

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# ADO.NET数据操作

讲师：邵志东

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 课前准备

- Dot Net Framework
- VS.NET 2002/2003
- C#/VB.NET
- Level 200

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- ADO.NET概述
- 数据绑定
- 数据操作
- 存储过程的使用

# ADO.NET概述

## 1、数据中心应用程序设计:

- 网络的发展和多媒体的应用导致数据资料的激增
- 数据量的增加导致数据存储的重要性日益突出
- 以数据为中心的应用程序设计正成为一种趋势

# ADO.NET概述

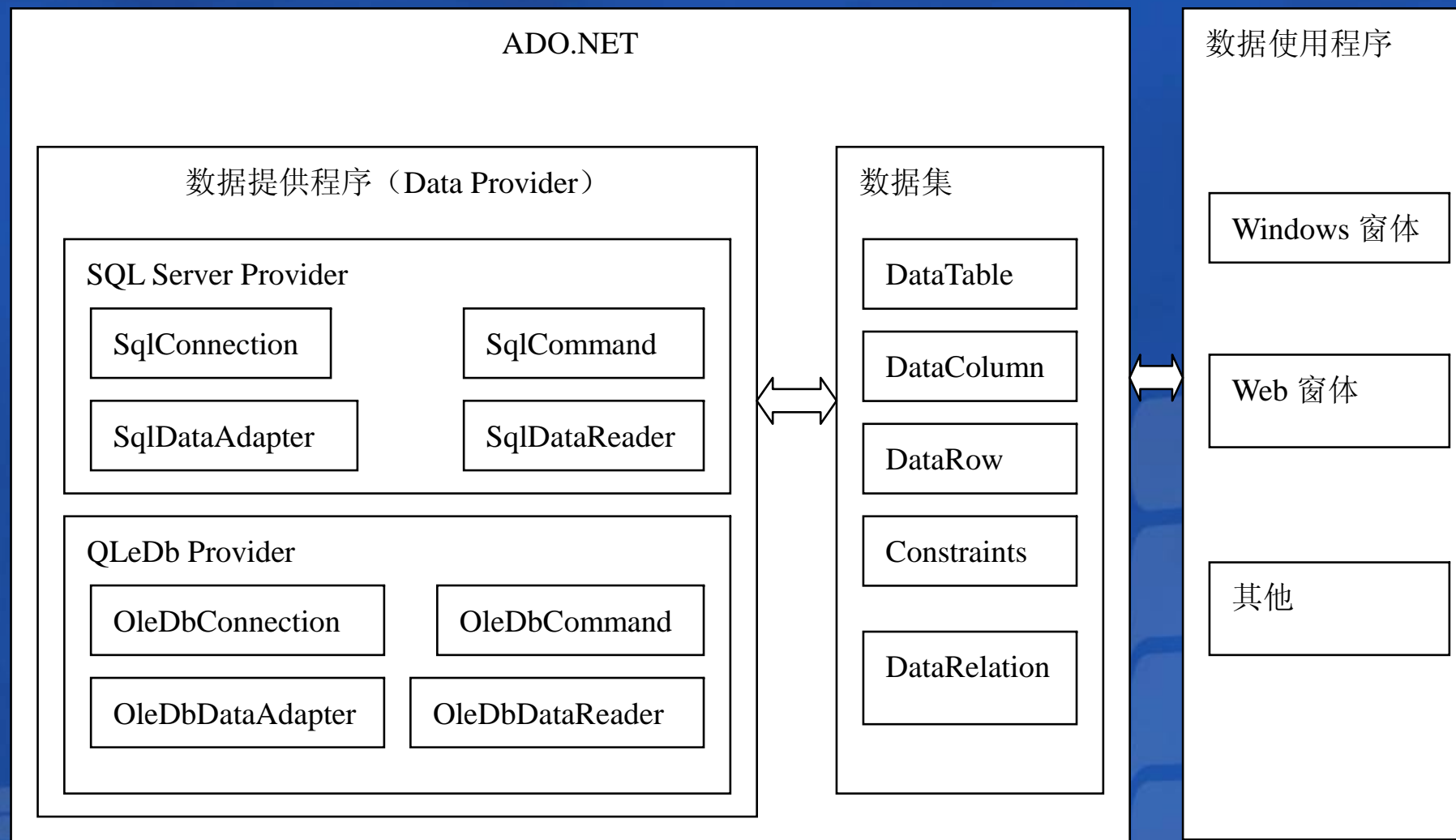
您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

## 2、ADO.NET架构

### 2.1 ADO.NET对象模型

- 数据提供程序 (Data Provider)
- 数据集 (DataSet)





## 2.2 对象概述

- **Connection**类: 在代码和容纳数据的数据存储之间提供了基本的连接
- **Command**类: 用于描述SQL语句或者通过其**Connection**类执行的存储过程
- **DataReader**: 从数据源中获取只读的数据流
- **DataAdapter**: 功能最强大、更复杂的对象, 可以读取、改变数据源

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- ADO.NET概述
- 数据绑定
- 数据操作
- 存储过程的使用



# 数据绑定

## 1、什么是数据绑定？

- 把数据绑定到窗体的过程
- 指 Microsoft 窗体使用 Microsoft ADO.NET 进行数据绑定的过程
- 可以编码实现绑定，也可以使用VS.NET实现
- 可以绑定的对象：
  - 数据库
  - 其他控件

## 2、数据绑定的类型

.NET窗体可以进行两种类型的数据绑定：简单数据绑定和复杂数据绑定。两种方法各有优点。

- 简单数据绑定：允许将控件绑定到单个数据元素。最常用的简单数据绑定是将单一数据元素（如表的列值）绑定到窗体的控件。可将此类数据绑定用于仅显示一个值的控件。使用简单数据绑定包括将数据绑定到文本框和标签。
- 复杂数据绑定：允许将多个数据元素绑定到一个控件。以列为例，复杂数据绑定可以绑定基础记录源中的多行或多列。支持复杂数据绑定的控件包括数据网格控件、组合框和列表框。

## 3、ASP.NET中的数据绑定

- 所有的DataBind都应该用DataBind() 函数来建立数据绑定。
- DataBind()是整个页面PAGE和所有控件的一个方法，也就是说，它可以被所有的控件使用。
- 建立数据绑定的时候，DataBind可以作为控件的一个子项，比如DataList1.DataBind(), 再比如Page.DataBind(), 会绑定整个页面。
- DataBind 常在页面载入时就被绑定。
- DataSource: 指定数据源。

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO1

## 数据绑定



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- ADO.NET概述
- 数据绑定
- 数据操作
- 存储过程的使用



# 1、建立Command对象

## 1)、创建方式:

- `SqlCommand command = new SqlCommand();`
- `SqlCommand command = new SqlCommand("SQL语句");`
- `SqlCommand command = new SqlCommand("SQL语句", Connection对象);`
- `SqlCommand("SQL语句", Connection对象, 指定事务);`

## 2)、Command对象的属性

属性	描述
CommandText	被执行的SQL语句或者存储过程
CommandTimeout	等待来自数据源响应的的时间
CommandType	指出CommandText数据如何解释
Connection	用于执行数据命令的Connection对象
Parameters	参数集合
Transaction	执行命令时所在的事务
UpdateRowSource	当DataAdapter的Update方法使用时, 该属性决定如何将结果对应到数据行中

# CommandType的值

- **StoredProcedure**: 存储过程名称
- **TableDirect**: 表名
- **Text**: SQL文本命令

## 3)、参数集合的使用

- 使用步骤
  - 在查询或者存储过程中指定参数
  - 在参数集合中使用参数
  - 设置参数值
- 占位符号
  - OleDbCommand:?
    - Select \* From Customers Where CustomID = ?
  - SqlCommand:@
    - Select \* From Customers Where CustomID = @custID

# 参数集的方法

- Add(): 添加新参数
- Clear(): 清除所有参数
- Insert(): 插入参数
- Remove(): 删除具有指定值的参数
- RemoveAt(): 删除指定的参数



# 在运行时添加和配置参数

- 新添加一个SqlCommand对象, 起名为cmdOrder
- 添加参数

```
cmdOrder.Parameters.Add("@empID",  
    System.Data.SqlDbType.Int, 4, "EmployeeID")  
mdOrder.Parameters.Add("@custID",  
    System.Data.SqlDbType.NVarChar, 5,  
    "CustomerID");
```

- 对CommandText属性赋值

## 2、Command对象的使用

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

方法	描述
Cancel	取消数据命令的执行
CreateParameter	创建一个新的参数
ExecuteNonQuery	执行命令并返回受影响的行数
ExecuteReader	执行命令并返回生成的DataReader
ExecuteScalar	执行查询并返回结果集中的第一行的第一列
ExecuteXmlReader	执行命令并返回生成的XMLReader
Prepare	在数据源上创建一个准备好的命令版本
ResetCommandTimeout	将CommandTimeout属性重置为默认值

# 返回DataReader

- 当数据命令返回结果集时，用DataReader来检索数据
- DataReader对象返回一个来自数据命令的只读的、只能向前的数据流
- 内存中每次仅有一个数据行，因此开销很少

# 返回一个值

- **ExecuteScalar**方法
- 它只返回结果集中第一行的第一列。使用 **ExecuteScalar** 方法从数据库中检索单个值（例如id号）。
- 与使用 **ExecuteReader** 方法，返回的数据执行生成单个值所需的操作相比，此操作需要的代码较少



# 返回受影响函数

- **ExecuteNonQuery**
- 对数据的更新不需要返回结果集
- 由于不返回结果集可省掉网络数据传输。它仅仅返回受影响的行数。如果只需更新数据用**ExecuteNonQuery**性能的开销比较小。



# 返回多行

- 通过DataSet实现
- 在本系列课程《数据集的使用方法和技巧》作详细介绍！

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO2

## 数据的查询、添加、删除、 修改

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- ADO.NET概述
- 数据绑定
- 数据操作
- 存储过程的使用

# 存储过程的调用

- 存储过程是SQLServer数据库的一个重要特色
- 存储过程执行效率比SQL文本命令要高的多
- 提高了程序的复用性
- 存储过程中可以使用变量和条件
- 可以在存储过程中使用参数

# 添加存储过程

```
CREATE PROCEDURE myProc AS  
select EmployeeID, FirstName, LastName  
from Employees  
where FirstName= 'Robert'  
GO
```



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO3

## 存储过程的使用

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 小结

- ADO.NET概述
- 数据绑定
- 数据操作
- 存储过程的使用

您的潜力，我们的动力


**Microsoft**  
微软(中国)有限公司

## 更多信息.....


- MSDN网站: <http://msdn.microsoft.com>
- 程序员大本营: [www.csdn.net](http://www.csdn.net)

# 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**