

WMI入门（一）：什么是WMI

WMI出现至今已经二十多年了，但很多人对它并不熟悉。知道它很好很强大，但不知道它从哪里来，怎么工作，使用范围是什么？现在我跟大家说说我的理解。。。

WMI有一组API。我们不管使用VBScript、PowerShell脚本还是利用C#的来访问WMI的类库，都是因为WMI向外暴露的一组API。这些API是在系统安装WMI模块的时候安装的，通过他们我们能够拿到我们想要的类。

WMI有一个存储库。尽管WMI的多数实例数据都不存储在WMI中，但是WMI确实有一个存储库，用来存放提供程序提供的类信息，或者称为类的蓝图或者Schema。

WMI有一个Service。WMI总是能够响应用户的访问，那是因为它有一个一直运行的Windows服务，名字叫Winmgmt。停止这个服务，所有对WMI的操作都将没有反应。

WMI是可扩展的。人人都知道WMI能干很多事情，读取本机硬盘信息、读取远程计算机的用户信息、读取域用户信息等等。基本上，你能想到的获取或者更改资源的操作，它都能干。可谓吃得少，干得多。它为什么这么能干呢？这基于WMI的可扩展性。WMI对资源的操作，不是它自己实现了什么方法，而完全取决于向它注册的提供程序。

WMI是管理员日常必备的强大工具之一，是脚本伴侣。当然也可以把一个大型系统建立在WMI以及WMI的提供程序之上。以下给两个使用WMI的例子：

示例一：

本示例利用VB Script检查本机的物理内存量：

```
strComputer = "."

Set wbemServices = GetObject("winmgmts:\\." & strComputer)
Set wbemObjectSet = wbemServices.InstancesOf("Win32_LogicalMemoryConfiguration")

For Each wbemObject In wbemObjectSet
    WScript.Echo "Total Physical Memory (kb): " & wbemObject.TotalPhysicalMemory
Next
```

示例二：

本示例利用PowerShell检查本机的逻辑硬盘情况：

```
PS C:\Users\v-lukez> Get-WmiObject -Class Win32_LogicalDisk
```

结语：

WMI功能强大，而且一直在发展，但是因为用的人不多，所以一直不温不火。希望更多的人能够慧眼识珠，看到的潜力。

WMI入门（二）：如何使用WMI

知道了WMI是什么以后，肯定想要知道如何使用WMI干活。它能写“Hello World”吗？WMI没那么无聊，它一上来就会问你“你想要什么？”或者“你想要更改什么？”。。。

WMI工具

Windows本身提供了几个WMI工具供用户使用，包括WMIC.exe、WBEMTest.exe和WMI Administrative Tools等。前两个是系统自带的，最后一个需要自己下载安装。不过说实话，我最喜欢WMI Administrative Tools。下载WMI Administrative Tools在[这里](#)。下面以WMI Administrative Tools为例说明如何使用工具使用WMI。

下载并安装WMI Administrative Tools之后，会有4个工具，最常用的是其中的WMI CIM Studio。在开始菜单里单击它，你会发现它是以网页形式在IE浏览器中打开的。

使用步骤如下：

- 1、单击页面上部的电脑图标，会显示一个对话框，单击“Connect”按钮，如图1所示；

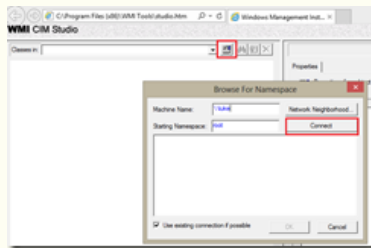


图1

- 2、任何时候出现如图2所示的登陆对话框，单击OK按钮即可；

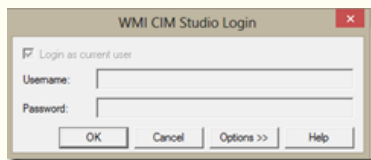


图2

- 3、选择合适的命令空间（这里选择常用的root\cimv2），然后单击OK按钮，如图3所示；

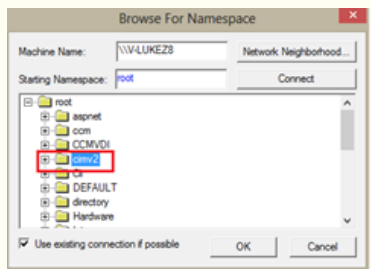


图3

- 4、直接双击右侧类名，或者单击上部的望远镜按钮，在弹出的查找对话框中输入你要查找的完整类名或者类名的一部分，单击Go按钮查找；并在查找结果中选择你要查看的类名，单击OK按钮，如图4所示；

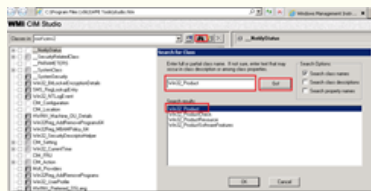


图4

- 5、单击右上部倒数第4个实例按钮，即可得到本机所有安装的软件列表，如图5所示；

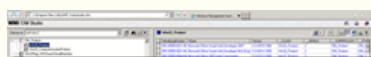


图5

任何WMI类以及实例的查看均可参考以上步骤。

脚本操作WMI

以前对WMI的操作都是通过VB Script进行的，现在更多人倾向于使用PowerShell来访问WMI类和对象。下面给出使用PowerShell命令查询本机安装的所有软件列表的示例：



```
PS C:\Users\luke> Get-WmiObject -Class Win32_Product | select name

name
----
HelpDesk
Windows Installer Clean Up
XML Notepad 2007
Microsoft Network Monitor 3.4
Microsoft Visual Studio 2010 Service Pack 1
ScreenRecorder
Microsoft Office Visual Web Developer 2007
Microsoft Office Visual Web Developer MUI (English) 2007
...
```



比使用带界面的WMI要简单的多，而且还可以有很多筛选条件。

.Net操作WMI

使用.Net中的高级语言来操作WMI也非常的方便。下面示例演示C#操作WMI获得本机安装软件的列表的代码：



```
using System;
using System.Management;

namespace WMIDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            ManagementPath wmiPath = new ManagementPath(@"\\.\root\cimv2:Win32_Product");
            ManagementClass wmiClass = new ManagementClass(wmiPath);

            ManagementObjectCollection instances = wmiClass.GetInstances();

            Console.WriteLine("Name, Vendor");
            foreach (ManagementObject product in instances)
            {
                Console.WriteLine(String.Format("{0}, {1} ", product["Name"], product["Vendor"]));
            }
        }
    }
}
```



结语

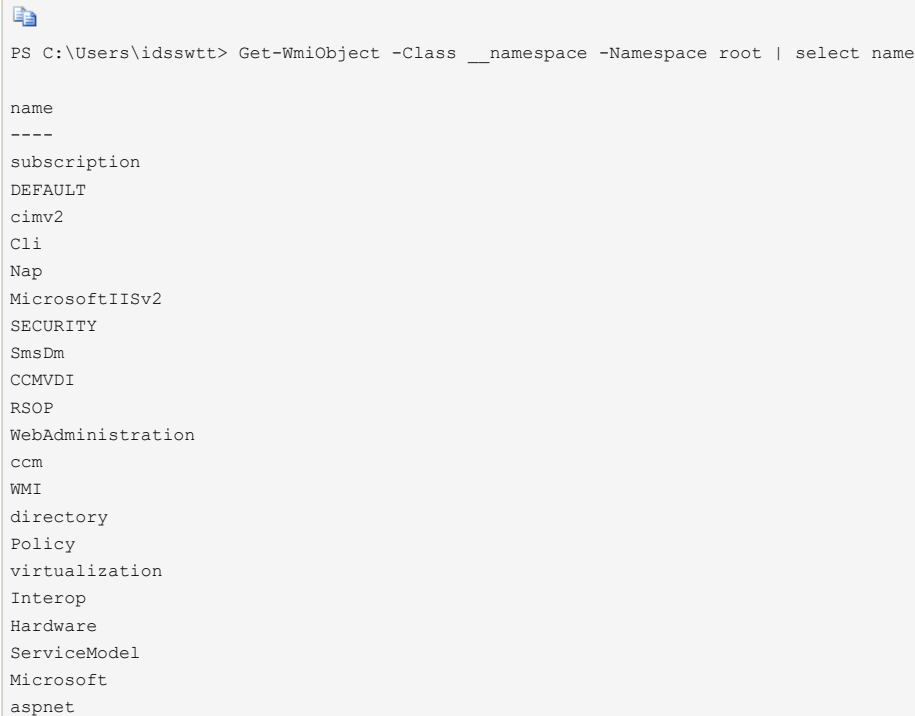
WMI是一个多面手，你可以从各个侧面来使用它。套用一句广告词：关于WMI，你怎么用？

WMI入门（三）：我需要的类在哪里？

使用WMI归根到底是对WMI类的访问，如何找到合适的类是其中关键的一步。我一直在为找不到合适的类和命名空间苦恼。你也有这样的问题吗？

类层次结构

WMI的类是以命名空间和继承层次方式组织的，呈树形结构。命名空间的根是root，在它的下面还有十几个命名空间，最常用的是root\cimv2。命名空间的信息存储在静态类__Namespace类中，要查询当前命名空间下的所有命名空间，可以查看__Namespace类的实例。以下是在我的机器上使用PowerShell查询到的命名空间：



```
PS C:\Users\idsswtt> Get-WmiObject -Class __namespace -Namespace root | select name

name
----
subscription
DEFAULT
cimv2
Cli
Nap
MicrosoftIISv2
SECURITY
SmsDm
CCMVDI
RSOP
WebAdministration
ccm
WMI
directory
Policy
virtualization
Interop
Hardware
ServiceModel
Microsoft
aspnet
```

关于命名空间更详细的信息，可以写一个递归的脚本来取得，或者从[WMI入门（二）：如何使用WMI](#)中提到的WMI CIM Studio工具中查到。

常用WMI类

WMI体系结构中有成百上千个类。我们经常用到的只有其中一部分。其中有些是用来查询实例的，另一些用来查询类的Schema的。除了上面提到的__Namespace类，我们还会用到__Win32Provider类，来查询当前系统中都有哪些提供程序。这两个类是系统类。

此外，还有两类WMI类我们经常用到：CIM_*类和Win32_*类。前者是WMI的核心类和公共类，很多时候也被称为CIM核心类和公共类。后者是Win32扩展类，我们查询实例多在Win32扩展类中查询。例如Win32_Product、Win32_LogicalDisk等。

WMI类查找手册？

很遗憾的是，到目前为止。我没有发现一个好用的WMI类查找手册。但在微软MSDN网站上，有一个分类。我们可以依照这个分类来做查询。以下几个链接是我经常用到的，贴在这里，希望你能用到。

[WMI Reference](#)

[Win32 Classes](#)

[WMI Providers](#)

结语

看完上面我写的这些东西，如果你还是找不到你要的类，那你就只能怪微软了。我觉得微软为WMI做的还是太少，至少在类查询这一块儿是这样。在微软为WMI投入更多之前，我们只能依赖经验和搜索引擎来寻找适当的类了。