

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

XMAL 啊 又一种语言

徐栋

北京中达金桥技术服务有限公司

xudong@itgoldenbridge.com

议程

- XAML简介
- XAML编程入门
- XAML里的精彩之处
- 总结

WPF

- **Windows Presentation Foundation**
- 是**WinFX**的三大基石之一, 未来界面设计的统一平台
- **XAML**是**WPF**界面设计的基础语言
- **XAML**利用了**WPF**强大的运行时环境
- 具有强大的功能和灵活的可移植性

XAML

- **Extensible Application Markup Language**
- 对应用界面的描述性语言，类似HTML
 - 界面设计可以和程序逻辑开发有效的分离
- 和.NET运行时环境中的对象模型良好交互
- **XAML**比传统的代码界面描述更直观，被大多数界面开发人员熟悉，且节省工作量

XAML的特点

- **XAML是真正的XML**（完整的对象模型）
 - 标签代表对象
 - 标签属性代表对象属性
 - **XAML界面定义被编译成partial classes**
- **但XAML不是其他编程语言的替代品**
 - 集中关注于界面编程
 - 不适合做流程处理和逻辑判断
 - 和其他语言配合使用

XAML编程入门

Hello World

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 用XAML向世界问好 – **Demo**
- `<Canvas xmlns="http://schemas.microsoft.com/winfx/avalon/2005">`
- `<TextBlock Margin="10,10,10,0" Background="BlanchedAlmond"`
`FontFamily="Comic sans MS" FontSize="36pt">Hello`
`World!</TextBlock>`
- `</Canvas >`
- 因为没有C#等逻辑代码, 所以不用编译, 可以直接被浏览器里的插件环境解释运行
- 当然, 也可以在编译后, 象应用程序一样运行

传统程序语言不适合界面创建

Hello World

- Hello World App in C#

```
using System;
using MSAvalon.Windows;
using MSAvalon.Windows.Controls;
using MSAvalon.Windows.Media;

class HelloWorldApp: Application { [STAThread] static void Main()
    { HelloWorldApp app = new HelloWorldApp();
      app.Run(); }

protected override void OnStartingUp(StartingUpCancelEventArgs args)
    { Window win = new Window();

      TextPanel tp = new TextPanel();
      tp.Background = Brushes.BlanchedAlmond;
      tp.FontFamily = "Comic sans MS";
      tp.FontSize = new FontSize(72f, FontSizeType.Point);
      tp.HorizontalAlignment = MSAvalon.Windows.HorizontalAlignment.Center;
      tp.TextRange.Text = "Hello, world";
      win.Children.Add(tp);
      win.Show();
    }
```

XAML也可以加入程序逻辑代码

- 在WPF中同时使用代码和XAML语言（需要编译）
- 可以把代码直接插入到XAML文件中

<Canvas>

```
—         <Button Click="Button_Click" Name="btn1">Click
—         Me!</Button>
—
—         <x:Code>
—             <![CDATA[
—                 void Button_Click(object sender,
—                 RoutedEventArgs e)
—                 {
—                     btn1.Background = Brushes.Red;
—                 }
—             ]]>
—         </x:Code>
—     </Canvas>
```


- 也可以把**XAML**和代码分开

XAML File

<Canvas>

<Button Click="Button_Click">Click Me!</Button>

</Canvas>

C# code-behind file

```
– namespace AvalonApplication1
– {
–     public partial class Window1 : Window
–     {
–         // Sample event handler:
–         // private void ButtonClick(object sender, RoutedEventArgs e) {}
–         void Button_Click(object sender, RoutedEventArgs e)
–         {
–             btn1.Background = Brushes.Red;
–         }
–     }
– }
```

您的潜力. 我们的动力

用VS.NET2005开发WPF应用

Microsoft
微软(中国)有限公司

- WinFX现在处于测试期间（January CTP）
- 安装WinFX Runtime
- 安装VS.NET2005
- 安装WinFX SDK
- 安装 VS2005 Extensions for WinFx

“统一的编程模型”

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 现在的情况是代码和界面描述不能很好的协调
 - 代码可以很好的处理逻辑流程，但对于描述文档的布局 and 用户界面就显得过于复杂和难于掌握
 - 标签语言（**HTML**等）非常适合进行文档和图片的定位整合，却不能很好的处理程序逻辑和用户界面交互
- **WPF**整合了上述两种技术的优点
- 把代码技术和标签语言技术进行无缝整合的统一模式
 - 大多数**WPF**应用程序都会整合**XAML**语言和程序逻辑代码

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

Demo

- 用VS.NET2005开发WPF应用

XAML开发的主要内容

- Controls
- Layout
- Graphics
 - Text
 - Imaging
 - 2D
 - 3D
 - Effects
 - Audio
 - Video
 - Animation and Timing
- Style

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

XAML里的精彩之处

- 灵活的平台级控件开发体系
- 更方便的控件布局能力
- 利用了图形加速能力的图象处理管道
- 灵活丰富的样式功能和样式模板
- 强大的声明式数据绑定

您的潜力, 我们的动力

界面控件: UI设计的基石

挑战

Microsoft
微软(中国)有限公司

传统控件并不灵活

“灰盒”子问题

对于传统控件, 你只能对其属性进行有限度的定制

全新的平台, 全新的控件

只提供基础的控件, 和强大的灵活性, 是平台级的

复杂结构

- “WPF” 控件可以再包含任意的控件和复杂结构
 - 使用统一平台和简单概念, 创建出满足各种需求的复杂控件
- 几乎可以淘汰自定义控件了
 - 很少的几个基本控件, 但可以有强大的组合功能
 - 按钮 == 图片按钮 == 菜单按钮?!?
- 甚至可以包含数据对象
 - 动态双向绑定
 - 界面开发与数据逻辑处理分开

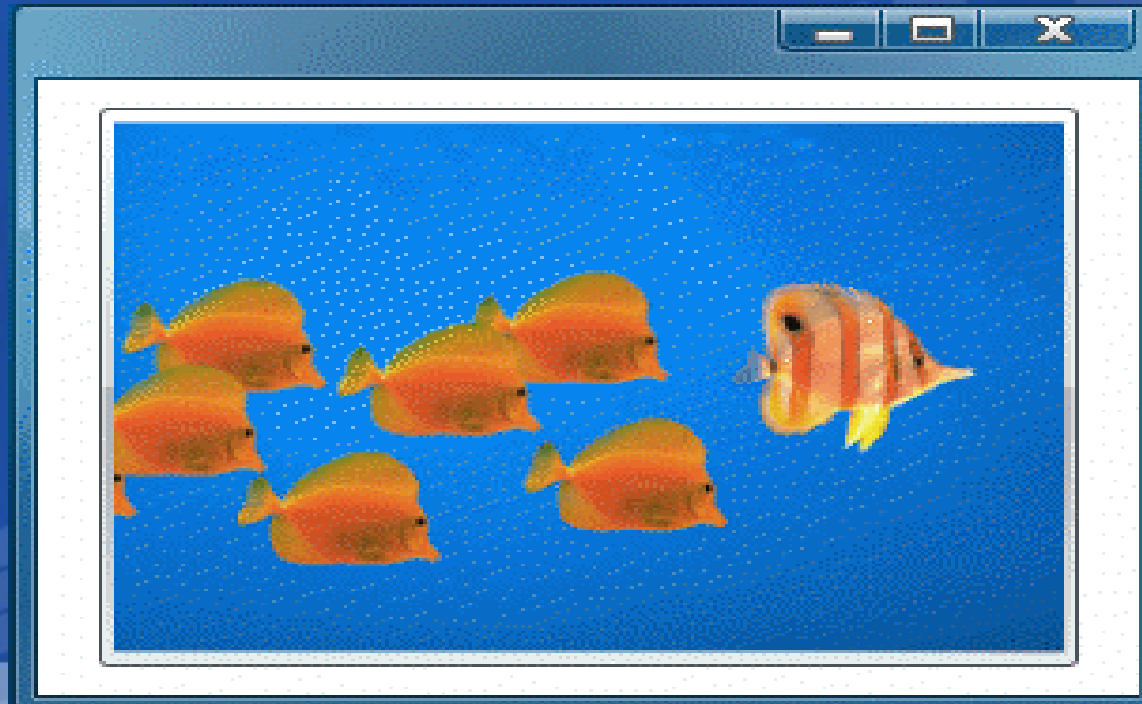
布局 (Layout)

<StackPanel>

<Button Width="260" Height="200" > <Image
Source="C:\WINDOWS\follow.jpg" />

</Button>

<StackPanel>



您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Canvas

DockPanel

Panel

StackPanel

Grid

WrapPanel

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

内容大小的适应

Scroll Viewer

对内容提供了滚动条

ViewBox

对内容提供了缩放自适应

图形处理 (Graphics)

- **WPF** 设计初始的目标就是全面替代旧的图形处理技术, 充分利用现代**PC**技术的新发展.
- 所有的图形元素统一到了统一的处理管道: **2D, 3D, text, imaging and video.**

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

变形

- 旋转

- 缩放

- 倾斜

- 转变

样式 (Style)

样式使设计人员创建出灵活界面

样式可以达到栩栩如生的效果

样式是 “WPF” 技术的粘合剂

您的潜力, 我们的动力

Microsoft®
微软(中国)有限公司

Simple Style

- 包含一组设置
- 每组设置都有属性和值

Extending style

- 基于简单的样式
- 基于已有WPF元素的样式

Trigger

- 允许样式根据用户的操作来改变

Event Trigger

- 支持故事板 (可以制作动画效果)

样式模板 (Template)

- 描述界面元素或数据类型的样式表现, 形成模板
- 可以大大的增加灵活性和界面的定制性
- 每个界面元素都有默认的显示模板, 都可以进行改变

数据绑定 (DataBinding)

Microsoft
微软(中国)有限公司

- 使界面元素（控件）的属性与后台数据源同步
- 有不同的绑定模式，适应不同的需求
- 采用声明式的绑定模式

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

绑定模式

动态绑定

双向绑定

变化绑定

ms

casts

总结


- **XAML**是一种新的完全基于**XML**的界面描述语言
- 其底层运行时环境由强大的**WPF**构成
- 语言模型完善, 功能丰富, 特性齐全
- 开发简便, 代码可理解性好, 管理方便
- 可以利用**VS.NET**强大的开发能力

开发机会

- 创建丰富用户体验的程序, 扩展了应用领域, 创造新机会
- 继承现有的开发思想, 开发人员学习、转型更容易
- 增强了界面设计功能, 进一步分离了界面设计和逻辑编程, 充分发挥了界面设计人员的能力
- 有完善的开发工具和环境, 在整个产品开发周期内提供强有力的支持
 - **Expression**
 - **VS2005**
 - **WinFX SDK**
 - **XML Tools**



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