PROGRAMMER TO PROGRAMMER™

Professional Windows PowerShell Programming

Snap-ins, Cmdlets, Hosts, and Providers

Windows PowerShell 高级编程

(美) Arul Kumaravel 等著
Jon White
冯权友 译

wrox™
WILEY
清华大学出版社

Windows PowerShell 高级编程

(美) Arul Kumaravel
Jon White 等著

冯权友 译

清华大学出版社

北 京

Arul Kumaravel, Jon White, Michael Naixin Li, Scott Happell, Guohui Xie, Krishna C. Vutukuri
Professional Windows PowerShell Programming: Snap-ins, Cmdlets, Hosts, and Providers
EISBN：978-0-470-17393-0

图书在版编目(CIP)数据
Windows PowerShell 高级编程/(美)库马拉尔(Kumaravel,A.)，(美)怀特(White,J.)等著；冯权友 译. —北京：清华大学出版社，2009.1
书名原文：Professional Windows PowerShell Programming: Snap-ins, Cmdlets, Hosts, and Providers
ISBN 978-7-302-18867-4
Ⅰ.W…　Ⅱ.①库…②怀…③冯…　Ⅲ.窗口软件，Windows—程序设计　Ⅳ.TP316.7
中国版本图书馆 CIP 数据核字(2008)第 173672 号

责任编辑：王　军　王　婷
装帧设计：孔祥丰
责任校对：成凤进
责任印制：杨　艳
出版发行：清华大学出版社　　地　　址：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　邮　　编：100084
社　总　机：010-62770175　　邮　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京四季青印刷厂
装 订 者：北京市密云县京文制本装订厂
经　销：全国新华书店
开　本：185×230　印　张：22　字　数：467 千字
版　次：2009 年 1 月第 1 版　　印　次：2009 年 1 月第1 次印刷
印　数：1～4000
定　价：48.00 元

译 者 序

在各种发行版的操作系统中，Windows 操作系统安装使用得最广泛、最受用户欢迎。但对系统管理工作者而言，Windows 一直有个小小的缺憾，就在于没能提供像 Unix Shell 程序那样的强大脚本支持，而使得系统管理员的工作效率不高。PowerShell 的问世彻底改变了这种局面。PowerShell 是构建在 Windows .NET 平台之上的，与.NET 框架紧密耦合，因此它不仅是一种脚本编程语言，还为系统管理员提供了功能丰富的编程开发环境，可以轻易操作系统中的 COM 组件。

目前，市场上已经出现了介绍如何使用 Windows PowerShell 来提高系统管理效率的书籍，但是从 API 层次介绍 PowerShell 编程的权威之作相当稀少。本书比 SDK 文档更加详尽地介绍了 cmdlet、提供者(provider)、snap-in、宿主(host)程序等方面的开发技术。本书针对 PowerShell snap-in 和宿主(host)程序开发人员而写，它从 Windows PowerShell 底层 API 开始介绍 PowerShell 程序设计。书中每个知识点都有配套的实例代码，可以从本书的合作站点上自由下载，极易验证。

本人是计算机专业的在读博士生，经常在 Linux 下进行脚本编程。接触 Windows PowerShell 一年多来，渐渐被它强大的功能所吸引。在 Windows 操作系统中，PowerShell 提供了能和 Linux 下的 bash 相媲美的强大功能。本书翻译过程中也加入了个人的使用体会，希望能和读者在 PowerShell 技术方面进行探讨交流，共同进步。

翻译过程中错误之处在所难免，敬请广大读者提供反馈意见，读者可以将意见发到 wkservice@vip.163.com，我们会仔细查读者发来的每一封邮件，以求进一步提高本书的质量。

译 者

2008 年 6 月

作者简介

Arul Kumaravel 是目前 Windows PowerShell 开发小组的负责人。从早期开始，他就参与了这个项目并领导该小组发布了第一版的 Windows PowerShell 产品。现在，他正领导着下一个版本的 PowerShell 产品的开发。Arul 从他年轻时学习 BASIC 编程开始，就着迷于计算机技术。他在 Iowa 大学和印度 Madras 工程学院都获得了计算机科学方面的硕士学位。在微软实习期间，他为 IE3 浏览器编写了第一个 JavaScript/VBScript 调试器，同时微软公司的氛围也留给了他很好的印象，他决定为微软效力而改变数以万计人的生活。在微软任职的 11 年里，他在各种各样的小组工作过，发布了很多个版本的产品，包括 IE 浏览器、Windows 操作系统、目录管理服务器。最近，由于对科学技术的商业运作感兴趣，Arul 开始在 Wharton 商学院刻苦攻读 M.B.A。他的邮件地址是 arulk@hotmail.com。

Jon White 是位软件工程师，生活工作在美丽的西雅图东郊，他也是微软 PowerShell 小组的创始成员。他的职业生涯开始于微软操作系统服务器版的管理工具组。10 多岁时，父亲在二手店给他买了一个 8088 型 PC，此后作为业余爱好者的他开始学习程序设计。那个二手 PC 内置 MS-DOS 2.0，它的 debug.exe 只含有 16 位反汇编器而没有汇编器。因此，Jon 在程序设计方面的第一次尝试就是把长长的字节表反汇编成一个反向查找的字典，以此将汇编程序手动地变成可执行二进制代码。更妙的是，他后来查出了 64 位 Windows 操作系统中 debug.exe 的漏洞。作为 PowerShell 小组的一员，在 2004 年，当他负责将小组的测试工作从 Perl 转移到 PowerShell 脚本上来时，他编写了该语言的第一个工作脚本。业余时间，他喜欢航海或者在自家后院里玩焰火。读者可以通过发送邮件至 jwh@microsoft.com 联系他。

Michael Naixin Li 是微软 PowerShell 小组产品测试方面的高级负责人，现在他正监管着 Windows PowerShell 2.0 的测试工作。加盟 PowerShell 小组前，Michael 在微软许多重大项目中工作过，这些项目包括 MSN 1.x 和 MSN 2.x 的开发、Windows2000 中 COM 服务组件的质量管理、NetDocs Web 用户接入、Hailstorm 中的 Web 服务以及 Windows Vista 中的软件许可证服务。在加入微软前，Michael 是上海科学与技术大学(现在叫上海大学)的助理教授，他在 Colorado 州立大学获得计算机科学博士学位。

Scott Happell 是一位有着 10 年工作经验的软件工程师与软件测试员。最初，他在新泽西参与一个 Internet 创业项目。由于对 PowerShell 小组的热爱，他加入微软并工作了 3 年。最近 Scott 离开微软而成为一名录音工程师和摇滚明星，同时他正试图通过使用 PowerShell 来帮助他创作音乐。

Guohui Xie(或称 George Xie)曾是微软 PowerShell 小组的高级开发人员，在该小组工作了 3 年，专注于 snap-in 模型和脚本语言方面的工作。最近他加入了微软移动设备小组从事移动设备管理方面产品的开发。在加盟微软前，George 曾在 Siebel 系统公司工作多年。

Krishna Chythanya Vutukuri 是微软 PowerShell 小组的软件开发人员。在加入 PowerShell 小组前，Krishna 在微软许多项目中工作过，其中包括 Windows Presentation Foundation 的开发。加盟微软前，Krishna 在惠普印度软件运营中心(HP-ISO)和 Wipro 科技公司承担过许多研发工作。他在印度 Pilani 的 Birla 科学与技术研究所获得信息系统方面理学硕士学位(工科)。

前　言

欢迎阅读本书。

2003 年，在微软的一个会议中心，来自微软管理控制台小组的工程师们组织了一个座谈会，他们举行了一个关于升级版 MMC 的原型演示活动，我有幸参加了这个会议。自微软发布第一个以 Internet 为中心的服务器操作系统，即 Windows 2000 Server 操作系统后，他们收到了来自 Windows 系统管理员用户大量的意见反馈，这些反馈并不都是称赞性的。那个原型演示就是微软公司早期针对这些反馈意见做出的回应之一。

自基于 DOS 的文本文件管理器开发以来，Windows 2000 Server 操作系统经过了长期的演化。在其发展过程中，有很长一段时间人们认为它只适合于用来处理邮件、管理一个 20MB 大小的硬盘，除此之外，别无他用。并且，Windows 2000 Server 操作系统的管理工作也是按照视窗概念的方式进行的，系统管理是通过大量的事件交互完成的。尽管 Windows 系统中已经存在大量的内置 API 函数和 COM 型 API 函数，遗憾的是，这两者之间并没有衔接起来。在 Linux 下，您可以书写外壳程序脚本代码来自动配置邮件和 DNS 服务器，但在 Windows 中，您要么手动配置，要么得先学习 C++和 COM 编程。

在 Windows 中，Visual Basic Script 和 JavaScript 技术的集成和合并一定程度上弥补了这个空白，但是 GUI 方式和命令行方式之间管理工作效率的差距并没有得到彻底的填补。传统的 Windows 脚本只是通过 COM 组件的一个子集与操作系统交互，而 GUI 程序则不但可以使用所有的 COM 组件、调用 Win32 API 函数，还可以直接调用内置的内核 API 函数(如果是任务管理器这类 GUI 程序的话)。因此，Windows 脚本的光芒一直以来都被 GUI 方式给掩盖了。

让我们回到那个原型演示会议现场。想象一下：听众们鱼贯而入，讲台上的工程师通过引用一个关于 PA 系统的笑话来作开场白。灯光暗了下去，演示开始了。他们演示的新型 MMC 原型系统是一个基于 GUI 的程序，该程序使用了一个命令行引擎作为它的 API 层。每一次节点扩展操作都变成了一次查询，每一次“确定”的点击操作都变成了一个命令。GUI 用户的每一个操作都百分百如实地以脚本方式显示在屏幕的底部。老程序员在座

位上紧张地微微颤动，那些高级经理人眼中则仿若看到了巨大的利润滚滚而来，会议的筹办人察觉到听众的反应后，手里拿着一块小点心，走到屋外悠闲地点燃了一支香烟。

这个演示宣告了三年后所谓的 Windows PowerShell 技术的开始。第一版的 Windows PowerShell 可以从网络上自由下载，同时也作为一个可选的组件集成到了 Windows Server 2008 操作系统中。Windows PowerShell 不但为各种类型的用户提供了一个丰富的编程环境，也为 Windows 用户从命令行方式过渡到 COM 组件以及其他未来技术奠定了基础。

在 PowerShell 有望成为年度黑马之际，本书也跟上时代的步伐，为您学习 Windows PowerShell 平台上的开发，提高系统管理的效率提供了全面指导。书中主要介绍了 PowerShell 1.0 开发平台中与 cmdlet 开发、扩展类型系统(Extended Type System，简称 ETS)、宿主(Host)、提供者(Provider)等概念有关的知识。

全书共八章。第 1 章简要介绍 PowerShell 入门知识；第 2 章介绍 snap-in 开发，给读者一个整体印象；第 3 章介绍扩展类型系统 ETS；第 4 章介绍 cmdlet 开发；第 5 章介绍提供者(provider)；第 6、7 章介绍宿主(host)API 及其运用；第 8 章介绍输出信息格式化配置文件。最后，附录中还给出了一些有用的内容。

本书读者对象

本书面向 PowerShell snap-in 和宿主程序开发人员，它从 API 层次起开始介绍 PowerShell 编程。本书的作者是 PowerShell 1.0 版的研发小组成员，本书比 SDK 文档更加详尽地介绍了 cmdlet、provider、snap-in、宿主程序以及可定制宿主等方面的开发技术。

源代码下载

在完成本书的示例时，可以选择手动输入代码或者使用本书附带的源代码文件。本书用到的所有源代码可以从 www.tupwk.com.cn/downpage 下载，也可以从 www.wrox.com 下载。进入该站点后，只需找到本书的名称(使用 Search 框或者书名列表)，单击本书的详细页面上的 Download Code 链接，就可以得到本书所有的源代码。

注意：

由于很多书有相似的名称，所以用 ISBN 搜索更为容易。本书的 ISBN 是 978-0-470-17393-0。

下载了代码后，用您喜欢的压缩工具把它解压缩。此外也可以去 Wrox 的主下载页面 www.worx.com/dynamic/books/download.aspx 找到本书或其他 Wrox 出版的书的代码。

勘误表

尽管我们竭尽所能来确保在正文和代码中没有错误，但错误难免会发生。如果您在 Wrox 出版的书中发现了错误(比如拼写错误或者代码错误)，我们将非常感谢您的反馈。发送勘误表将节省其他读者的时间，同时也会帮助我们提供更高质量的信息。

到 www.wrox.com 站点上，用 Search 框或者标题列表找到本书的名称，在详细页面上点击 Book Errata 链接就能找到本书的勘误表。在这个页面中可以看到所有被提交的本书的勘误表，它们是由 Wrox 的编辑发布的。在 www.worx.com/misc-pages/booklist.shtml 中有完整的书的列表，其中包括每本书的勘误表。

如果您在书的勘误表页面上没有看到您发现的错误，请将错误发送至 wkservice@vip.163.com。我们会检查这些信息，如果属实就把它添加到本书的勘误表页面上，并在本书的后续版本中更正错误。

p2p.wrox.com

如果想和作者或者其他人讨论，请加入在 http://p2p.worx.com 的 P2P 论坛。该论坛是基于 Web 的系统，您可以发布关于 Wrox 出版的书和相关技术的消息，与其他读者或技术人员交流。该论坛有预定功能，在您选择的感兴趣的主题有新帖子时，会邮件通知。Wrox 的作者、编辑、其他业界专家和像您一样的读者都会出现在这些论坛中。

在 http://p2p.wrox.com，您会找到很多不同的论坛，它们不但有助于您阅读本书，还有助于您开发自己的应用程序，加入论坛的步骤为：

(1) 到 http://p2p.wrox.com 上单击 Register 链接。
(2) 阅读使用说明，单击 Agree 按钮。
(3) 填写加入必需的信息和其他您愿意提供的信息，单击 Submit 按钮。
(4) 您将收到一封 email，描述如何验证您的账户和完成加入过程。

注意：

不加入 P2P 也可以阅读论坛里的消息。但是如果要发布自己的消息，就必须加入。

加入之后，就可以发布新的消息和回复其他用户发布的消息。可以随时在 Web 上阅读

论坛里的消息。如果想让某个论坛的新消息以 E-mail 的方式发给您，可以单击论坛列表里论坛名字旁边的 Subscribe to this Forum 图标。

要了解如何使用 Wrox P2P 的更多信息，请阅读 P2P FAQs，其中回答了论坛软件如何使用的问题，以及许多与 P2P 和 Wrox 出版的书相关的问题。要阅读 FAQs，单击任何 P2P 页面里的 FAQ 链接即可。

目　录

第 1 章 PowerShell 简介

Windows PowerShell 是.NET 平台之上基于对象的命令行外壳程序和脚本语言。PowerShell 为 Windows 平台上的 IT 事务管理工作提供了更高级的控制和自动化支持，更有利于提高 IT 专业人士和开发人员的工作效率。

市场上向 IT 专业人士介绍 Windows PowerShell 的书已经屡见不鲜，但从 cmdlet、提供程序(Provider)类和宿主(Host)类开发方面来介绍 PowerShell 开发技术的书却寥寥无几。本书从使用 Windows PowerShell 软件包的过程入手，向读者介绍其中的基本概念、常用组件以及开发技术，试图弥补前面提到的图书市场空白。对于那些意图扩展 Windows PowerShell 的功能或者使用 PowerShell 扩展自己程序功能的开发人员来说，本书是最佳的选择。

一般来说，在书写命令行工具时，程序员需要书写代码来完成参数解析、赋值绑定工作。此外，程序员还要书写代码，用来格式化命令输出信息。Windows PowerShell 为程序员提供了一个自带解释器的运行时引擎，简化了参数解析、赋值绑定等繁琐的工作。当输出对象需要显示时，PowerShell 也为程序开发人员提供了可定制的格式化功能。使用 Windows PowerShell 完成那些开发命令行工具时的常规事务，开发人员可以集中精力关注程序的业务逻辑问题，从而摆脱那些琐碎的小问题。

1.1 Windows PowerShell 设计原则

多年以来，广大用户对微软 Windows 操作系统上的系统管理工作提了许多意见，Windows PowerShell 就是对这些用户反馈信息的响应。最初，许多用户常常询问为什么像一些传统 Unix 系统上的外壳程序没有被授权并包含在 Windows 操作系统中。对于这个问题，我们认为，只有开发一种独立于那些传统外壳程序的全新外壳程序才能解决这个问题。这个想法进而分解为四条指导原则，这四个原则构成了设计 PowerShell 的指导思想。

1.1.1 保留用户已有的投资

一项新技术发布后，必须经历一定的时间才能被广泛采用。此外，客户很可能在原有技术上已经投资了很多，要他们抛弃已有的投资是不现实的。因此，PowerShell 在本质上与现有的 Windows 管理技术完全兼容，保留了用户已有的投资。

事实上，在 PowerShell 环境中，操作系统原有的命令和脚本可以直接运行。PowerShell 除了和.NET 紧密结合外，它和 COM、WMI 和 ADSI 技术也几乎是无缝集成的。PowerShell 提供了一个统一的操作环境，用户可以处理前面提到的各种对象，这是 PowerShell 的最大特色。在本章后面的 PowerShell 快速入门部分，您可以看到本设计原则和其他设计原则相关的代码演示。

1.1.2 提供一个功能强大、面向对象的外壳程序

CMD.exe 和其他的外壳程序都是基于文本的，也就是说在这些外壳程序中，命令接收文本输入，产生文本输出。在内部处理时，这些命令把文本转化为其他对象，但仍以文本方式进行输出。在传统外壳程序中，当许多简单的命令通过管道连接起来时，命令之间需要进行许多的文本处理工作才能产生需要的输出。这方面的工具比如 SED、AWK 和 Perl，由于它们出色的文本处理能力，很受命令行脚本程序员的青睐。

.NET 之上的 PowerShell 是一种面向对象的外壳程序和脚本语言。当用管道连接命令时，PowerShell 在命令管道上传递的是各种对象。各 cmdlet 之间可以直接处理和传递对象。由于与.NET 紧密结合，IT 专业人士不用学习掌握如 C#和 VB.NET 之类的高级编程语言，就可以通过 PowerShell 来体验.NET 的强大功能和程序设计体验的一致性。

1.1.3 扩展性是第一位的

PowerShell 设计目标之一是提供更高级的 Windows 操作系统环境管理控制功能并且加速系统管理工作的自动化，提高 IT 管理人士的工作效率。PowerShell 兼容已有的命令和脚本，很容易推广，因此系统管理员不用学习任何新知识，就可以立即使用它。系统管理员可以方便地使用外壳程序和脚本语言。

PowerShell 中所有的命令都叫作 cmdlet(读作“commandlet”)，它采用动-名词结合的命名方式，比如 Start-Service、Stop-Service 或者 Get-Process、Get-WMIObject 等。动-名词结合的方式使得各种命令简单易学。PowerShell 系统内部包含了一百多个与系统管理有关的命令和工具。此外，PowerShell 还是一门强大的脚本语言，支持编写各种风格的脚本，不论是简单的还是复杂的。系统管理员可以一边编写简单的脚本，一边学习该语言。由于其复合的功能和简单易上手的特性，PowerShell 为系统管理员的日常工作提供了一个强大的管理环境。