HZ BOOKS

WMI Essentials for Automating Windows Management

Windows 技术丛书

WMI

(美) Marcin Policht 著
智慧东方工作室 译

WMI 技术指南

机械工业出版社
China Machine Press

SAMS

Windows 技术丛书

WMI 技术指南

（美） Marcin Policht 著

智慧东方工作室 译

机械工业出版社
China Machine Press

Windows管理设备(WMI)是微软提出的Windows环境自动管理方案,它建立在"公共信息模型"(CIM)这个工业标准的基础上。本书内容包括Windows脚本编制基础、WMI基础知识、WMI脚本编制方法、利用WMI脚本管理Windows环境、WMI和Microsoft Server产品介绍、WMI的未来展望,等等。

本书内容丰富、通俗易懂,为系统管理员和初级开发者更有效地管理Windows环境并进行应用软件开发提供了极具实用性的参考。

本书版权登记号:图字:01-2001-4289

图书在版编目(CIP)数据

WMI技术指南/(美)波利奇(Policht. M)著;智慧东方工作室译. -北京:机械工业出版社,2002.2

(Windows技术丛书)

书名原文:WMI Essentials for Automating Windows Management

ISBN 7-111-09499-9

Ⅰ.W... Ⅱ.①波... ②智... Ⅲ.软件包,WMI Ⅳ.TP317

中国版本图书馆CIP数据核字(2001)第078847号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:路晓村 张鸿斌

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年2月第1版第1次印刷

787mm×1092mm 1/16·30.75印张

印数:0 001-4000册

定价:48.00元

前　言

计算机领域是当今发展得最迅速的技术领域之一。客户不断要求更智能化、更快和功能更全面的计算产品，由此也增加了复杂性。作为这一要求所带来的直接结果，科学和技术的不断进步，一次又一次地证明了人的创造性是无穷无尽的。然而，计算机领域发生的快速变革也不是没有缺点的。这个过程的随机性和自发性使我们很难确定一种标准方法，实现新系统的开发、实现以及管理。

系统管理技术简介

计算机系统管理的标准化最近成为许多业内人士关注的重点。这一现象并不令人意外，因为对于这样的标准来说，一经建立和接纳，便会使牵涉到的所有人获益。针对各类基础结构，程序员可采用"插件"的形式，开发出能重复使用的软件组件。最终用户可享受到越来越高的可靠性，同时不会由于频繁的升级、维护或者不可预期的宕机时间而影响自己的工作效率。管理者可利用它有效地缩短自己的"总体拥有成本"(Total Cost of Ownership，TCO)。最后，系统管理员和IT支持人员可从一个位置部署、配置和监视数百台计算机——无论它们的配置或操作系统是什么。因此，这些人可从以往繁重的工作中解脱出来，在缩短工作时间的同时，轻松地完成自己的管理任务。

微软一直都在努力地使Windows操作系统易于管理，这是通过几种不同的策略来实现的。自从在Windows NT 4.0 Option Pack中采用了Windows Script Host以来，脚本功能得到了显著的改进。1994年，微软发布了版本号为1.0的Systems Management Server，它提供了硬件和软件库存、软件分发以及远程诊断功能。Zero Administration Initiative for Windows(Windows零管理，ZAW)不仅把这两种功能合为一体，同时还合并了其他一些技术(如微软管理控制台、Windows管理规范、基于Web的企业管理、系统策略和零管理包等)。ZAW最初面向的是Windows NT/9x操作系统，允许实现自动化管理任务，并可对客户机的桌面进行集中式管理。后来，Systems Management Server 2.0进一步增强了这些特性。

随着Windows 2000的问世，微软采取更积极的手段开发出高效的管理策略。根据下面这几个设定好的目标，Windows 2000成为其中最主要的支持平台：

- 管理同活动目录服务的集成。
- 将.NET企业服务器(SQL Server 2000、Exchange Server 2000和BizTalk Server 2000)包括到管理范畴内。
- 将XML(可扩展标记语言)和SOAP(简单对象访问协议)规范作为交换管理数据的标准方法使用。

要实现这些目标，必须对Windows零管理(ZAW)进行扩展，加入Windows 2000基本结构的

核心元素，比如 IntelliMirror(智能镜像)、Active Directory(活动目录)和 Windows Installer Service(Windows 安装服务)等等，另外还要使用 Windows 管理设备(WMI)和基于 Web 的企业管理组件(WBEM)。

目前有明显的趋势可以证明，在下一个版本的 Windows 中，必然会提供对这个策略的支持。Windows 管理设备必将成为.NET 方案的管理基础。另外，微软也允许第三方软件开发商开发支持该结构的产品。

此外，微软还成立了 Microsoft Management Alliance(微软管理联盟)，目的是帮助其他厂商构建相应的产品来支持新结构。

业内其他竞争者也没有放慢自己的脚步。Unicenter(来自 CA 国际公司)和 Enterprise(来自 Tivoli Systems 公司)均为十分成熟的产品，支持多种平台上的管理功能。NetIQ 的 Operations Manager(已获微软授权)能在 Windows 系统上提供实时的性能管理与监视。

对那些预算吃紧的小公司来说，即便在这些“精简”方案面前，它们的费用也通常超出了能够接受的限度。即使在较大的公司里，对管理控制台的访问都是由一个中心的 IT 小组小心地看管着，极少被纳入日常的管理任务中。

幸运的是，我们还有一个备用方案可供使用。“专业”管理工具提供的大多数特性其实都可通过一系列相对简单的脚本来实现。其中要用到的核心组件(它提供了所需的功能)便是 WMI。WMI 是 Windows 2000 以及 Windows .NET 的一种内建服务，这证明了微软试图将 WMI 作为自己管理策略的一个密不可分的部分提供给用户。WMI 也可在其他 Win32 平台上使用，同时可作为加载项或免费工具下载。

WMI 的优点

通常，WMI 可提供三种类型的管理功能：

- 数据收集——访问来自多种不同来源(操作系统、性能计数器、事件日志、注册表、硬件、驱动程序和目录服务)的系统信息。这些信息可针对资产管理进行分析与总结，以便创建性能基准，进行可用性分析(为实现所需的服务级别)或进行安全方面的跟踪。
- 系统配置——通过一种集中化的方式来修改系统信息，其中包括操作进程、服务和软件组件，进行系统维护或升级，以及执行作业，等等。
- 事件管理——被监视系统组件的属性一旦发生改变，可以及时知道。这种侦测既可以通知作为基础(发生时再通知)，也可以时间间隔作为基础(定时通知)。在主动和被动解决问题、错误隔离、控制系统可用性和系统健康监视过程中，这一功能是十分有用的。

下面我们进一步举例介绍使用 WMI 脚本可以实现的功能：

- 收集 Windows 性能监视器数据(比如剩余磁盘空间的百分比、处理器利用率)，对其进行总结或自动采取相应的行动(比如清除临时文件夹、Internet Explorer 缓存以及中止无用进程等等)。
- 收集 Windows 事件日志条目，并定义特定的行动，针对特定事件做出响应。数据也可以进行总结，而事件日志可进行清空或备份处理。

- 查询被管理系统的不同特征，比如操作系统版本、NetBIOS 名称、域成员、物理和逻辑盘、交换文件的信息、日期和时间设置等等。
- 启动、暂停、停止、恢复和删除服务。
- 启动、中止和查询进程。
- 关机或重启系统。
- 注销用户。
- 收集与打印机和打印队列有关的信息。
- 获取和编辑 Windows 注册表信息。
- 获取和编辑目录服务信息(包括活动目录)。
- 控制网络配置(显示和修改路由表、网卡和协议设置的内容)。
- 访问 WDM 设备驱动程序信息。

相同的脚本可同时应用在本地和远程系统中(只要提供足够的许可权限)。

如果你觉得上述任何特性值得自己花一些时间来实现，请继续阅读本书。本书将展示几种不同的方法来访问 WMI。通过本书，你有机会熟悉 WMI SDK 中的一些易于使用的、基于浏览器的工具。另外，你还会学习由其他工具(比如 WbemTest 或 WbemDump)提供的更强大的特性，只是用户界面没有这么友好而已。然而，本书的重点仍然放在由 VBScript 脚本、Windows Script Host 和其他 Windows 组件对象模型提供的功能上面。

你会发现在大多数时间里，一个简单的脚本里只需包含少数几行代码，便可充分利用全面的 WMI 功能。

脚本编制方法论

我们的脚本将采用下述组件进行构建：

- Visual Basic 脚本版(VBScript)是我们采用的脚本语言。它比较简单，只需几个钟头，你便能熟悉它的大多数基本特性。同时，和 Windows 管理员为自动化任务而使用的 Windows NT 命令相比它的功能比较强，足够达到我们的目的。
- Windows Script Host(WSH)将用于提供脚本编制环境。利用 WSH，我们能访问 Windows COM 对象(WMI 只是 Windows COM 的一个例子；我们还会对其他例子进行操作)。
- WMI 用于提供对管理数据的访问。
- 在 Windows Script Host 2.0 中引入的可扩展标记语言(XML)特性，用于提供辅助脚本格式。XML 和 WSH 2.0 的联合使用并不一定是必需的。但是遵守 XML 规范仍然是有一些好处的(详情参见第 1 章)。

从概念上说，大多数脚本中沿用的基本思想都是相当简单的。我们需要的数据由 WMI 提供，不必过份操心这些数据是如何生成的，它的生成流程我们会进一步解释，我们将利用由 Windows Script Host 和 WMI 对象模型提供的技术来访问它。有了一定程度的熟悉后，将用 VBScript 中的一系列命令来处理它，并生成自己需要的结果。

格式和编码约定

书中文字将采用一系列格式规则,目的是提高脚本的可读性。这些脚本是你开发自定义方案的良好起点。

- 代码采用一种特殊计算机字体,同普通正文区分。要求读者自行录入的代码采用粗体。
- 如有可能,会采用缩进、空行和大小写混合的名字。分配给变量的名字将被选中,以反映其用途。
- 按照被业内人士所普遍采纳的约定,变量名将用一个前缀来标志它的数据类型(尽管 VBScript 采用的肯定是"变体"数据类型——这方面的详情参见第 1 章)。举个例子来说,intCounter 这个变量名指出它的变量保存的是"整型"值。下面列出本书准备采用的变量类型:

bln	布尔类型(TRUE 或 FALSE)
dtm	日期(时间)
err	错误
int	整数
obj	对象
str	字串
arr	数组
col	对象集合

- 代码中使用的每个变量都会被声明(明确列出)。此外,使用 Option Explicit 语句还可以强制进行这种声明(详见第 1 章)。每个变量的用途都会得到恰当的解释。
- 代码结构(比如函数或子函数)将得到正确的编档。在注释中将提供下列信息:
  - 代码功能说明。
  - 输入参数列表。
  - 返回信息类型。
  - 在外部变量上的代码执行结果。
- 尽可能使用常量,使脚本的功用一目了然。常量名将直接反映其用途。
- 在一行代码的末尾,如下划线后跟随一个空格,表明同一条语句将在下一行延续。在你的计算机上录入脚本时,则不应换行,并省去那个下划线。
- VBScript 代码中的注释将用一个单引号字符(')开头。
- XML 代码中的注释在 <! - - 和 - - > 符号内(与 XML 规范相符)。不过,这一规则也有例外,详见第 1 章。

提示 *在全书中,作者会根据内容随时提供一些有用的提示。这些提示均采用这种格式。*

目　录

第1章 Windows脚本编制基础

像本书讨论的其他技术一样，Windows管理设备也是以面向对象设计的原理为基础的，同时大量借用了它的术语和概念。我们将通过讨论这些起源开始。

本章介绍面向对象设计的基本术语，比如对象、类、属性和方法、类的层次结构和继承、限定词以及键。当你理解了抽象模型之后，会发现它是如何与Windows组件结构相互联系的。特别是我们将讨论组件对象模型(COM)的基本原理，并简要回顾它的前身(DDE和OLE)，以及从它派生出来的新对象模型(DCOM和COM+)。

因为本书的很大部分是讨论脚本，所以本章也介绍了创建基于WMI脚本所需的工具集。“VBScript基础”一节向你介绍了语言中用于创建代码的基本语法。专门介绍Windows Script Host的1.6节帮助你熟悉通过脚本方法访问COM对象的过程。最后，讲述可扩展标记语言(XML)的1.7节阐释了它们提供的额外功能。本章还介绍了XML格式的原则，这些格式为你的代码提供了专业化和一致性的外观。

由于你的背景不同，你会发现本章展示的一些材料既可能非常基础，也可能相当复杂。这是可以理解的，因为无论对于程序员还是系统管理员，WMI都提供了颇具吸引力的特性。

1.1 面向对象模型原理

让我们从“模型”这个术语开始讨论。模型是对一些现实世界环境的一种抽象表示。模型的基本构建单元是对象。到底是什么构成一个对象？这在很大程度上取决于模型处理的是现实世界的哪个部分。你选择的对象应使你足以代表与你试图模拟的那个环境对应的每个部件。

对象是按类型分组，并基于一系列通用的特性。这些类型称为类。对象也称为它们的类的实例。

例如，如果你的任务是要提供一个大学学生注册的程序设计方案，那么学生(Student)、课程(Course)、教室(Classroom)和教员(Faculty Member)很有可能就是我们要设计的类。它们每一个都有一系列特定的属性，用于将对象与相同类的其他对象区分开。Student类可能包含这样的一些属性：学生的姓名、年级、专业、社会保险号、完成学分数，等等。想要修读American History 101的Ben Franklin(大学三年级，物理专业，SS # 199 - 99 - 0000，70个已完成学分)就是Student类的一个实例。课程本身可能是Course类的一个实例，可能有自己的时间表、总座位数、开放座位数等等。

除属性之外，类(及其对象)还可能有一系列方法(或动作)，它们以某种形式影响着模型——一般通过改变对象的一个属性或来自另一个类对象的属性来产生这种影响。同样地，到底选择哪一个来实现，这取决于现实世界的情况。在我们的例子中，可能有一个方法让学生选读一个班

级。不同类的对象可能通过联合,相互关联在一起。在我们的例子中,这种联合存在于学生和课程之间(把 Ben Franklin 和 American History 101 关联起来)。

注意,尽管一些类没有共同的属性或方法(Student 和 Course),但其他类可能共享属性和方法(Student 和 Faculty Member)。只要发生这种情况,就可创建类的一个层次结构。其中,子类继承了父类的属性和方法(见图 1-1)。这有助于维持类之间的关系。Student 和 Faculty Member 可派生自另一个所谓的抽象类(它没有有任何实例),比如可派生自一个 Person 类。在 Person 类中,将包含 Student 和 Faculty Member 类共同的属性和方法。

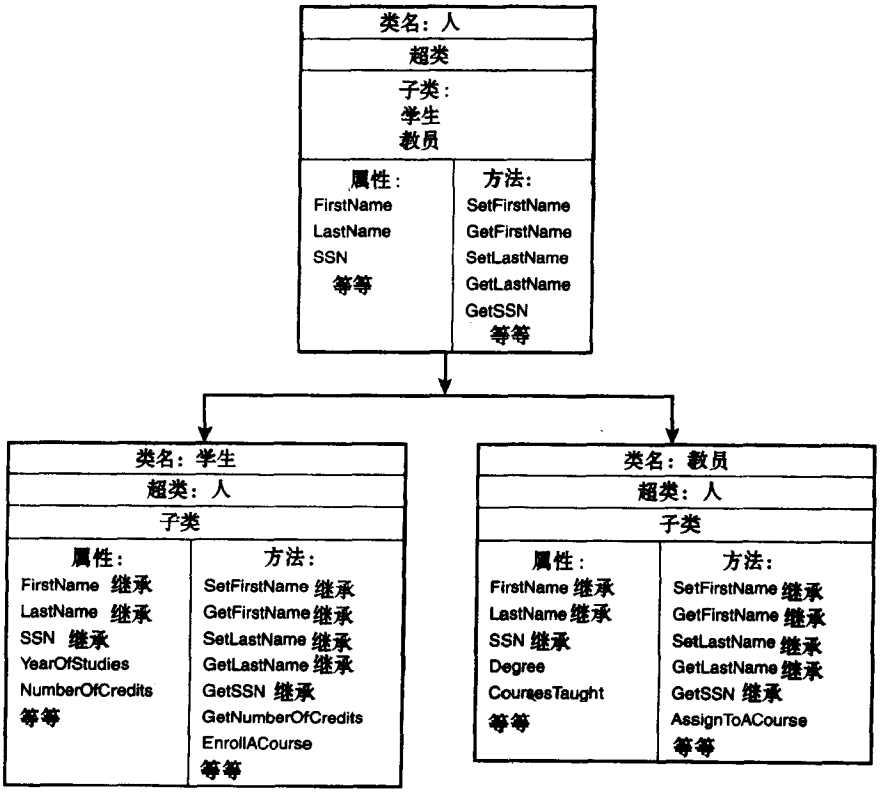

图 1-1 包括一个父类和两个子类的面向对象的类层次结构

有一种方法,它通过使用限定词来提供与类特征、方法和属性有关的额外信息。键便是这样的一个例子,它描述了一个属性,可惟一性地标识类的一个实例。通常,每个类都有一个键属性,尽管也有例外。Singleton(独身)类只允许有一个实例,所以不需要一个键。

限定词可通过指定限定词派生系统(flavors)进行更精细的调整。例如,派生系统可描述类继承如何影响限定词——换句话说,应该确定一个父类属性的限定词是否由它的子属性继承。

类的属性和方法是通过一个所谓的接口对外界公开的。其他类和外部程序可以访问接口。然而这完全取决于设计者,他们决定了哪些属性和方法保持隐藏(私有)以及哪些可由外部类访问(公共)。

面向对象方法得到了如此广泛的应用,主要有几方面的原因。

当应用到软件开发时,这种类型的设计将重点从针对对象采取的行动,转移到这些行动所针对的对象上面。如前面所解释的,每个对象都有一组自己的方法和属性。因为二者都可以通过一个接口来访问,所以它们的内部实现是不相关的。编写处理对象的程序时,只需知道接口的情况就可以了。这种类型的信息一般都由类的开发者很好地进行了归档。

此外,因为一个程序仅能通过接口访问允许的内容,所以存储在对象中的数据更加安全。

通常实现隐藏允许你自由地更新方法和属性,而不会影响外部程序使用它们的方式。

对象和类也能容易地在处理类似实体的任意类型程序中重复使用。

请想像一下,在完成你的注册(Enrollment)项目后,你被要求在另一个项目中工作,它涉及为另一所大学开发一套会计系统。你可以容易地把现有类和已写好的代码集成到里面。

1.2　Windows DDE、OLE、COM、DCOM 和 COM +

Windows 组件对象模型(或简称 COM)是一种对象模型,它是 Windows 脚本和程序设计的基础。COM 是以我们刚刚讨论的面向对象设计的原理为基础的。它定义了一个环境,其中包括多个独立的软件部件(对象)。通过接口,任何脚本或程序都可访问这些对象的属性和方法。也可容易地确定一个特定的接口是否可用,因为每个 COM 部件在进行安装的时候,都会向操作系统注册。

提示　注册信息输入到注册表的 HKEY _ CLASSES _ ROOT 部分。接口一般由两个参数来标识,它们是 ProgID 和 CLSID。第一个参数是由两个或多个部分组成的一个名称,一般描述了接口所属的部件和它的功能。第二个参数是一个 128 位的数字,一般用 32 个十六进制字符组成的字串来表示。尽管定义接口时只有 CLSID 是必需的,但我们脚本中使用的所有 COM 对象也有它们对应的 ProgID。在脚本中引用 COM 对象时,要使用 ProgID。

例如,我们将一直使用的一个 WMI 接口称为 SWbemSink。它的 ProgID 是 Wbem-Scripting.SWbemSink,CLSID 是{75718C9A - F029 - 11D1 - A1AC - 00C04FB6C223}。只要你在计算机上安装了 WMI,就可启动注册表编辑器(Regedt32.exe 或 Regedit.exe),然后在 HKEY _ CLASSES _ ROOT 目录中找到这两个条目。

当找到一个接口时,一个对象的属性和方法可通过一系列标准查询从这个接口派生出来(这是在程序设计级发生的,所以我们不必关心其细节)。

构建一个由多个独立部件组成的计算环境,且部件之间可以通信并能利用彼此之间的能力,这种想法在 Windows 发展的早期就以不同的形式体现出来了。

起初,它用于开发动态数据交换(DDE),也就是通过使用共享内存空间,在不同的应用程序之间交换数据。

它的下一种形式称为对象链接和嵌入(OLE),不但允许在应用程序间进行通信,还允许把应用程序合并到一起。这可以通过如下两种方法之一来完成:要么在不同类型的文件(比如一个 Word 文档和 Excel 电子表格)间创建一个链接;要么将文件中嵌入另一个文件。访问一个链接

或内嵌的文件时,会激活原始应用程序。

OLE 2.0 支持就地激活,它可展示出一个原始应用程序的特性,而不必真正启动它。这种能力是以自动化的概念为基础的。自动化允许一个应用程序(称为控制器对象)访问为另一个应用程序开发的函数(用 OLE 2.0 的术语,这是一个服务器应用程序)。自动化控制器可以使用 Windows 脚本语言(VBScript、JScript 或 VBA)或程序设计语言(比如 Visual Basic、C、C++和 Visual C++)开发。像 Excel 或 Word 的 Office 应用程序是由 Microsoft 提供的自动化服务器。

这些 OLE 2.0 服务器应用程序具有等价 COM 对象的功能。然而,COM 把自动化的概念应用到不如应用程序复杂的一部分软件(称为对象或控件)上。

COM 发展历程的下一个里程碑已经成形。在 Windows NT 4.0 中,它以分布式组件对象模型(DCOM)的形式被首次引入。它不仅允许在本地系统中访问 COM 对象,还允许通过网络访问。相应地,这允许编写分布式多层应用程序,以利用多台计算机的处理能力和资源。它同时增强了容错能力,因为应用程序不必再依赖于单独一个系统的可用性。

Windows 2000 采用的是对 COM 的一种扩展,名为 COM+。COM+增加了如下的支持:分布式事务处理、扩展的安全模型,同时以前的一些特性集的性能也得到了增强(比如线程池)。它同时还提供了更好的配置和管理能力。

1.3 Windows 脚本编制原理

在相当长的一段时间内,Windows 缺乏脚本编制的能力。即使对于相对简单的任务,大多数系统管理员也必须编制烦琐的批文件。字串处理、文件编辑、注册表访问、处理图形接口的大多数任务、标准的程序设计结构(比如循环)、条件语句、变量和子程序都是非常有限的。

解决办法是 Windows Scripting Host(WSH),它是作为 Windows NT 4.0 Option Pack 的一部分引入的。WSH 的强大特性是通过组合 3 个独立部件的功能来实现的。

第一个部件是脚本语言。WSH 提供了对 Visual Basic 脚本版本(VBA 的一个简化版本)和 JScript(JavaScript 的一个简化版本)的支持,后者常用在客户端 Web 脚本中。

第二个部件是脚本引擎。它用于解析和执行脚本。每个脚本语言需要它自己的脚本引擎,因为它们各自的语法是不相同的。这就是脚本有别于程序的地方。程序在执行前要编译,以便把它们转换成一组二进制的机器指令。一旦编译,它们就可以不用做任何的进一步转换而执行。另一方面,脚本是普通的文本文件。它们是通过脚本引擎来读取的,在每行的执行前,脚本引擎会校验它们的语法正确性。尽管程序设计语言执行起来更快,它们也需要更加复杂的开发工具,对于创建者来说,也要求他们具有更加高级的技术水平。另一方面,脚本可以使用任何文本编辑器来编写,而且它们的简化语法也可以相对快速地掌握。

第三个部件是一个环境。它是以提供自动化的对象模型为基础的。每个模型由处理不同功能的 COM 对象层次结构组成。例如,一个模型允许脚本处理目录服务,另一个模型提供了一种方法,用于读取和修改文件系统和注册表,它也可用于访问关系数据库和非关系数据库。任何脚本可以使用自动化机制来利用所有的这些功能。

这种环境可以通过宿主程序访问。Windows Script Host(WSH)可以作为如下的两个程序之

一运行:cscript.exe(命令行版本)或 wscript.exe(基于图形用户接口的版本)。脚本使用这些程序中的一个来启动。这自动生成了 WSH 对象模型的 WScript 对象,只要脚本继续执行,对象就保持有效。相应地,这个对象具备了访问其他对象模型的方法和属性。它们的 ProgID 注册表设置提供了定位它们所需要的信息。

对象模型属于这两个类别中的一个,这取决于它们操作在 Windows 的哪个区域。系统类别对访问操作系统函数的模型进行分组。应用程序类别中的模型揭示了 Microsoft Office 应用程序的特性。

系统类别包括了下面的对象模型:

- Windows Script Host(WSH)——处理图形接口部件(文件系统和 URL 快捷方式,特殊文件夹):本地打印机、远程共享打印机和文件夹、注册表和系统环境变量。
- 活动数据对象(ADO)——提供了一种手段用于读取和修改关系型和面向对象的数据库(或其他数据源,只要它们可以通过一种称为 OLE DB 的较低级别的接口来访问)。
- 活动目录服务接口(ADSI)——允许访问不同的目录服务(Windows NT SAM Windows 2000 Active Directory、第三方基于 LDAP 的目录、NDS、NetWare bindery、Exchange 5.x DS)。
- 协作数据对象(CDO)——处理消息和协作应用程序。
- FileSystemObject(FSO)——一般用于处理物理文件系统(驱动器、文件夹和文件)。
- Windows 管理设备(WMI)——主要处理系统管理信息和事件监控。

在我们的脚本中,可以使用 WSH、FSO、WMI 以及在较小的范围内使用的 ADSI。有关这些模型的更详细的介绍请继续阅读下一章。

1.4 脚本编辑器

在我们开始进一步探究脚本语言和对象模型的基本原理前,让我们快速浏览一组工具,它们可以使编写脚本更加容易。

如前面所提到,脚本只不过是一些文本文件,因而任何 ASCII 文本编辑器都可以进行编写。Windows Notepad 可能是比较通用的编辑器,因为任何版本的 Windows 都免费配送了这个编辑器。它的主要缺点是缺少行编号,当调试较长脚本时,就变得特别的麻烦。如果你尝试运行包含一个错误的脚本,脚本引擎会生成一条错误信息,指出发生错误的代码行。如果使用 Notepad,你就要被迫一行一行地计算行数。如果你计划定期地创建更复杂的脚本,你可能想使用下面介绍的程序之一。

同样请记住,WordPad 或 Microsoft Word 都不是很好的选择,因为在默认情况下,它们都不是以 ASCII 格式保存文件。你可能会得到扩展的 ASCII 或 Unicode 字符,而脚本引擎是不能正确解释这些字符的。

除了行编号外,你也可能对一项被称为"语法着色"的特性感兴趣。当打开这种特性时,它导致了脚本语言的保留字高亮显示。这就给你提供了语法正确性的即时确认。

例如,Visual InterDev 6.0 提供了"智能感应",当你输入代码时,会显示可用的语法选项列表。

下面是一些可从网上下载获得的较高级的编辑器。

- PFExpress(免费软件):www.winsite.com/info/pc/win95/misc/pfe101i.zip。
- NoteTab(共享软件):www.notepad.com。
- EditPlus(商业软件):www.editplus.com。
- TextPad(商业软件):www.textpad.com。
- UltraEdit 和 UltraEdit32(商业软件):www.idmcomp.com。
- EditPad(商业软件):www.editpadpro.com。
- WinEdit(商业软件):www.windowware.com。
- PrimalScript(商业软件):www.primalscript.com。

1.5 VBScript 基础

对于部分读者来说,可能是第一次接触 Windows 脚本。为了理解和能够修改书中的脚本,有必要熟悉一下 VBScript、Windows 组件对象模型和 Windows Script Host。

下面的几节介绍了基本 Windows 脚本概念。然而请记住,这本书不想作为 VBScript 或 Windows Script Host 的手册或参考资料。如果你决定学习更多有关这方面的内容,可以参考这个主题的很多专著和网上资源。查看附录 B,以获取进一步的资料。

VBScript 来源于 Visual Basic for Applications,它的初衷是要让因特网环境中的 Web 开发者使用。正因如此,为了以更加安全的形式和较少的开销操作,需要修改它。这将导致删除了 I/O 能力,并失去了显式数据类型的支持。换句话说,使用 VBScript,直接访问本地驱动器是不可能的,代码中的每个变量都是作为 variant 创建的。在一般的程序设计语言中,使用变量前,你需要指定变量中将存储哪种数据类型(比如整型或字串)。Variant 类型可以存储任何类型的数值。

为了能运行起来,VBScript 脚本需要一个带有脚本引擎和解析器的主机环境。可以提供这些部件的软件有,Internet Explorer 和一个基于 HTML 的应用程序(在 Web 客户端)、运行在 IIS (Internet Information Server)上的活动服务器页或运行在 32 位 Windows 系统上的 Windows Script Host。解析器一行行地读取代码,然后校验它的正确性,接着把它传给运行时系统,最后运行时系统在宿主程序提供的环境中执行代码。

当使用 Windows Script Host,解析工作由 vbscript.dll 或 jscript.dll 完成(作为 WSH 安装程序的一部分包括进来),这取决于脚本是使用 VBScript,还是使用 JScript 编制的。运行时引擎可以是基于 GUI(wscript.exe)或基于控制台(cscript.exe)的。此外,WSH 环境包括了一个 WSH 对象模型(大多数对象在 wshom.ocx 中实现)。其他的对象模型可以通过 COM 自动化访问。

脚本构建单元

脚本由多行文本组成,它们是根据脚本语言的严格语法规则编写的。这些规则定义了各种部件,在脚本执行创建期间,它们用作构建单元。部件在复杂性和大小上会发生变化,从非常简单(比如变量和数据类型)到比较复杂(比如函数和子程序)。本节介绍了理解后面章节介绍的脚本所需要的部件。

1. 变量和数据类型

变量是值的容器,在脚本执行期间,容器中的值可以修改。变量的数据类型确定了变量中将

存储哪种值(整数、浮点数、字符串、对象,等等)。

变量可以使用 Dim 语句显式说明。例如:

```
Dim intVal
```

这条语句说明了名字为 intVal 的一个变量。

说明不是强制性的,但它可以通过在脚本的开始处放置如下的代码行来实现这种强制性。

```
Option Explicit
```

使用 Option Explicit 是好的编程习惯,因为它有助于预防误敲变量名而产生的错误。本书出现的脚本都一致地遵循这一规则。

如前面所提到的,VBScript 没有显式地把数据类型赋予变量。然而,你可以使用几个 VBScript 函数来检查存储在变量中的值类型:

```
Option Explicit
Dim intX
Dim blnY
intX = 5
blnY = IsNumeric(intX)
MsgBox blnY
```

在这个例子中,IsNumeric 将返回布尔类型值(True),并把它赋予 blnY 变量。这个值将通过 MsgBox 函数生成的消息框显示出来。

即使 variant 数据类型适用于所有变量,你也不能在一个单独操作中自由组合变量。例如,把一个整数和一个字符添加进来也是不允许的。你需要使用转换函数把变量中的值从一种数据类型转换到另一种数据类型。例如:

```
Option Explicit
Dim intX
intX = 65
MsgBox Chr(intX)
```

上述代码使用 Chr 函数把一个整数转换成相应的 ASCII 字符(消息框将显示 A,因为这是 ASCII 表中的第 65 个字符)。Asc("A")将做相反的转换,也就是,它将返回整数值 65。Cstr 函数将把一个整数值转换成一个字串,也就是,CStr(65)将产生一个两个字符的字串 65。

下面的例子将产生一个类型不匹配的错误信息,因为我们尝试把一个整数和一个字符添加进来——这不是一个合法操作:

```
Option Explicit
Dim intX
Dim strY
intX = 65
strY = "B"
MsgBox intX + strY
```

然而,如果我们使用转换,将生成正确的结果(字串 AB)。