

Date

# Visual Studio 2010 DSL 开发与设计实践系列

## 第 5 讲 DSL 的图形符号系统（下）

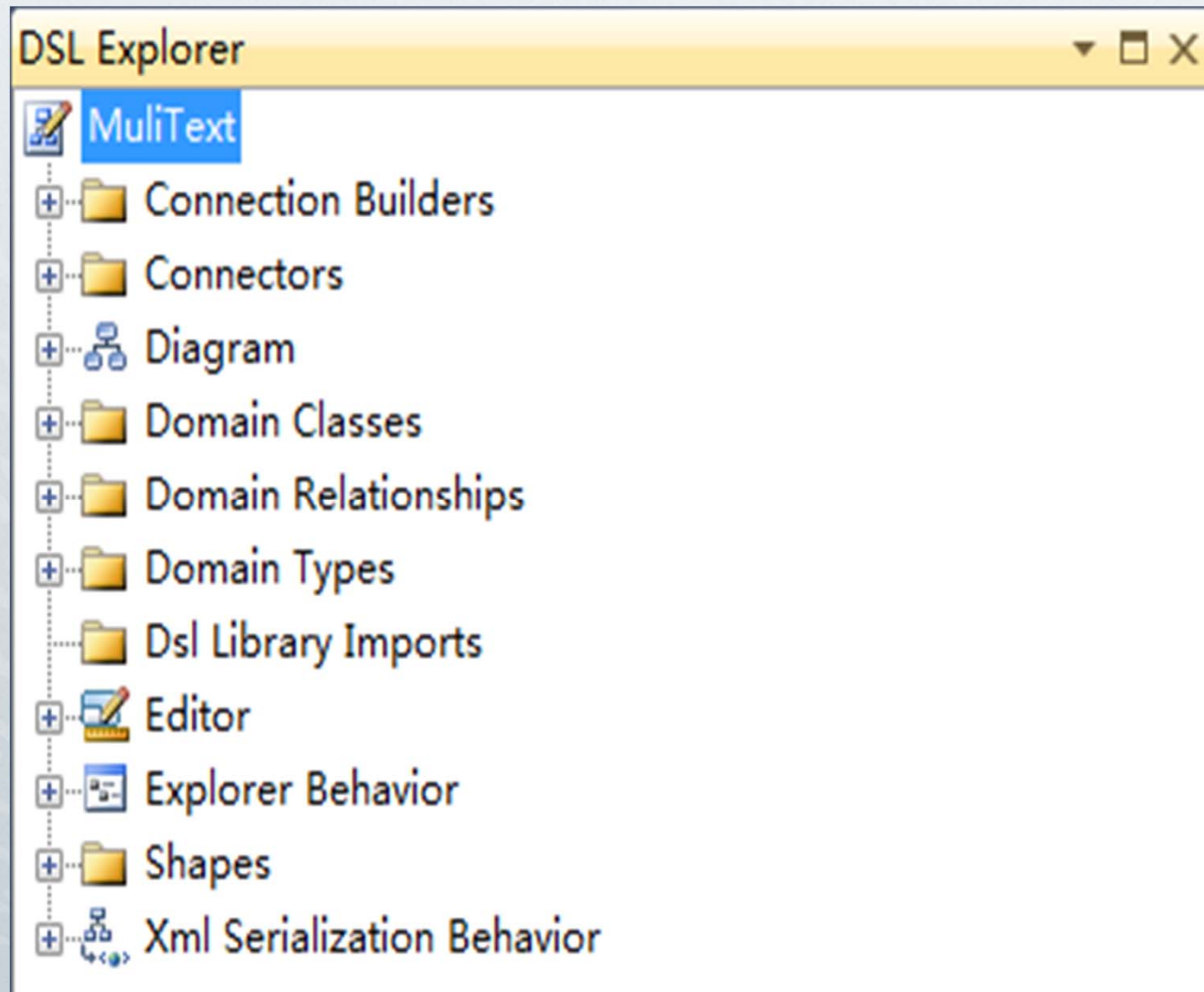
# 自我介绍

- 陈俊先
- .NET 高级讲师

# 内容提要

- **DSL Explorer**介绍
- 设置背景图片
- 动态添加图形元素
- 工具箱
- 编辑器
- 属性窗口

# DSL Explorer



# 设置背景图片

- 背景图片只能通过程序设置

```
- public partial class BackgroundDiagram
- {
-     protected override void InitializeInstanceResources()
-     {
-         ImageField backgroundField = new ImageField("background",
-             Background.Properties.Resources.Tulips);

-         backgroundField.DefaultFocusable = false;
-         backgroundField.DefaultSelectable = false;
-         backgroundField.DefaultVisibility = true;

-         this.ShapeFields.Add(backgroundField);

-         backgroundField.AnchoringBehavior.SetTopAnchor(AnchoringBehavior.Edge.Top, 0.01);
-         backgroundField.AnchoringBehavior.SetLeftAnchor(AnchoringBehavior.Edge.Left, 0.01);
-         backgroundField.AnchoringBehavior.SetRightAnchor(AnchoringBehavior.Edge.Right, 0.01);
-         backgroundField.AnchoringBehavior.SetBottomAnchor(AnchoringBehavior.Edge.Bottom, 0.01);

-         base.InitializeInstanceResources();
-     }
- }
```



# Demo(Background)

# 动态添加图形元素

```
- public partial class DynamicAddElementDiagram
- {
-     public override void OnDoubleClick(DiagramPointEventArgs e)
-     {
-         base.OnDoubleClick(e);
-         ExampleModel model = this.ModelElement as ExampleModel;
-         Transaction trans = model.Store.TransactionManager.BeginTransaction();
-         for (int i = 0; i < 10; i++)
-         {
-             ExampleElement element = new ExampleElement(this.Store);
-             element.Name = "Example" + i;
-             model.Elements.Add(element);
-         }
-
-         trans.Commit();
-     }
- }
```

# Demo (DynamicAddElement)



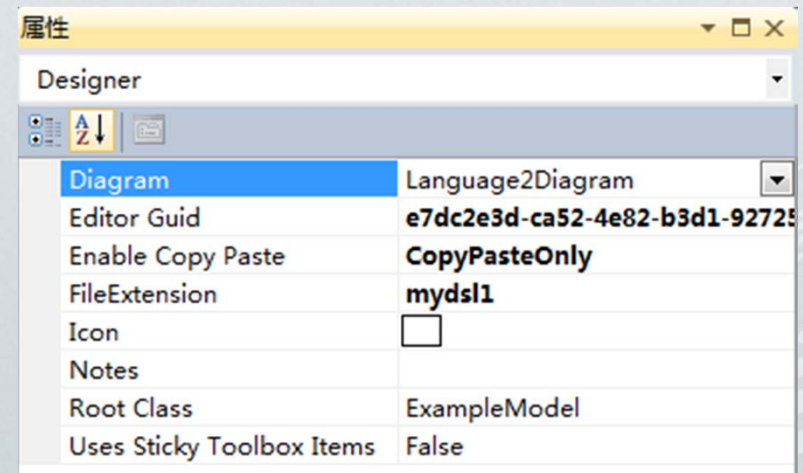
# 工具箱

- 如何添加Tab
- 如何添加域类元素工具
- 如何添加关系元素工具
- 调整工具的排列顺序

# Demo (ToolboxSample)

# 编辑器

- 默认是把设计器作为编辑器，但可以自定义编辑器。编辑器的Diagram属性必须是设计器图表类



# 自定义编辑器

- 自定义编辑器必须提供一个Win Form控件
- override DslPakager 下DocView类的Window属性，添加自定义控件。

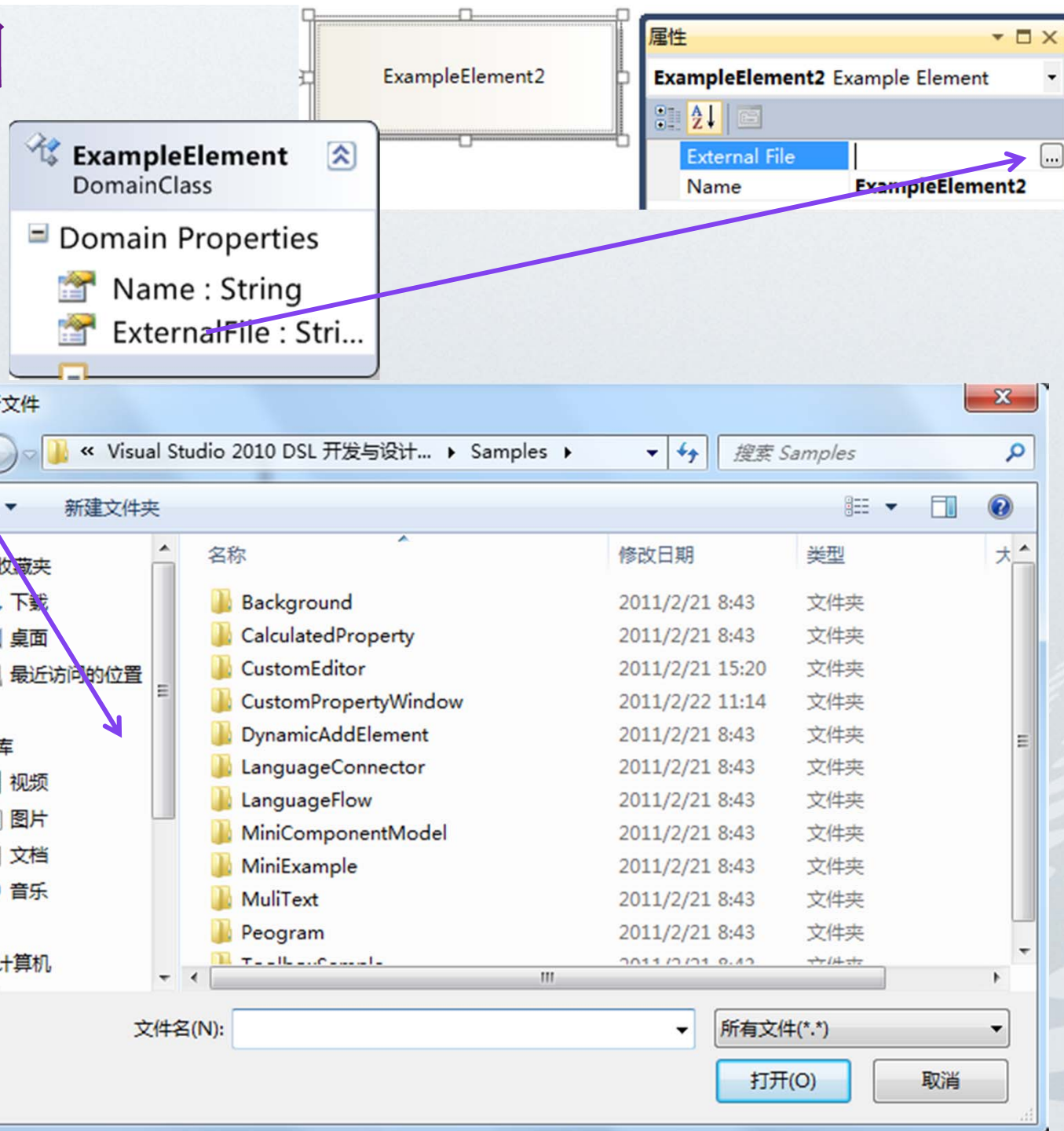


# Demo (CustomEditor)



# 属性窗口

- ExternalFile属性
- 需要弹出“打开文件”窗口



# 属性窗口

## – 生成的程序

- [System.ComponentModel.EditorAttribute(typeof(System.Windows.Forms.Design.FileNameEditor),  
typeof(System.Drawing.Design.UITypeEditor))]
- public global::System.String ExternalFile
- {
  - [global::System.Diagnostics.DebuggerStepThrough]
  - get
  - {
    - return externalFilePropertyStorage;
  - }
  - [global::System.Diagnostics.DebuggerStepThrough]
  - set
  - {
    - ExternalFilePropertyHandler.Instance.SetValue(this, value);
  - }
- }

# Demo (CustomPropertyWindow)



© 2009 Microsoft Corporation. All rights reserved. Microsoft, MSDN, the MSDN logo, and [list other trademarks referenced] are trademarks of the Microsoft group of companies. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.