

基于 WMI 的计算机管理技术的研究与实现

宋 昕¹, 盛 晨², 王新华²

(1. 浙江科技学院 信息与电子工程学院, 杭州 310023; 2. 武警杭州指挥学院 计算机教研室, 杭州 310023)

摘 要: 随着计算机及网络技术的飞速发展, 实时高效安全地管理计算机, 更好地共享资源的各种技术, 已经成为人们研究的重点。为此, 针对传统 C/S 管理模式存在的问题, 提出了采用 WMI 技术实现对计算机进行高效管理的方法, 并给出获取远程计算机信息和实现使用 WMI 远程控制计算机的具体技术措施。

关键词: WMI; C/S; 远程控制; WMI 脚本; VB. NET

中图分类号: TP315

文献标识码: A

文章编号: 1671-8798(2007)01-0023-04

Development of WMI-Based Computer Management Technology

SONG Xin¹, SHENG Chen², WANG Xin-hua²

(1. School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China;
2. Teaching and Research Section of Computer, Hangzhou Command College of Armed Police Force, Hangzhou 310023, China)

Abstract: With the rapid development of the computer and network technology, it has become important to do research on how to manage computer efficiently and safely in real time and on how to share various techniques more conveniently. Aiming at the problem of the traditional management of C/S, we put forward the method which adopts a WMI technique to carry on computer managing efficiently, and also provide the concrete realization on obtaining information from the remote computers and the remote control of computers.

Key words: WMI; C/S; remote control; WMI script; VB. NET

随着计算机技术和网络技术的飞速发展, 计算机应用已经广泛普及到各行各业, 同时, 为了更好地共享使用设备, 对计算机管理也提出了越来越高的要求。传统的管理方法是采取严格的 C/S 模式, 客户机必须运行特定的终端程序, 时刻监听服务器才能保证客户机实时响应^[1]。如果客户机因为某种原因退出终端程序, 那么服务器就无法管理该客户机。因此, 传统管理计算机的方法不仅增加了客户端的

系统开销, 而且对各种故障难以检测, 其管理效率很难得到较大改善。本文首先介绍一种高效管理计算机的技术——WMI(Windows Management Instrumentation)——Windows 管理规范^[2], 然后提出一种基于 WMI 技术对计算机进行高效管理的方法, 并给出获取远程计算机信息和实现使用 WMI 远程控制计算机的具体方法, 而且结合应用实例说明这种方法的有效性。

收稿日期: 2007-02-03

作者简介: 宋 昕(1982—), 女, 浙江金华人, 助教, 主要从事数据库、软件工程研究。

1 WMI 技术概述

WMI 是 Microsoft 基于 Web 的企业信息管理 (WBEM) 的实现, 同时也是一种基于标准的系统管理接口。WMI 技术是微软提供的 Windows 下的系统管理工具, 通过该工具可以在本地管理客户端系统中几乎一切的信息, 很多专业的网络管理工具都是基于 WMI 开发的。WMI 是一种轻松获取系统信息的强大工具, 信息源可能是一个硬件、操作系统或软件应用程序, 数据源提供的信息称为规范。

WMI 作为 Windows 2000 操作系统的一部分, 提供了可伸缩、可扩展的管理架构。公共信息模型 (Common information model, CIM) 是由分布式管理任务标准协会 (DMTF) 设计的一种可扩展的、面向对象的架构, 用于管理系统、网络、应用程序、数据库和设备。Windows 管理规范也称作 CIM for Windows, 提供了统一的访问管理信息的方式。利用 WMI, 可以监视、跟踪和控制有关软件应用程序、硬件组件和网络的系统事件, 将来自不同来源的数据用通用、标准且逻辑上有组织的方式映像出去, 以便在管理数据之间建立相互关系和关联, 而不必考虑这些数据的类型、内容或来源。

使用 WMI 可以获得统一的编程 API, 远程管理, 可发现和导航, 过滤查询, 强大的事件发布和订阅等功能。它由管理架构、WMI 提供者、WMI 使用者、CIM 储存库和 CIMOM (Common information model object manager, 公共信息模型对象管理器) 五部分组成^[2]。在现代计算机管理中, 系统管理员更为关注的是监视应用程序的运行状况、检测瓶颈或故障、管理和配置应用程序、查询应用程序数据、执行无缝的本地或远程管理等操作, 并能在最短时间里最有效率地处理或规避上述问题。而 WMI, 即: 视窗管理规范技术提供了一个统一的、基于标准的、可扩展的面向对象接口, 除此之外, WMI 还提供了与系统管理信息和基础 WMI API 交互的标准方法, 这些标准都是本例中基于 WMI 的计算机管理技术平台实现的基础。

2 基于 WMI 的计算机管理技术的实现

笔者在研究与开发学校机房管理系统的过程中, 充分利用了 WMI 技术在计算机管理方面的强大功能, 采用 Web 形式对机房和实验室设备进行综合管理。本文给出了基于结合 WMI 和 Web 服务

技术来获取远程计算机信息, 并且实现使用 WMI 远程控制计算机, 其他的管理功能如监视应用程序的运行状况、管理和配置应用程序等, 均可采用类似的方法实现。本例的系统结构如图 1 所示。

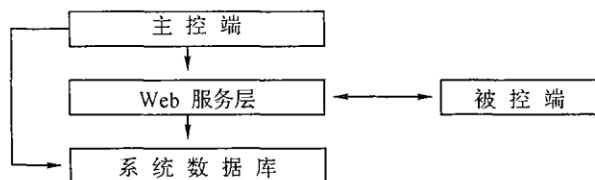


图 1 系统结构示意图

在图 1 的三层结构中, 主控端是指本系统的使用者与 Web 服务层的应用接口, 即可视化操作界面。Web 服务层在本例中并不是单纯地提供 Web 服务 (作为用户和数据库的一个中间接口层, 负责响应用户的请求, 并把实际的数据库操作结果反馈给用户), 而更为关键的是它还负责实现了系统中的管理逻辑, 并且有效地解决了主控端和被控端的通信问题。

2.1 使用 WMI 获取远程计算机信息

现在的计算机以及网络组成十分复杂, 例如系统硬件方面就有主板、硬盘、网卡等; 软件方面有操作系统、系统中安装的软件、正在运行的进程等; 网络方面有域、工作组等等。对这些信息的收集和数据的处理是现代计算机管理中非常重要的任务。WMI 的功能十分强大, 可以管理 Windows 的方方面面, 从常见的文件操作到硬件管理, 都可以通过 WMI 来实现。以下介绍笔者使用 WMI 和 .NET 框架一起来获取远程计算机软硬件的实现方法。

首先需要引入 System. Management 命名空间。在 .NET 中, 使用 System. Management 命名空间, 通过对 class 的操作, 可以查询系统软硬件的信息。此命名空间是 .NET 框架中的 WMI 命名空间, 它提供的类对象为访问各种管理对象提供了面向对象的编程接口^[3], 并包括下列支持 WMI 操作的第一级类对象:

- 1) ManagementObject 或 ManagementClass: 分别为单个管理对象或类;
- 2) ManagementObjectSearcher: 用于根据指定的查询或枚举检索 ManagementObject 或 ManagementClass 对象的集合;
- 3) ManagementEventWatcher: 用于预订来自 WMI 的事件通知;
- 4) ManagementQuery: 用作所有查询类的基础。

System. Management 类的使用编码范例对 .NET 框架环境很适合,并且 WMI 在任何适当的时候均使用标准基框架。应用程序和服务可以通过使用从 ManagementObjectSearcher 和 ManagementQuery 派生的类,查询其感兴趣的管理信息(例如在磁盘上还剩多少可用空间、当前 CPU 利用率是多少、某一应用程序正连接到哪一数据库等等);或者它们可以通过使用 ManagementEventWatcher 类来预订各种管理事件。

笔者在学习了上述相关理论知识后,了解在 .NET 中如何与 WMI 进行通讯以便达到获取信息的目的,并着手编写了一些利用 System. Management 下面的类来获取系统相关信息的代码,在此做一简单介绍。

2.1.1 获取软件信息 以下是一段列出所有系统当前共享目录的代码:

```
Imports System. Management
Dim searcher As New ManagementObjectSearcher("SELECT * FROM Win32_share")
Dim share As ManagementObject
For Each share In searcher.Get()
    MessageBox.Show(share.GetText(TextFormat. Mof))
Next share
```

上述代码采用 WQL(WMI Query Language) 语句^[4],也就是标准 SQL 的一个子集加上了 WMI 的扩展,进行 SQL 操作,代码简单、通俗易懂。由于 WQL 是个只读的查询语言,只能查询响应的数据,不能用 UPDATE,INSERT 等更新操作,并且采用了一种 MOF(Meta Object Facility),托管对象格式的显示。

2.1.2 获取硬件信息 以下是一段获取主板的序列号以及制造商的代码:

```
Dim searcher As New ManagementObjectSearcher("SELECT * FROM Win32_BaseBoard")
Dim share As ManagementObject
For Each share In searcher.Get()
    Debug.WriteLine("主板制造商:" & share("Manufacturer"))
    Debug.WriteLine("主板型号:" & share("Product"))
    Debug.WriteLine("主板序列号:" &
```

```
share("SerialNumber"))
```

Next share

此段代码同样采用 WQL 语句实现。同理,还可获取网卡硬件地址、硬盘 ID 和 CPU 序列号等信息,在此就不一一列举。

2.2 使用 WMI 远程控制计算机

WMI 提供了大量的规范以便为许多高端应用程序(例如 Microsoft Exchange、Microsoft SQL Server 和 Microsoft Internet 信息服务(IIS))实现几乎任何管理任务。管理员可以执行下列任务:

- 1) 监视应用程序的运行状况;
- 2) 检测瓶颈或故障;
- 3) 管理和配置应用程序;
- 4) 查询应用程序数据(使用对象关系的遍历和查询);
- 5) 执行无缝的本地或远程管理操作。

一般对计算机的控制主要包括:重新启动计算机、关闭计算机、注销当前用户、启动或终止某个进程(某项服务)。

使用 WMI 实现远程控制的最简单方法是编制 WMI 脚本程序,然后在命令行中运行该脚本,其命令格式为:Cscript. exe 脚本文件名,Cscript. exe 是执行脚本的宿主程序,当运行该程序时,它自动执行作为输入参数的脚本文件。

笔者通过编写 WMI 脚本程序来实现计算机远程控制的过程中发现,编写 WMI 脚本的很大一部分工作涉及到读取和设置属性值,WMI 脚本的语法比较简单,编制起来较容易。脚本程序在任何一个具有 WSH(Windows Script Host)环境的系统中都可以运行,由于它不依赖于特定的开发工具和运行环境,因此具有良好的可移植性。只要在脚本中指定了要控制的远程计算机名,且该计算机是可访问的,同时拥有远程系统的管理员权限,则在任何时候都可以对它进行远程控制。相对于传统的 C/S 控制模式,这种方法显得相当简单和可靠。

对比传统的 C/S 控制模式,人们需要能更高效地管理计算机,因此笔者又尝试在软件中实现远程控制功能,发现同样通过简单的代码即可实现。以下介绍笔者在 .NET 环境下远程控制计算机的实现方法。

以远程关闭计算机为例(VB. NET^[5]编写的核心代码):

```
Dim scope As ManagementScope = New ManagementScope("\\\" + RemoteHostName + "\
```

Root\CIMV2", options) //以给定的管理员帐号连接给定计算机名或者 IP 地址的计算机

```
Try
    scope.Connect() //连接远程计算机
    Dim oq As System.Management.ObjectQuery
    oq = New System.Management.ObjectQuery
    ("Select * From Win32_OperatingSystem")
    //调用远程计算机进行 WMI 操作
    Dim gwo As ManagementObjectSearcher
    gwo = New ManagementObjectSearcher
    (scope, oq)
    //从已经完成远程连接的计算机中,得到 WMI 操作集合
    Dim pwo As ManagementObjectCollection
    pwo = gwo.Get() //存放得到 WMI 操作
    Dim mo As ManagementObject //定义 WMI 操作对象
    For Each mo In pwo
        Dim ss() As String = {" "}
        mo.InvokeMethod("Shutdown", ss)
    Next //调用 WMI 控制,实现关闭远程计算机
    Console.WriteLine("指定的设备已经被关闭!")
Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.WriteLine("不能与指定设备建立连接,设备已经关闭或出现网络故障!")
```

End Try

当触发某个事件,执行上述代码后,就实现了关闭远程计算机的功能,不需要管理员一一去关闭每台远程计算机。这种方法同样不需要在被控制的计算机上运行任何终端程序,只要该计算机是可访问的且拥有该计算机的管理员权限,即可被远程控制。根据此例中的程序稍做修改就可得到远程重启计算机等其他一系列控制程序来进行远程控制。

3 结 语

与传统 C/S 模式相比,使用基于 WMI 的计算机管理技术将会带来更可靠的计算环境和更高的系统可靠性,提高了管理的效率,从而节省了系统的开销。WMI 技术的应用把计算机管理推向了一个新的境界,在当今高度信息化的时代,只有充分利用计算机和网络的先进技术,才能使计算机的管理日趋实时高效。

参考文献:

- [1] 张茜,张银福. 基于 Web 服务器的设备管理技术研究
与实现[J]. 计算机工程与应用, 2003, 39(24): 169-171.
- [2] MARCIN Policht. WMI 技术指南[M]. 智慧东方工作室, 译. 北京:机械工业出版社, 2002.
- [3] 姜边. 一种基于 .NET 元数据的对象持久化框架[J]. 计算机应用, 2005, 25(4): 833-836.
- [4] DAVID Chappell. .NET 大局观[M]. 2 版. 荣耀, 译. 北京:电子工业出版社, 2006.
- [5] BILL Evjen, BILLY Hollis. VB.NET 高级编程[M]. 3 版. 杨浩, 杨智, 译. 北京:清华大学出版社, 2005.