

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

使用Remoting开发分布式应用 ——Remoting入门

任旻

微软特约讲师

RenMin@bjut.edu.cn

个人简介

- 2002年从事.Net开发
 - 中国民航中文适航指令管理系统
 -
 - 中国电信互联星空三期
- 微软ISV支持工作
 - .Net培训讲师, 技术支持工程师
- 北京工业大学计算机学院
 - 分布式技术, 网格技术的研究
- TechEd 2005
 - Dev370 着眼未来的分布式设计

本次课程

Level 100

- 难度100（容易）
- 预期听众
 - .Net技术的初学者
 - 没有使用或很少使用Remoting技术的开发者
 - 希望对Remoting有系统了解的程序员
- 具备条件
 - 了解.Net1.1和C#

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

本次课程内容包括

- Remoting技术简介
- Remoting开发过程

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

议程

- Remoting技术简介
- Remoting开发过程

议程

- Remoting技术简介

- 出现背景

- Xml

- Soap

- 序列化

- Remoting开发过程

Remoting出现的契机

- 分布式应用的需求迅速增长
 - 进程之间通讯
 - 局域网中计算机通讯
 - 互联网中的通讯
 - 各个领域: 商业, 娱乐, Peer-to-Peer, 网格(Grid)
- 原有的C/S, B/S模式和技术已经不能胜任
 - 串口RS232, Socket, RPC, DCOM

什么是Remoting

- Remoting的词根——Remote

- Remote Object
- 分布式对象

- Remoting的优势

- 性能
- 扩展性
- 可配置性
- 安全
- 生存周期管理

Remoting使用的技术

- XML
- SOAP
 - 简单对象传输协议
- 序列化
 - 添加 **SerializableAttribute**
 - 实现 **ISerializable** 接口

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

DEMO

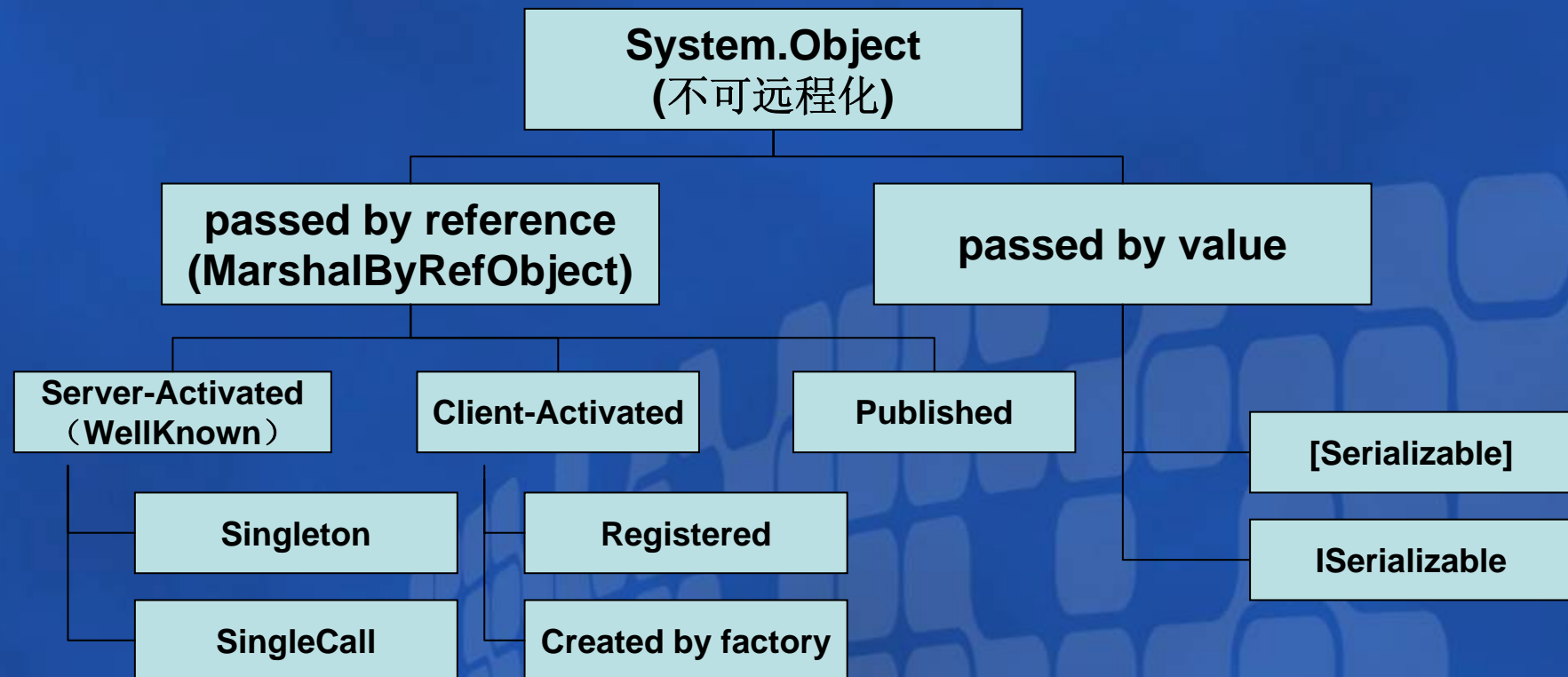
将对象序列化

- 1、序列化为XML
- 2、序列化为二进制数据

议程

- Remoting技术简介
- Remoting开发过程
 - Remoting框架
 - 开发Remoting程序的三个步骤
 - 服务器端激活
 - 客户端激活
 -

Remoting 框架图



远程对象的两个含义

- 操作远程对象

- 对象运行在远程, 客户端向他发送消息。
- MarshalByRefObject

- 传递远程对象

- 将远程的对象拿到本地, 或者将本地对象发送过去。
- 对副本进行操作
- [Serializable] 或 ISerializable

谁来激活对象？

- 服务器激活 (WellKnown)
 - Singleton
 - SingleCall
- 客户端激活

通道 (Channels)

- 一个远程对象使用通道发送和接收消息
 - 服务器选择一个通道来监听请求 (request)
 - 客户端选择通道来和服务器通讯
- Remoting提供了内置的通道
 - TCP通道和HTTP通道
 - 你也可以编写自己的通道

开发Remoting三步走

- 创建远程对象
- 创建一个应用程序作为“宿主（host）”，以接收客户请求
- 创建一个客户端调用远程对象

第一步：创建远程对象

● 继承 System.MarshalByRefObject

```
public class HelloServer : MarshalByRefObject  
{  
    .....  
}
```

第二步：创建宿主应用程序

- 注册通道
 - 内置通道：TCP，HTTP
- 注册服务器激活的远程对象（WellKnown）
 - Singleton，SingleCall
 - URL
- 运行宿主程序

第三步：建立客户端程序

- 注册通道
 - 内置通道：TCP，HTTP
- 根据URL得到对象代理
- 使用代理调用远程对象

传递参数

- 传递简单类型
 - int, double, string, enum.....
- 传递可序列化的类型
 - ArrayList, Hashtable, DataSet.....
- 传递自定义类型
 - [Serializable]

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

DEMO

传递自定义类型的对象

Singleton VS. SingleCall

- 都是服务器激活的对象 (WellKnown)
- Singleton (单实例)
 - 在服务器端只实例化一次。
 - 以后每次调用都访问同一个实例。不论同一客户端还是不同客户端。
 - 可以维持状态。
- SingleCall (单调用)
 - 每次调用都实例化新的实例。
 - 更好的支持无状态编程模型。

您的潜力. 我们的动力

Microsoft[®]
微软(中国)有限公司

DEMO

Singleton VS. SingleCall

Serialization VS. MarshalByRefObject

- 按值列集 (Serialization)
 - 得到远程对象的副本。
 - 对副本的操作不影响远程对象。
 - 不论远程对象是Singleton还是SingleCall
- 按引用列集 (MarshalByRefObject)
 - 得到远程对象引用, 本地创建代理 (Proxy)
 - 通过 (Proxy) 对远程对象访问。
 - Singleton记录更改, SingleCall无状态。

您的潜力. 我们的动力

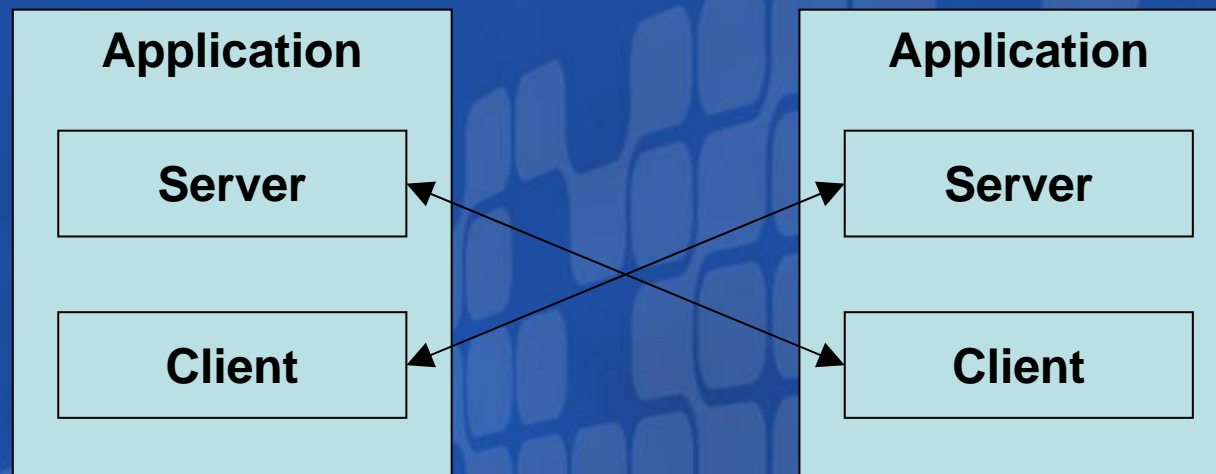
Microsoft[®]
微软(中国)有限公司

DEMO

Serialization
VS.
MarshalByRefObject

Remoting的模式

- 服务器/客户端模式
- 如果实现端对端 (Peer-to-Peer)
 - 则每个段既是服务器又是客户端




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