

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 谁动了我的 NET CF 应用的性能?

张欣  
中软国际



## 本次课程内容包括

- 引言
- 分析Windows Mobile设备上的程序性能
- 远程性能监视器
- .NET CF CLR Profiler
- 总结

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

## 收听本次课程需具备的条件

- 具有一定的.NET开发经验
- 了解Windows Mobile设备开发

# Level 300

# 引言

- 什么是性能?
- 用户对性能的度量手段?



延迟

完成时间

# 引言

- 性能非常重要

- 没有人因为您的程序有着良好的性能而向您祝贺
- 每个人都会因为您的程序性能不好而抱怨 ☹

- 应用程序的性能应该是一种需求

- 常见的一些想法

- 托管程序太慢
- 我不相信垃圾回收机制
- C#的解析器和C++的一样
- 托管程序中永远不需要考虑释放内存

# Mobile设备上的性能

- 为什么需要我們考虑性能
  - 低速的CPU
  - 小内存
- 更多的考虑一下我们的程序
  - 少做不必要的运算
    - 维护电量
  - 程序的大小
    - 程序小一点更好

# 影响性能的地方

- 程序加载时间
  - 程序集的大小
  - 窗体上控件的多少
- JIT 编译时间
  - 第一次调用某一个方法
  - 方法的后继调用
- 垃圾回收
  - 内存分配失败
  - 分配超过1M的内存
  - 程序转到后台
  - 调用 GC.Collect()

## Tips

-  限制每一次操作的计算量
-  String vs. StringBuilder
-  了解装箱/拆箱
-  集合 vs. 泛型集合
-  控制P/Invokes调用的数量
-  虚方法调用 vs. 非虚方法调用
-  使用简单属性
-  小心调用 GC.Collect
-  知道何时需要实现Finalizers

您的潜力. 我们的动力

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

# DEMO

一个性能差的程序的例子

# 性能分析

- 移动设备程序的性能监测在 Visual Studio 2005 中并不被直接支持
- 如果您的程序可以直接运行在桌面环境上的话可以使用 Visual Studio 2005 中的性能分析器
- 使用 Stopwatch 对象来度量方法的执行时间
  - `System.Diagnostics.Stopwatch` 在 .NET CF 中不被支持
  - 那就自己建立一个吧 ☺

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# Visual Studio 2005性能分析器

The screenshot displays the Visual Studio 2005 Performance Analyzer interface. The left pane shows the 'WinFormTest' project structure with folders for '目标' (Targets) and '报告' (Reports). The right pane displays a '性能报告摘要' (Performance Report Summary) for the 'WinFormTest071012(1).vsp' file. The report includes two tables: '最频繁调用的函数' (Most frequently called functions) and '大部分时间单独工作的函数' (Functions that work alone for most of the time). The bottom of the window shows navigation buttons: '< 上一步(B)' (Previous), '下一步(N) >' (Next), '完成(F)' (Finish), and '取消' (Cancel).

### 性能报告摘要

#### 最频繁调用的函数

名称	调用数	%
...tem.Threading.Thread.Sleep	10	18.868
...FormTest.Form1.TestMethod	10	18.868
...ws.Forms.Control.set_Text	2	3.774

#### 大部分时间单独工作的函数

名称	时间 (毫秒)	%
...mTest.Form1.button1_Click	0.064180	20.150
...ws.Forms.Form.set_ClientSize	0.055778	17.512
...ws.Forms.Control.set_Name	0.052427	16.460

#### 耗时最长的函数

名称	时间 (毫秒)	%
...tem.Threading.Thread.Sleep	50479.3272...	66.827
...ndows.Forms.Application.Run	15019.6312...	19.884
...ws.Forms.MessageBox.Show	8553.458316	11.323

# Stopwatch使用的最佳实践

- 排除JIT编译的干扰
  - 在分析一个方法前先调用它一次
- 不要依赖于一次度量
  - 度量一个你要分析的方法多次，然后取其平均值
- 理解你要分析的源代码
  - 为什么这个方法需要那么长的时间？
  - 不要忘了深入调查你所监测的方法中所调用的其他方法，它们也许是影响性能的地方

您的潜力. 我们的动力

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

# DEMO

## 分析和度量

# 远程性能监视器

 运行在桌面上，连接到移动设备上

 实时显示性能计数器

 监视以下类别

- 内存和垃圾回收
- 线程和同步
- 图像和窗口
- P/Invoke调用

 使用Windows性能分析器来分析数据

 察看托管堆的快照



# 安装远程性能监视器

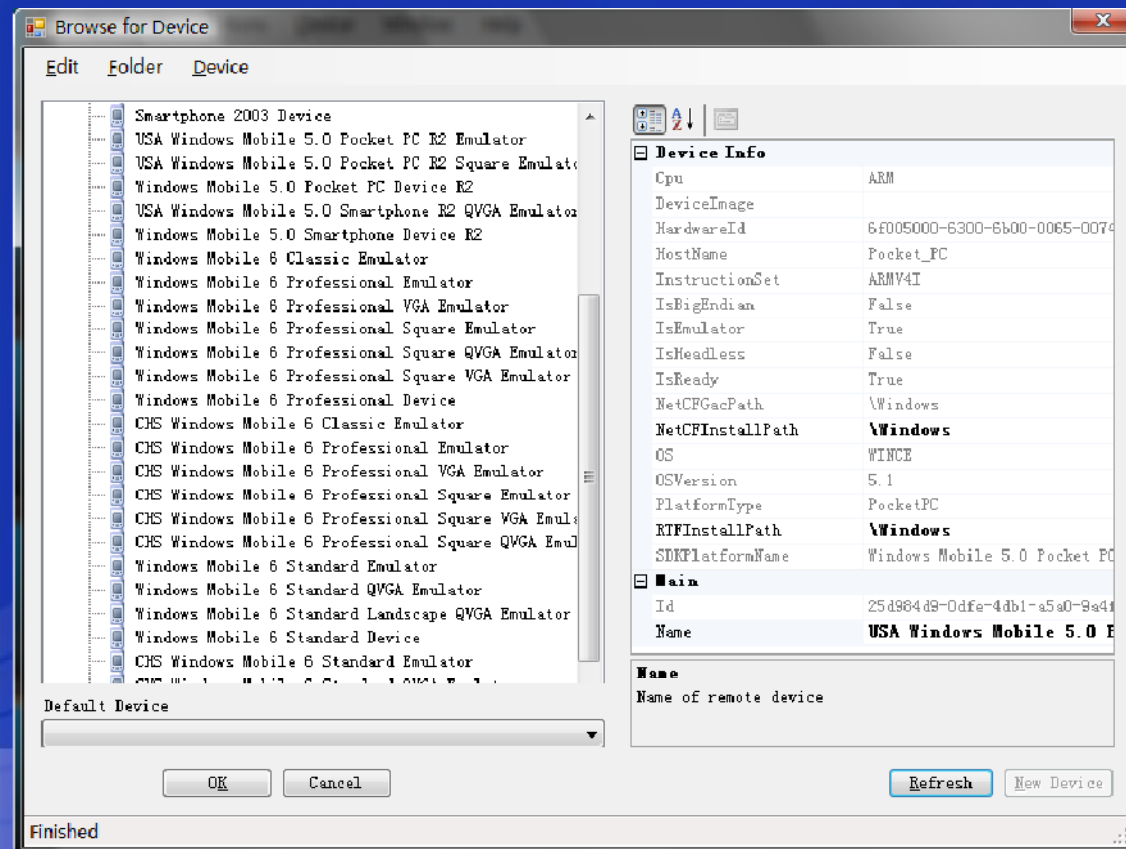
- 下载.NET Compact Framework 3.5 Power Toys
  - <http://www.microsoft.com/downloads/details.aspx?FamilyId=C8174C14-A27D-4148-BF01-86C2E0953EAB&displaylang=en>
- 包含以下组件
  - Remote Performance Monitor
  - CLR Profiler
  - Application Configuration Tool
  - ServiceModel Metadata Utility
  - Remote Logging Configuration Tool
  - Network Log Viewer

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

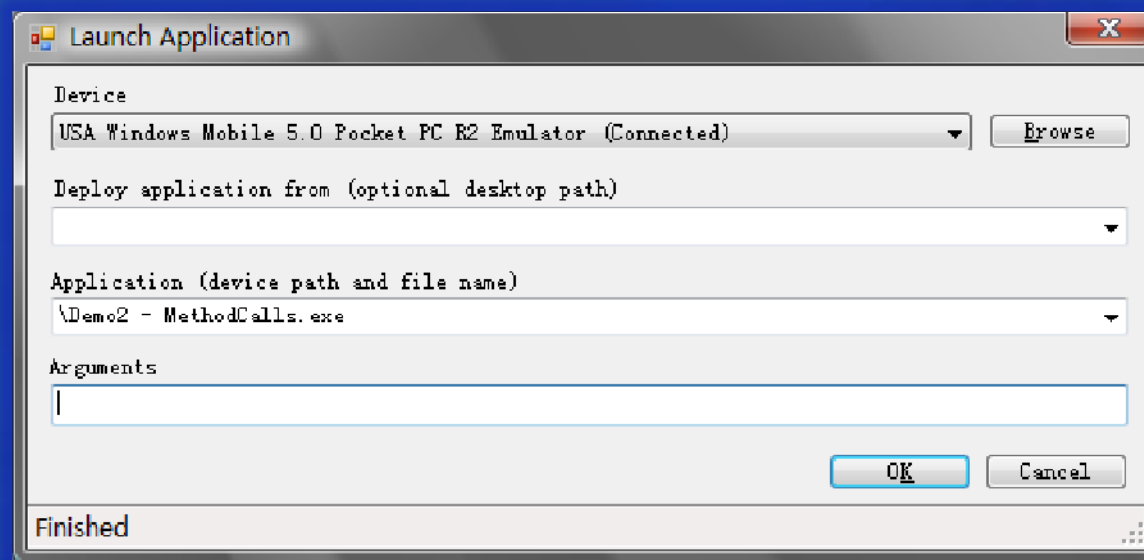
# 运行远程性能监视器

- 运行 .NET CF Remote Performance Monitor
- 连接到设备上



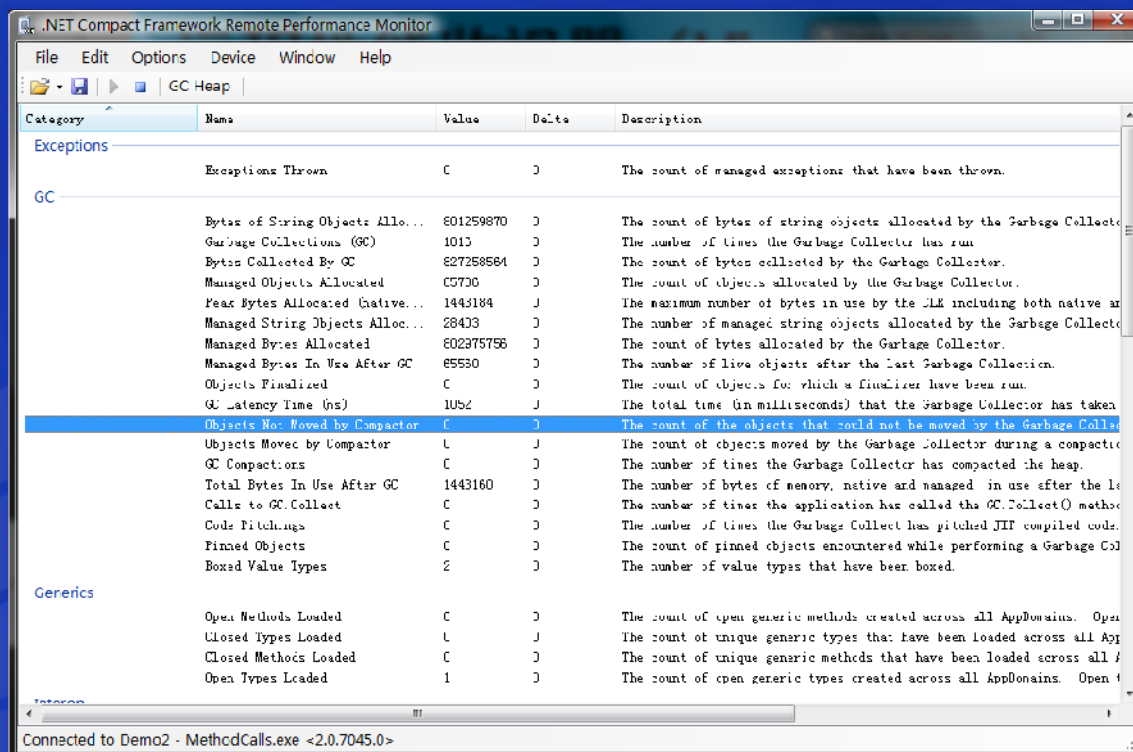
## 运行远程性能监视器（续）

- 把要监测的程序部署到设备上
- 设定要监测的程序的完全路径,然后运行



# 运行远程性能监视器 (续)

- 远程性能监视器会自动调用您的程序并监测
- 查看各项数据



The screenshot shows the .NET Compact Framework Remote Performance Monitor window. The 'GC' category is selected, displaying a list of performance metrics. The table has columns for Category, Name, Value, Delta, and Description. The 'Exceptions' category is also visible above the GC data.

Category	Name	Value	Delta	Description
Exceptions	Exceptions Thrown	0	0	The count of managed exceptions that have been thrown.
GC	Bytes of String Objects Allocated by the Garbage Collector (GC)	801259870	0	The count of bytes of string objects allocated by the Garbage Collector.
	Bytes Collected By GC	1015	0	The number of times the Garbage Collector has run.
	Managed Objects Allocated	827258564	0	The count of bytes collected by the Garbage Collector.
	Peak Bytes Allocated (Native)	50730	0	The count of objects allocated by the Garbage Collector.
	Managed String Objects Allocated	1443184	0	The maximum number of bytes in use by the JLM including both native and managed.
	Managed Bytes Allocated	28403	0	The number of managed string objects allocated by the Garbage Collector.
	Managed Bytes In Use After GC	802375756	0	The count of bytes allocated by the Garbage Collector.
	Objects Finalized	85530	0	The number of live objects after the last Garbage Collection.
	GC Latency Time (ms)	0	0	The count of objects for which a finalizer have been run.
	Objects Not Moved by Compactor	1052	0	The total time (in milliseconds) that the Garbage Collector has taken.
	Objects Moved by Compactor	0	0	The count of the objects that could not be moved by the Garbage Collector.
	GC Compactions	1	0	The count of objects moved by the Garbage Collector during a compaction.
	Total Bytes In Use After GC	0	0	The number of times the Garbage Collector has compacted the heap.
	Calls to GC.Collect	1443160	0	The number of bytes of memory, native and managed, in use after the last collection.
	Code Pitches	0	0	The number of times the application has called the GC.Collect() method.
	Pinned Objects	0	0	The number of times the Garbage Collector has pitched JIT compiled code.
	Boxed Value Types	0	0	The count of pinned objects encountered while performing a Garbage Collection.
Generics	Open Methods Loaded	2	0	The number of value types that have been boxed.
	Closed Types Loaded	0	0	The count of open generic methods created across all AppDomains.
	Closed Methods Loaded	1	0	The count of unique generic types that have been loaded across all AppDomains.
	Open Types Loaded	0	0	The count of unique generic methods that have been loaded across all AppDomains.
		1	0	The count of open generic types created across all AppDomains.

Connected to Demo2 - MethodCalls.exe <2.0.7045.0>

您的潜力. 我们的动力

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

# DEMO

## 远程性能监视器



**MSDN Webcasts**

# 托管代码不会造成内存泄露？

- 垃圾回收仅仅回收无根对象
  - 大多数对象在跳出程序作用域后就不可达
- 常见的内存泄露场景
  - Event handlers
  - 静态变量

# Event Handler的内存泄露

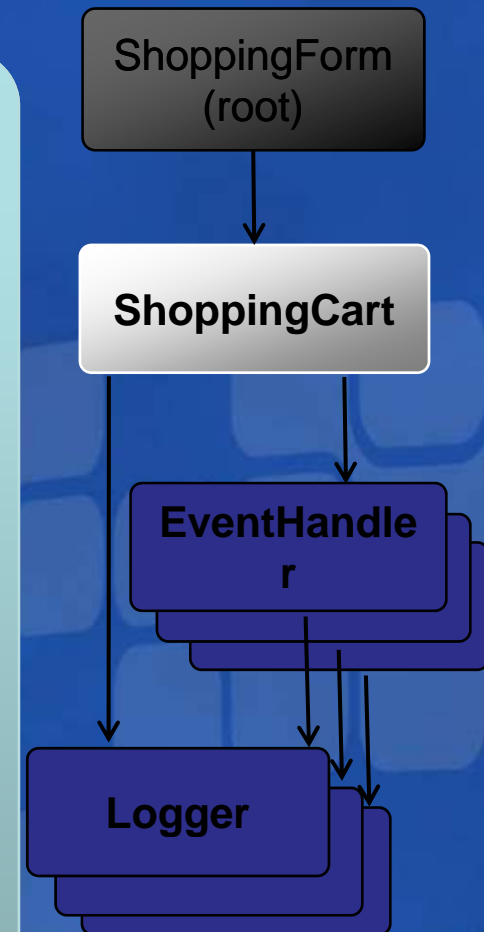
```
public class ShoppingCart
{
    public event EventHandler<EventArgs> OnChange;

    public void ProcessCart()
    {
        Logger l = new Logger();

        // hook up event handler
        this.OnChange +=
            new EventHandler<EventArgs>(l.OnChangeHandler);

        // process order....
    }

    public class Logger
    {
        public void OnChangeHandler(object sender, EventArgs
e)
        {    // handle log event ... }
    }
}
```



# 修正Event Handler的内存泄露

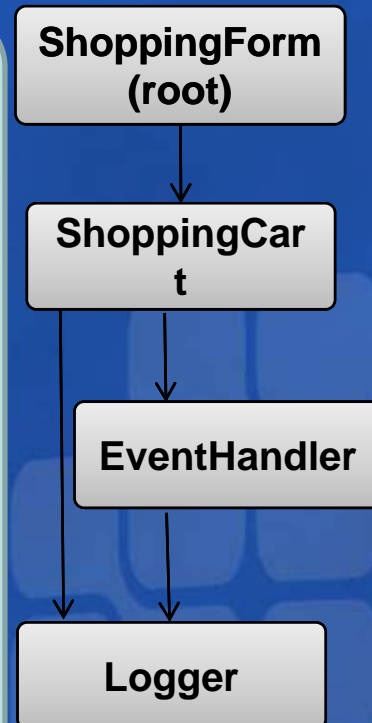
```
public class ShoppingCart
{
    public event EventHandler<EventArgs> OnChange;

    public void ProcessCart()
    {
        Logger l = new Logger();

        // hook up event handler
        this.OnChange +=
            new EventHandler<EventArgs>(l.OnChangeHandler);

        // process order....

        // detach event handler
        this.OnChange -=
            new EventHandler<EventArgs>(l.OnChangeHandler);
    }
}
```



您的潜力. 我们的动力

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

# DEMO

## 内存泄露

# 解析函数

- 如果只是一个简单的托管对象，**Dispose**和解析函数对其用处不大。
- 如果使用了非托管资源（例如图像、网络等），只是使用解析函数无法有效释放资源。
  - **GC**并不会立即释放解析函数中的资源
  - 应该使用**Dispose**模式来通知**GC**释放资源
- 解决方案
  - 实现 **Dispose**
  - 调用 **object. Dispose**

您的潜力. 我们的动力

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

# DEMO

## Dispose

# 总结

- 应用程序的性能非常重要，它甚至应该列入功能需求
- 性能在很大程度上取决于你的内存分配/使用
- 善于利用工具来分析和诊断您的程序的性能

## 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>

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

## 我的Blog

- **MobileSide.cn – Your Mobility! Our Passion!**

<http://MobileSide.cn>

- 我的回收站@cnblogs


<http://twodays.cnblogs.com>

## 联系我们

- 您有任何关于**Windows Mobile**的建议和想法, 或关于技术的问题, 请发送邮件给我们 (**WindowsMobility@hotmail.com**), 这里将有**Mobile**技术专家为您解答。
- 订阅 **Windows Mobile** 新闻稿, 并了解从基于 **Windows Mobile** 的 **Pocket PC**、**Smartphone** 和 **Portable Media Center** 中获得最大利益! 此新闻邮件, 每月发行一次, 重点介绍有关新设备、提示与技巧、实用下载和特价优惠信息, 以及可帮助您借助 **Windows Mobile** 提高效率的其他资源。  
<http://www.microsoft.com/china/windowsmobile/newsevents/newsletters.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**