

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

创建动态和可配置智能客户端 应用程序

付仲恺
微软特邀开发专家

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

课程准备

- 了解Microsoft .NET[®] Framework 2.0 和 Microsoft[®] Visual Studio[®] 2005
- 有构建GUI应用程序的经验

Level 200

议题

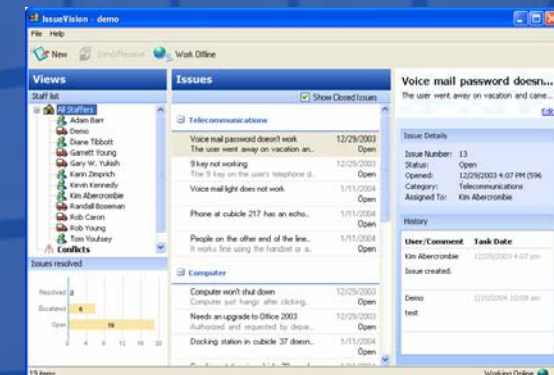
- **动机**
- 读 / 写配置设置
- 动态创建对象
- 设计可插拔对象
- Microsoft 配置应用程序块

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

动机

- 满足用户需求的变化
- 提高应用程序的可扩展性
 - 易于修改
 - 界面
 - 数据源
 - 易于满足用户的自定义需求



您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

demo

动态, 可配置应用程序

演示概述

- 设计动态, 可配置应用程序:
 1. 只读, 应用程序范围内设置: *app.exe.config*
 2. 读 / 写 用户指定设置: *user.config*
 3. 动态创建对象: *CreateInstance()*
 4. 可插拔对象: *ICommonInterface*

议题

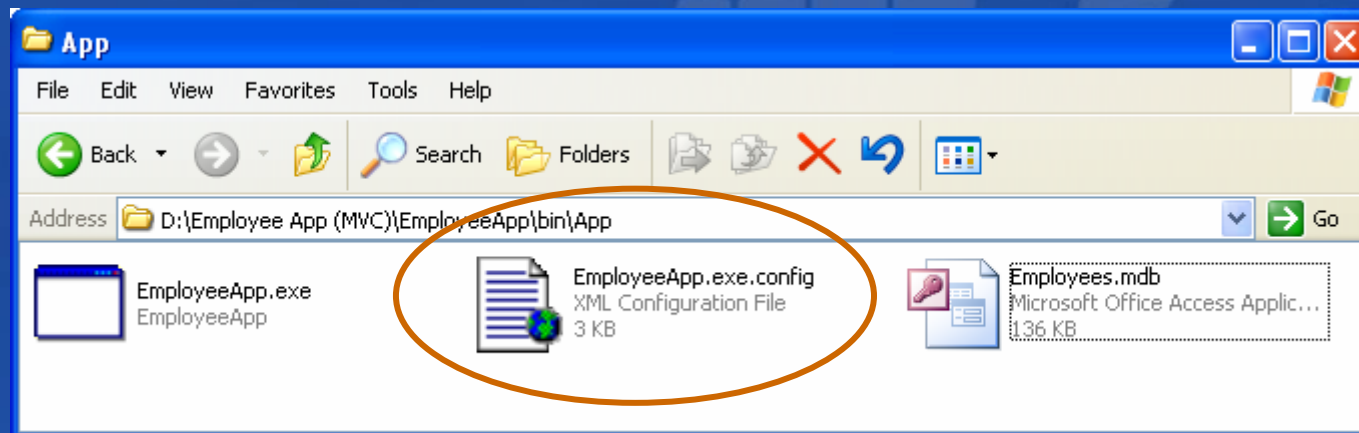
- 动机
- **读 / 写配置设置**
- 动态创建对象
- 设计可插拔对象
- Microsoft 配置应用程序块

配置设置

- 应用程序范围的配置设置:
 - 连接字符串
 - 服务器名称
 - Web Service URLs
 - **只读**
- 用户指定配置设置:
 - 字体和颜色偏好
 - 界面控件选择
 - GUI状态等各种个性化信息
 - **可读写**

应用程序范围配置文件

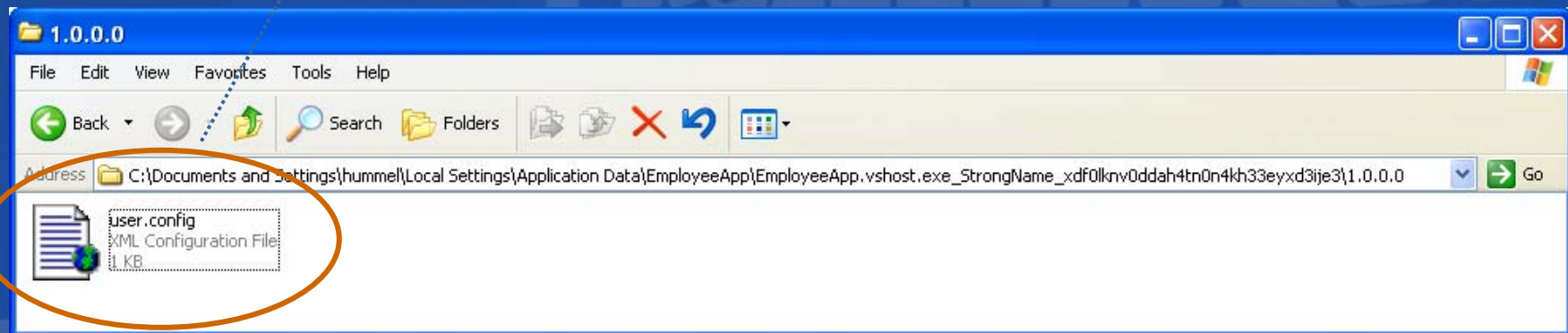
- 应用程序范围配置文件存储在 *appname.exe.config*
 - 以XML文件的形式存储在与.EXE相同的目录
 - 这些配置在程序的运行过程中为只读，不可修改



用户指定配置文件

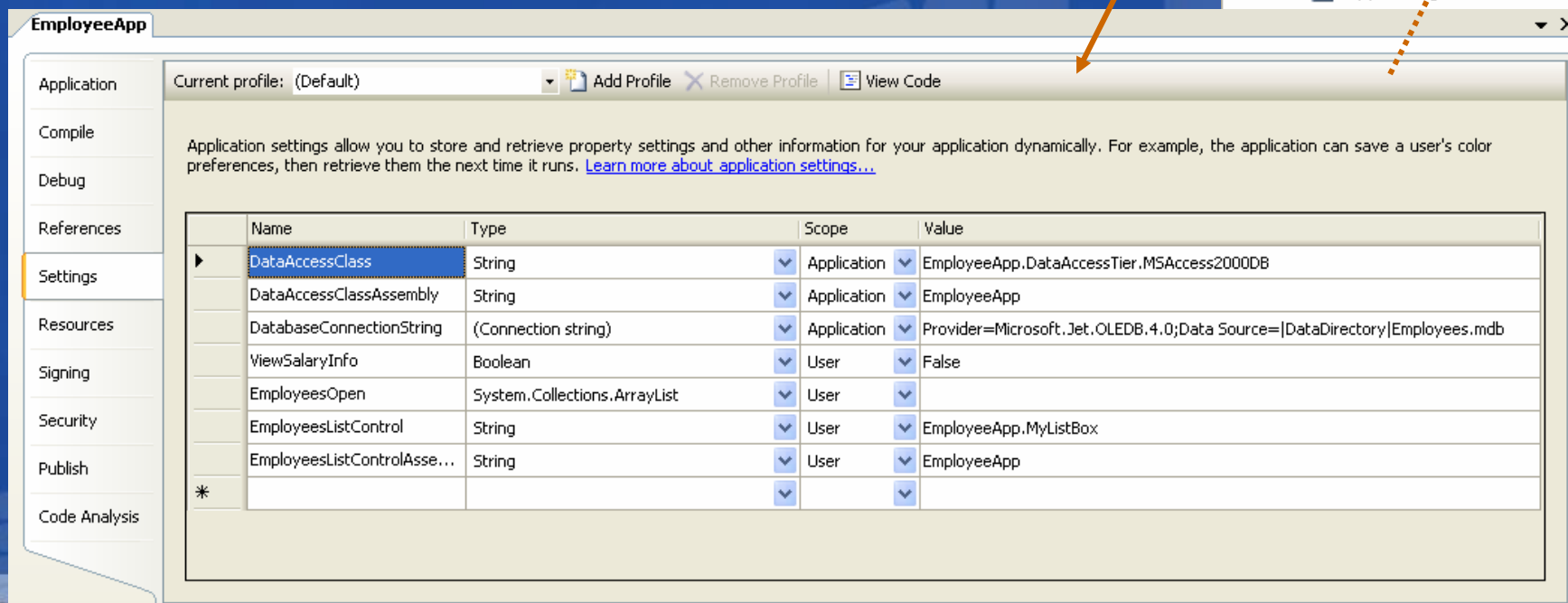
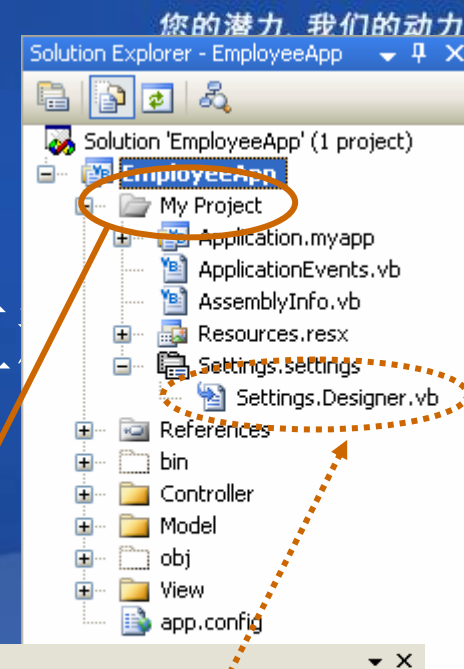
- 用户指定配置文件存储在 *user.config*
 - 以XML文件的形式存储在用户的本地目录
 - 配置在程序执行过程中可读 / 写

C:\Documents and Settings\username\Local Settings\Application Data\appname\



创建配置项

- 在Visual Studio 2005中，使用工具创建配置项：



访问配置项

- Visual Studio会自动生成配置的访问代码
- 读取:

```
** VB:  
s = Settings.Default.DatabaseConnectionString  
  
// C#:  
s = Properties.Settings.Default.DatabaseConnectionString;
```

- 写入:

```
** VB:  
Settings.Default.ViewSalaryInfo = True  
Settings.Default.Save()  
  
// C#:  
Properties.Settings.Default.ViewSalaryInfo = true;  
Properties.Settings.Default.Save();
```

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

demo

创建和访问应用程序和用户配置信息...

重置为用户缺省设置

- Visual Studio也可以生成代码将设置恢复为用户缺省设置

```
** VB:  
s = Settings.Default.Reset()  
  
// C#:  
s = Properties.Settings.Default.Reset();
```

议题

- 动机
- 读 / 写配置设置
- **动态创建对象**
- 设计可插拔对象
- Microsoft 配置应用程序块

动态实例化

- 问题:
 - 使用new创建对象需要类名称
 - 如何实例化未知类名并且带有参数的对象
- 解决方案:
 - .NET反射机制
 - *asm.CreateInstance(...)*

样例

- 让我们动态创建数据访问对象
 - 使用 *DataAccessClass* 和 *DataAccessClassAssembly* 设置
 - 允许我们在应用程序中使用不同的数据库:

```
// C#
System.Reflection.Assembly asm;
object obj;

//获取数据访问类所在的程序集
asm = System.Reflection.Assembly.Load(Settings.Default.DataAccessClassAssembly);

//动态创建这个类的实例, 并且调用对应的构造函数
object[] args; //传给构造函数的参数
args = new object[1];
args[0] = Settings.Default.DatabaseConnectionString;...
obj = asm.CreateInstance(Settings.Default.DataAccessClass, false,
System.Reflection.BindingFlags.CreateInstance, null, args, null, null);

// 转换类型为数据访问类型!
this.data = obj as DataAccessTier.GenericDataAccess;
ds = data.SelectQuery("Select * From Employees");
```

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

demo

动态创建对象...

议题

- 动机
- 读 / 写配置设置
- 动态创建对象
- **设计可插拔对象**
- Microsoft 配置应用程序块

可插拔对象

- 问题:
 - 智能客户端提供可扩展GUI
 - 智能客户端提供可互换的UI元素
 - 对不同程序集中和不同命名空间下的对象进行替换
- 解决方案:
 - 使用 **interfaces** 来确保对象可插拔
 - 将对象的命名空间, 类名等参数保存在配置文件中
 - 采用.NET的反射机制进行对象的实例化

考虑

- 应用程序提供多组控件供用户挑选
- 定义接口并且使程序可替换使用！

```
obj = asm.CreateInstance(Settings.Default.UIControl, ...)  
  
ICommonInterface uiControl;  
uiControl = obj as ICommonInterface;  
uiControl.Add(...);  
.  
.  
.  
uiControl.Show(...);
```



```
public interface ICommonInterface  
{  
    void Show(Form frm) //在form中显示控件  
    Sub Add(Object obj) //添加用于显示的数据  
    .  
    .  
    .  
}
```

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

demo

可插拔对象...

演示概述

- 应用程序提供了两个控件用于显示Employee
 - 用户设置可以记录用户更偏好于哪个控件
 - 通过实现相同的接口使得控件可以相互替换

```
public delegate void IEmployeeListClickHandler(string name);  
  
public interface IEmployeeList  
{  
    void Add(string name);  
    void Remove(string name);  
    event IEmployeeListClickHandler ClickOnEmployee;  
}
```



```
public class MyListBox : ListBox, IEmployeeList
```

```
class ...
```


议题

- 动机
- 读 / 写配置设置
- 动态创建对象
- 设计可插拔对象
- **Microsoft 配置应用程序块**

配置应用程序块

- Microsoft为管理配置设置提供了两个应用程序块:
 - 配置管理应用程序块 (CMAB) (单独安装, .NET 1.1):
<http://www.code-magazine.com/article.aspx?quickid=0505071>
<http://www.code-magazine.com/Article.aspx?quickid=0505061>
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/config.asp> (在下载区域可以找到链接)
 - 配置应用程序快 (CAB) (Microsoft *patterns & practices* Enterprise Library的一部分, .NET 1.1):
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/config.asp>
- CMAB是版本1, CAB是版本2
- 将Visual Studio 2005中的配置管理带到了Visual Studio 2003中

总结


- 智能客户端应该能够可配置
- 智能客户端应该能够被自定义
- 使用下列手段扩展您的架构:
 1. 只读应用程序范围设置: *app.exe.config*
 2. 读 / 写 用户指定设置: *user.config*
 3. 动态实例化对象: *CreateInstance()*
 4. 可插拔对象: *ICommonInterface*

获取更多MSDN资源

- **MSDN中文网站**
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.msp>



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