

基于 Windows 脚本技术的计算机信息采集方案实现

刘 珊 (北京市丰台区职工大学 计算机教研室 北京 100161)

摘 要: 使用 Windows 脚本编程,并结合数据库实现 C/S 模式下的网络内计算机设备信息的获取,展示了在网络环境下 Windows 脚本结合数据库编程技术的强大功能,善用 Windows 脚本编程技术将提高系统管理员日常管理和维护效率。

关键字: Windows 脚本; VBS; WSH; WMI; 数据库

Implementation of Information Acquisition Solution Based on Windows Script

LIU Shan(Teaching and Research Division for Computer Science, Employees' College of Fengtai in Beijing, Beijing 100161, China)

Abstract: This paper introduces a database-supported information acquisition application and its implementation using Windows Script programming. It proves the strong functionalities the database-supported Windows Script application could provide in web development environment. This technique could also facilitate daily maintenance for systems and increase working productivity.

Keywords: Windows script; VBS; WSH; WMI; database

1 引言

现在,计算机设备已成为人们日常办公、与其他部门沟通协作的基本工具,大量的计算机设备分散到企业的各部门办公场所,计算机设备科学化管理显得越来越重要。大多数的单位可能会使用软件进行设备的管理,但设备信息的采集无疑是一个费时、费力,还有可能出错的工作。另外,由于计算机设备的特殊性,常常会有硬件配置的更新,如何确保设备信息的准确性是我们系统管理人员面临的一个问题。

借助编程工具,有多种途径来解决这个问题。在本文中采用一种简单易行的方式,使用 Windows 脚本编程和数据库技术来实现计算机硬件相关信息的获取。

2 Windows脚本技术简介

在 Windows 操作系统中内置了 Microsoft VBScript(即 VBS) 和 JScript 脚本引擎。在早期,脚

本广泛地用于 Web 开发方面,这是因为早期脚本是在客户端的 IE 浏览器、服务器的 IIS 这样的环境中运行的。但自从 Windows98 之后,在 Windows 操作系统中内置了 WSH(Windows Script Host),即 Windows 脚本宿主环境,在客户端上使用脚本编程成为可能。WSH 从最早的 1.0 版直至今在 5.6 版本,WSH 脚本宿主环境变得越来越成熟,应用越来越广泛^[1-3]。

WSH 是一种基于面向对象的编程语言环境,其内置对象封装了大量的 Windows API 函数,构成了以根为 Wscript 对象、包含 14 个对象组成的 Windows 脚本宿主对象模型。利用 WSH 可以轻松地完成一系列繁琐的工作,例如,批量进行文件改名、删除;映射网络驱动器、打印机;访问注册表等等。还可以结合强大的 COM 组件技术控制 OFFICE 成员对象,结合 ODBC 技术访问各类数据库等等^[3]。

当执行脚本文件时,Windows 操作系统会启动 Windows 脚本宿主,由 Windows 脚本宿主调用与

收稿时间:2010-01-20;收到修改稿时间:2010-02-11

之相关的脚本引擎来解释，并运行该脚本文件。就 VBScript 和 JScript 脚本而言，VBScript 更受编程人员的喜爱。VBScript 是 Visual Basic 的一个抽象子集，是一个高效、强大而易于学习的编程语言，最方便的是它可以不需要专用的开发环境，使用记事本工具就可以编写 VBS 脚本并直接执行。

3 设计思路及实现

为了实现计算机硬件及相关信息的采集，解决方案可分为服务器和客户端两部分。服务端使用 MS SQL Server 数据库储存数据，而客户端使用 Windows 脚本编程来获取本机硬件及相关数据信息，并通过网络将数据信息递交数据库服务器上。下面，分别进行服务器和客户端的设计。

3.1 数据库结构设计

使用网络中的一台机器作为服务器，在服务器上安装 MS SQL Server 数据库服务器。使用 SQL Server 企业管理器，注册并连接 SQL Server 数据库服务器，从【工具】菜单上选择【SQL 查询分析器】，执行下面生成数据库及表结构的 SQL 语句。

```
/*创建数据库 DB*/
CREATE DATABASE [DB] ON (NAME = N'DB_Data',
FILENAME = N'D:\DB_Data.MDF') LOG ON (NAME =
N'DB_Log', FILENAME = N'D:\DB_Log.LDF')
GO
use [DB]
/*在数据库 DB 中创建 ComputerConfig 表*/
CREATE TABLE [dbo].[ComputerConfig] (
    [ID] [bigint] IDENTITY (1, 1) NOT NULL ,
    [MAC] [nvarchar] (50) NULL ,
    [Memory] [nvarchar] (50) NULL ,
    [NetworkCard] [nvarchar] (100) NULL ,
    [ComputerName] [nvarchar] (50) NULL ,
    [Account] [nvarchar] (50) NULL ,
    [OSName] [nvarchar] (100) NULL ,
    [UserName] [nvarchar] (20) NULL ,
    [Number] [nvarchar] (50) NULL ,
    [Dept] [nvarchar] (50) NULL ) ON [PRIMARY]
ALTER TABLE [dbo].[ComputerConfig] ADD
    CONSTRAINT [PK_ComputerConfig] PRIMARY
KEY CLUSTERED
```

([ID]) ON [PRIMARY]

执行完上面的 SQL 脚本，在数据库服务器上将会创建一个名为“DB”的数据库，D 盘根目录下亦生成对应的物理文件 DB_Data.MDF 和 DB_Log.LDF。在“DB”数据库中创建了一个用于存储计算机硬件及相关配置信息的表“ComputerConfig”，如图 1 所示。该表共设计有 10 个字段，用于存储计算机的 MAC 地址、计算机名、计算机的物理内存配置大小、网卡的型号、计算机安装的操作系统的名称、计算机用户账号、以及计算机存放的物理位置、使用者和所属部门。当然也可以根据实际情况增减这些信息，例如还可以采集计算机的硬盘配置情况、IP 和 DNS 等等。

列名	数据类型	长度	可为空值	说明
ID	bigint	8	NOT NULL	ID
MAC	nvarchar	50	NULL	MAC地址
Memory	nvarchar	50	NULL	计算机内存
NetworkCard	nvarchar	100	NULL	网卡型号
ComputerName	nvarchar	50	NULL	计算机名称
Account	nvarchar	50	NULL	计算机账号
OSName	nvarchar	100	NULL	操作系统名称
UserName	nvarchar	20	NULL	使用者姓名
Number	nvarchar	50	NULL	办公室门牌号
Dept	nvarchar	50	NULL	部门

图 1 ComputerConfig 表结构设计

上述是以执行 SQL 语句的方式来创建数据库和相关表结构的，也可以很方便地使用 SQL Server 企业管理器的可视化工具来创建上述的数据库和相关的表结构。

另外，还需要为该表创建一个 SQL Server 的登录账户，并赋予对“ComputerConfig”表具有读、更改、增加、删除的权利，用于在客户端执行脚本提交数据信息。下面的语句创建了具有这种权限的 SQL Server 登录用户，其用户名为“dba”、密码为“dba”。

```
/*创建登录用户 dba*/
EXEC sp_addlogin 'dba', 'dba', 'DB'
/*为用户 dba 分配对 ComputerConfig 表的权限
EXEC sp_grantdbaccess N'dba', N'dba'
GRANT SELECT , UPDATE , INSERT , DELETE
ON [dbo].[ComputerConfig] TO [dba]
```

也可使用 SQL Server 企业管理器的可视化工具来创建用户和分配权限。定义好用户之后，双击“ComputerConfig”表，在其属性窗口中点击【权限】按钮，进入权限设置面板，对相关用户选择相应的权限即可，如图 2 所示。

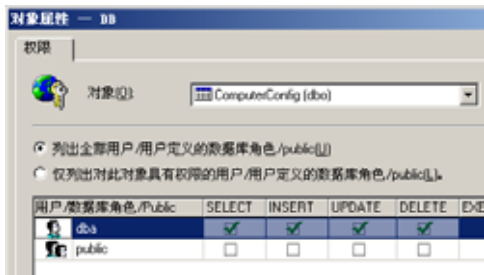


图 2 ComputerConfig 表用户权限设置

3.2 Windows 脚本设计

Windows 脚本用于在客户端机器上的使用,通过运行该脚本来收集“ComputerConfig”中所设计字段的信息。其中,“使用者姓名”、“办公室门牌号”和“所属部门”这几个信息在客户机上以人机交互的方式来实现,因此在 Windows 脚本中采用 InputBox 函数来获取这些信息。相关脚本如下所示,只有获取了这几个字段的内容,才能进行后续的硬件相关信息的采集。

```
strUsername = InputBox("使用者姓名: ", "提示:")
If Not (strUsername = Empty) Then
    strNumber = InputBox("办公室门牌号: ", "提示:")
    If (strNumber = Empty) Then
        WScript.echo "输入信息错误, 未收集本地计算机的相关信息!"
        WScript.Quit '终止脚本执行
    else
        strDept= InputBox("所属部门: ", "提示:")
        If (strDept= Empty) Then
            WScript.echo "输入信息错误, 未收集本地计算机的相关信息!"
            WScript.Quit '终止脚本执行
        End If
    End If
else
    WScript.echo "输入信息错误, 未收集本地计算机的相关信息!"
    WScript.Quit '终止脚本执行
End If
```

接下来的脚本是设置连接 SQL Server 数据库服务器的四个参数,另外“.”表示客户端本机。将这些作为自定义过程 WriteInfo 的参数带入,由 WriteInfo

过程来实现客户端信息的采集。在这里,设定安装了数据库服务器机器的 IP 地址为 192.168.20.2。

strDBServer = "192.168.20.2" '数据库服务器的 IP

strDBuid="dba" '数据库服务器的登录用户

strDBpwd="dba" '登录用户 dba 的密码

strDBName="DB" '数据库名称

strComputer="." 本地机器

Call

WriteInfo(strComputer,strDBServer,strDBuid,strDBpwd,strDBName,strUsername,strNumber) '调用自定义过程

WScript.Echo "本地计算机的相关信息已被收集"

在 WriteInfo 自定义过程中,使用 ADODB 组件实现对 SQL Server 连接、读写记录的操作。因为每台机器的 MAC 地址是唯一的,所以从 GetMac 自定义函数中获取本机的 MAC 地址作为从数据库检索的条件,当没有检索到该 MAC 地址的记录时,表示需要采集该台客户机的相关硬件信息,用 objRecordset 对象的 AddNew 方法实现对记录的增加操作,并递交到数据库中。在 WriteInfo 自定义过程中,定义了多个自定义函数来获取客户端机器的相关信息。

```
'将本地计算机的相关信息写入 SQL 数据库
Sub
WriteInfo(strComputer,strDBServer,strDBuid,strDBpwd,strDBName,strUsername,strNumber)
    Const adOpenStatic = 3 '启动 Static 类型的游标
    Const adLockOptimistic = 3 '只有在调用 Update 方法时才锁定记录集
    Const adUseClient = 3 '使用由本地游标库提供的客户端游标
    Set objConnection = CreateObject("ADODB.Connection") '建立数据库连接对象
    Set objRecordset = CreateObject("ADODB.Recordset") '建立记录集对象
    objConnection.Open _
        "Provider=SQLOLEDB;Data Source=" & strDBServer & ";" & _
        "uid=" & strDBuid & ";pwd=" & strDBpwd & ";Initial Catalog=" & strDBName '建立数据库连接
    objRecordset.CursorLocation = adUseClient
```

```

objRecordset.Open "SELECT * FROM
ComputerConfig", objConnection, _
adOpenStatic, adLockOptimistic ' 获取
ComputerConfig 表的记录
strSearchCriteria = "MAC = "" &
GetMac(strComputer) & "" 建立查找条件表达式
objRecordSet.Find strSearchCriteria
'查找不到有相同 MAC 的记录, 则增加一条记录
If objRecordset.EOF Then
objRecordset.AddNew
End If
'填充该记录的各个字段内容:
objRecordset("MAC") = GetMac(strComputer) 'MAC 地
址
objRecordset("Memory") =
GetPhysicalMemory(strComputer) '物理内存
objRecordset("NetworkCard") =
GetNetworkCard(strComputer) '网卡型号
objRecordset("Account") = GetUsername() '计算机账
号
objRecordset("ComputerName") =
GetComputerName() '计算机名
objRecordset("OSName") =
GetOSName(strComputer) '操作系统名
objRecordset("UserName") = strUsername '使用
者
objRecordset("Number") = strNumber '办公室牌
号
objRecordset("Dept") = strDept '部门
objRecordset.Update '递交该记录到数据库
objRecordset.Close '关闭记录集
objConnection.Close '关闭数据库连接
End sub

```

自定义函数 GetUsername 和 GetCom-
puterName 利用 WSH 的 Network 对象的属性获取
当前用户名和计算机名。脚本如下所示:

```

Function GetUsername() '获取当前用户
set wshNetwork =
WScript.CreateObject("WScript.Network")
GetUsername =wshNetwork.UserName
End Function

```

```

Function GetComputerName() '获取计算机名
set wshNetwork =
WScript.CreateObject("WScript.Network")
GetComputerName =wshNetwork.ComputerName
End Function

```

自定义函数 GetMac、GetOSName、GetPhysi-
calMemory 和 GetNetworkCard 实现对客户端机器
的硬件信息获取。这四个自定义函数分别获取该机器
的 MAC 地址、安装的操作系统、本机内存大小和网
卡型号。这些自定义函数都是使用 WMI 组件对象^[4],
并利用它的 ExecQuery 方法来分别获取相应的信息,
脚本如下所示。

```

Function GetMac(strComputer) '获取 MAC 地址
Set objWMIService = GetObject("winmgmts:" _
&
"{impersonationLevel=impersonate,(Shutdown)}!\\"
& strComputer & "\root\cimv2")
Set colMac = objWMIService.ExecQuery("Select *
from Win32_NetworkAdapterConfiguration where
IPEnabled=TRUE")
GetMac=""
For Each Mac In colMac
GetMac = Mac.MacAddress
Exit For
Next
End Function
Function GetOSName(strComputer) '获取操作系统
Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\" &
strComputer & "\root\cimv2")
Set colOperatingSystems = objWMIService.ExecQuery
-
("Select * from Win32_OperatingSystem")
For Each objOperatingSystem in colOperatingSystems
GetOSName = objOperatingSystem.Caption
Next
End Function
Function GetPhysicalMemory(strComputer) '获取物理
内存
Set objWMIService = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\\" &

```

```
strComputer & "\root\cimv2")
    Set colSettings = objWMIService.ExecQuery _
        ("Select * from Win32_ComputerSystem")
    For Each objOperatingSystem in colSettings
        GetPhysicalMemory=objOperatingSystem.TotalP
hysicalMemory
    Next
End Function
Function GetNetworkCard(strComputer) '获取网卡名
    Set objWMIService = GetObject("winmgmts:" _
        & "{impersonationLevel=impersonate}!\\" &
strComputer & "\root\cimv2")
    Set colItems = objWMIService.ExecQuery("Select
* from Win32_NetworkAdapter")
    For Each objItem in colItems
        If objItem.AdapterType = "Ethernet 802.3"
Then
            GetNetworkCard = objItem.Name
            Exit function
        End if
    Next
End Function
```

把上面这些脚本保存在一个 vbs 文件中, 执行该脚本文件即可将客户端机器的信息提交到 SQL Server 数据库中, 实现对客户端信息的采集。

4 小结

采用 Windows 脚本技术与数据库相结合, 可以很方便地采集网络中的计算机硬件信息, 用简单的技术就能完成那些使用高级开发工具才能完成的功能。另外, 还可以进一步地借助 Excel 数据透视表的功能, 很方便地充分利用采集到的数据信息, 提供灵活可变的数据分析报表, 掌握网络中计算机设备的情况。

参考文献

- 1 Microsoft 帮助文档, Microsoft Windows 脚本技术.
- 2 TechNet 脚本中心[2010-1-20]. <http://www.microsoft.com/china/technet/community/scriptcenter/default.aspx>,
- 3 马文刚. WSH 实用技术与应用. 北京: 清华大学出版社, 2002.
- 4 Policht M. WMI 技术指南. 北京: 机械工业出版社, 2002.