

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

ASP.NET 2.0 第十五讲 扩充你的ASP.NET应用

MSDN特约讲师 苏鹏



MSDN Webcasts

内容介绍

- 页面框架生命周期扩展
- 服务器端控件运行时扩展
- 页面状态扩展
- 服务器端控件设计扩展
- 应用服务供应者扩展
- 表达式扩展

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

扩展你的ASP.NET应用概述

ASP.NET 2.0尽管为我们提供了很多功能非常完善的功能，但是我们依然经常希望增加我们自己的功能，为此，微软在asp.net架构中提供了很多新的可扩展功能。下面我们来一一介绍一下。

页面框架生命周期

页面框架本身的生命周期包含了很多函数。

Page.InitializeCulture

Page.OnPreInit, Page_PreInit

Control.LoadControlState,

Control.SaveControlState

页面框架生命周期

- 部件初始化
- 加载显示用数据
- 加载返回数据
- 加载控件
- 加载回调事件
- 处理客户端回调事件
- 交付控件
- 保存显示状态
- 交付html
- 撤销

用户控件运行时管理

- ASP.NET中的用户控件`ascx`控件使用的是和页面一样的技术, 所以具有相同的生命周期和类似的加载方式, 同时也提供可视化的设计方法
- `<%-- Register the controls defined in an assembly that resides in the Bin directory --%>`
- `<%@ Register TagPrefix="Demos" NameSpace="DemoControls" Assembly="MyDemoControls" %>`
- `<%-- Register the controls defined in code that resides in the App_Code directory --%> <%@ Register TagPrefix="Demos" NameSpace="DemoControls" %>
<Demos:ControlFromBinAssembly runat="server" ID="CustomControl1" /> <Demos:ControlFromFile runat="server" ID="CustomControl2" />`

控件类的属性

- 用户自定义的控件类可以包含很多其他性质的特性，这些特性可以在开发中由用户自行拓展
- 本地化属性
- 主题属性
- Url属性
- Web资源属性
- 可过滤属性

控件类

ASP.NET提供了Control和WebControl两个基础类，asp.net 2.0提供了一个新的控件基础类CompositeControl

- CompositeControl 类的每个实例都通过CompositeControlDesigner 机制来确保每个类的实例都被创建
- Public Class MyCompositionControl Inherits CompositeControl Protected Overrides Sub CreateChildControls() Controls.Clear() ' Create control collection .. End Sub ' Additional implementation End Class

状态管理

```
Protected Overrides Sub OnInit(ByVal e As System.EventArgs)
    MyBase.OnInit(e) Page.RegisterRequiresControlState(Me) End Sub
Protected Overrides Function SaveControlState() As Object Dim o As
Object = MyBase.SaveControlState() ' Additional implementation Return
New Pair(o, ViewIndex) End Function Protected Overrides Sub
LoadControlState(ByVal savedState As Object) Dim p As Pair = savedState
If Not (p Is Nothing) Then MyBase.LoadControlState(p.First) ViewIndex =
CInt(p.Second) End If End Sub
```

控件与页面适配器

```
<!-- New browser node links into parent --> <browser  
  id="My_PIE_PPC" parentID="PIEnoDeviceID">  
  <identification> <capability name="browser"  
    match="Pocket IE" /> <capability name="majorversion"  
    match="4" /> </identification> <!-- Define new text-writer  
  for this browser and define control-adapter mapping -->  
  <controlAdapters  
    markupTextWriterType="MyNamespace.MyCustomText  
    Writer" > <adapter  
      controlType="MyNamespace.MyControl"  
      adapterType="MyNamespace.MyControlAdapter" />  
  </controlAdapters> </browser>
```

数据源和数据绑定控件

- ASP.NET 2.0提供了方便的扩张方法, 可以快速的扩展asp.net的控件类, 实现自己的数据源控件。
- ```
public interface IDataSource { event
 EventHandler DataSourceChanged { add;
 remove; } DataSourceView GetView(string
 viewName); ICollection GetViewNames(); }
public abstract class DataSourceControl :
 Control, IDataSource, IListSource { // Additional
 implementation }
```

# Web 资源

在asp.net 1.0时代，所有的资源都必须放到asp.net 运行的虚拟目录下面

这样既占用空间，有影响扩展性

' Mark the assembly with the resource

```
<assembly: WebResource("CustomControlScript.js",
"text/javascript")>
```

```
Public Class CustomControl Inherits WebControl ' Additional
implementation Protected Overrides Sub OnPreRender(ByVal e As
System.EventArgs)
```

```
Me.Page.ClientScript.RegisterClientScriptResource(GetType(Custom
Control), "CustomControlScript.js"); Me.Attributes.Add("onmouseover",
"MouseOverScript()") MyBase.OnPreRender(e) End Sub End Class
```



# 用户端脚本控制

```
Public Class MyButton Inherits Button Protected
 Overrides Sub OnPreRender(ByVal e As
 System.EventArgs) Dim sScript As String =
 "function DoAlert(){alert('Hello World');}"
 Me.Page.ClientScript.RegisterClientScriptBlock(
 GetType(MyButton), _ "ScriptFunction", sScript,
 True) OnClientClick = "javascript:DoAlert();"
 MyBase.OnPreRender(e) End Sub ' Additional
 implementation End Class
```



# 回调函数

Public Class CustomControl

Inherits CompositeControl

Implements ICallbackEventHandler Private Sub PrepareCallbackEvent(ByVal eventArgument As String) Implements System.Web.UI.ICallbackEventHandler.PrepareCallbackEvent

**PrepareCallbackEvent(eventArgument)** End Sub Private Function

RenderCallbackResult() As String \_ Implements

System.Web.UI.ICallbackEventHandler.RenderCallbackResult Return RenderCallbackResult()

End Function Private \_callbackEventArg As String; Protected Overridable Sub

PrepareCallback(ByVal eventArgument As String) \_callbackEventArg = eventArgument End Sub

Protected Overridable Function RenderCallbackResult() As String Dim result As String If

(\_callbackEventArg = "theArg") Then result = "Some data" Else result = "Only theArg allowed!"

End If Return result End Function Private sButtonCallBack As String = \_ "function

ButtonCallBack(result, context){ alert(result);}" Private sButtonCallBackError As String = \_

"function ButtonErrorCallBack(result, context){ alert(result);}" Protected Overrides Sub

OnInit(ByVal e As System.EventArgs) MyBase.OnInit(e)

Page.ClientScript.RegisterClientScriptBlock(GetType(MyControl), \_ "ButtonCallBack",

sButtonCallBack, True); Page.ClientScript.RegisterClientScriptBlock(GetType(MyControl), \_

"ButtonCallBackError", sButtonCallBackError, True); End Sub ' Additional implementation

Protected Overrides Sub OnPreRender(ByVal e As EventArgs) Me.Attributes("OnClick") = \_

Page.ClientScript.GetCallbackEventReference(Me, \_ "theArg", "ButtonCallBack", "null",

"ButtonErrorCallBack", True) MyBase.OnPreRender(e) End Sub End Class

# 表达式建构器


- ExpressionBuilders 在asp.net 2.0中用户可以自由扩展。
- 链接字符串
- 应用设置
- 资源文件访问

# 总结

- 页面框架生命周期扩展
- 服务器端控件运行时扩展
- 页面状态扩展
- 服务器端控件设计扩展
- 应用服务供应者扩展
- 表达式扩展



# 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**