

您的潜力，我们的动力

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

# ASP.NET事务 和异常处理

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

- 事务处理简介
- 事务处理的方法
- 异常处理简介
- 异常处理的技术

# 1.1 什么是事务处理？

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

- 事务是一组组合成逻辑工作单元的数据库操作，虽然系统中可能会出错，但事务将控制和维护每个数据库的一致性和完整性。
- 如果在事务过程中没有遇到错误，事务中的所有修改都将永久成为数据库的一部分。
- 如果遇到错误，则不会对数据库作出任何修改。

# 1.2 事务处理过程

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

1. 开始一个事务。进入“事务待命”状态。
2. 在“事务待命”状态，记录事务中改变的数据库记录。此改变不能直接改变数据库中的值，必须先用一个顺序的“事务日志”记录在一边。同时，对于要改变的原始记录加锁，让其它用户无法读和写。如果记录已经被其它事务加锁，则报错。
3. 在“事务待命”，如果用户给出**commit transaction**命令，则进入“事务拷贝”状态，拷贝所有加锁的记录成备份。
4. 上面3执行完，则进入“事务更新”状态，用“事务日志”中记录一一更新实际的数据库记录。
5. 上面4执行完，则进入“事务结束”状态，释放所有的记录锁，然后抛弃“事务日志”和备份的原数据库记录。
6. 上面5做完后，事务被删除



# 事务处理过程

但是，最为关键的是，事务系统必须执行以下过程：一旦数据库由于软件、硬件问题发生故障，重新启动后，一旦有事务没正常删除，则：

- 7. 如果在“事务待命”、“事务结束”状态，则重新从5中结束事务的动作开始执行。
- 8. 如果在“事务更新”状态，则重新从4开始更新记录，并继续想下执行。结果，虽然系统崩溃过，但事务仍然能正常提交。

## 1.3 事务处理举例

甲



乙



A、判断余额

B、取钱

C、扣除

## 1.4事务处理有关事项

Microsoft®  
微软(中国)有限公司

- 事务处理的关键是在提交事务或者取消事务时，万一系统崩溃了，数据库在再次启动时，仍然需要保持数据可逻辑一致性。
- 应用中包含的事务应当尽量让它“瞬间”完成，避免在比较忙时造成用户进程的互锁。
- Informix、Oracle、DB2等数据库的实际事务处理流程更复杂，目的是具有更好的抵抗系统错误性质，因为事务保护是业务系统安全稳定的最后一道防线。



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- 事务处理简介
- 事务处理的方法
- 异常处理简介
- 异常处理的方法

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 事务处理方法

- 直接写入SQL
- 通过ADO.NET实现
- COM+事务（分布式事务）

## 2.1 直接写入SQL

- 在存储过程中使用 BEGIN TRAN, COMMIT TRAN, ROLLBACK TRAN 实现
- 优点:
  - 所有事务逻辑包含在一个单独的调用中
  - 拥有运行一个事务的最佳性能
  - 独立于应用程序
- 限制:
  - 事务上下文仅存在于数据库调用中
  - 数据库代码与数据库系统有关

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO1

## 直接写入SQL

## 2.2 ADO.NET 事务处理

在 ADO.NET 中, 可以使用 **Connection** 和 **Transaction** 对象来控制事务。若要执行事务, 请执行下列操作:

- 调用 **Connection** 对象的 **BeginTransaction** 方法来标记事务的开始。
- 将 **Transaction** 对象分配给要执行的 **Command** 的 **Transaction** 属性。
- 执行所需的命令。
- 调用 **Transaction** 对象的 **Commit** 方法来完成事务, 或调用 **Rollback** 方法来取消事务。



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO2

## ADO.NET事务处理

# ADO.NET事务处理

- 优点:
  - 简单性
  - 和数据据事务差不多的快
  - 独立于数据库, 不同数据库的专有代码被隐藏了
- 缺点:
  - 事务不能跨越多个数据库连接
  - 事务执行在数据库连接层上, 所以需要在事务过程中维护一个数据库连接

## 2.3 COM+事务

一般的数据库事务控制要求事务里所做的操作必须在同一个数据库内，这样在出现错误的时候才能回滚（RollBack）到初始状态。这就存在一个问题，在分布式应用程序中，我们往往需要同时操作多个数据库，使用数据库本身的事务处理，很难满足程序对事务控制的要求。在COM+中，提供了完整的事务服务，我们可以利用它来完成在分布式应用程序中的事务控制。

# 创建参与自动事务的类

1. 将**TransactionAttribute** 类应用于您的类, 指定组件请求的自动事务类型。事务类型必须是**TransactionOption** 枚举的成员。
2. 从**ServicedComponent** 类派生您的类。**ServicedComponent** 是所有使用 COM+ 服务的类的基类。
3. 用强名称 (strong name) 标记 (sign) 程序集 (assembly), 确保程序集包含唯一的密钥对。
4. 在 COM+ catalog 中注册包含您的类的程序集。



- 定义一个COM+事务处理的类  
[Transaction(TransactionOption.Required)]  
public class  
DataAccess:System.EnterpriseServices.ServicedComponent  
{  
    }  
}
- TransactionOption枚举类型支持5个COM+值
  - Disabled 忽略当前上下文中的任何事务。
  - NotSupported 使用非受控事务在上下文中创建组件。
  - Required 如果事务存在则共享事务，并且如有必要则创建新事务。
  - RequiresNew 使用新事务创建组件，而与当前上下文的状态无关。
  - Supported 如果事务存在，则共享该事务。



# 强名称

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

为使该组件正确运行，该组件必须有一个强名称。生成一个强名称，然后使用该强名称对程序集进行签名。为此，请按照下列步骤操作：

- 在 命令提示符下，键入 **sn.exe -k snEnterprise.snk** 以创建一个密钥文件。
- 将 **snEnterprise.snk** 复制到您的项目文件夹中。
- 在 **AssemblyInfo.vc** 中，将以下代码添加到其他程序集属性语句之前或之后：**[assembly: AssemblyKeyFileAttribute("..\\..\\snEnterprise.snk")]**
- 进行保存，然后生成您的项目
- 使用**regsvcs.exe**将DII注册到COM+ Services里面

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO3

## 使用 COM+ 事务处理

# 注意事项

- 确保使用 COM+ 服务的所有项目都有一个强名称。
- 使用 COM+ 服务的所有类都必须继承服务组件。服务组件位于 **System.EnterpriseServices** 命名空间中。
- 进行调试时，事务在提交或终止前可能会超时。要避免出现超时，请在事务属性中使用一个超时属性。在下面的示例中，在完成任何事务时，关联的方法在超时前都有 1,200 秒的执行时间。  
[Transaction(TransactionOption.Required,timeout:=1200)]

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- 事务处理简介
- 事务处理的方法
- 异常处理简介
- 异常处理的方法



# 什么是异常处理

- 异常是正在执行的程序所遇到的任何错误情形或者意外行为。
- 很多原因都可以引起异常，例如，代码中错误、操作系统资源不可用、公共语言运行时（**common language runtime**）中的意外情况等。
- 然而应用程序能够从上述的一些情况中恢复执行，但是大多数运行时异常是不可恢复的。在这种情况下，需要一种有效的方法来处理这些异常并给调用者提供相同的异常。



# 用结构化的异常处理方法 来处理异常

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

- 在.NET Web服务中，对异常处理支持的关键点是由try...catch..finally语句提供的。
- 关键字try放在可能抛出异常的普通处理代码块之前。
- 关键字catch放在异常处理代码块之前。
- 关键字finally放在那些经常在异常处理后还需要执行的代码块之前。
- 一旦异常从try代码块中抛出，程序流切换到后面的第一个catch代码块。

# 异常类

- **Exception** 所有异常对象的基类
- **SystemException** 运行时产生的所有错误的基类
- **IndexOutOfRangeException** 当一个数组的下标超出范围时运行时引发
- **NullReferenceException** 当一个空对象被引用时运行时引发
- **InvalidOperationException** 当对方法的调用对对象的当前状态无效时, 由某些方法引发
- **ArgumentException** 所有参数异常的基类
- **ArgumentNullException** 在参数为空(不允许)的情况下, 由方法引发
- **ArgumentOutOfRangeException** 当参数不在一个给定范围之内时, 由方法引发
- **InteropException** 目标在或发生在CLR外面环境中的异常的基类
- **ComException** 包含COM 类的HRESULT信息的异常
- **SEHException** 封装win32 结构异常处理信息的异常

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO4

## 结构化异常处理

# 优化异常

- 理解异常是一定会发生的
  - 大多数的软件系统都不是百分之百可靠的！
  - 要站在异常一定可能会发生的角度来编写异常处理程序，应对程序有可能发生的错误。
  - 建立一个良好的异常处理策略
- 处理未预料的异常
  - 确保所有程序的入口都使用了try-catch
  - 在catch中截获所有的异常



# 异常处理注意事项

- 当引发异常时，要提供有意义的文本。
- 要引发异常仅当条件是真正异常；也就是当一个正常的返回值不满足时。
- 如果你的方法或属性被传递一个坏参数，要引发一个 **ArgumentException** 异常。
- 当调用操作不适合对象的当前状态时，要引发一个 **InvalidOperationException** 异常。
- 要引发最适合的异常。
- 要使用链接异常，它们允许你跟踪异常树。
- 不要为正常或预期的错误使用异常。
- 不要为流程的正常控制使用异常。
- 不要在方法中引发 **NullReferenceException** 或 **IndexOutOfRangeException** 异常。



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 议程

- 事务处理简介
- 事务处理的方法
- 异常处理简介
- 异常处理技术

# 异常处理技术

- 记录异常
  - 在文件中记录异常
  - 在数据库中记录异常
  - 在eventlog中记录异常
- 发送email通知异常
- 异常产生时, 用友好(user-friendly)的方式通知用户

# 处理错误

- Page\_Error事件
- Application\_Error事件
- 利用配置文件，自定义错误页面
  - `<customErrors defaultRedirect="url" mode="RemoteOnly">`  
    `<error statusCode="code" redirect="url"`  
    `></error>`  
    `</customErrors>`

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# DEMO5

编写到windows错误日志  
把错误发送邮件到管理员  
自定义错误页面



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 小结

- 事务处理简介
- 事务处理的方法
- 异常处理简介
- 异常处理的技术

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**