

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

**ADO.NET 2.0 开发实践系列课程**

# 用 **ADO.NET 2.0** 进行数据库应用程序开发（上）

徐长龙

# 本系列课程简介

- **ADO.NET 2.0 开发实践系列课程**

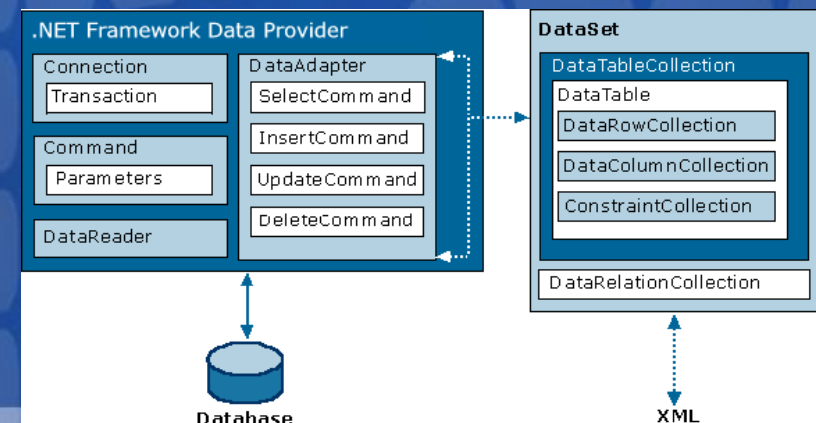
<b>2005.9.7</b> <b>(today)</b>	用 <b>ADO.NET 2.0</b> 进行数据库应用程序开发 (上)	本讲为 <b>ADO.NET</b> 入门课程，包括 <b>ADO.NET 2.0</b> 介绍及数据访问的新功能介绍。
<b>2005.9.21</b>	用 <b>ADO.NET 2.0</b> 进行数据库应用程序开发 (下)	重点介绍 <b>ADO.NET 2.0</b> 的新特性，以及何时如何使用这些新特性。

# 议程

- **ADO.NET 2.0 简介**
- 基于基类的提供者模型
- **Drag-Once** 数据绑定
- **Binding to Objects**
- 数据绑定的相关组件(控件)介绍
- 数据集设计器(**DataSet Designer**)
- 表适配器(**TableAdapters**)
- 保存连接字符串到程序配置文件

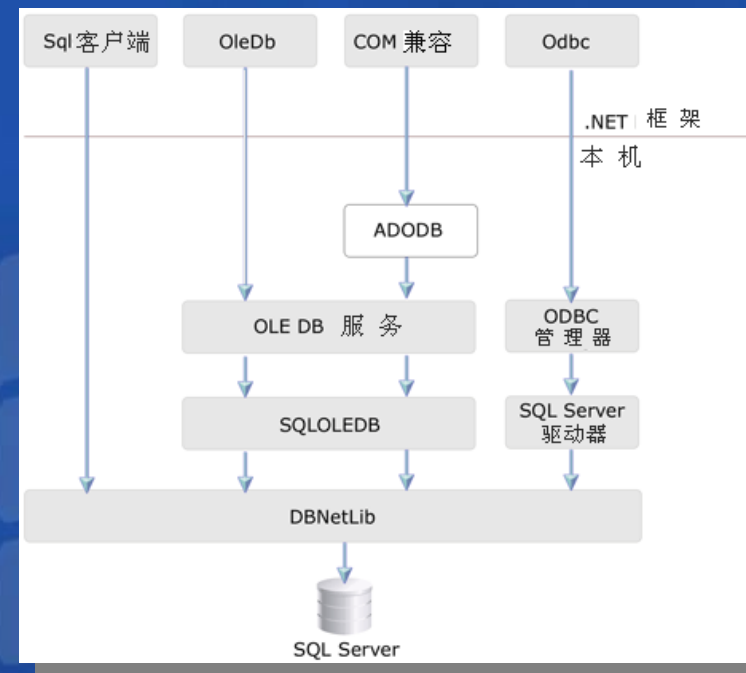
# ADO.NET 2.0 简介

- 在即将发布的**Visual Studio.NET 2005**中, 作为**DOT NET**平台重要组成部分的数据访问组件(**ADO.NET**)也将由 **1.1**版升级到**2.0**版
- **ADO.NET 2.0** 带来了一系列的新特性, 比如: 基于基类的数据提供者(**Data Provider**)模型, **DataSet**的性能改进, 更方便快捷的数据绑定组件, 以及大批量数据操作等
- **ADO.NET**主要由以下两个部分组成:
  - .NET Framework data providers
  - DataSet



# .NET Framework data providers

- ADO.NET 2.0 Data Provider:
  - SqlClient Data Provider
    - 命名空间: `System.Data.SqlClient`
  - OleDb Data Provider
    - 命名空间: `System.Data.OleDb`
  - Odbc Data Provider
    - 命名空间: `System.Data.Odbc`
  - OracleClient Data Provider
    - 命名空间: `System.Data.OracleClient`
- SQL Server 7.0或以上推荐使用SqlClient Data Provider





# 议程

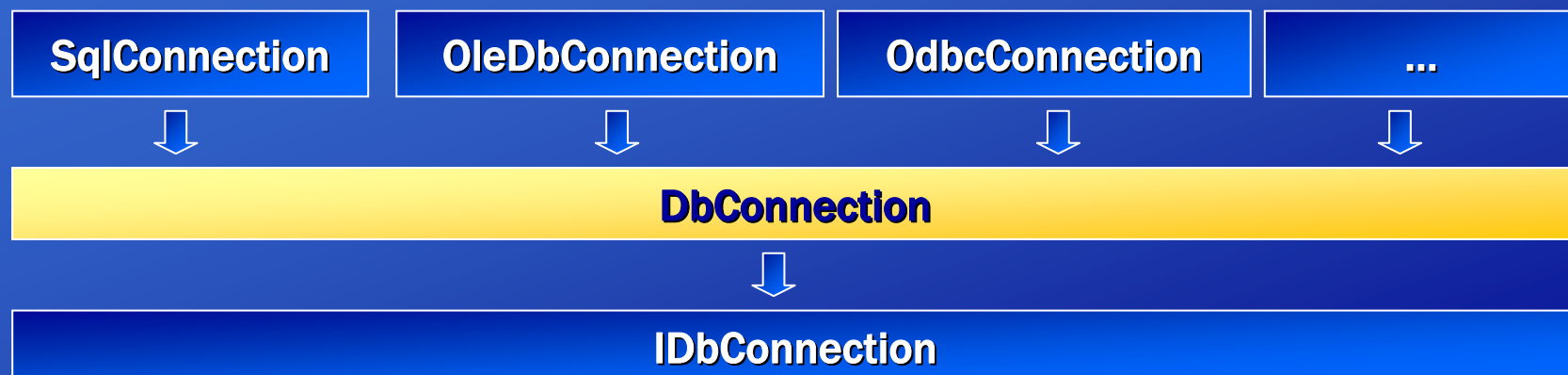
- ADO.NET 2.0 简介
- 基于基类的提供者模型
- Drag-Once 数据绑定
- 数据绑定的相关组件(控件)介绍
- 数据集设计器(DataSet Designer)
- 表适配器(TableAdapters)
- 保存连接字符串到程序配置文件

# 基于基类的提供者模型

## ADO.NET 1.0/1.1



## ADO.NET 2.0



# 基于基类的提供者模型

SqlClient类 (System.Data.SqlClient)	基类 (System.Data.Common)	接口 (System.Data)
SqlConnection	DbConnection	IDbConnection
SqlCommand	DbCommand	IDbCommand
SqlDataReader	DbDataReader	IDataReader/IDataRecord
SqlTransaction	DbTransaction	IDbTransaction
SqlParameter	DbParameter	IDbDataParameter
SqlParameterCollection	DbParameterCollection	IDataParameterCollection
SqlDataAdapter	DbDataAdapter*	IDbDataAdapter
SqlCommandBuilder	DbCommandBuilder	
SqlConnectionStringBuilder	DbConnectionStringBuilder	
SqlPermission	DBDataPermission*	

带\*的基类在**ADO.NET 1.0/1.1**中已经存在。



# 基于基类的提供者模型

**ADO.NET 1.0/1.1**创建一个通用连接对象

注: 创建何种连接对象从配置文件中获得

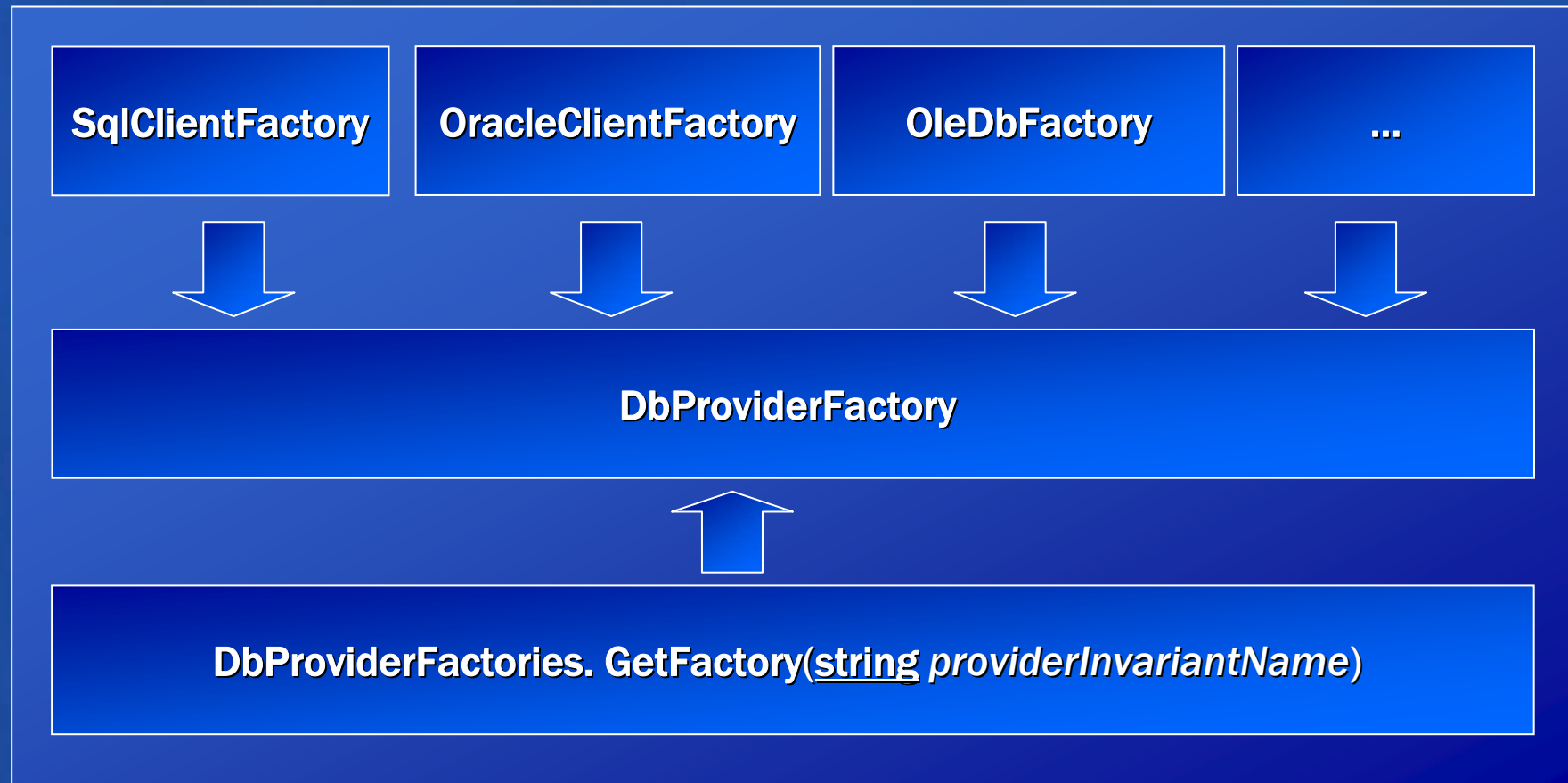
```
enum provider {sqlserver, oracle, oledb, odbc};  
// determine provider from configuration  
provider prov = GetProviderFromConfigFile();  
IDbConnection conn = null;  
switch (prov) {  
    case provider.sqlserver:  
        conn = new SqlConnection(); break;  
    case provider.oracle:  
        conn = new OracleConnection(); break;  
    // add new providers as the application supports them  
}
```

# 基于基类的提供者模型

## 用 ADO.NET 2.0 实现

```
// get ProviderInvariantString from configuration
enum provider {sqlserver, oracle, oledb, odbc};
public DbConnection GetConnectionBaseClass()
{
// determine provider from configuration
provider prov = GetProviderFromConfigFile();
DbConnection conn = null;
switch (prov) {
case provider.sqlserver:
conn = new SqlConnection(); break;
case provider.oracle:
conn = new OracleConnection(); break;
// add new providers as the application supports them
}
return conn;
}
```

# Provider Factories



# DbProviderFactory的方法

CreateCommand	创建命令行对象
CreateCommandBuilder	创建命令行生成器对象
CreateConnection	创建连接对象
CreateConnectionStringBuilder	创建连接字符生成器对象
CreateDataAdapter	创建数据适配器对象
CreateDataSourceEnumerator	创建数据源枚举对象
CreateParameter	创建参数对象
CreatePermission	创建权限对象

# DbProviderFactories的方法

- **GetFactoryClasses()**
  - 返回所有已注册的提供程序的列表DataTable对象
  - Machine.config

```
<system.data>  
  <DbProviderFactories>  
    <add name="Odbc Data Provider" invariant="System.Data.Odbc" description=".Net Framework Data  
    <add name="OleDb Data Provider" invariant="System.Data.OleDb" description=".Net Framework Dat  
    <add name="OracleClient Data Provider" invariant="System.Data.OracleClient" description=".Net  
    <add name="SqlClient Data Provider" invariant="System.Data.SqlClient" description=".Net Fram  
    <add name="SQL Server CE Data Provider" invariant="Microsoft.SqlServerCe.Client" support="3F  
  </DbProviderFactories>  
</system.data>
```

- **GetFactory(string providerInvariantName)**
  - 根据提供程序名返回DbProviderFactory对象
- **GetFactory(System.Data.DataRow providerRow)**
  - 根据提供程序数据行返回DbProviderFactory对象



# 使用基于基类的提供者模型的好处

- 可使用同一份代码支持不同的数据库，可作为大型应用程序中的数据访问层组件，从而将业务逻辑与数据库分离。
- 可针对基类编写数据访问的代码，而不用关注具体的数据库类型，有些接口不支持属性的存取
  - 例如：**DbDataReader.HasRows**，而**IDataReader**接口则没有**HasRows**属性

# 基于基类的提供者模型

## • DEMO预览（随后统一Demo）



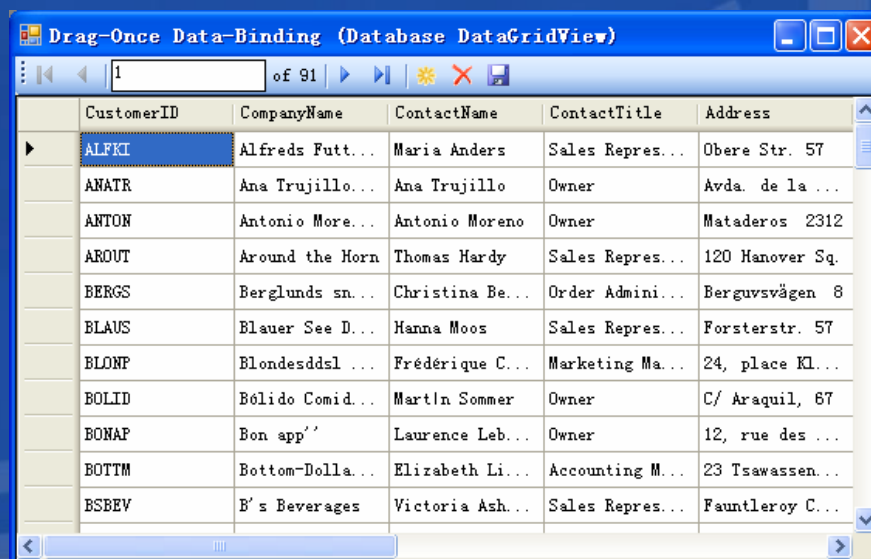
# 议程

- ADO.NET 2.0 简介
- 基于基类的提供者模型
- **Drag-Once 数据绑定**
- 数据绑定的相关组件(控件)介绍
- 数据集设计器(DataSet Designer)
- 表适配器(TableAdapters)
- 保存连接字符串到程序配置文件

# Drag-Once 数据绑定

Microsoft®  
微软(中国)有限公司

- 何为Drag-Once数据绑定？
  - Visual Studio 2005提供的一种简单快捷的数据绑定的编程方式：一次性拖放数据绑定。只需要将要绑定的数据库对象或自定义的对象拖至窗体或控件即可完成数据的逐条浏览、新增、编辑、删除等操作。



# Drag-Once 数据绑定

- **Drag-Once**可绑定至以下几种类型:
  - **Database**
    - 可绑定到表、视图和存储过程
    - 还可绑定到主子表等
  - **Web Service**
    - <http://api.google.com/GoogleSearch.wsdl>
  - **Object**
    - 可绑定到任何自定义的对象
    - 当前程序集中的对象
    - 外部程序集中的对象
      - 添加外部程序集的引用



# 制作

1. 新
2. 设

3. 将即

Form1

1 of 91

Customer ID: ALFKI

Company Name: Alfreds Futterkiste

Contact Name: Maria Anders

Contact Title: Sales Representative

Address:

City: Berlin

Region:

Postal Code: 12209

Country: Germany

Phone: 030-0074321

Fax: 030-0076545

运行效果

您的潜力. 我们的动力

Microsoft

CustomersTableAdapter

MSDN Webcasts

msdn

# Drag-Once 数据绑定

**Microsoft**  
微软(中国)有限公司

- DEMO预览 (随后统一Demo)

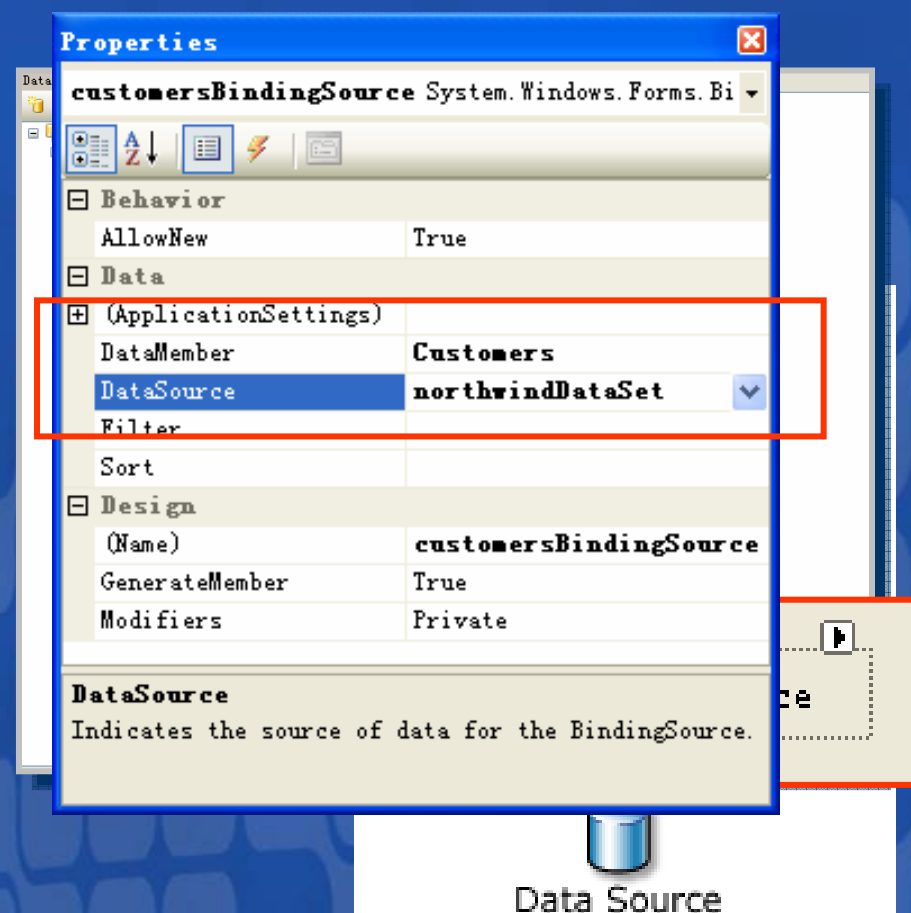


# 议程

- **ADO.NET 2.0 简介**
- 基于基类的提供者模型
- **Drag-Once 数据绑定**
- **数据绑定的相关组件(控件)介绍**
- 数据集设计器(DataSet Designer)
- 表适配器(TableAdapters)
- 保存连接字符串到程序配置文件

# 数据绑定的相关组件(控件)介绍

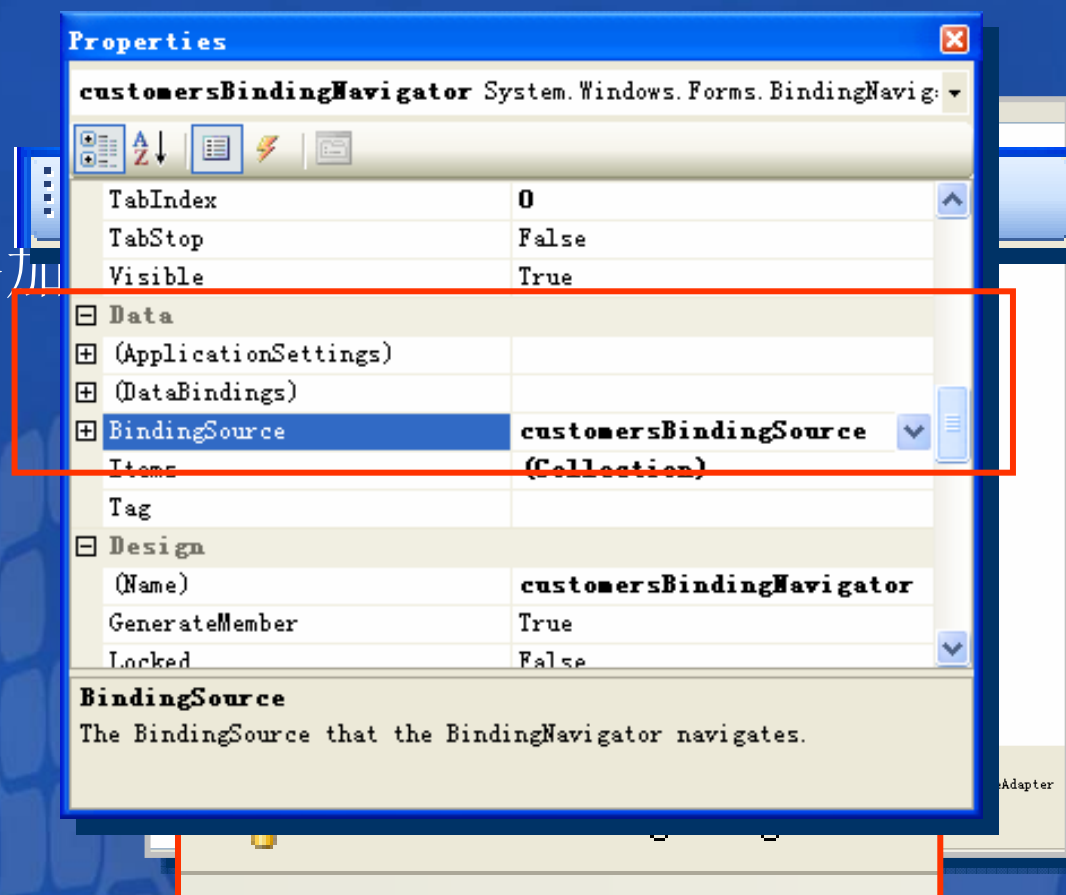
- **BindingSource Component**
  - 控件的数据源
    - 某个对象的单一属性
    - 集合对象 (ArrayList)
  - 复杂的数据源
    - 数据表
  - 扮演中间的角色
    - 当前记录位置
    - 新增、修改、删除操作
    - 排序和过滤



# 数据绑定的相关组件(控件)介绍

- **BindingNavigator Control**

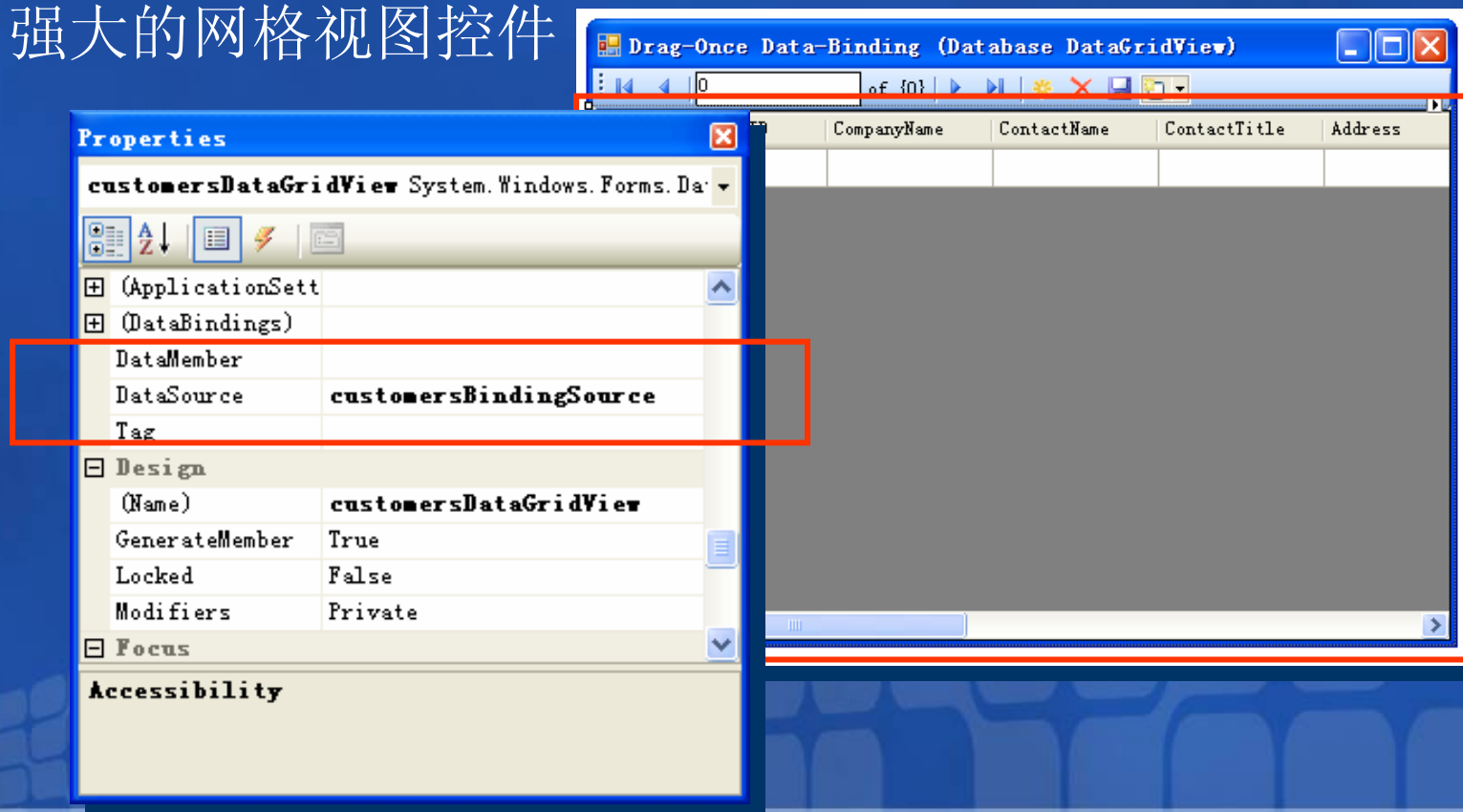
- 提供数据导航、添加、删除数据等操作
- 与 **BindingSource Component** 对应





# 数据绑定的相关组件(控件)介绍

- DataGridView Control
  - 强大的网格视图控件

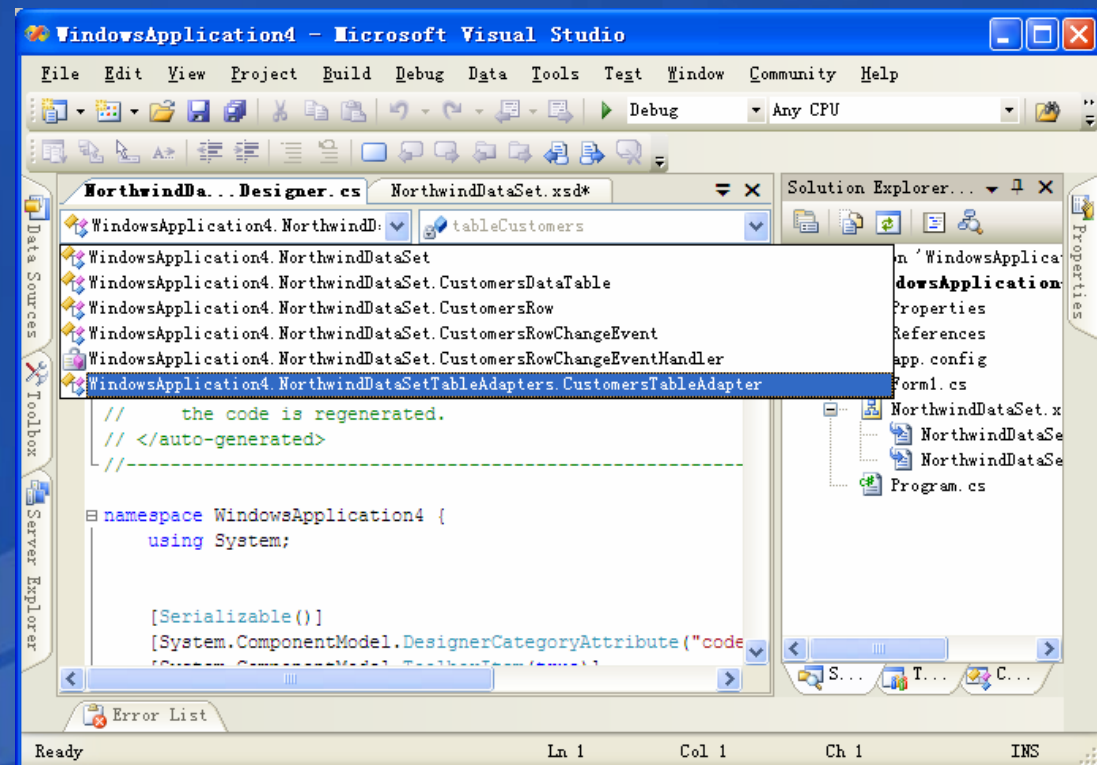


# 议程

- ADO.NET 2.0 简介
- 基于基类的提供者模型
- Drag-Once 数据绑定
- 数据绑定的相关组件(控件)介绍
- **数据集设计器(DataSet Designer)**
- 表适配器(TableAdapters)
- 保存连接字符串到程序配置文件

# 数据集设计器(DataSet Designer)

- 改进点
  - 完整的DataSet设计器, 建立/修改 DataTable, Data Relation 等等
  - 提供对新的 TableAdapter 的支持



# 议程

- **ADO.NET 2.0 简介**
- 基于基类的提供者模型
- **Drag-Once** 数据绑定
- 数据绑定的相关组件(控件)介绍
- 数据集设计器(DataSet Designer)
- **表适配器(TableAdapters)**
- 保存连接字符串到程序配置文件

# 表适配器(**TableAdapters**)

- **ADO.NET 2.0** 全新的表适配器对象, 用于填充 **DataSet**, 执行查询和存储过程
- 设计器生成的组件(**designer-generated components**), 对象浏览器中找不到 **TableAdapter** 的命名空间



# TableAdapters成员

- **TableAdapter.Fill**
  - 将数据填充至DataTable对象
- **TableAdapter.Update**
  - 将数据更新至数据源
- **TableAdapter.GetData**
  - 获取已填充数据的DataTable对象
- **TableAdapter.Insert**
  - 插入一笔数据
- **TableAdapter.Delete**
  - 删除一笔数据
- **TableAdapter.ClearBeforeFill**
  - 填充数据时是否先清空DataTable, 支持多次填充

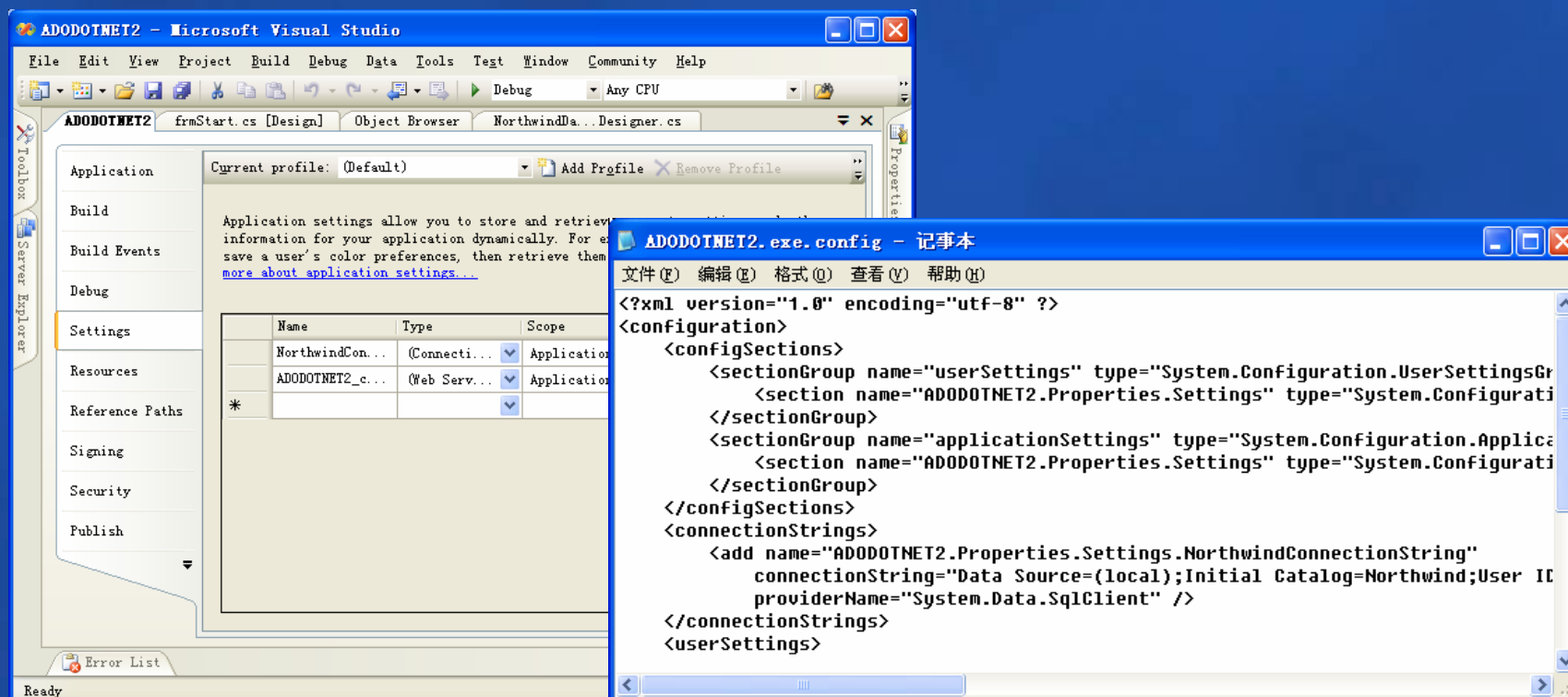
# 议程

- **ADO.NET 2.0 简介**
- 基于基类的提供者模型
- **Drag-Once 数据绑定**
- 数据绑定的相关组件(控件)介绍
- 数据集设计器(DataSet Designer)
- 表适配器(TableAdapters)
- 保存连接字符串到程序配置文件

# 保存连接字符串到 程序配置文件

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司



- 简易的访问App.config文件的方法
  - ADODOTNET2.Properties.Settings.Default.NorthwindCo  
nnectionString

# DEMO

- 基于基类的提供者模型
- **Drag-Once** 数据绑定
- 数据绑定的相关组件
- 数据集设计器(**DataSet Designer**)
- 表适配器(**TableAdapters**)
- 保存连接字符串到程序配置文件

# 小结

- **ADO.NET 2.0 简介**
- 基于基类的提供者模型
- **Drag-Once** 数据绑定
- 数据绑定的相关组件(控件)介绍
- 数据集设计器(**DataSet Designer**)
- 表适配器(**TableAdapters**)
- 保存连接字符串到程序配置文件

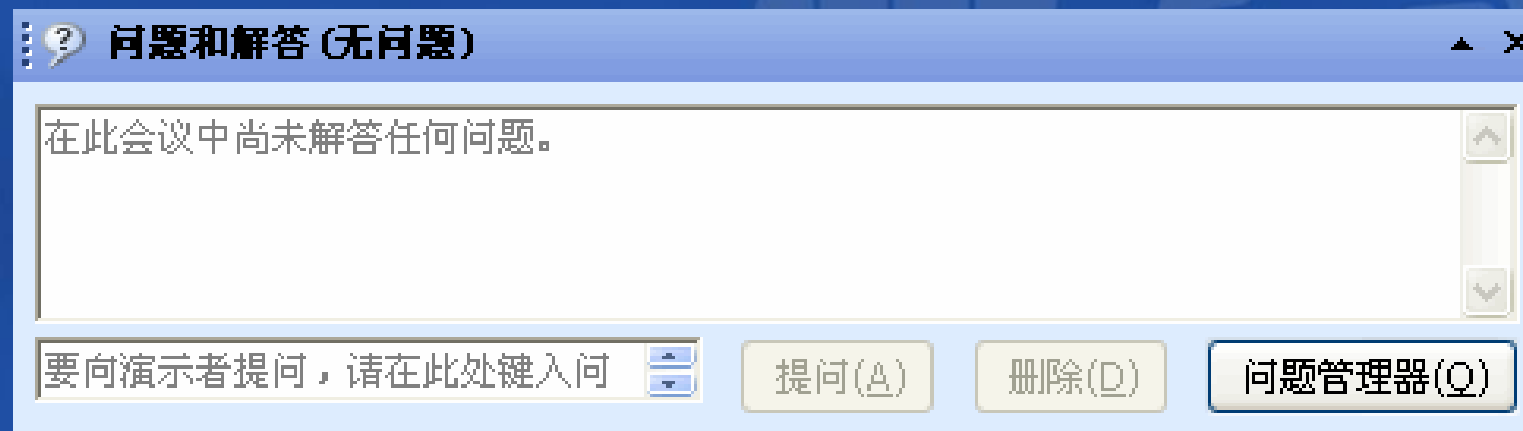


# 2005.9.21 课程预览

- **DataSet 改进**
- **异步处理(Asynchronous Processing)**
- **Multiple Active Result Sets (MARS)**
- **大批量数据操作(Bulk Copy Operations)**
- **New SQL Server Max Data Types**
- **SQL Server 用户自定义类型**
- **SQL Server Notifications**
- **Transactions with SQL Server Snapshot Isolation**
- **SQL Server 2005 数据库镜像**
- **批量处理(Batch Processing)**

# Q&A

- 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。



The screenshot shows a window titled "问题和解答 (无问题)" (Questions and Answers (No Questions)). Inside the window, there is a large text input area with the placeholder text "在此会议中尚未解答任何问题。" (No questions have been answered in this meeting yet.). Below the input area, there are four buttons: "要向演示者提问, 请在此处键入问" (To ask a question to the presenter, enter the question here), "提问(A)" (Ask), "删除(D)" (Delete), and "问题管理器(Q)" (Question Manager).

您的潜力. 我们的动力

**Microsoft®**  
微软(中国)有限公司

**Microsoft®**

msdn  


**MSDN Webcasts**