第 2 章　初识 PowerShell

本章将协助读者选择一种最适合的 PowerShell 界面（不错，你可以做出选择）。如果你曾经使用过 PowerShell，可以直接跳过本章，但是你阅读依旧可以从本章中找到一些对你有帮助的信息。

2.1　选择你的“武器”

微软提供了两种（如果你是很严谨的人，可以认为是四种）使用 PowerShell 的方式。图 2.1 显示了【开始】菜单中的【所有程序】界面，其中包含四种 PowerShell 图标。可以通过图中划线部分快速找到这些图标。

提示：在旧版本的 Windows 中（本书环境基于 Windows Server 2012），这些图标位于【开始】菜单中，可以通过依次选择【所有程序】>【附件】>【Windows PowerShell】来找到它们。除此之外，还可以在【开始】菜单中运行“PowerShell.exe”，然后单击【确认】，打开 PowerShell 的控制台应用程序。在 Windows 8 和 Windows Server 2012 中，使用 Windows 键（通常位于 Ctrl 键和 Alt 键之间的 Windows 图标）加 R 打开运行对话窗口，或者单击 Windows 键，然后在输入框中输入 PowerShell，即可快速打开 PowerShell 图标。

在 32 位操作系统中，最多只有两个 PowerShell 图标。在 64 位系统中，最多有 4 个。它们分别是：

- Windows PowerShell——64 位系统上的 64 位控制台和 32 位系统上的 32 位控制台。
- Windows PowerShell(x86)——64 位系统上的 32 位控制台。
- Windows PowerShell ISE——64 位系统上的 64 位图形化控制台和 32 位系统上的 32 位图形化控制台。
- Windows PowerShell(x86)——64 位系统上的 32 位图形化控制台。

图 2.1 你可以选择四种 PowerShell 启动方式的其中一种

换句话说，32 位操作系统仅有 32 位的 PowerShell 应用程序，而 64 位操作系统可以有 32 位和 64 位两个版本的 PowerShell 应用程序，其中 32 位应用程序在图标名中会包含“x86”字样。需要注意的是，有些扩展程序只支持 32 位环境，不支持 64 位。微软现在已经把全部精力放到 64 位系统中，而 32 位仅用于向后兼容。

提示： 在 64 位系统中，人们经常会错误地打开 32 位应用程序，此时应该注意窗体的标题，如果显示“x86”，证明你在运行 32 位程序。另外，64 位扩展程序不能运行在 32 位应用程序中，所以建议用户把 64 位应用程序以快捷方式的形式固定在开始菜单中。

2.1.1 控制台窗口

图 2.2 展示了 PowerShell 控制台窗口界面，这是大多数人第一次见到的 PowerShell 界面。接下来从使用简单的 PowerShell 控制台命令和参数开始本小节。

- PowerShell 不支持双字节字符集，也就是说，大部分非英语语言不能很好地展示出来。
- 剪切板操作（复制和粘贴）使用的是非标准键，意味着使用起来较为不便。
- PowerShell 在输入时会提供少量帮助信息（这个相对于 ISE 而言，在下面即将介绍）。

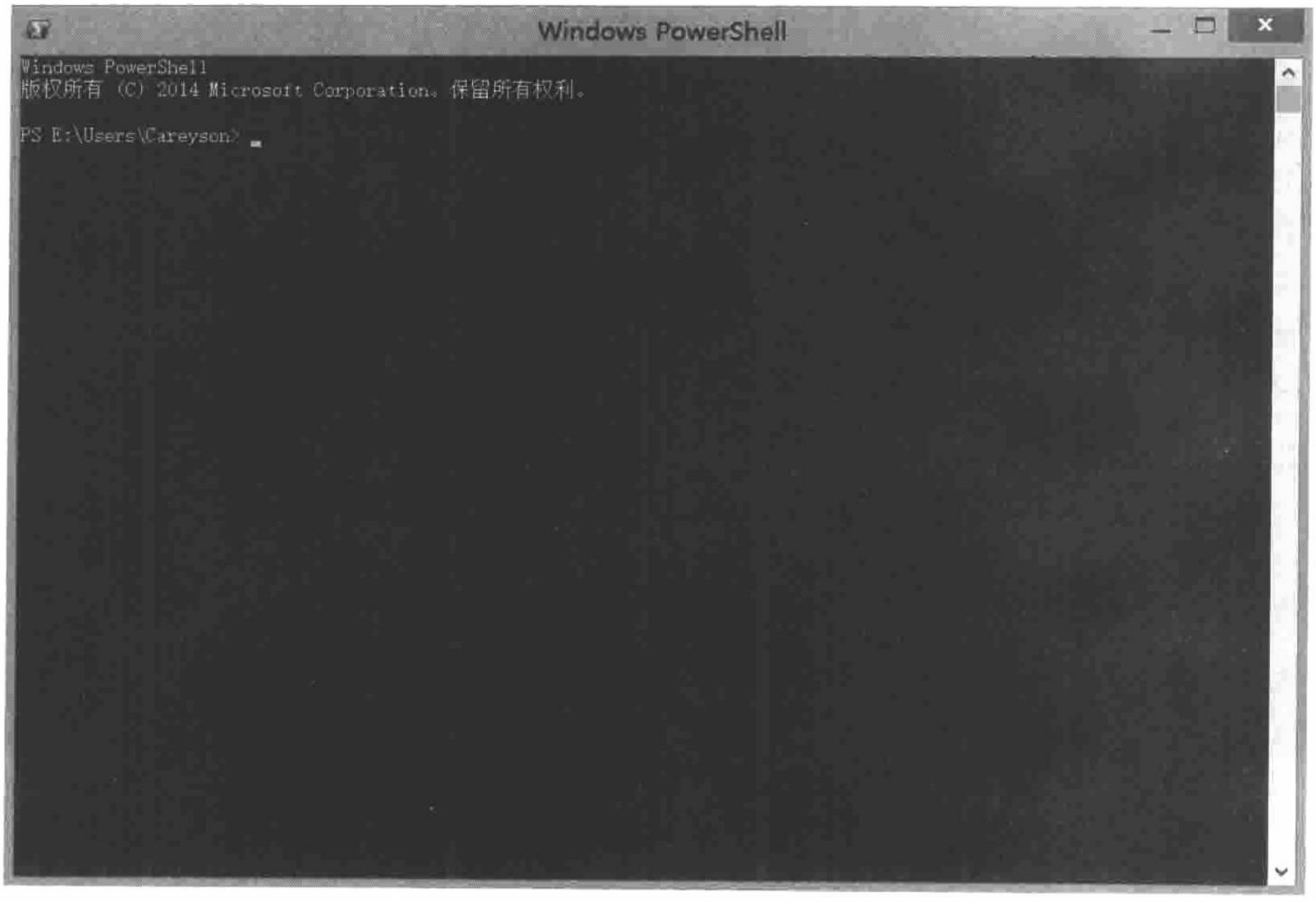

图 2.2 标准的 PowerShell 控制台窗口：PowerShell.exe

综上所述，PowerShell 控制台应用程序将是你在没有安装 GUI Shell 的服务器上运行 PowerShell 的唯一选择（如一些“服务器核心功能”安装或者 Windows Server 中服务器 GUI Shell 功能被移除或没有安装的情景）。其优点是：

- 控制台程序非常轻量，可以快速加载且不需要太多内存。
- 不需要任何非 PowerShell 自身必需的.NET Framework 之外的资源。
- 可以在黑色背景中设置绿色字体，正如在 20 世纪 70 年代的机器上工作一样。

如果你打算使用控制台应用程序，在你配置时会有些建议可供参考。可以通过单击窗体左上角的图标，并选择【属性】来实现，如图 2.3 所示。

在【选项】标签页，可以调大“命令记录”的缓冲区大小。这个缓冲区可以记住你在控制台输入的命令，并且通过键盘的上、下键重新调用它们。你也可以通过按 F7 键来弹出命令列表。

在【字体】标签页，选择稍微大于默认 12 像素的字体。不管你是否拥有 1.5 的视力，稍微提高一下字体大小也没什么坏处。PowerShell 希望你能在大量类似的字符中快速区分它们，比如'（撇号或单引号）和`（重音符）。如果使用小像素字体，识别这类字符将比较困难。

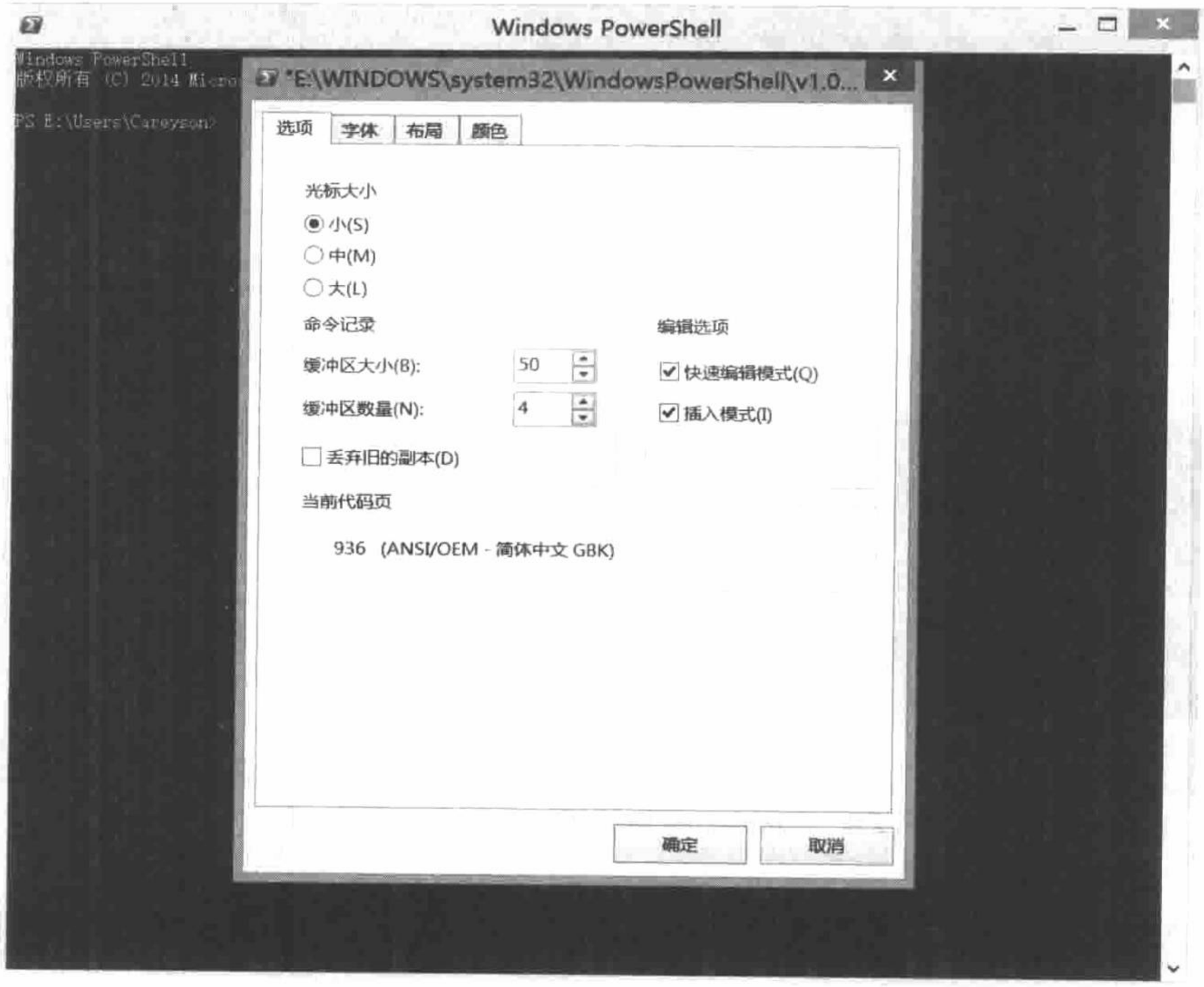

图 2.3 配置控制台应用程序的属性

在【布局】标签页，把所有“宽度”设为相同的数值，并且确保结果窗体能适合你的显示屏。如果设置不合理，会导致 PowerShell 窗体下方出现水平滚动条。这可能导致一部分输出结果被挡住，从而忽略了它们的存在。作者的学生就曾经花了半小时来运行命令，但是却完全没有输出，实际上输出被隐藏在右边。

最后，在【颜色】标签页，强烈建议不要修改，保持高度反差将有助于阅读。如果你不喜欢默认的蓝底白字，可以考虑中灰底黑字的形式。

需要记住一件事：这个控制台应用程序并不是真正的 PowerShell，仅仅是你和 PowerShell 交互的界面。控制台应用程序本身可以追溯到大约 1985 年，所以你不要指望能从中得到流畅的体验。

2.1.2 集成脚本环境（ISE）

图 2.4 展示了 PowerShell 集成脚本环境，也称 ISE。

提示：如果你不经意打开了普通的控制台应用程序，可以输入 “ise” 并按回车键，从而打开 ISE。

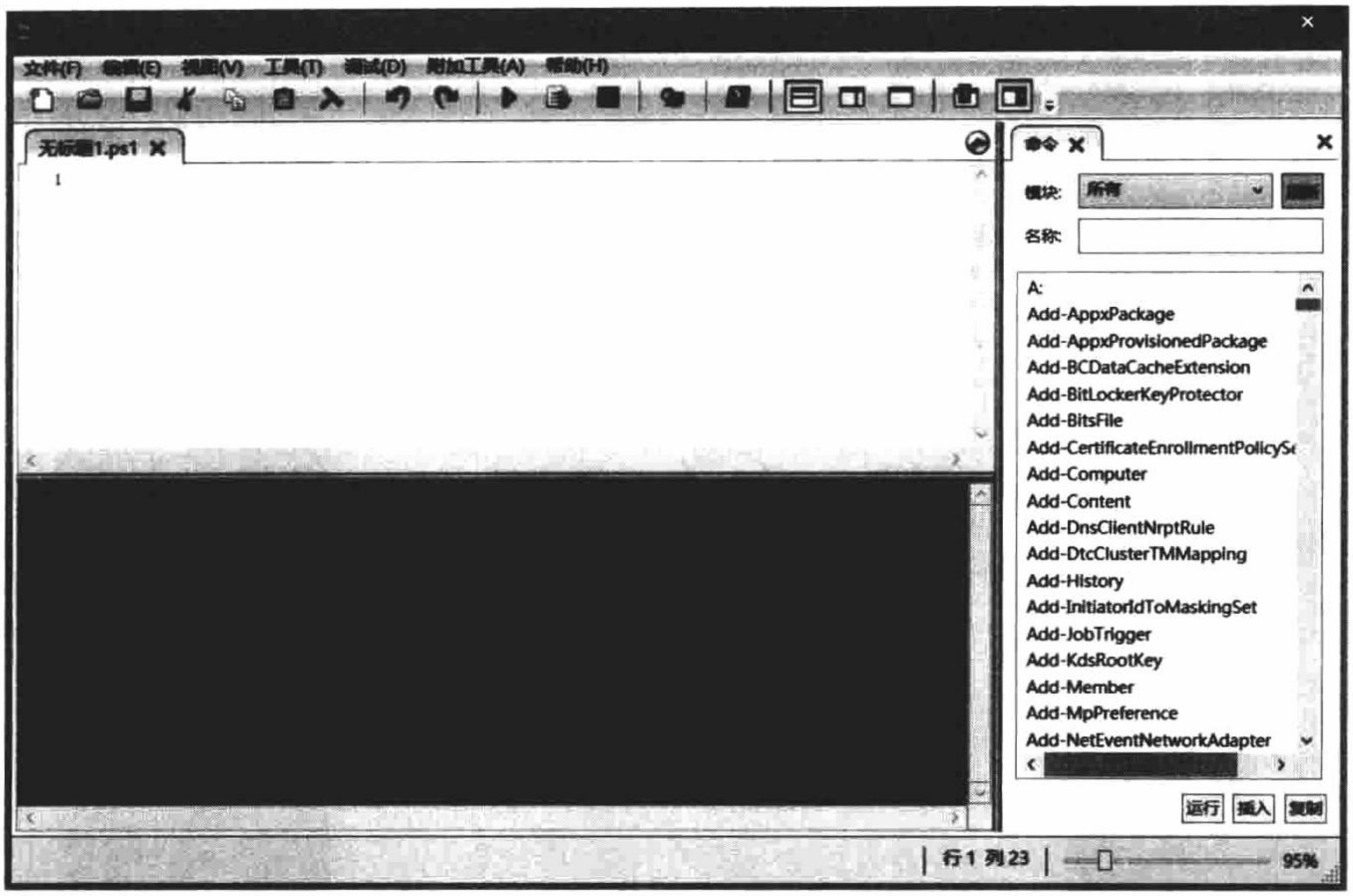

图 2.4 PowerShell ISE（PowerShell_ISE.exe）

表 2.1 ISE 的优缺点

优点	缺点
ISE 界面友好且支持双字节字符集	ISE 要求 Windows Presentation Foundation（WPF），意味着不能在没有安装 GUI 的服务器上运行 ISE
在后续章节可以看到 ISE 能在你创建 PowerShell 命令和脚本时提供更多的帮助	启动和运行需要较长时间，但是这通常只是几秒的差异
ISE 使用常规的复制、粘贴按键	ISE 不支持转录

表 2.1 列出了 ISE 的优缺点，从中可以得到大量背景信息。

下面从一些基本定位开始。图 2.5 展示了 ISE 的三个主要区域，图中划线部分即为 ISE 的工具栏。

在图 2.5 中，最上方的区域是【脚本编辑窗格】，直到本书最后才会用到。在它的右上角，可以看到一个蓝色的小箭头，单击它可以最小化【脚本编辑窗格】并最大化【控制台窗格】。控制台窗口是我们将要使用的地方。右边是【命令管理器】，可以通过它最右上方的“X”打开或者关闭这个窗口。除此之外，可以通过工具栏倒数第二个按钮来浮动【命令管理器】。如果你已经关闭【命令管理器】又想让它重新出现，可以单击工具栏的最后一个按钮。工具栏中的前三个按钮用于控制【脚本编辑器】和【控制台窗格】的布局。可以通过这些按钮把窗体设置为【在顶部显示脚本窗格】、【在右侧显示脚本窗格】和【最大化显示脚本窗格】。

在 ISE 窗口的右下角，可以发现用于改变字体大小的滚动条。在【工具】菜单中，可以找到一个【选项】项用于配置定制化的颜色方案和其他显示配置——这完全根据你的喜好而定。

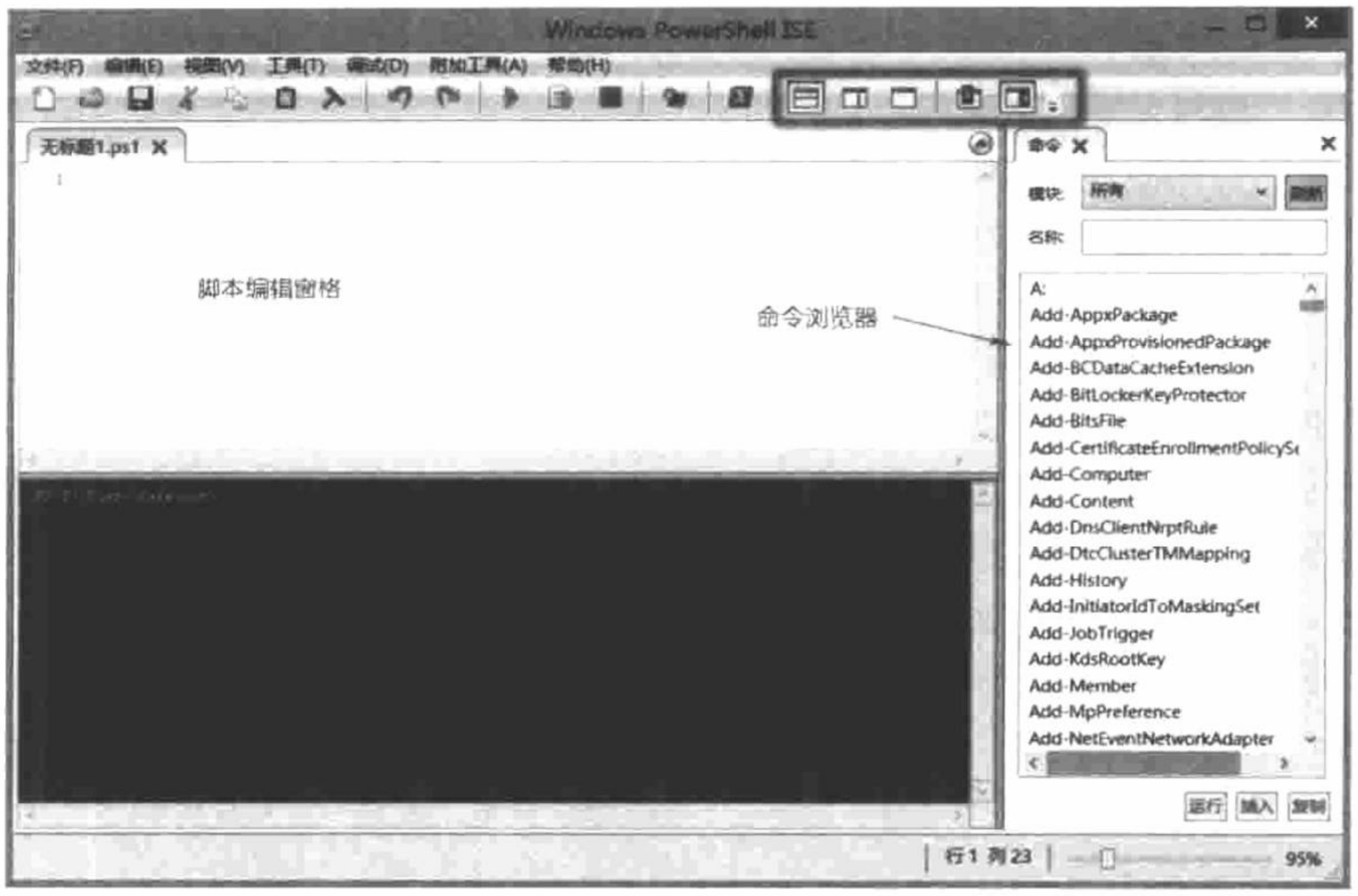

图 2.5 ISE 的三个主要区域及控制它们的工具栏

动手实验： 首先我们假设读者将会在余下章节中只使用 ISE，然后隐藏【脚本编辑窗格】。如果你愿意，也可以把【命令管理器】隐藏。把字体大小设置到你喜欢的样子。如果你不能接受默认的颜色方案，请自行选择。如果你更喜欢控制台应用程序，请放心使用，本书的绝大部分内容同样能在控制台中运作。一些仅在 ISE 中才能使用的功能将会额外标注。

2.2 重新认识代码输入

PowerShell 是一个命令行接口，意味着你需要大量输入代码。然而输入命令就意味着可能出现错误，例如拼写错误。幸运的是，所有 PowerShell 应用程序都提供了最小化错别字的方式。

动手实验： 接下来的例子在本书中可能显得不太实际，但是在本节看来却很炫。读者可以在自己的环境中尝试一下。

控制台应用程序支持四种“Tab 键补全”。

- 输入“Get-S”，然后按几下“Tab”键，再按 Shift+Tab 组合键。PowerShell 会循环地显示以 Get-S 开头的所有命令。然后不停按 Shift+Tab 组合键，直到出现你期望的命令为止。

- 输入“Dir”，按空格键，然后输入 C:\，再按“Tab”键，PowerShell 会从当前文件夹开始循环遍历所有可用的文件和文件夹名。
- 输入“Set-Execu”，按“Tab”键，然后输入一个空格和横杠（-），再开始按“Tab”键，可以看到 PowerShell 循环显示当前命令的所有可用参数。另外，也可以输入参数名的一部分，如-E，然后按“Tab”键，开始循环匹配的参数名。按“Esc”键可以清空命令行。
- 再次输入“Set-Execu”，按“Tab”，再按空格键，然后输入-E，再次按“Tab”键，然后按一次空格键，再按“Tab”键。PowerShell 会循环显示关于这些参数的合法值。这个功能仅对那些已经预设了可用值（称为枚举）的参数有效。按“Esc”键同样可以清空命令行。

PowerShell ISE 提供了类似功能，甚至可以说比“Tab 补全”功能更好的功能——智能提示。这个功能在前面四个情况下都能运作。图 2.6 演示了如何通过弹出菜单来实现你在使用“Tab”键时完成的功能。可以使用上下箭头按钮来滚动菜单，找到你想要的选项，然后按“Tab”或者按“Enter”键来选择，再继续输入剩余代码。

智能提示可以工作在 ISE 的控制台窗格和脚本编辑窗格中。

警告： 当你在 PowerShell 中输入时，请极其小心。在某些情况下，一个错位的空格、引号或者单引号都会带来错误或者失败。如果出现了错误，请再三检查你的输入内容。

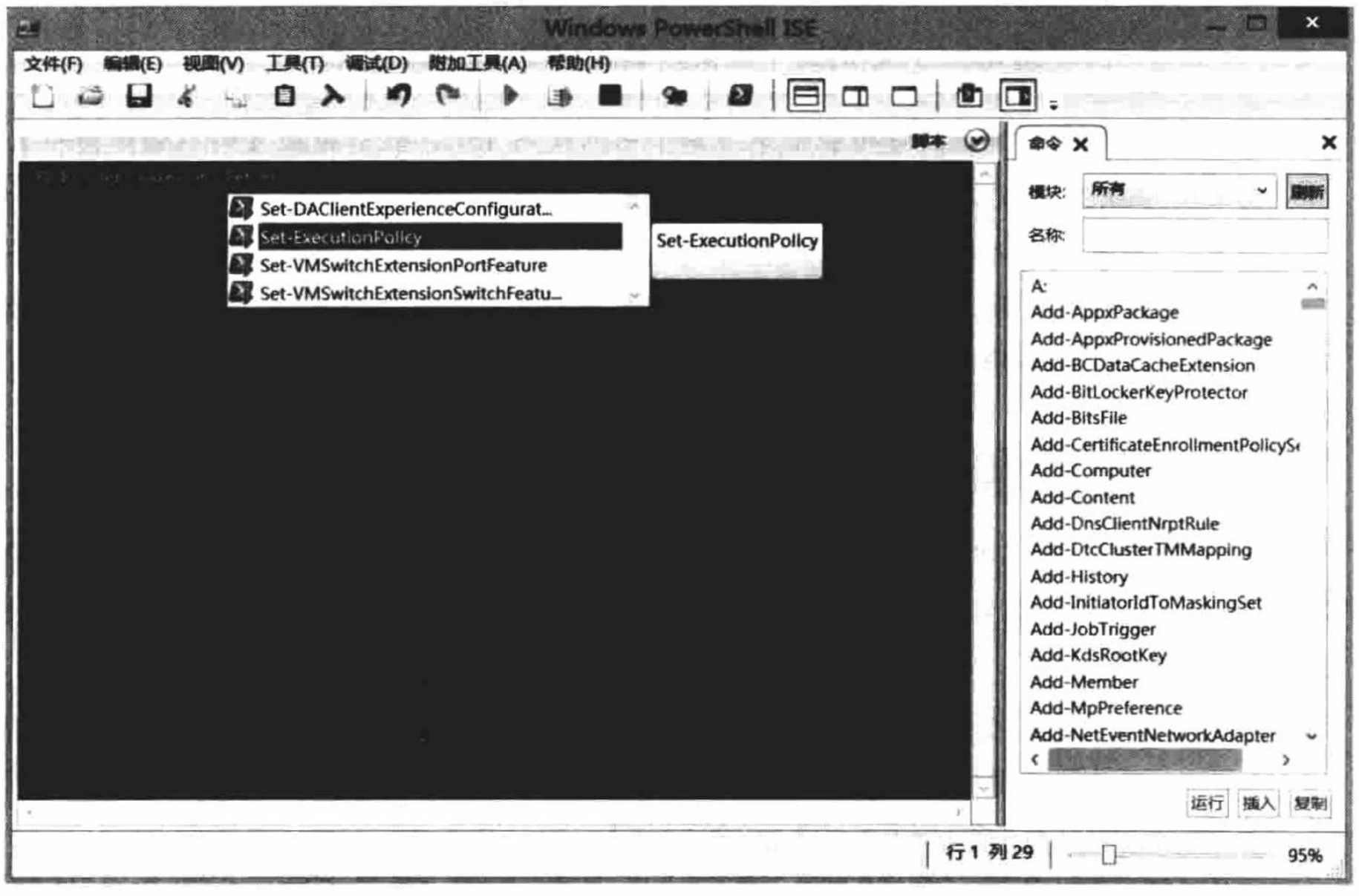

图 2.6 在 ISE 中类似 Tab 自动补全功能的智能提示功能

2.3 常见误区

接下来让我们快速回顾一些会影响你享受 PowerShell 旅途的绊脚石。

- 在控制台应用程序中的水平滚动条——从多年的教学经验中我们得知，正如前面提到过，配置控制台的窗口，使其不出现水平滚动条是非常重要的。
- 32 位 vs 64 位——建议你使用 64 位的 Windows 并使用 64 位的 PowerShell 应用程序（没有出现“x86”字样的应用程序）。虽然对于某些人来说，购买 64 位的计算机和 64 位的 Windows 可能是件大事，但是如果你希望 PowerShell 高效运行，那么这些投资还是必须的。虽然在本书中我们尽可能覆盖 32 位环境，但是这些内容在 64 位的生产环境上将带来很大的差异。
- 确保 PowerShell 应用程序的窗体标题显示“管理员”——如果你发现打开的窗体上没有“管理员”字样，关闭窗体并右键单击 PowerShell 图标，选择“以管理员身份运行”。在生产环境中，不一定总是要这样。本书后面将演示如何使用特定的凭据运行命令。但是通常情况下，为了避免运行时出现一些问题，最好确保以管理员身份运行 PowerShell。

2.4 如何查看当前版本

在很大程度上，找出当前使用的 PowerShell 版本不是件容易的事，因为每个发布版本都安装在“1.0”的目录下面（1.0 是引用的 Shell 引擎语言版本，即所有版本都向后兼容到 v1）。针对 PowerShell，有一种简单的方式检查：

```
PS C:\> $PSVersionTable
Name                           Value
----                           -----
PSVersion                      3.0
WSManStackVersion              3.0
SerializationVersion           1.1.0.1
CLRVersion                     4.0.30319.17379
BuildVersion                   6.2.8250.0
PSCompatibleVersions           {1.0, 2.0, 3.0}
PSRemotingProtocolVersion      2.2
```

输入$PSVersionTable，然后按“Enter”键。可以看到每个 PowerShell 相关技术的版本号，包括 PowerShell 自身的版本号。如果命令不能运行，或者显示最少需要 PSVersion 3.0 等字样，则需要使用第 1 章中展示的方式安装最少 3.0 的版本。

动手实验：现在就开始使用 PowerShell，首先检查你的 PowerShell 版本是否达到最少 3.0 版本，如果不是，请先升级到最少 3.0 版本。

2.5 动手实验

因为这是本书第一个实验，所以我们会花一些时间去描述它们是如何运作的。对于后续的每个实验，我们会给出一些任务，以便你可以自己动手尝试和完成。一般我们只给出一些提示或者方向指引。所以从现在开始，你只能靠自己了。

我们保证所有需要用于完成实验的知识仅限于当前章节或之前的章节（对于之前章节的知识，我们一般采用提示的方式给出）。我们不会把答案说得太明显，更多地，当前章节会告诉你如何发现你所需要的信息，你需要自己去发现这些问题的答案。虽然看起来有点让人沮丧，但强迫自己去完成，从长远来说绝对可以让你在 PowerShell 的世界里面走得更好。

顺带提醒，你可以在 MoreLunches.com 中找到一些示例答案。这些答案不一定完全匹配你的问题，但是当我们一步一步地深入之后，答案将变得越来越准确。实际上，我们会发现 PowerShell 针对几乎所有的问题都能提供几种甚至更多的方式实现。我们会尽可能地使用最常用的方式，但是如果你尝试使用另外一些不同的方式，并不代表你是错误的。任何能实现结果的方式都是正确的。

注意：本实验需要 PowerShell v3 或以上版本。

我们从简单的例子开始：希望你能从控制台和 ISE 的配置中实现相同的结果。然后按照下面五步进行。

1. 选择适合你自己的字体和颜色。

2. 确保控制台应用程序下方没有水平滚动条。（本章中已经第三次提到，可见其重要性。）

3. 在 ISE 中，最大化控制台窗格，移除或最小化命令管理器。

4. 在所有应用程序中，输入一个单引号“'”和一个重音符“`”，确保你可以轻易识别出来。在美式键盘中，重音符位于左上角，在“Esc”键下面，和波浪号“~”位于同一个键中。

5. 同样输入括号（），中括号[]，尖括号<>和花括号{}，确保你所选择的字体和大小能很好地展示这些符号，足以让你马上识别出来。否则，请选择其他字体或者加大字体大小。

前面已经提到过这些步骤，所以你对此应该没有任何疑问，你要做的只是一步一步地完成。

2.6 进一步学习

除了微软提供 PowerShell 之外，你会发现其他针对 PowerShell 定制的免费或

商用的编辑工具。下面提供常见的几种。你可以去试用。就算商业版也会有试用期，所以不妨去尝试一下。

- *SAPIEN PrimalScript* 和 *PrimalForms*——来自 http://primaltools.com 的两个商业版工具。
- *Idera PowerShell Plus*——来自 http://idera.com 的编辑器与控制台环境。

你可能找到其他工具，但是这四种工具的用户群是最多的，而且是我们最常用的工具。我们和这些公司没有任何关系（仅仅是欣赏他们的成果）。经常有人问，这些软件里面哪个用得最多。此时我们只能说使用 ISE 最多，因为我们经常重建虚拟机，并且懒得重新安装这些编辑器（而且我们也没有时间去为此专门写一个 PowerShell 脚本）。即便如此，当我们的确要确定使用一个第三方工具时，通常是 PowerShell Plus，因为我们喜欢它提供的增强版控制台而不仅仅是脚本编辑器。对于这种集成，我们深表欢迎。但是还是建议读者根据需要和预算选择这些工具。

2012 年 1 月，Don 评论了各种各样的 PowerShell 环境。假如你在 2014 年 1 月开始阅读本书（此时文章可能过时，并且根据当前可用产品而更新），可以从下面链接中查看：http://ConcentratedTech.com。它成文于 PowerShell v3 时代，并且适用于文章中提到的产品。不管怎样，它还是可以作为学习 Shell 脚本的不错的起点。