

ASP.NET 2.0 绑定高级技巧

苏鹏

网通技术支持与项目部开发经理

MSDN特约讲师

内容概述

- 简单数据绑定
- 数据源控件
- 数据控件
 - GridView 与 DetailsView 控件
- 缓存
 - 配置缓存

简单数据绑定

- 数据绑定变的更简单，并支持XML数据源绑定

```
<!-- ASP.NET 1.x data binding expression -->  
<%# DataBinder.Eval (Container.DataItem, "Price") %>
```

```
<!-- Equivalent ASP.NET 2.0 data binding expression -->  
<%# Eval ("Price") %>
```

```
<!-- XML data binding -->  
<%# XPath ("Price") %>
```

数据源控件

● 无代码的数据绑定

| 控件名 | 控件描述 |
|--------------------------|--------------------------|
| SqlDataSource | 一切支持 SQL 语句的数据源控件 |
| AccessDataSource | Access 数据源控件 |
| XmlDataSource | XML 数据源控件 |
| ObjectDataSource | 自行编写组件的数据源控件 |
| SiteMapDataSource | 页面导航控件的数据源控件 |

SqlDataSource

- 支持数据绑定到 **SQL** 数据库
 - 任何支持SQL语句的数据库都在支持之列
- 两种数据绑定方式
 - SelectCommand 定义查询操作
 - InsertCommand, UpdateCommand, and DeleteCommand 定义更新操作
- 可以对查询结果使用缓存
- 可以对参数添加参数

使用 *SqlDataSource*

```
<asp:SqlDataSource ID="Titles" RunAt="server"  
    ConnectionString="server=localhost; database=pubs; integrated security=true"  
    SelectCommand="select title_id, title, price from titles" />  
<asp:DataGrid DataSourceID="Titles" RunAt="server" />
```


SqlDataSource 关键属性

| 名称 | 描述 |
|------------------|---|
| ConnectionString | 连接数据库的连接字符串 |
| SelectCommand | 用于执行查询的命令 |
| InsertCommand | 用于执行插入的命令 |
| UpdateCommand | 用于执行更新的命令 |
| DeleteCommand | 用于执行删除的命令 |
| DataSourceMode | 指定数据源类型是 DataSet 或 DataReader (默认值 = DataSet) |
| ProviderName | 指定供应商 (默认值 = SQL Server .NET provider) |

SqlDataSource 与缓存

- SqlDataSource 支持通过以下属性进行数据缓存:

| 属性名 | 描述 |
|-----------------------|---|
| EnableCaching | 指定是否打开缓存 (默认值 = false) |
| CacheDuration | 指定结果被缓存多少秒 |
| CacheExpirationPolicy | 指定缓存间隔是 sliding 还是 absolute |
| CacheKeyDependency | 使缓存依赖于一个特定键值 |
| SqlCacheDependency | 使缓存依赖于一个特定数据库实体 |

缓存查询结果

```
<asp:SqlDataSource ID="Countries" RunAt="server"
    ConnectionString="server=localhost; database=northwind; ..."
    SelectCommand="select distinct country from customers order by country"
    EnableCaching="true" CacheDuration="60" />
<asp:DropDownList ID="MyDropDownList" DataSourceID="Countries"
    DataTextField="country" AutoPostBack="true" RunAt="server" />
```

参数化命令

- 参数属性允许数据库命令带参数执行
 - 例如:通过下来框获得查询语句的参数
 - 例如:从GridView控件中获得删除命令的参数
- **XxxParameter** 类型指定参数来源

XxxParameters 属性

| 名称 | 描述 |
|------------------|------------|
| SelectParameters | 为查询命令指定参数 |
| InsertParameters | 为插入命令指定参数 |
| UpdateParameters | 为更新命令指定参数 |
| DeleteParameters | 为删除命令指定参数 |
| FilterParameters | 为过滤器命令指定参数 |

XxxParameter 类型

| 名称 | 描述 |
|----------------------|-------------------|
| Parameter | 为数据源绑定一个参数 |
| ControlParameter | 指定一个源自于控件的参数 |
| CookieParameter | 指定一个源自于cookie的参数 |
| FormParameter | 指定一个源自于表单的参数 |
| ProfileParameter | 指定一个源自于profile的参数 |
| QueryStringParameter | 制定于一个来源于查询字符串的参数 |
| SessionParameter | 指定一个源自于session的参数 |

使用 *ControlParameter*

```
<asp: Sql DataSource ID="Countries" RunAt="server"
  Connecti onStri ng="server=l ocal host; database=northwi nd; . . . "
  Sel ectCommand="sel ect di stinct country from customers order by country" />
<asp: Sql DataSource ID="Customers" RunAt="server"
  Connecti onStri ng="server=l ocal host; database=northwi nd; . . . "
  Sel ectCommand="sel ect * from customers where country=@Country">
  <Sel ectParameters>
    <asp: Control Parameter Name="Country" Control I D="MyDropDownLi st"
      PropertyName="Sel ectedVal ue" />
  </Sel ectParameters>
</asp: Sql DataSource>
<asp: DropDownLi st ID="MyDropDownLi st" DataSourceI D="Countries"
  DataTextFiel d="country" AutoPostBack="true" RunAt="server" />
<asp: DataGri d DataSourceI D="Customers" RunAt="server" />
```

调用存储过程

```
<asp:SqlDataSource ID="Countries" RunAt="server"
  ConnectionString="server=localhost; database=northwind; ... "
  SelectCommand="proc_GetCountries" />
<asp:SqlDataSource ID="Customers" RunAt="server"
  ConnectionString="server=localhost; database=northwind; ... "
  SelectCommand="proc_GetCustomers">
  <SelectParameters>
    <asp:ControlParameter Name="Country" ControlID="MyDropDownList"
      PropertyName="SelectedValue" />
  </SelectParameters>
</asp:SqlDataSource>
<asp:DropDownList ID="MyDropDownList" DataSourceID="Countries"
  DataTextField="country" AutoPostBack="true" RunAt="server" />
<asp:DataGrid DataSourceID="Customers" RunAt="server" />
```

```
CREATE PROCEDURE proc_GetCustomers
@Country nvarchar (32) AS
  SELECT * FROM Customers
  WHERE Country = @Country
GO
```

```
CREATE PROCEDURE proc_GetCountries AS
  SELECT DISTINCT Country
  FROM Customers
  ORDER BY Country
GO
```


SqlDataSource

demo

XmlDataSource

- 使用 **XML** 作为数据源
- 支持缓存与 **XSL** 转换
- 只支持查询绑定，不支持更新

```
<asp:XmlDataSource ID="Rates" DataFile="Rates.xml" RunAt="server" />  
<asp:TreeView ID="MyTreeView" DataSourceID="Rates" RunAt="server" />
```

XmlDataSource 的关键属性

| 名称 | 描述 |
|-----------------------|---|
| DataFile | XML 数据文件的路径 |
| TransformFile | 含有 XSL 风格定义的数据文件路径 |
| EnableCaching | 指定是否开启 cache (默认值 = false) |
| CacheDuration | 以秒为单位的时间间隔 |
| CacheExpirationPolicy | 指定时间间隔是 sliding 还是 absolute |
| CacheKeyDependency | 创建缓存依存于某个键 |
| XPath | XPath 表达式用来确认数据 |

ObjectDataSource

- 从数据组件绑定数据
 - 提供中间件的数据绑定
 - 使数据访问和UI脱离
- 两种数据绑定方式
 - SelectMethod, InsertMethod, UpdateMethod, and DeleteMethod
- 可选择是否使用缓存
- 可选择是否使用参数

ODS 关键属性

| 名称 | 描述 |
|----------------------|----------------------|
| TypeName | 数据组件的类型名称 |
| SelectMethod | 调用查询方法 |
| InsertMethod | 调用插入方法 |
| UpdateMethod | 调用更新方法 |
| DeleteMethod | 调用删除方法 |
| EnableCaching | 指定是否缓存 (默认值 = false) |

ODS 关键属性

| 名称 | 描述 |
|--------------------|-------------|
| CacheDuration | 缓存间隔时间以秒为单位 |
| SqlCacheDependency | 基于某个数据实体的缓存 |
| SelectParameters | 指定查询方法参数 |
| InsertParameters | 指定插入方法参数 |
| UpdateParameters | 指定更新方法参数 |
| DeleteParameters | 指定删除方法参数 |

创建和清除

- **ObjectDataSource.SelectMethod** 可以使用静态方法也可以使用一个类的新实例
- 如果使用实例方法:
 - ODS 在每次调用的时候创建一个新实例
 - 类必须具有公共的构造函数
- 使用 **ObjectCreated** 和 **ObjectDisposing** 元素可以 初始化和撤销函数

ObjectDataSource

demo

GridView 控件

- 增强的 **DataGrid** 控件
 - 使用 HTML 表格来显示数据
- 内置支持排序, 分页, 查询, 更新, 以及 删除操作
- 支持复杂的数据单元格类型, 包括 **CheckBoxFields**
 - `<Columns>` 元素中声明
- 高可定制的用户界面

GridView 示例

```
<asp:SqlDataSource ID="Employees" RunAt="server"
    ConnectionString="server=localhost; database=northwind; ..."
    SelectCommand="select lastname, firstname, title from employees" />
<asp:GridView DataSourceID="Employees" Width="100%" RunAt="server" />
```

输出

http://localhost/samples/northwindgridview1.aspx - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Media

Address http://localhost/samples/northwindgridview1.aspx Go Links

| lastname | firstname | title |
|-----------|-----------|--------------------------|
| Davolio | Nancy | Sales Representative |
| Fuller | Andrew | Vice President, Sales |
| Leverling | Janet | Sales Representative |
| Peacock | Margaret | Sales Representative |
| Buchanan | Steven | Sales Manager |
| Suyama | Michael | Sales Representative |
| King | Robert | Sales Representative |
| Callahan | Laura | Inside Sales Coordinator |
| Dodsworth | Anne | Sales Representative |

Done Local intranet

GridView 列类型

| 名称 | 描述 |
|----------------|---|
| BoundField | 显示数据库中取出的文本 |
| ButtonField | 显示按钮 |
| CheckBoxField | 使用 check boxes 显示一个 boolean 型变量 |
| CommandField | 显示一个查询或者编辑按钮 |
| HyperLinkField | 显示一个超链接 |
| ImageField | 显示一个图片 |
| TemplateField | 显示一个自定义的 HTML 模板 |

指定区域类型

```
<asp:SqlDataSource ID="Employees" RunAt="server"
    ConnectionString="server=localhost; database=northwind; ..."
    SelectCommand="select photo, lastname, firstname, title from employees" />
<asp:GridView DataSourceID="Employees" Width="100%" RunAt="server"
    AutoGenerateColumns="false" >
    <Columns>
        <asp:TemplateField HeaderText="Name">
            <ItemTemplate>
                <%# Eval ("firstname") + " " + Eval ("lastname") %>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField HeaderText="Title" DataField="title" />
    </Columns>
</asp:GridView>
```

输出

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address <http://localhost:1056/DataAccess/testtt.aspx> Go Links

| Name | Title |
|------------------|--------------------------|
| Nancy Davolio | Sales Representative |
| Andrew Fuller | Vice President, Sales |
| Janet Leverling | Sales Representative |
| Margaret Peacock | Sales Representative |
| Steven Buchanan | Sales Manager |
| Michael Suyama | Sales Representative |
| Robert King | Sales Representative |
| Laura Callahan | Inside Sales Coordinator |
| Anne Dodsworth | Sales Representative |

Done Local intranet

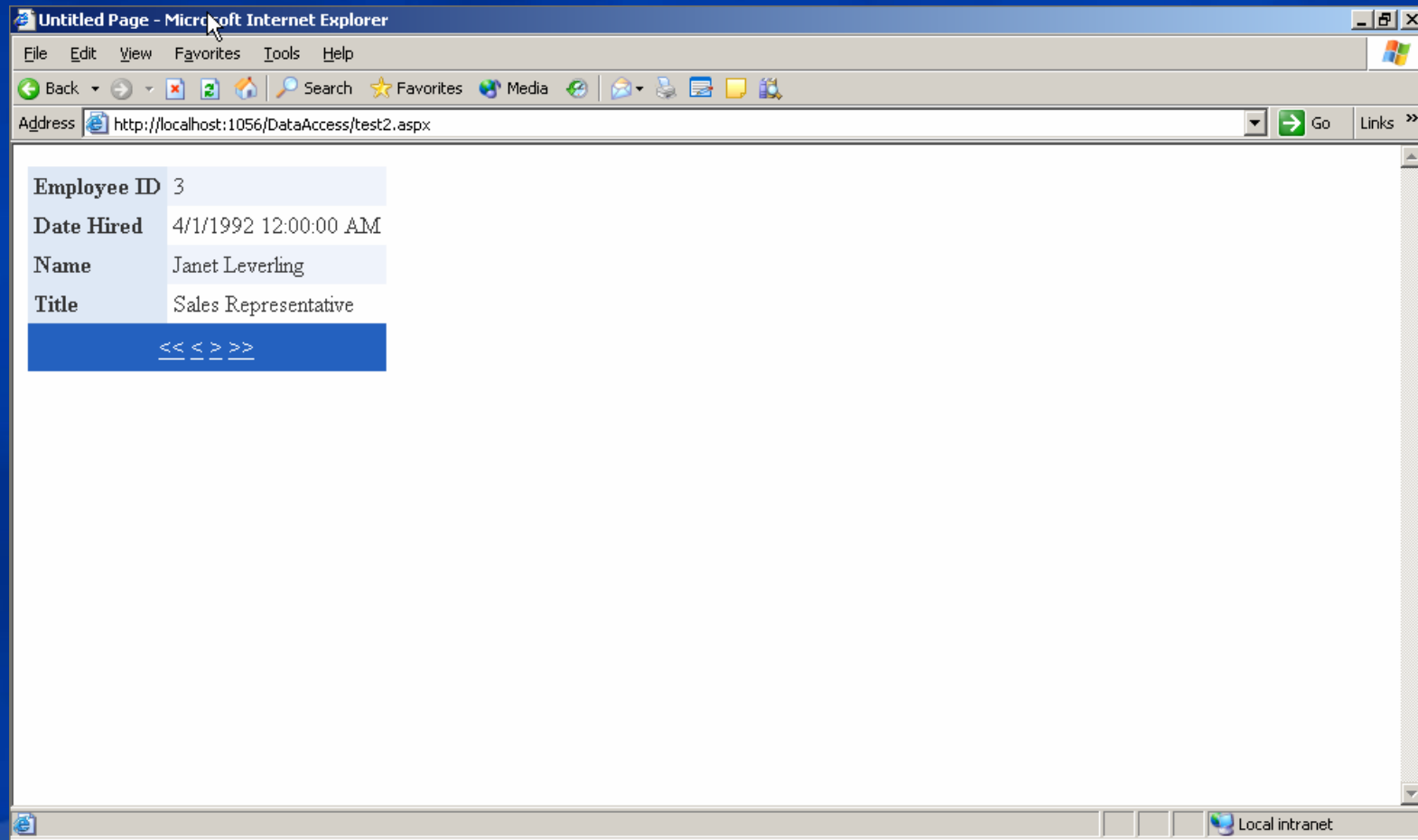
DetailsView 控件

- 显示个别的记录
 - 配合 GridView 显示master-detail 视图
 - 显示一个单独的记录
- 内建了分页，查询，添加，删除操作
- 使用与 **GridView**完全相同的单元格类型
 - 在 <Fields> 元素中声明
- **Highly customizable UI**

DetailsView 示例

```
<asp:SqlDataSource ID="Employees" RunAt="server"
    ConnectionString="server=localhost;database=northwind;..."
    SelectCommand="select employeeid, photo, ... from employees" />
<asp:DetailsView DataSourceID="Employees" RunAt="server"
    AllowPaging="true" AutoGenerateRows="false"
    PagerSettings-Mode="NextPreviousFirstLast">
    <Fields>
        <asp:BoundField HeaderText="Employee ID" DataField="employeeid" />
        <asp:BoundField HeaderText="Date Hired" DataField="hiredate" />
        <asp:TemplateField HeaderText="Name">
            <ItemTemplate>
                <%# Eval ("firstname") + " " + Eval ("lastname") %>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField HeaderText="Title" DataField="title" />
    </Fields>
</asp:DetailsView>
```

Output



Master-Detail Views

demo

插入, 更新, 和删除

- 数据显示控件支持从用户界面进行编辑
 - AutoGenerateXxxButton 属性
 - Insert/EditRowStyle 属性
- 数据源控件支持编辑逻辑
 - Insert/Update/DeleteCommand 属性
 - Insert/Update/DeleteParameters 属性
 - Inserting/ed, Updating/ed, Deleting/ed 事件
- **Visual Studio** 支持两者的结合

使用 *GridViews* 进行数据编辑

Update 命令

Update 参数

```
<asp:SqlDataSource ID="Employees" RunAt="server"
  ConnectionString="server=localhost;database=northwind;..."
  SelectCommand="select employeeid, lastname, firstname from employees"
  UpdateCommand="update employees set lastname=@lastname, firstname=
    @firstname where employeeid=@original_employeeid">
  <UpdateParameters>
    <asp:Parameter Name="EmployeeID" Type="Int32" />
    <asp:Parameter Name="Lastname" Type="String" />
    <asp:Parameter Name="Firstname" Type="String" />
  </UpdateParameters>
</asp:SqlDataSource>
```

```
<asp:GridView DataSourceID="Employees" Width="100%" RunAt="server"
  DataKeyNames="EmployeeID" AutoGenerateEditButton="true" />
```

Primary key

编辑按钮

冲突判定

- 先入胜利
 - 如果数据在取出之后被改变，则修改失败
 - UpdateCommand 结构构成
 - 指定 ConflictDetection="CompareAllValues" 来实现
- 后入胜利
 - 无论数据是否被修改，该修改都会成功
 - UpdateCommand 结构构成
 - 指定 ConflictDetection="OverwriteChanges" 来实现

先入胜利法则更新

```
<asp:SqlDataSource ID="Employees" RunAt="server"
  ConnectionString="server=localhost; database=northwind; ..."
  SelectCommand="select employeeid, lastname, firstname from employees"
  UpdateCommand="update employees set lastname=@lastname, firstname=
    @firstname where employeeid=@original_employeeid and lastname=
    @original_lastname and firstname=@original_firstname"
  ConflictDetection="CompareAllValues">
  <UpdateParameters>
    <asp:Parameter Name="EmployeeID" Type="Int32" />
    <asp:Parameter Name="lastname" Type="String" />
    <asp:Parameter Name="firstname" Type="String" />
  </UpdateParameters>
</asp:SqlDataSource>

<asp:GridView DataSourceID="Employees" Width="100%" RunAt="server"
  DataKeyNames="EmployeeID" AutoGenerateEditButton="true" />
```

错误检测

- 数据更新后控件调用的事件
 - GridView.RowUpdated
 - DetailsView.ItemUpdated
 - SqlDataSource.Updated, etc.
- 处理“**status**”的事件
 - 无论数据库是否异常
 - 允许数据库异常被处理或者再次抛弃
 - 显示多少数据库行被修改

处理更新错误

```
<asp: Sql DataSource ID="Employees" RunAt="server" ...  
    UpdateCommand="..." OnUpdated="OnUpdateComplete">  
    ...  
</asp: Sql DataSource>  
    ...  
void OnUpdateComplete (Object source, Sql DataSourceStatusEventArgs e)  
{  
    if (e.Exception != null) {  
        // Exception thrown. Set e.ExceptionHandled to true to prevent  
        // the Sql DataSource from throwing an exception, or leave it set  
        // to false to allow Sql DataSource to rethrow the exception  
    }  
    else if (e.AffectedRows == 0) {  
        // No exception was thrown, but no records were updated, either.  
        // Might want to let the user know that the update failed  
    }  
}
```

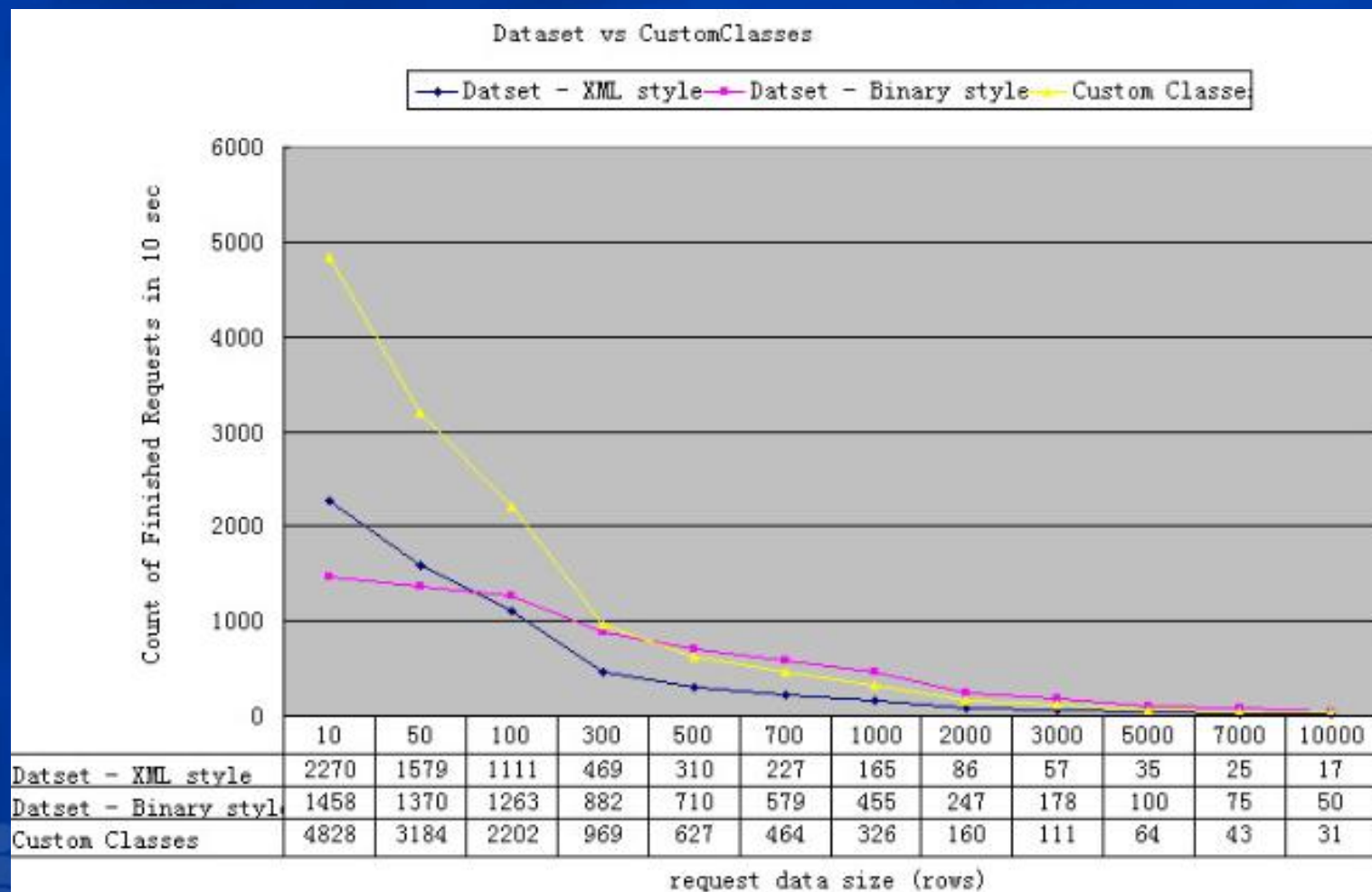

使用 *GridViews* 和 *DetailsViews* 编辑数据

demo

DataSet的性能

- DataSet能支持真正意义上的Binary序列化了, 而我们需要做的工作呢, 只用写一句话:
- `ds.RemotingFormat = SerializationFormat.Binary`

- 对一个包含3个column, 大约10000条数据的Dataset, 采用binary序列化生成的文件大小是采用xml序列化的28% (457kb : 1.6MB), 花费的时间只是xml的35%. 看上去真好极了.



SQL 缓存

- 新的缓存依赖属性类型
 - 包含于 SqlCacheDependency 类
 - 通过 <sqlCacheDependency> 来配置
- 连接数据库实体
 - ASP.NET 应用缓存
 - ASP.NET 输出缓存
- 兼容 SQL Server 7, 2000, 2005

数据库准备

- 使用 **Aspnet_regsql.exe** 或者 **SqlCache-DependencyAdmin** 做好数据库准备*

```
aspnet_regsql -S local host -E -d Northwi nd -ed
```

服务器名称
连接方式，这里使用致信连接
数据库名称
启用数据库

* SQL Server 2005 不需要配置

数据表准备

- 使用 **Aspnet_regsql.exe** or **SqlCacheDependencyAdmin** 准备数据表*

```
aspnet_regsql -S local host -E -d Northwind -t Products -et
```

服务器名称
可信任的连接
数据库名称
表名称
启用表监视

* SQL Server 2005 不需要配置

准备 Web.config

```
<configuration>
  <connectionStrings>
    <add name="Northwind"
      connectionString="server=localhost; database=northwind; ..." />
  </connectionStrings>
  <system.web>
    <caching>
      <sqlCacheDependency enabled="true" pollInterval="5000">
        <databases>
          <add name="Northwind" connectionStringName="Northwind" />
        </databases>
      </sqlCacheDependency>
    </caching>
  </system.web>
</configuration>
```

使用系统缓存

```
Cache.Insert ("Products", products,  
    new Sql CacheDependency ("Northwind", "Products"));
```

数据库名称

表格名称

使用输出缓存

```
<%@ OutputCache Duration="60" VaryByParam="None"  
    Sql Dependency="Northwind: Products" %>
```

数据库名称

表格名称

对 *SqlDataSource* 使用缓存

```
<asp:SqlDataSource ID="Countries" RunAt="server"
  ConnectionString="server=localhost; database=northwind; ..."
  SelectCommand="select distinct country from customers order by country"
  EnableCaching="true" CacheDuration="60000"
  SqlCacheDependency="Northwind: Customers" />
<asp:DropDownList ID="MyDropDownList" DataSourceID="Countries"
  DataTextField="country" AutoPostBack="true" RunAt="server" />
```

Database name

Table name

缓存配置

- **<cache>**
 - Enable/disable 应用缓存
 - Enable/disable 元素是否到期
- **<outputCache>, <outputCacheSettings>**
 - Enable/disable 显示缓存
 - Enable/disable 基于磁盘的缓存
 - 为每个应用程序指定最大的缓存空间
- **<sqlCacheDependency>**

回顾

- 简单数据绑定
- 数据源控件
- 数据控件
 - GridView 与 DetailsView 控件
- 缓存
 - 配置缓存
 - Cache configuration

更多信息

- MSDN China

<http://www.microsoft.com/msdn/china>

- MSDN中文网络广播


<http://www.msdnwebcast.com.cn>

- MSDN Flash

<http://www.microsoft.com/china/newsletter/case/MSDN.asp>



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)

Microsoft®

Microsoft®

Your potential. Our passion.™

msdn
