

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

使用MSMQ 创建Windows Mobile远程通讯程序

马宁

Windows Mobile MVP



本次课程内容包括

- MSMQ概述
- Windows CE和.NET CF 2.0的MSMQ支持
- 安装与配置MSMQ
- 发送与接收消息
- 支持HTTP的MSMQ
- 可靠性与性能
- 安全性

MSMQ: 概述

- MSMQ是一种灵活、可靠的通讯方式
 - MSMQ基础结构支持Windows和Windows CE操作系统
 - 轻量级编程模式
 - 应用程序开发者不必为以下情况写代码
 - 建立和维持网络连接
 - 选择使用的通讯管道
 - 持久性数据缓冲区
- 对于分布式系统, MSMQ是一个理想的解决方案, 也包括“间歇在线”设备
 - 应用程序更加容易使用和升级
 - 分布式应用范围更加广泛, 也包括不稳定的网络
 - 维持数据的完整性

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

MSMQ In Action

demo

MSMQ Essentials



- MSMQ支持应用程序间通讯, 不需要一个持续的连接
- MSMQ支持
 - 可靠的消息传递
 - 路由
 - 安全性
 - 消息优先级
- MSMQ接受发送程序的消息, 并将其放置在一个接受程序可以访问的队列中
- 如果队列无法访问(也许是由于网络无法连接), MSMQ将会再试
- 如果目的地队列在远程机器上, 并且无法访问, MSMQ将消息缓存到本地, 直到网络连接恢复
- **保证消息送达!**

Windows CE的MSMQ 支持

关键差异

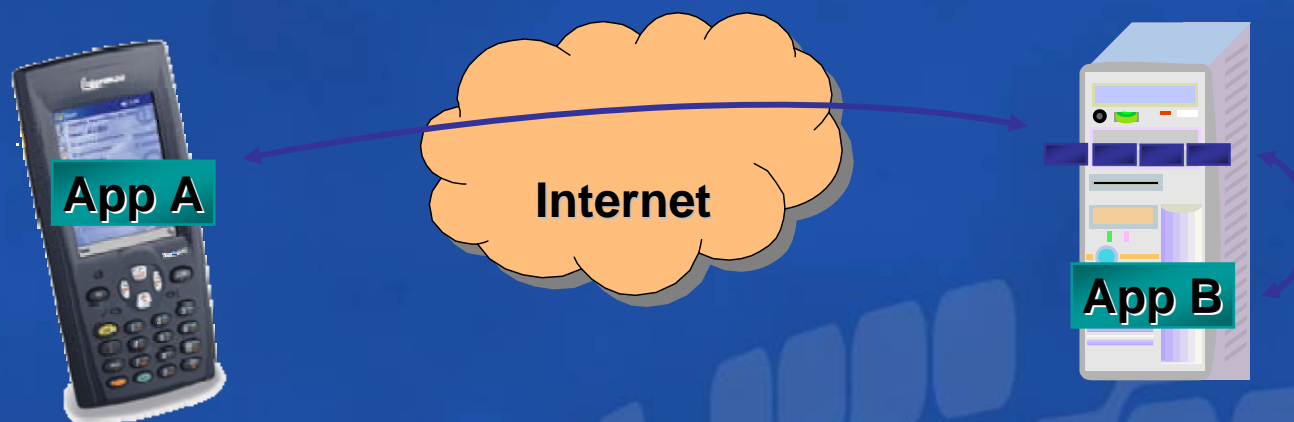
- Windows CE的MSMQ是桌面实现的一个子集
- 主要差异
 - 不能从远程机器的队列中读取消息
 - 只能作为独立的客户端使用
 - 不需要访问域控制器或消息队列服务器进行操作
 - 不能查询Active Directory来定位队列，因此必须知道队列位置
 - 不能分享通过MS DTC的coordinated transactions
 - 事实上，支持Single-Message Transactions—保证发送消息（并非接收）只有一次、按次序传递
 - 可以只创建本地、私有队列
- 支持NIC tracking
 - 超过桌面级MSMQ性能
 - 当MSMQ发现网络再次连接后，可以立刻（或者等待指定的时间）重试传递未送达的消息，例如当一个无线网络重新连接

MSMQ Support In Windows CE

Connectivity scenarios

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司



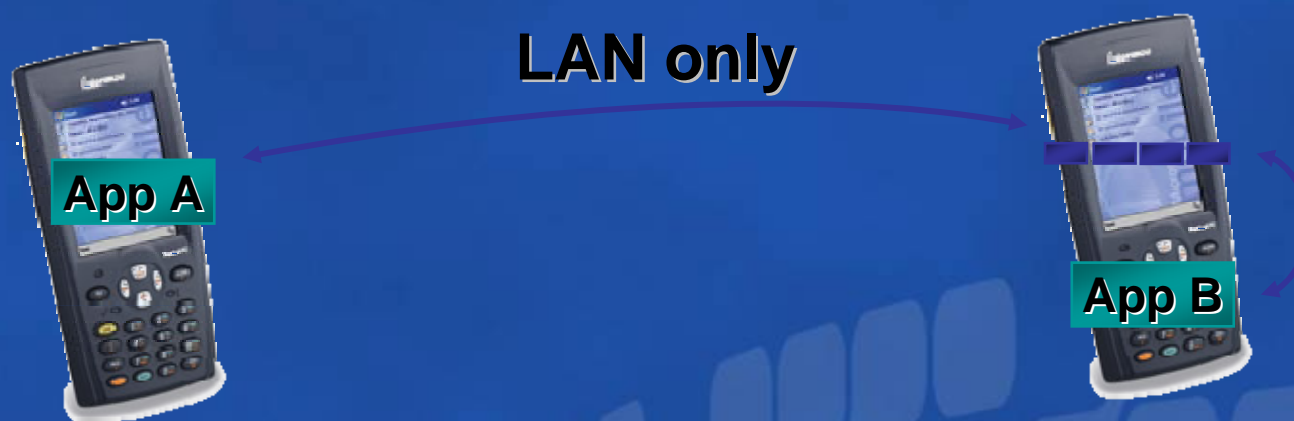
- 设备上的应用程序 A 发送消息到server上的public或private队列
- Server上的应用程序 B 从相同队列接收消息
- 基于HTTP的Internet 访问支持Windows CE 4.0 及后续版本和Windows Mobile 5.0 Pocket PC/Smartphone, 但是不支持Pocket PC 2003

MSMQ Support In Windows CE

Connectivity scenarios

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司



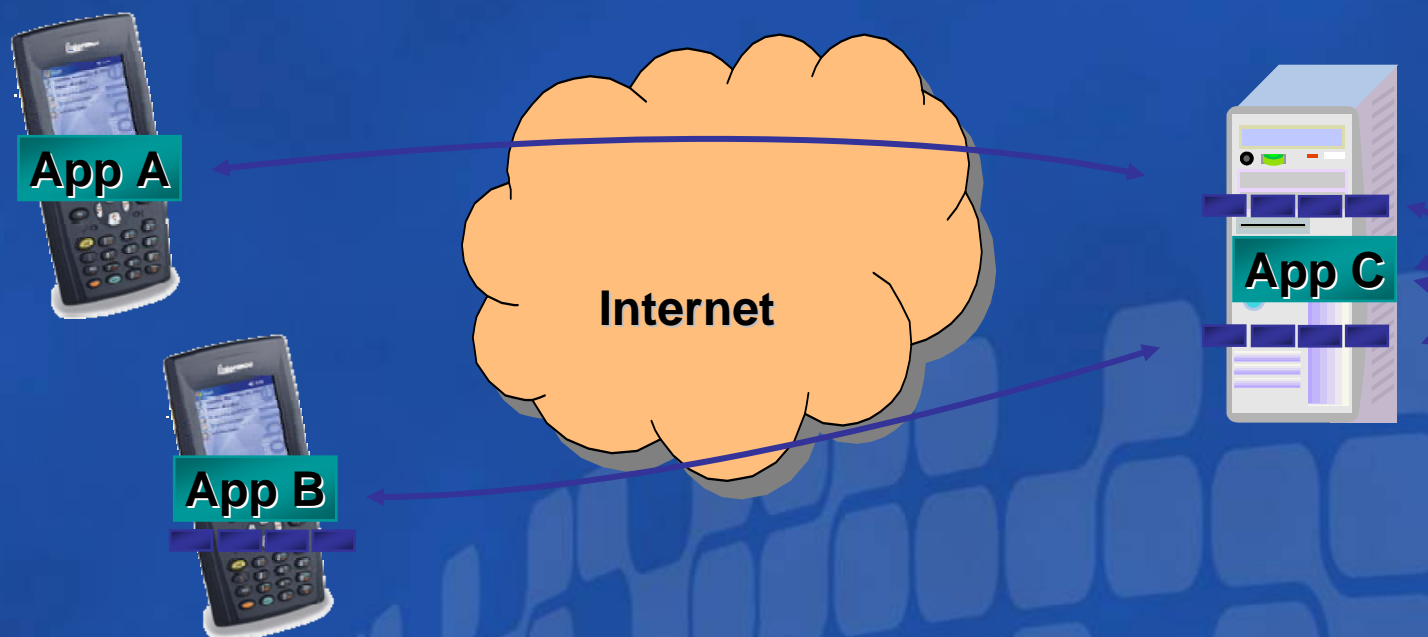
- 应用程序 B 在本地设备中创建private队列
- 设备上的应用程序 A 发送消息到设备中的队列
- 设备上的应用程序 B 从相同的队列中接收消息
- 注意:Windows CE的MSMQ对于远程队列可以写,但是不能读

MSMQ Support In Windows CE

Connectivity scenarios

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司



- 设备A上的应用程序A发送消息到路由server的public或private队列
- 应用程序C接收消息，并存储到某种数据存储器中
- 设备B上的应用程序B周期性地发送poll消息到server的队列中；应用程序C接收到poll消息，并响应发送来自设备A的消息到B

MSMQ Applications In .NET CF 2.0

What you need

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Visual Studio 2005
- 目标设备或模拟器:
 - Windows CE 4.2 或 5.0 内置MSMQ支持
 - Windows Mobile 2003 for Pocket PC/Smartphone
 - Pocket PC 2003 SDK 或 Smartphone 2003 SDK的Native组件
 - Windows Mobile 5.0 for Pocket PC/Smartphone
 - Cab文件包括Windows Mobile Redistributable Components package
 - 从 <http://msdn.microsoft.com/windowsmobile> 下载

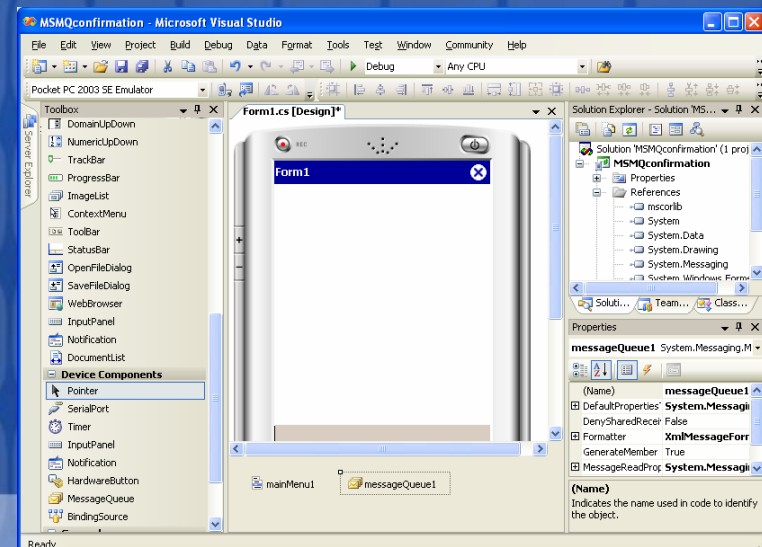
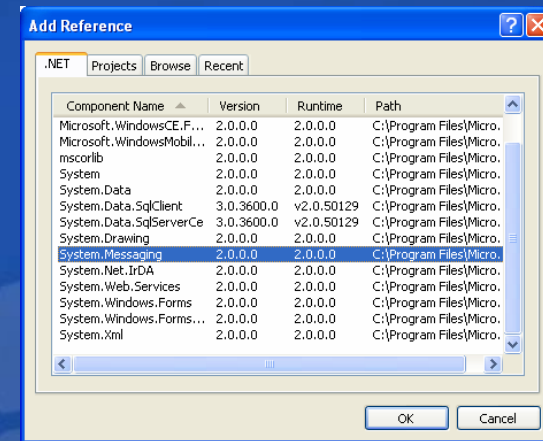
MSMQ Applications In .NET CF 2.0

Getting started

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 创建project
 - Add reference 到 System.Messaging
 - 或者...
 - 将MessageQueue组件拖拽到Form



MSMQ Applications In .NET CF 2.0

Send a message

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

```
using System.Messaging;
...
//Send and receive a message in button1's click event handler
private void button1_Click(object sender, EventArgs e)
{
    //The Queue on the device we will be sending messages to
    string strDestQ = @".\private$\queueOnDevice";
    try
    {
        // Create the queue if needed
        if (!MessageQueue.Exists(strDestQ))
            MessageQueue.Create(strDestQ);
        MessageQueue mq = new MessageQueue(strDestQ);

        // Build message
        Message m = new Message();
        m.Label = "Hello World";
        m.Body = "Hello World";

        // Send and close queue
        mq.Send(m);
        mq.Close();
        MessageBox.Show("Message sent!");
    }
}
```


MSMQ Applications In .NET CF 2.0

Receive a message

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

//continued...

```
// Create an Xml formatter for a message that
// is a String type
mq.Formatter =
    new XmlMessageFormatter(new Type[] { typeof(String) });

// Receive message synchronously
Message messageReceived = mq.Receive();

MessageBox.Show((string)messageReceived.Body);
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
}
```

MSMQ Applications In .NET CF 2.0

Testing the application

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 应用程序触发异常
System.InvalidOperationException
– “Message Queuing has not been installed on this computer.”
- 在尝试使用MSMQ之前，必须启动操作系统中的Message Queuing services



Starting MSMQ On The Device

Install runtime components

- 支持MSMQ的Windows CE设备已经存在runtime组件
- 如果目标设备为Pocket PC或Smartphone, 需要下列文件:
 - `msmqadm.exe`
 - `msmqadmext.dll`
 - `msmqd.dll`
 - `msmqrt.dll`
 - `visadm.exe`
- 在Windows Mobile 2003 SDK中, 文件位于
 - `C:\Program Files\Windows CE Tools\wce420\{device}\Support\msmq\armv4`
 - 对于真实设备和模拟器, 都使用 `\armv4` 中的版本, 而不是 `\x86` 中的 – Visual Studio 2005 emulators 运行 armv4 指令集!
 - 拷贝到设备的\Windows目录中
- 对于Windows Mobile 5.0
 - 安装 MSMQ CAB 文件
 - Redistributable Server Components for Windows Mobile 5.0
 - <http://www.microsoft.com/downloads/details.aspx?FamilyID=cdfd2bb2-fa13-4062-b8d1-4406ccddb5fd&DisplayLang=en>

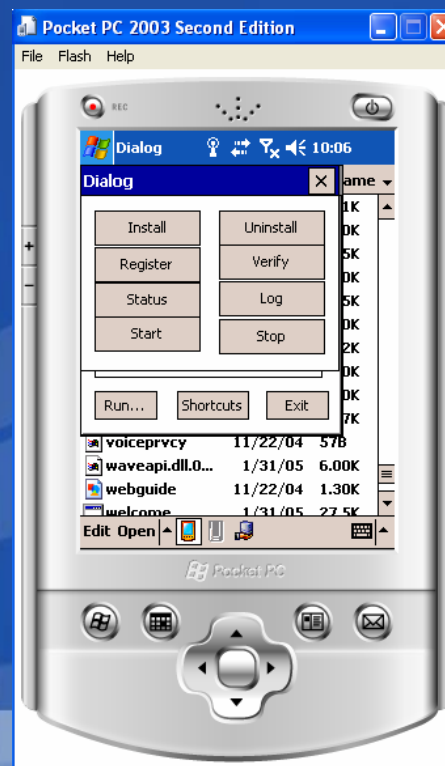
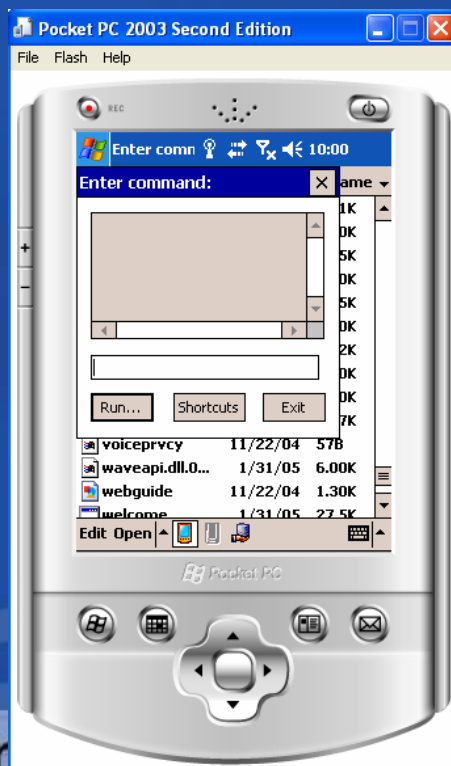
您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Starting MSMQ

Visual MSMQ administrator console

- 在File Explorer中, 执行 `\windows\visadm.exe`
- 点击 Shortcuts 按钮



- Install
- Register
- Verify
- Start
- Status

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Installing And Starting MSMQ On Pocket PC

demo

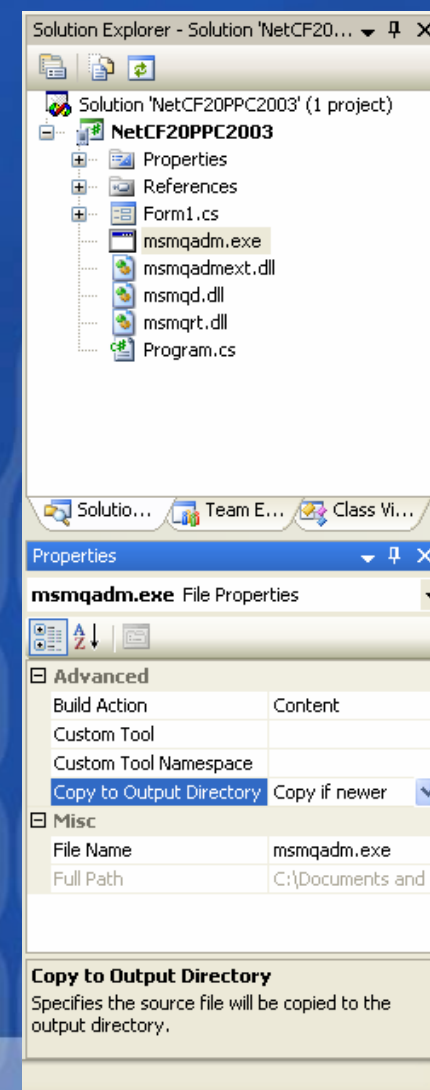
Starting MSMQ On The Device

Start MSMQ programmatically - 1

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Visadm.exe 不适合大规模部署
 - 将runtime文件与project一起部署
 - 写代码拷贝到 **\Windows**
 - 调用 **CreateProcess** 以运行 **msmqadm.exe** 来通过代码安装和启动 service
- 使用应用程序安装到目标设备中
 - 设置 **Build Action** 为 **Content**
 - 设置 **Copy to Output Directory** 为 **Copy if Newer**
- Reference: Mark Ihimoyan's blog at <http://blogs.msdn.com/ihimmar>



Starting MSMQ On The Device

Start MSMQ programmatically - 2

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 使用System.Diagnostics.Process.Start() 运行msmqadm.exe

```
using System.Diagnostics;
public class Utilities
{
    //Execute commands on the MSMQAdm.exe utility
    public static bool ExecuteMSMQAdm(String CmdLine)
    {
        Process prc = Process.Start(@"\windows\msmqadm.exe", CmdLine);
        if (prc == null)
            throw new ApplicationException("Process already running");

        prc.WaitForExit();
        if (prc.ExitCode != 0)
            return false;
        else
            return true;
    }

    public static void StartMSMQ()
    {
        //Check status of MSMQ (is it installed and running yet?)
        if (!(ExecuteMSMQAdm("status")))
        { ...

```

Starting MSMQ On The Device

Start MSMQ programmatically - 3

- 调用msmqadm.exe清除注册表, 安装MSMQ作为设备驱动, 创建注册表中的配置, 并启动protocol

```
// Check status of MSMQ (is it installed and running yet?)
if (!(ExecuteMSMQAdm("status")))
{
    try {
        //Deletes the MSMQ registry key and store directory.
        //All messages are lost.
        ExecuteMSMQAdm("register cleanup");

        //Installs MSMQ as device drivers.
        ExecuteMSMQAdm("register install");

        //Creates the MSMQ Configuration in Registry
        ExecuteMSMQAdm("register");

        //Enables the native MSMQ protocol
        ExecuteMSMQAdm("enable binary"))
    }
    catch (Exception exc) {
        MessageBox.Show("MSMQ Install Failed! Error: " + exc.Message);
        return;
    }
    ...
}
```


Starting MSMQ on the Device

Start MSMQ programmatically - 4

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 启动service,并在Pocket PC/Smartphone加载设备驱动程序
 - P/Invoke ActivateDevice()

```
[DllImport("CoreDll.dll")]
private extern static IntPtr ActivateDevice(...);
[DllImport("CoreDll.dll")]
private extern static Int32 CloseHandle(IntPtr hProcess);
...

//Starts the MSMQ service
if (CreateProcess(MSMQ_ADM, "start"))
{
    MessageBox.Show("Started MSMQ Successfully");
}
else
{
    //This is one additional step that is needed for PocketPCs
    //The Device Drivers have to be loaded before the service
    //can be started
    //ActivateDevice will load the device drivers
    IntPtr handle = ActivateDevice(@"Drivers\BuiltIn\MSMQD", 0);
    CloseHandle(handle);
}
```

Starting MSMQ On The Device

Start MSMQ during application install

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Windows Mobile 5.0
 - 在安装过程中调用MSMQ CAB
- Windows Mobile 2003
 - 在.inf文件中包括MSMQ可执行文件和必要的注册表键值
 - 使用eVC++创建setup.dll运行msmqadm.exe来安装MSMQ
 - Example: 查看Ken Rabold文章“Programing MSMQ on the Pocket PC using eVB”中的示例代码
http://www.devbuzz.com/content/zinc_eVB_programming_MSMQ_pg1.asp

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; .inf file for MSMQ app
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
...
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Specifies the files to copy
[Files.Windows]
; Copy only if newer
"msmqd.dll",,,0xA0000001
"msmqrt.dll",,,0xA0000001
"netregd.dll",,,0xA0000001
"msmqadm.exe",,,0xA0000001
"msmqcom.dll",,,0xA0000001

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Specifies the registry entries
;
[Reg.MSMQ]
; MSMQ Driver
HKLM,Drivers\BuiltIn\MSMQD,Dll,...
HKLM,Drivers\BuiltIn\MSMQD,Entry,...
HKLM,Drivers\BuiltIn\MSMQD,Keep,...
HKLM,Drivers\BuiltIn\MSMQD,Order,...
```

您的潜力. 我们的动力

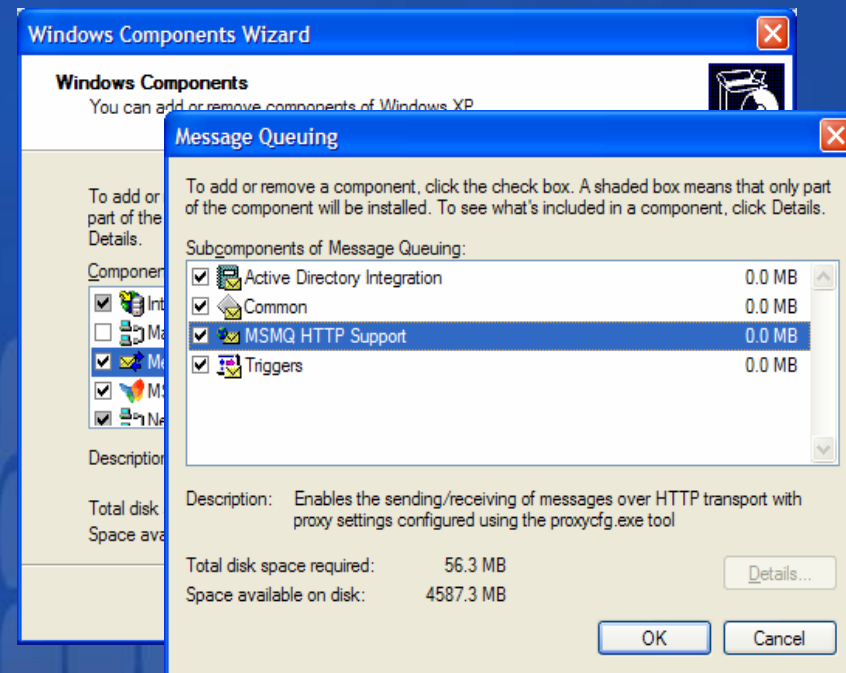
Microsoft
微软(中国)有限公司

Installing And Starting MSMQ Programmatically

demo

Setting Up MSMQ On A Server

- 在Windows XP和Windows 2003 Server中, MSMQ都不是默认安装的
 - 使用“添加或删除程序”中的“添加或删除Windows组件”“消息队列”
 - MSMQ HTTP 支持没有被默认选择
 - 选择 详细信息...



Managing Queues On A Server

- 使用System.Messaging API创建或删除队列

Try

```
Dim queueAsMessageQueue
```

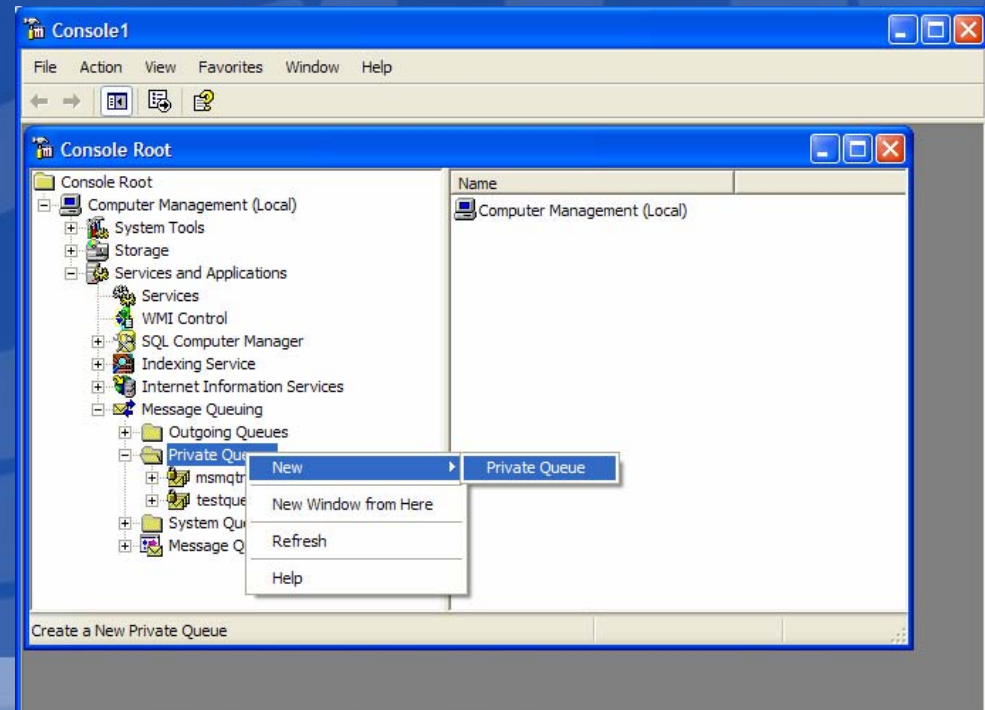
```
queue = MessageQueue.Create( ".\Private$\MyNewQueue" )
```

```
Catch ex As MessageQueueException
```

```
...
```

End Try

- 或者使用控制面板
 - 管理工具
 - 计算机管理



Queue Naming

- MSMQ 支持两种类型的消息队列
 - Public
 - 队列必须在一个域中的一台计算机上
 - 队列名称和属性发布在Active Directory中
 - Public队列被命名为 *machineName\queueName*
 - Private
 - 与public队列有相同的属性，但是你不能在directory中查找名称
 - 比public队列更有效率，并且不依赖于AD
 - Private 队列被命名为 *machineName\Private\$\queueName*
- 使用 Direct Format names
 - *TCP:192.168.0.11\Private\$\MyQueue*
 - *OS:MyServer\Private\$\MyQueue*
 - Direct format names 提供了最佳的性能

Configure Name Resolution On Device

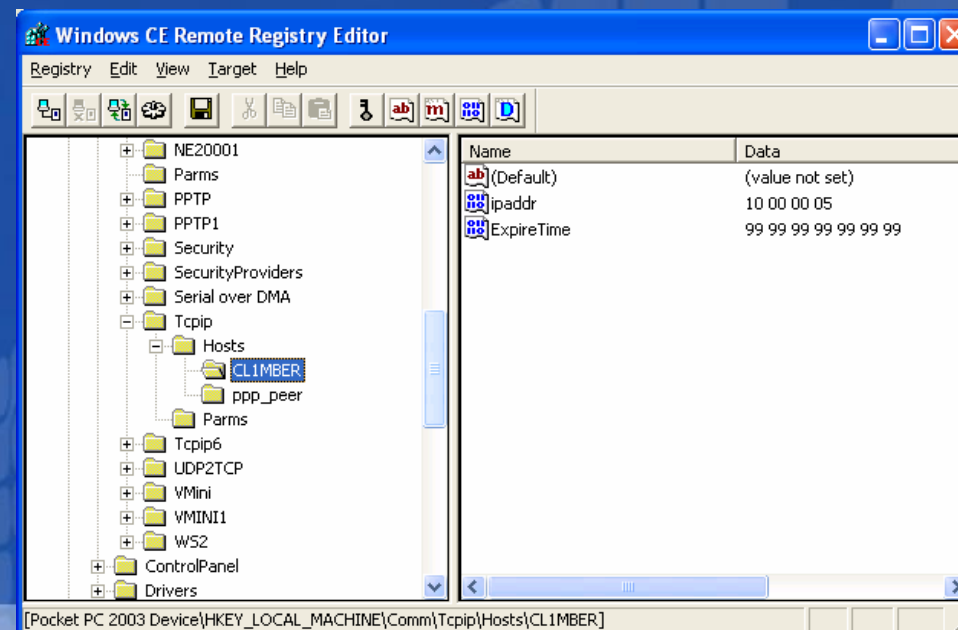
您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Windows CE MSMQ依赖于下列条件：
 - 根据计算机名获得IP地址的功能— DNS 或 WINS
 - 根据IP地址获得计算机名称的功能 — DNS 反向查找或者通过netbios node status
 - 路由消息内部使用计算机名，优于使用IP地址
 - 每个设备必须有一个唯一名称
- 三种操作配置名称
 - WINS
 - DNS
 - Registry Subkey

Configure Name Resolution In The Registry

- 如果WINS 或 DNS 无效, 最简单的办法是在注册表中配置host
 - 在 **HKEY_LOCAL_MACHINE\Comm\Tcpip\Hosts** 键值中定义host
 - 两个二进制变量:
 - ipaddr – hex nn nn nn nn
 - ExpireTime – 99 99 99 99 99 99 99
- Tip: Download **Pocket Hosts** from <http://www.handango.com>



Sending Messages Across The Network

您的潜力, 我们的动力

Microsoft
微软(中国)有限公司

- 远程队列寻址, 使用 DIRECT format names
 - Private 队列 hostname
FormatName:Direct=OS:hostname\Private\$\queueName
 - Private 队列 IP address
FormatName:Direct=TCP:nn.nn.nn.nn\Private\$\queueName
 - Public 队列:
FormatName:Direct={OS:hostname|TCP:nn.nn.nn.nn}\queueName
- 特殊示例: 发送消息到 public 队列的, 但是不知道队列所在的机器名称
 - Windows CE 不能查询 Active directory 来定位队列
 - 设置 **HKEY_LOCAL_MACHINE\Software\Microsoft\MSMQ\SimpleClient\OutFRSQueue** 为 Falcon Routing Server (FRS) 的名称, 可以用来发送消息

Sending Messages

```
using System.Messaging;
...
//The remote queue we will be sending messages to
string strDestQ =
    @"FormatName:Direct=OS:CLIMBER\private$\MyQueueOnSvr";
try
{
    // Open the queue context
    MessageQueue mq = new MessageQueue(strDestQ);

    // Build message
    Message m = new Message();
    m.Label = "Hello World";
    m.Body = "Hello World";

    // Send and close queue
    mq.Send(m);
    mq.Close();
    MessageBox.Show("Message sent!");
}
catch (MessageQueueException ex)
{
    ...
}
```

Synchronous Message Reception

```
try
{
    MessageQueue mq = new MessageQueue(@".\Private$\MyQueueOnSvr");

    // Create an Xml formatter for a message that
    // is a String type
    mq.Formatter =
        new XmlMessageFormatter(new Type[] { typeof(String) });

    // Receive message synchronously - 30 second timeout
    Message messageReceived =
        mq.Receive(new TimeSpan(0, 0, 30));
}
// Catch exception - would include a timeout
catch (MessageQueueException ex)
{
    MessageBox.Show(ex.ToString());
}
```

Asynchronous Message Reception

BeginReceive

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

```
Private MessageQueue asyncMQ;

public void ReceiveAsync() {
    try {
        asyncMQ = new MessageQueue(@".\Private$\MyQueueOnSvr");

        // Create an Xml formatter for a message that
        // is a String type
        asyncMQ.Formatter =
            new XmlMessageFormatter(new Type[] { typeof(String) });

        // Add an event handler for the ReceiveCompleted event.
        asyncMQ.ReceiveCompleted +=
            new System.Messaging.ReceiveCompletedEventHandler(
                MyOwnReceiveCompleted);

        // Begin an asynchronous read operation
        asyncMQ.BeginReceive ( ) ;
    }
    catch (MessageQueueException) { ... }
}
```


Asynchronous Message Reception

Receive completed event handler

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

```
public void MyOwnReceiveCompleted(Object source,
    System.Messaging.ReceiveCompletedEventArgs asyncResult) {
    try {
        // End the asynchronous receive operation.
        System.Messaging.Message msg =
            asyncMQ.EndReceive(asyncResult.AsyncResult);

        // Convert received message to object that we think was sent
        String text = (String) msg.Body;

        // Display it to the user
        MessageBox.Show(text, "Asynchronous Read Complete");

        // Restart the asynchronous receive operation.
        asyncMQ.BeginReceive();
    }
    catch (MessageQueueException) { ... }
}
```

Sending Objects Over MSMQ

- 在full .NET Framework中有三种Formatter
 - ActiveXMessageFormatter
 - BinaryMessageFormatter
 - XmlMessageFormatter
- 在.NETCF中只支持XmlMessageFormatter
- 你可以发送支持XML Serialization的任何对象

```
MyCustomObject myObj = new MyCustomObject();  
myMessageQueue.Send( myObj );
```

- 在接收端, 你必须设置 TargetTypes 或 TargetTypeNames 属性

```
XmlMessageFormatter myFormatter = new XmlMessageFormatter();  
myFormatter.TargetTypes =  
    new Type[] { typeof(MyCustomObject) };  
myMessageQueue.Formatter = myFormatter;  
Message msg = myMessageQueue.Receive();  
MyCustomObject receivedObj = (MyCustomObject) msg.Body();
```

Custom Message Formatter

- 你可以创建自定义Formatter
 - 创建一个实现了 **IMessageFormatter** 接口的类
 - **IMessageFormatter** 只有三个 public methods
 - Write
 - Read
 - CanRead
 - **IMessageFormatter** 也实现了 **ICloneable**, 因此必须实现 Clone()
 - 在你的自定义Formatter, 你可以
 - 加密和解密你的 message body
 - 实现压缩
 - 详细信息查看 <http://support.microsoft.com/default.aspx?scid=kb;EN-US;310683>

MSMQ Protocols

- MSMQ 1.0, 2.0 只支持二进制
- MSMQ 3.0 (Windows XP, Windows Server 2003) 开始支持HTTP/XML transport
- Windows CE 中的HTTP/XML 支持：
 - Windows CE 4.0
 - 在 Pocket PC 2003 中不支持
 - HTTP transport 需要 MSXML3, 但是 Pocket PC 2003 为了节省 ROM, 装载了MSXML2
 - Windows Mobile 5.0
- HTTP 使用 Soap Reliable Messaging Protocol (SRMP)
 - 设置注册表键值
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\SimpleClient\SrmpEnabled to “Yes”
 - 或者使用msmqadm.exe
 - `msmqadm enable srmp`

MSMQ Over HTTP

- Direct format names 可以包括URL到队列
 - `FormatName:Direct=HTTP://URLAddressSpecification/msmq/private$/myQueue`
 - Note HTTP format names 可以使用 forward slashes (/) 或者 back slashes (\) 来分割 Message Queuing 信息, 但是你可以在 URLAddressSpecification/msmq 之后只使用 back slash (\)
 - 例如, 下列两个示例都是有效的:
 - `FormatName:Direct=HTTP://URLAddressSpecification/msmq/private$/myQueue`
 - `FormatName:Direct=HTTP://URLAddressSpecification/msmq\private$\myQueue`
- HTTP 允许 Windows CE 客户端来访问 public 队列
 - `FormatName:Direct=HTTP://URLAddressSpecification/msmq/myQueue`
 - 不需要通过 Falcon Routing Server 路由, 不像 native protocol
- 使用 HTTPS 来加密通讯

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Accessing Remote Queues With MSMQ

demo

可靠性与性能

Persistent message store

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 队列信息和消息存储使用的位置
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\SimpleClient\BaseDir
 - 默认为 \Temp\MSMQ
- 改变为持久存储, 避免hard reset时消息丢失
 - \Storage Card compact flash 或 SD card
 - Smartphone和未来的 Pocket PC中的 \Storage flash memory
 - 使用慢速存储介质会损失性能

可靠性与性能

Recoverable messages

- **Message.Recoverable** 属性

```
// Build message
Message m = new Message();
m.Body = orderObject1;

m.Recoverable = true; // guaranteed delivery
```

- **True:** 消息的传递是有保证的（通过在路由的硬盘中保存消息的方式）
 - 速度慢，消耗更多资源
 - 在系统故障的情况下，消息可以被重新获得
- **False:** 不确定消息是否被交付（默认）
 - 速度快，消耗较少资源
 - 消息有丢失的可能，如果运行队列的计算机发生故障

可靠性与性能 Transactions

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Recoverable消息是保证会被传递到的, 但是
 - 也许会被多次传递
 - 也许会被打乱传递顺序
- 使用事务消息队列
 - 保证只传递一次, 并按顺序传递
 - Windows CE MSMQ只支持Single Internal Transactions; 不支持外部 (MTS/COM+)
- 创建Transactional Message Queue(Server或Device)

```
// Create transactional message queue  
MessageQueue mq = new MessageQueue(@".\Private$\TranQ", true);
```

- 发送消息到事务队列

```
Message m = new Message();  
m.Body = orderObject1;  
mq.Send(m, MessageQueueTransactionType.Single);
```

可靠性与性能

Message Timeouts

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

Property	Description
Message.TimeToBeReceived	设置消息在中目标队列读取的时间限制 保证所有计算机有正确的时间，否则引起不确定的结果
Message.AdministrationQueue	定义非事务队列的名称，用于接收MSMQ acknowledgement消息(如果需要) Acknowledgement queue必须存在
Message.AcknowledgementType	定义所需要的acknowledgement 类型： NegativeReceive, PositiveReceive 等.
Message.UseDeadLetterQueue	false – 未送达的消息将被丢弃 true –未送达的消息拷贝将被发送到Dead Letter 队列 Dead Letter 队列将被调用 machine\Deadletter\$

可靠性与性能

Message Timeouts Example - 1

```
// Create an administration queue
MessageQueue.Create(@".\private$\myAdministrationQueue");

// Create a new message.
Message myMessage = new Message("Original Message");
myMessage.AdministrationQueue =
    new MessageQueue(@"PPCDEVICE\private$\myAdministrationQueue");
myMessage.AcknowledgeType = AcknowledgeTypes.FullReceive;
myMessage.TimeToBeReceived = TimeSpan.FromSeconds(10.0);

// Create MessageQueue object for a queue on a remote computer.
MessageQueue myQueue =
    new MessageQueue(@"CLIMBER\private$\myQueue");
myQueue.Send(myMessage);

// save the message Id of the sent message
string messageId = myMessage.Id;
```

可靠性与性能

Message Timeouts Example - 2

```
// save the message Id of the sent message
string messageId = myMessage.Id;

...

// Process acknowledgement
MessageQueue admQueue =
    new MessageQueue(@".\private$\myAdministrationQueue");
admQueue.MessageReadPropertyFilter.CorrelationId = true;
admQueue.MessageReadPropertyFilter.Acknowledgment = true;

// Wait 30 seconds for an acknowledgement
Message myAcknowledgmentMessage = admQueue.
    ReceiveByCorrelationId(
        myMessageId, TimeSpan.FromSeconds(30.0));

// Output acknowledgment message information.
MessageBox.Show("Acknowledgment Type: " +
    myAcknowledgmentMessage.Acknowledgment.ToString());
```


安全性

- Windows CE MSMQ不支持
 - 用于authentication的数字签名
 - Message encryption
 - 消息加密
 - 使用证书的Client authentication
- 安全的HTTP transactions
 - 使用HTTPS加密transport channel
 - 在Windows Server中, 配置IP地址限制来限制允许访问的IP地址
 - 可以使用自定义formatter 在消息中嵌入自定义的credentials
 - Native MSMQ API允许包含自定义的SOAP headers
- Windows CE MSMQ对Denial of Service (DoS)攻击提供保护
 - 当消息大小超过定义在注册表中的限制, 将拒绝通过HTTP发送的基于SOAP的消息
 - DefaultQuota, DefaultLocalQuota, MachineQuota

NETCF Messaging Comparison

	HTTP Support	Handling Network Availability	Security	Ease of development	Server configuration
MSMQ	✓ ✓ ¹	✓ ✓ ✓	✓ ✓	✓ ✓ ✓	✓ ✓
Web services	✓ ✓ ✓	✓ ³	✓ ✓ ✓	✓ ✓	✓ ✓ ✓
Sockets	✓ ✓ ²	✓ ³	✓	✓	✓ ✓ ✓
SQL Mobile RDA	✓ ✓ ✓	✓ ³	✓ ✓ ✓	✓ ✓	✓

Notes:

1. MSMQ HTTP 不支持 Windows Mobile 3.0
2. System.Net.HttpWebRequest\WebResponse 类使用sockets
3. Web services, sockets 和 SQL Mobile 全部需要开发者检测和响应网络的连接状态

Summary

- MSMQ提供一套易用、直观的API
- 在分布式系统中，理想的松耦合通讯方式
- 在Windows CE中最容易使用的消息传递技术
- Windows Mobile 5.0提供的HTTP support 扩展了对Internet-connected Pocket PCs/Smartphones的支持
- 安全性被限制于基本功能，尽管可以通过自定义formatter来改善

Useful Tools

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Cambridge Computer Corporation vxUtil
 - Ping, DNS, Route etc...
 - Other tools...
 - <http://www.cam.com/windowsce.html>
- Marc Zimmerman's Pocket Hosts
 - Create hosts in registry
 - Other tools...
 - <http://www.zimac.de/cestuff.htm>
- Visual Studio 2005 Device Command Shell
 - Visual Studio "Command Window" add-in
 - Supports remote device commands such as
 - Copying files
 - Starting/stopping/listing processes
 - Working with the device registry
 - Remote XML provisioning and more
 - <http://www.goldotnet.com/workspaces/workspace.aspx?id=50618f79-c7b1-4588-9c0a-cf4ddae8092a>

References

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- Microsoft Message Queuing Center
<http://www.microsoft.com/windows2000/technologies/-communications/msmq/default.asp>
- Mark Ihimoyan's Blog: System.Messaging (MSMQ) in .NET CF 2.0
<http://blogs.msdn.com/ihimmar>
- C Sharp corner article:
MSMQ on Pocket PC 2003
<http://www.c-sharpcorner.com/Code/2004/Sept/-MSMQonPocketPC2003.asp>
- MSDN Magazine:
MSMQ AND .NET Send MSMQ Messages Securely Across the Internet with HTTP and SOAP
<http://msdn.microsoft.com/msdnmag/issues/03/12/-MSMQandNET/default.aspx>


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



Microsoft
微软(中国)有限公司

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