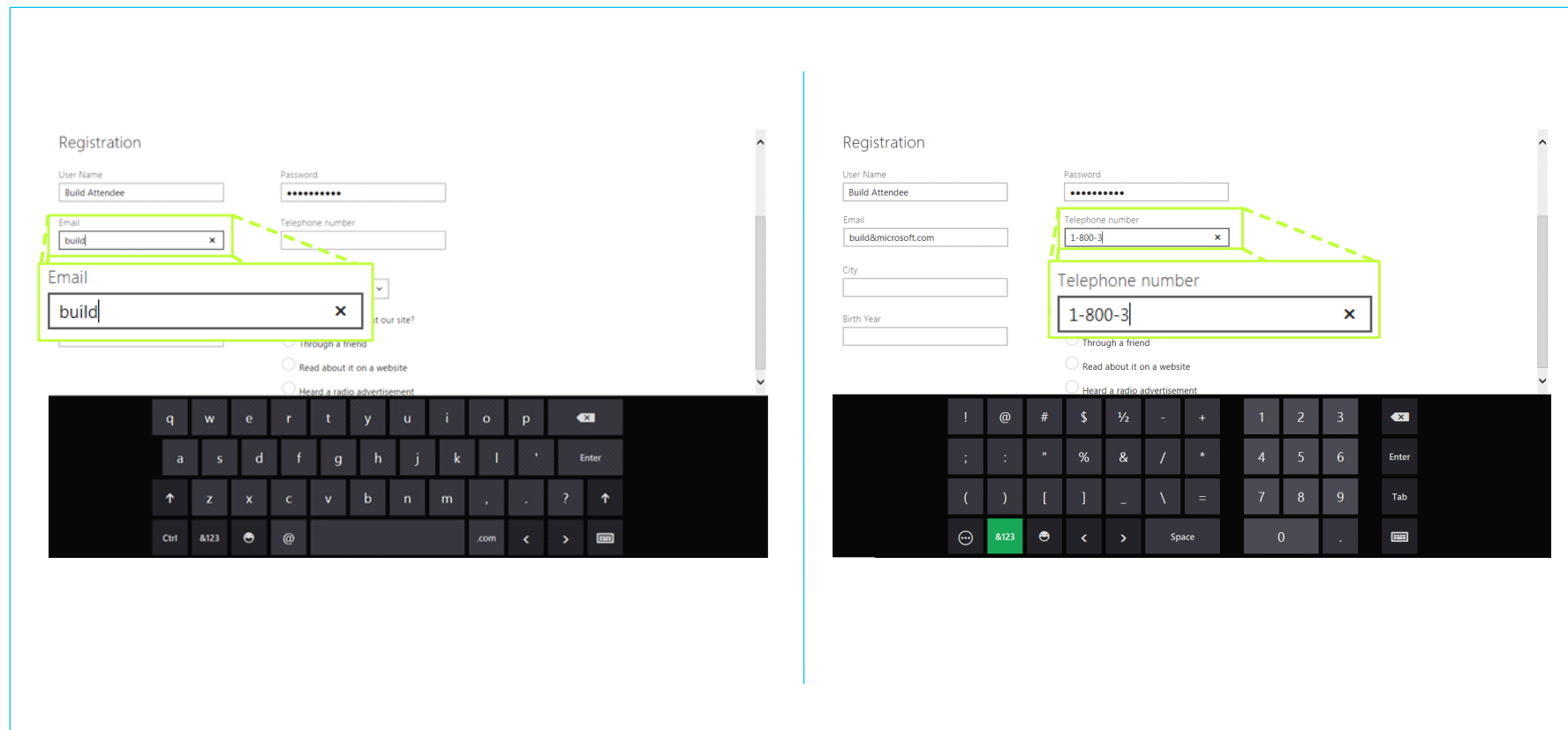
attendees'

Add to dictionary

Ignore

Hello BUILD attendies!

文本编辑控件 - 触摸键盘



TextBlock

```
<TextBlock Style="{StaticResource BasicTextStyle}" Height="250" Width="400" IsTextSelectionEnabled="True"
TextWrapping="Wrap" TextTrimming="WordEllipsis" Text="....." />
```

TextBlock

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ac mi ipsum. Phasellus vel malesuada mauris. Donec pharetra, enim sit amet mattis tincidunt, felis nisi semper lectus, vel porta diam nisi in augue. Pellentesque lacus tortor, aliquam et faucibus id, rhoncus ut justo. Sed id lectus odio, eget pulvinar diam. Suspendisse eleifend ornare libero, in luctus purus aliquet non. Sed interdum, sem vitae rutrum rhoncus, felis ligula ultrices sem, in eleifend eros ante id neque. Vivamus quam lorem, varius vitae porta mollis, placerat quis est. Aenean eget sagittis massa. Sed tellus turpis, ullamcorper eget imperdiet vel, faucibus vel nisl. Nunc sed suscipit quam. Vivamus convallis faucibus dignissim. Fusce sollicitudin, quam...

TextBox

```
<TextBox Height="250" Width="400" TextWrapping="Wrap" AcceptsReturn="True" Text="....."/>
```

TextBox

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ac mi ipsum. Phasellus vel malesuada mauris. Donec pharetra, enim sit amet mattis tincidunt, felis nisi semper lectus, vel porta diam nisi in augue. Pellentesque lacus tortor, aliquam et faucibus id, rhoncus ut justo. Sed id lectus odio, eget pulvinar diam. Suspendisse eleifend ornare libero, in luctus purus aliquet non. Sed interdum, sem vitae rutrum rhoncus, felis ligula ultrices sem, in eleifend eros ante id neque. Vivamus quam lorem, varius vitae porta mollis, placerat quis est. Aenean eget sagittis massa. Sed tellus turpis, ullamcorper eget imperdiet vel, faucibus vel nisl. Nunc sed suscipit quam. Vivamus

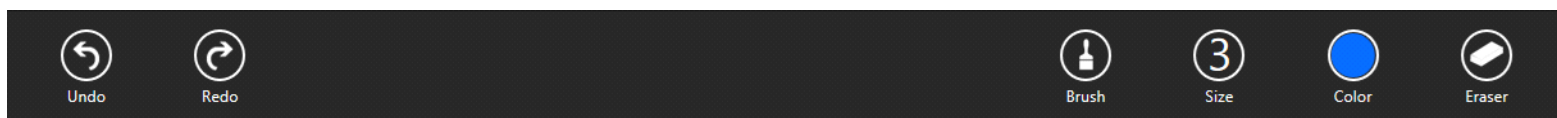
The background is a solid orange color. It is decorated with several white outline shapes that resemble stylized leaves or droplets. These shapes are scattered across the slide, with a cluster in the top left, a few in the top right, one in the center, and a group in the bottom right.

文本编辑控件

演示

命令界面

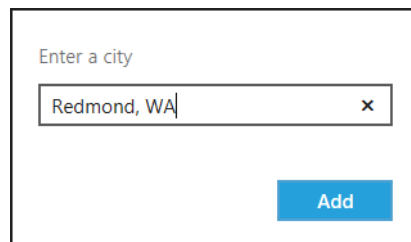
应用栏



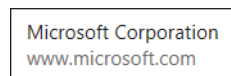
上下文菜单



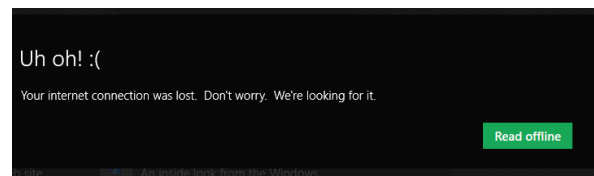
弹出窗口



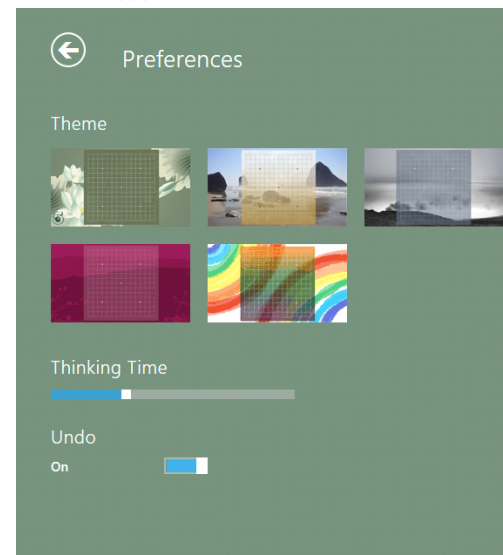
工具提示



消息对话框



设置窗格



AppBar

```
<AppBar x:Name="BottomAppBar1" Padding="10,0,10,0">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="50*" />
      <ColumnDefinition Width="50*" />
    </Grid.ColumnDefinitions>
    <StackPanel x:Name="LeftPanel" Orientation="Horizontal" Grid.Column="0" HorizontalAlignment="Left">
      <Button x:Name="Edit" Style="{StaticResource EditAppBarButtonStyle}" Tag="Edit" />
      <Button x:Name="Save" Style="{StaticResource SaveAppBarButtonStyle}" Tag="Save" />
      <Button x:Name="Delete" Style="{StaticResource DeleteAppBarButtonStyle}" Tag="Delete" />
    </StackPanel>
    .....
  </Grid>
</AppBar>
```



Edit



Save



Delete



Refresh



Previous

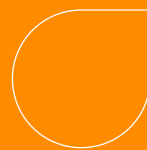


Next



Help

应用栏 演示



总结

XAML概览：XAML简介、为什么学习XAML、XAML作用

基础语法：命名空间、对象声明、属性设置、事件、标记扩展、大小写区分、空白规范化

常用控件：布局控件、基础控件、文本编辑控件、命令界面

相关资源

[XAML Overview](#)

Windows 8 Sample Code

[《Programming Windows Sixth Edition》](#)

<http://shop.oreilly.com/product/0790145369079.do>

[《Programming Windows Phone 7》](#)

<http://shop.oreilly.com/product/0790145316707.do?intcmp=ba-ms-books-int-search-windows-phone-ct>



Q&A

© 2011 Microsoft Corporation。保留所有权利。Microsoft、Windows、Windows Vista 及其他产品名称是或者可能是在美国和 /或其他国家/地区的注册商标和 /或商标。此处包含的信息仅供参考，并不代表 Microsoft Corporation 截至本演示文稿发布之日的最新观点。由于 Microsoft 必须响应不断变化的市场条件，所以不应将本文视为 Microsoft 一方的承诺，Microsoft Corporation 也无法保证所提供信息在本文发布之后的准确性。MICROSOFT 对本演示文稿中包含的信息不做任何明示、暗示或法定的担保。

The Microsoft logo is centered in a white, italicized serif font. The background is a solid blue color. Scattered around the logo are several white-outlined squares of various sizes, some of which are connected by thin white lines, creating a geometric pattern.

Microsoft[®]

© 2011 Microsoft Corporation。保留所有权利。Microsoft、Windows、Windows Vista 及其他产品名称是或者可能是在美国和 /或其他国家/地区的注册商标和 /或商标。此处包含的信息仅供参考，并不代表 Microsoft Corporation 截至本演示文稿发布之日的最新观点。由于 Microsoft 必须响应不断变化的市场条件，所以不应将本文视为 Microsoft 一方的承诺，Microsoft Corporation 也无法保证所提供信息在本文发布之后的准确性。MICROSOFT 对本演示文稿中包含的信息不做任何明示、暗示或法定的担保。