МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Безопасность информационных и автоматизированных систем»

WMI-ИНСТРУМЕНТАРИЙ УПРАВЛЕНИЯ WINDOWS

Методические указания
к выполнению лабораторной работы
по дисциплине «Операционные системы»
для студентов специальностей 090105, 230105, 090303, 090900

Курган 2012

Кафедра: «Безопасность информационных и автоматизированных систем»

Дисциплина: «Операционные системы» (специальности 090105, 230105, 090303, 090900)

Составил: ст. преподаватель А.Г. Рабушко

Утверждены на заседании кафедры «16» марта 2012 г.

Рекомендованы методическим советом университета «30» марта 2012 г.

ПРИБОРЫ, ОБОРУДОВАНИЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1 Операционная система (Microsoft Windows Xp).
2 Используемая среда программирования (Visual Studio C++).

ЦЕЛЬ РАБОТЫ

Изучение архитектуры WMI и основных логических классов.

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

WMI – инструментарий управления Windows, представляет собой реализацию Web-Based Enterprise Management (WBEM) – стандарда, который индустриальный консорциум Distributed Management Task Force (DMTF) определяет. WMI предоставляет развитые и гибкие средства для сбора информации и конфигурирования корпоративной сети. Архитектуру WMI можно отобразить следующим образом:

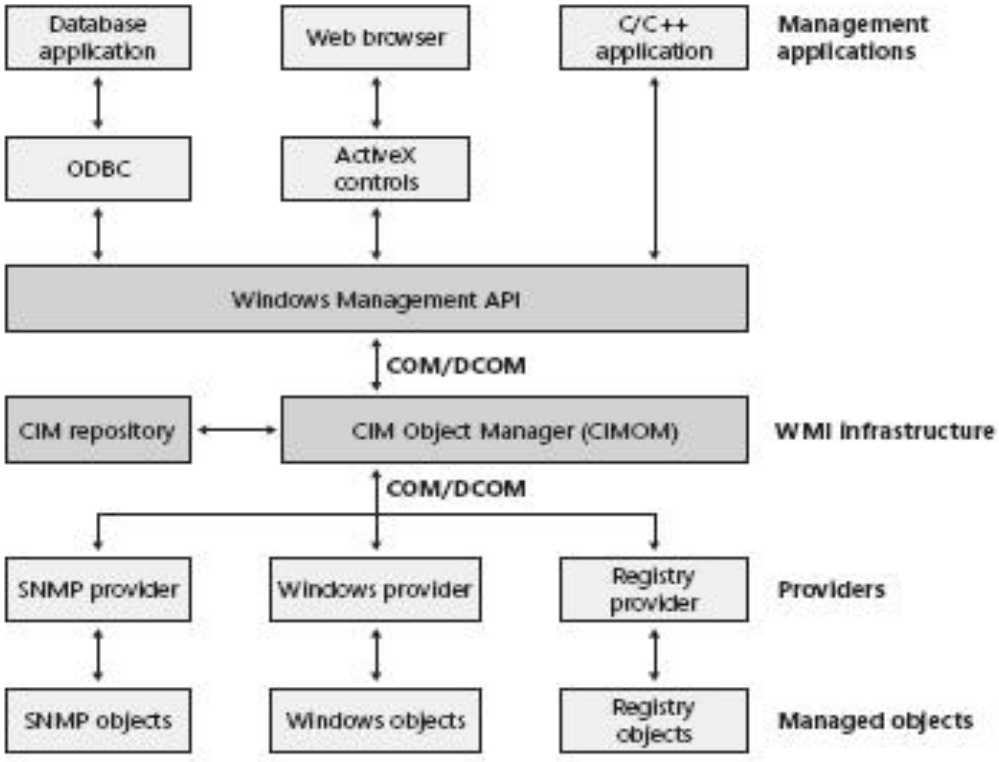
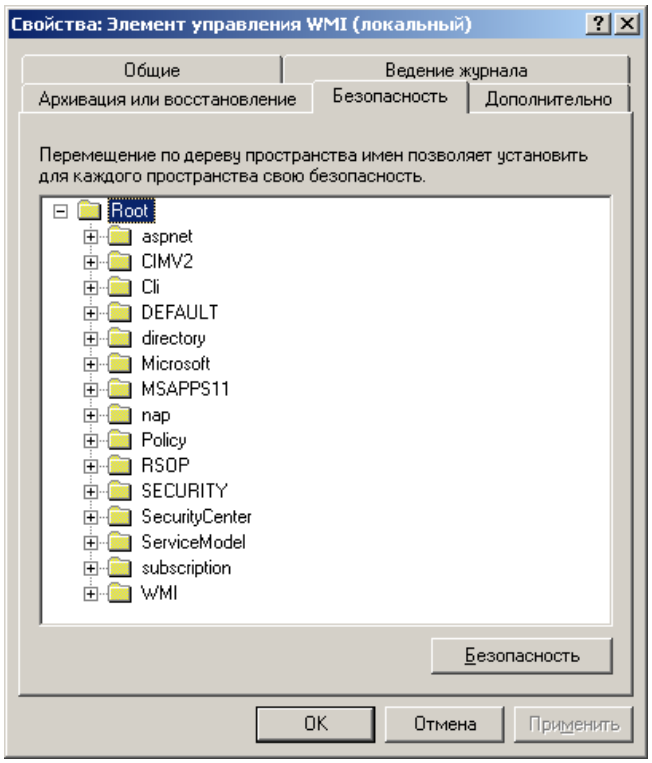


FIGURE 4-18 WMI architecture

Большинство логических и физических ресурсов операционных систем представляются в виде классов. Провайдеры обеспечивают, в часности, информацию об объектах поддерживаемых ими классов. Классы определяются в пространстве имен WMI. Пространство имен WMI – это раздел (директория) репозитория WMI, которая призвана группировать классы и объекты WMI по назначению, а также определять атрибуты безопасности при доступе к классам и объектам в каждом таком контейнере. Фактически это полная аналогия со структурой файловой системы. Все пространства имен начинаются с корня, который в WMI обозначается ключевым словом root. После имени корня через

косую черту указывается пространство имен. Пространства имен могут быть вложенными. Пример того, как выглядит пространство имен: root/mynamespace/subnamespace. Подавляющее большинство классов и объектов размещается в пространстве имен root/CIMv2. Для просмотра структуры можно воспользоваться утилитой **wmimgmt.msc**.



Провайдеры реализуются в виде dll, расположенные в каталоге \WINDOWS\system32\wbem

Provider	DLL	Namespace
Active Directory provider	dsprov.dll	root\directory\ldap
Event Log provider	ntevt.dll	root\cimv2
Performance Counter provider	wbemperf.dll	root\cimv2
Registry provider	stdprov.dll	root\default
SNMP provider	snmpincl.dll	root\snmp
WDM provider	wmiprov.dll	root\wmi
Win32 provider	cimwin32.dll	root\cimv2
Windows Installer provider	msiprov.dll	root\cimv2

Для примера рассмотрим класс, представляющий текущие процессы

Win32_Process. Этот класс производен от класса **CIM_Process**, который, в свою очередь, производен от класса **CIM_LogicalElement.**

```
class Win32_Process : CIM_Process
{
  string Caption;
  string CommandLine;
  string CreationClassName;
  datetime CreationDate;
  string CSCreationClassName;
  string CSName;
  string Description;
  string ExecutablePath;
  uint16 ExecutionState;
  string Handle;
  uint32 HandleCount;
  datetime InstallDate;
  uint64 KernelModeTime;
  uint32 MaximumWorkingSetSize;
  uint32 MinimumWorkingSetSize;
  string Name;
  string OSCreationClassName;
  string OSName;
  uint64 OtherOperationCount;
  uint64 OtherTransferCount;
  uint32 PageFaults;
  uint32 PageFileUsage;
  uint32 ParentProcessId;
  uint32 PeakPageFileUsage;
  uint64 PeakVirtualSize;
  uint32 PeakWorkingSetSize;
  uint32 Priority;
  uint64 PrivatePageCount;
  uint32 ProcessId;
  uint32 QuotaNonPagedPoolUsage;
  uint32 QuotaPagedPoolUsage;
  uint32 QuotaPeakNonPagedPoolUsage;
  uint32 QuotaPeakPagedPoolUsage;
  uint64 ReadOperationCount;
  uint64 ReadTransferCount;
  uint32 SessionId;
  string Status;
  datetime TerminationDate;
  uint32 ThreadCount;
  uint64 UserModeTime;
```

```
  uint64 VirtualSize;
  string WindowsVersion;
  uint64 WorkingSetSize;
  uint64 WriteOperationCount;
  uint64 WriteTransferCount;
};
```

В этом классе определены следующие методы:

```
AttachDebugger
Create
GetOwner
GetOwnerSid
SetPriority
Terminate
```

Доступ к WMI может осуществляться через интерфейсы COM+ и .NET Framework. Это означает, что любой язык программирования, который поддерживает взаимодействие с Microsoft Windows COM+ и .NET Framework, может использоваться для работы с WMI. К перечню таких языков, в частности, относятся: VBScript, Visual Basic и Visual Basic .NET, Java Script, Python, Perl, PHP, C#, C++, Pascal, TCL и другие. Обращение к объектам и методам WMI в разных языках может немного отличаться из-за специфики синтаксиса работы с объектами и типами для каждого конкретного языка, но в целом все приемы очень сходны. Для иллюстрации выведем список процессов и их идентификаторов, используя VBScript:

```
strComputer = "."

Set objWMIService = GetObject("winmgmts:\\" & strComputer &
"\root\CIMV2")

Set colItems = objWMIService.ExecQuery( _
   "SELECT * FROM Win32_Process")

For Each objItem in colItems
      Wscript.Echo objItem.name+"---"+objItem.handle
Next
```

и C#:

```
using System;
using System.Management;
using System.Windows.Forms;

namespace WMISample
{
```

```
public class MyWMIQuery
{
   public static void Main()
{
      ManagementObjectSearcher searcher =
         new ManagementObjectSearcher("root\\CIMV2",
         "SELECT * FROM Win32_Process");

      foreach (ManagementObject queryObj in searcher.Get())
      {

      Console.WriteLine("Name---pid {0} {1}",
queryObj["Name"],queryObj["Handle"]);

      }

}
}
}
```

Инструментарий управления Windows реализуется как сервисная dll и стартует как C:\WINDOWS\System32\svchost.exe -k netsvcs.

Естественно, WMI предоставляет средства для получения информации о самом себе. С помощью следующего скрипта можно получить список всех провайдеров.

```
For Each pr In GetObject("winmgmts:").InstancesOf("__Win32Provider")
   WScript.Echo pr.Name
Next
```

Далее – пример рекурсивного прохода пространства имен:

```
comp = "."
Call rec("root")
Sub rec(txt)
  WScript.Echo txt
  Set x = GetObject("winmgmts:\\" & comp & "\" & txt)
  Set coll = x.InstancesOf("__NAMESPACE")
  For Each i In coll
    Call rec(txt & "\" & i.Name)
  Next  End Sub
```

Таким образом можно получить все подклассы :

```
set x = Getobject("winmgmts:\\.\root\cimv2")
set y = x.SubclassesOf("CIM_LogicalElement")
wscript.echo y.count
for each z in y
wscript.echo   z.Path_
next
```

Аналогично предоставляется информация о свойствах и методах конкретного класса.

Крайне полезной представляется возможность WMI обрабатывать события. События WMI – это очень удобный и эффективный механизм выявления изменений в системе и экземплярах объектов WMI. Обработка событий может быть синхронной и асинхронной. Синхронная обработка событий – это когда процесс ожидает события и более ничем не занят. Обычно это ожидание – бесконечный цикл проверки условия: поступило событие или нет. Асинхронная обработка подразумевает, что процесс регистрирует обработчик события (подписывается на событие) и далее продолжает выполнять различные задачи. Когда событие возникает, нормальная работа процесса прерывается, запоминается место, где произошло прерывание, а управление передается на зарегистрированный обработчик событий. После обработки события обработчиком управление возвращается на то действие основного процесса, которое было прервано.

С помощью следующего скрипта отслеживается порождение процесса с именем cmd.exe.

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\" & strComputer &
"\root\CIMV2")
 Set objEvents = objWMIService.ExecNotificationQuery _
("SELECT * FROM Win32_ProcessStartTrace WHERE ProcessName =
'cmd.exe'")
 Wscript.Echo "Waiting for events ..."
Do While(True)
   Set objReceivedEvent = objEvents.NextEvent
    Wscript.Echo "CMD.EXE started"
 Loop
```

СПИСОК ЗАДАНИЙ

Для выполнения задания необходимо изучить свойства следующих классов и на удобном языке программирования вывести полную информацию о соответствующих объектах.

1 Win32_PageFileUsage
2 Win32_Registry
3 Win32_SoftwareFeature
4 Win32_Process
5 Win32_PortResource
6 Win32_DMAChannel
7 Win32_IRQResource
8 Win32_Environment
9 Win32_BIOS
10 Win32_ServerConnection
11 Win32_ComputerSystem

12 Win32_SystemDriver
13 Win32_Service
14 Win32_TerminalService
15 Win32_PrinterDriver
16 Win32_Share
17 Win32_OperatingSystem
18 Win32_MappedLogicalDisk
19 Win32_LogicalDisk
20 Win32_DiskPartition
21 Win32_MotherboardDevice
22 Win32_Keyboard
23 Win32_USBController
24 Win32_IDEController
25 Win32_VideoController
26 Win32_SoundDevice
27 Win32_Processor
28 Win32_DiskDrive
29 Win32_NetworkConnection
30 Win32_Thread

СПИСОК ЛИТЕРАТУРЫ

1 Руссинович, М. Внутреннее устройство Microsoft Windows. Мастер-класс / М. Руссинович, Д. Соломон. – СПб. : Питер, 2005.
2 Дейтел, Х.М. Операционные системы. Ч. 2: Распределенные системы, сети, безопасность / Х.М. Дейтел, П.Дж. Дейтел, Д.Р. Чофнес. – М. : Бином, 2006.
3 Дейтел, Х.М. Операционные системы. Ч.1: Основы и принципы / Х.М. Дейтел, П.Дж. Дейтел, Д.Р. Чофнес. – М.: Бином, 2006.
4 Гордеев, А.В. Операционные системы : учебник для вузов /А.В. Гордеев. – СПб. : Питер, 2004. – 416 с.
5 Олифер, В.Г. Сетевые операционные системы /В.Г. Олифер, Н.А. Олифер. – 2-е изд. - СПб.: Питер, 2001. – 544 с.
6 Танненбаум, Э. Современные операционные системы /Э. Танненбаум. – СПб. : Питер, 2002. – 1040 с.
7 Кастер, Х. Основы Windows NT и NTFS. Русская редакция /Х. Кастер. – М., 1996.
8 Проскурин, В.Г. Защита в операционных системах /В.Г. Проскурин, С.В. Крутов, И.В. Мацкевич. – М. : Радио и связь, 2000.

Рабушко Артур Германович

WMI-ИНСТРУМЕНТАРИЙ УПРАВЛЕНИЯ WINDOWS

Методические указания
к выполнению лабораторной работы
по дисциплине «Операционные системы»
для студентов специальностей 090105, 230105, 090303, 090900

Редактор О.Д. Постовалова

Подписано к печати	Формат 60×84 1/16	Бумага тип. №1
Печать трафаретная	Усл. печ.л. 0,75	Уч.-изд.л.0,75
Заказ	Тираж э/в	Цена свободная

РИЦ Курганского государственного университета.
640669, г. Курган, ул. Гоголя, 25.
Курганский государственный университет.