



VMware 脚本与命令 (PowerCLI) 管理手册

VMware 脚本与命令 (PowerCLI) 管理手册

由于 VMware 是个成熟的虚拟化平台，所以它拥有几个自带的和第三方的管理选项。图形用户界面 GUI 提供了直观的、概念上的管理 VMware 环境的方式，但这些对于大型或者重复的任务来说就略有不足。如大量贮藏和虚拟机配置这些任务最好使用 VMware 脚本和命令套件。在本期虚拟化技术手册中，我们将详细介绍 VMware 脚本和命令工具，如 PowerShell 与 PowerCLI，以便有效监控 VMware 环境。

PowerCLI 入门

VMware 管理任务是一项耗时和易出错的工作。但是 vSphere PowerCLI 的扩展，可以帮助管理员完成一些控制工作。本部分将探究 VMware 脚本工具 PowerCLI 的概念以及其中五个比较重要的脚本。

- ❖ 如何使用 VMware vSphere PowerCLI?
- ❖ 五大必备 vSphere PowerCLI 脚本

PowerCLI 使用与管理

如果您刚刚开始应用 PowerCLI 管理 VMware 环境，需要学习的内容很多。首先，最常见和最重要的 cmdlets 是 Get-VM。那么它该如何使用？如何设置能自动化主机服务器任务呢？要创建 host profiles，PowerCLI 能做些什么？这部分中，我们 TechTarget 中国的特约作者 Hal Rottenberg 将详细介绍 PowerCLI 的使用与管理技巧。

- ❖ 掌控 PowerCLI：使用 Get-VM 来管理虚拟机
- ❖ 使用 PowerShell 与 PowerCLI 自动化主机服务器任务

-
- ❖ 跳出框外 巧解 PowerShell 与 PowerCLI 难题
 - ❖ 如何使用 vSphere PowerCLI 创建 host profiles?
 - ❖ 在 vSphere PowerCLI PowerShell 界面使用 host profiles

PowerCLI 技巧

如何把 VMware PowerCLI 脚本功能应用到 VMware SRM (Site Recovery Manager) 恢复计划中? 如何借助 PowerCLI 配置标准交换机? 本部分将给出解决方法与步骤。

- ❖ 如何用 VMware PowerCLI 脚本来定制 SRM 恢复计划
- ❖ VMware 管理员必备: 借助 PowerCLI 配置标准交换机

VMware 脚本和命令

无论是使用 VMware ESX 还是 ESXi, VMware 命令行和脚本——如 vMA、Power CLI 和 vSphere Health Check——都可以辅助配置虚拟机、更好地管理 VMware vSphere 基础架构和通过命令行自动完成管理工作。

- ❖ 最佳免费的 VMware 脚本和命令行工具介绍
- ❖ ESX 与 ESXi 管理员必备 25 个命令

如何使用 VMware vSphere PowerCLI？

尽管我要再次申明乏味的 VMware 管理任务是一项耗时和易出错的工作。但是 vSphere PowerCLI 的扩展，可以帮助管理员完成一些控制工作。在这篇文章中，TechTarget 中国的特约虚拟化专家 Hal Rottenberg 将探究 VMware 的这个脚本工具是如何帮助我们自动地完成管理任务的。

作为一名拥有多年经验的系统管理工程师，我认为，管理员的主要工作（除了那些没有显示屏也不需要人员值守的环境）就是通过移动鼠标或者在键盘上键入指令来完成人机交互的过程。您可能此时会有疑问，这有什么问题吗？我要告诉你的是，这会导致很多的问题发生。首先，阐明几件作为 VMware 管理员都会面对的事情。

- 我们的工作需要非常的细致
- 重复工作会导致人为错误发生，而且
- 无论您的输入速度多么快，计算机总是更快

我想以上的这些情况是大家都认可的事实，无论在什么情况下，作为一名计算机用户都要面对的操作。和我们需要完成的任务是什么而无关，对吧？接下来让我们回到之前的假想环境来说明为什么这个问题会困扰我们。需要指出的，最关键的一点是由于自动化或者是缺少自动化而引起的。

为什么是自动化？

人类创造了计算机用于解决问题。通常讲，您的工作是基于其他的开发者进行的（开发者很多时候也是问题的创造者，他们发现了问题，但没有让问题一直遗留）。开发人员希望通过提供底层的指令系统来解决问题，同时提供给上层使用者一个更高层的和计算机交互的接口和工具。我知道这些都是最基本的理论，请跟着我的思路继续。

有一个非常流行的设想：系统管理员所需要的，仅仅是最上层的接口程序，所有的内脏都隐藏在那些漂亮的图形界面接口（GUI， graphical user interfaces）后面。例如，让我们以 vSphere client 来举例说明。我喜欢这个应用，因为它可以很好的工作并解决问题。但是如果您需要改变十二台虚拟机的端口设置时我们需要怎么完成？让我来为您做一道算术：每台虚拟机的设置需要点击六次鼠标，总共需要重复这个无聊的鼠标点击动作 72 次。这种无聊的重复过程可能并且经常会导致人为错误的发生。无论您点击得有多快，都需要有您的眼睛跟踪 GUI 控制界面的时间和通知您的大脑把鼠标移动到正确地点的时间。所有的这些工作如果某次点击出现了几个象素的偏差，可能会使您选择一个错误的网络。

我想说到这里您应该明白我想说什么了。我有更好的解决办法，当然，那就是通过 PowerCLI 实现。

什么是 PowerCLI

让我们来假设，此刻，您不是什么 Perl 语言的狂热爱好者（开个玩笑）。总之，如果您现在用 Perl 来管理您的 vSphere 环境，您可能比较难以想象自动化，在使用脚本语言时的重复过程，这当然很好。我将呈现给您的将是一种简短和便捷的操作方式，所以请继续往下读。

在开始讲述 PowerCLI 之前，我们必须首先了解关于 PowerShell 的一些内容。PowerShell 是一种由微软设计开发的，面向对象的脚本语言程序和命令行 shell，用于取代 VBScript 和以前的 CMD. EXE 程序；它是面向对象的（什么是对象，稍后讲述）；它是一种和其他面向对象系统（如：虚拟架构）建立连接的非常有效的方式。一台虚拟机就是一个对象，也是一个主机。和 VBScript 不同，PowerShell 是一个 shell，也就是说您可以运行它并且直接通过它发布命令。

但是，更重要的是，PowerShell 是一个脚本语言。脚本语言和程序语言类似，我们通过文本的形式提供指令，然后计算机运行它并完成您需要的工作。脚本语言又与程序语言不完全相同，脚本语言相比而言是一种更高层次的计算机交互方式，所以程序员可以较少地关注那些理论上的概念。换句话说，脚本语言位于程序语言和图形界面之间。同样的，它也是两者之间的粘合剂，或者是在一个复杂的 IT 系统环境中的位于两者之间的中间层。（PowerShell 可以比作是饭后的甜点，或者是地板上打得蜡。）

好了，背景故事已经讲了不少！PowerCLI 是 PowerShell 的扩展。PowerCLI 内部带有 165 个 cmdlets（读作 command-lets），它们每一个都是一个独立的可以分解为字节大小的 vSphere 管理任务。Cmdlets 使用了一些简短的逻辑词条的名称，以便于人们学习和使用。接下来让我来简单展示一下。

Show me the money

让我们先来简短地通过这个核心短语来理解“面向对象”的含义。对象指的是像汽车、房子和虚拟机等等这样的物品。对象是由其他部分所组成的，例如方法和属性。举例来说，方法指的是“加速汽车”，“打开房门”和“启动虚拟机”这样的操作。属性指的是：“320 马力”，“5 个房门”和“2GB 内存”这样的一些问题。好的，现在是否依然还跟我的思路继续？

PowerCLI 的一个简单的例子就是，Get-VM cmdlet，一个事实上非常有用的命令。下面是这个命令下达后的执行情况：

(Note that "PS >" indicates the shell prompt, and the text that immediately follows is what I typed. Everything else is output from a previously typed command.)

```
PS > Connect-VIServer 192.168.0.109
Name          Port          User
----          ----          ----
192.168.0.109      443          root

PS > Get-VM
Name          PowerState  Num CPUs Memory (MB)
----          -----
vm1          PoweredOff  1          4
vm2          PoweredOff  1          4
vm3          PoweredOff  1          4
```



点击图片本身放大

或许您已经猜到，如果不先跟 vSphere 环境连接并且通过认证，是无法和 vSphere 交互的。这个过程是由 Connect-VIServer cmdlet 所掌控的。这里我不再演示命令执行后的情况，这个 cmdlet 将提示您输入认证证书。

在连接建立之后，我输入 Get-VM 命令而不带任何参数。“Get”在这个短语中是动词的角色，它的意思是从某处获得某个对象。“VM”是 cmdlet 命令中的名词部分，当然它指代的就是所面对的虚拟机。结果理解起来相当的简单，当然，这也是 PowerShell 的标志。您可以看到一个带有标题的列表在屏幕上显示，如果您曾经在虚拟机环境中工作过，我想这些参数特性都是一目了然的，所以我不再进一步对该命令的细节做说明。

对于每一行文字而言，它不仅仅代表字面的意思，更多的是表示一个对象。在这个例子中，我们所讨论的对象就是虚拟机。但是 PowerShell 的这个规制还适用于任何其他对象，当我们讨论 FileInfo 时对象是文件系统，或者是虚拟机，也可能是主机服务器或任何其他的东西。

最后所提到的，在某些情况对象可以指代任意东西，这个课题确实非常大，所以我对它的讲述到此为止。同时，我希望您已经获得关于 PowerCLI 足够多的内容，使您已经产生兴趣并将继续学习使用它。我也希望你可以同意我的观点：关于自动化是将来唯一的发展方向。所以如果您和您的小组还没有搭上这个潮流班车，我想现在这是个很好的切入契机。我接下来的系列将继续讲述有关内容，所以请保持关注。

下载 PowerShell，访问[微软的 PowerShell 站点](#)。同样，要获得 PowerCLI，请访问[VMware 的 PowerCLI 站点](#)。

(作者: Hal Rottenberg 译者: 李哲贤 来源: TechTarget 中国)

原文标题: 如何使用 VMware vSphere PowerCLI?

原文链接: http://www.searchvirtual.com.cn/showcontent_27966.htm

五大必备 vSphere PowerCLI 脚本

VMware vSphere PowerCLI 是一个可以帮助虚拟化管理员创建脚本的命令行工具，它可以在 vSphere 环境中自动执行管理员任务而且无需更多人为参与。

PowerCLI 指代 Microsoft PowerShell，是一个在很多数据中心产品中普遍存在的面向对象的脚本语言和命令行编辑器，如 Microsoft Windows、Exchange 和 SQL servers 中都有。PowerCLI 可以通过如下方式自动执行管理员任务：

- cmdlets，系统动作或指令
- 脚本，一组 cmdlet 指令及其关联关系
- 可执行文件，独立的应用程序，以及
- .NET 类

借助 vSphere PowerCLI，通过由 165 个 VMware 专用 cmdlets 组合形成脚本，IT 管理员可以自动执行所有的虚拟环境相关管理任务。对于虚拟化管理员而言，脚本无疑是非常实用的，因为它可以无需人为参与执行进程（这样可以释放 IT 管理员去做其它工作），而且在资源紧张的环境中可以用于规划在非高峰期完成某些任务，在执行大量重复性的烦杂工作时还可以避免人为错误的发生。

在熟悉了 PowerCLI 的使用之后，如下五个脚本是您必备的：

一、[vCheck \(Daily Report\) V3](#) *Alan Renouf 制作*

想象一下您可以在喝早咖啡的时候顺便查看一下关于虚拟环境的详细报告。通过 Alan Renouf 制作的这个流行脚本，可以每天自动生成虚拟环境概要的报告并发送到您的收件箱。

除了可以列举出宿主机、虚拟机、集群数量和数据存储量以外，新版的 vCheck 还可以显示出虚拟机动作警报、废弃的 SCSI LUN 等更多内容。

二、[Who created that VM?](#) *Alan Renouf 制作*

您是否遭遇过虚拟机在数据中心蔓延的情况，或者您无法跟踪到是谁在创建虚拟机？现在您无需采用那些诸如撤消管理员权限等的极端方式来解决问题，使用“who created that VM?” 脚本可以解决。

通过为每个虚拟机添加定制字段，这个 PowerCLI 脚本可以在虚拟机的注释中提供关于每个虚拟机的创建者和创建日期等信息。另外，您还可以使用这个脚本来查询一个月内创建的虚拟机数量，并把这些信息输出形成清单。

三、vSwitch port overview *Hugo Peeters 制作*

在迁移过程中，VMware VMotion 并不会检查新的宿主机是否有足够的可用虚拟交换机端口。更为糟糕的是，如果端口不够用，VMotion 依然会完成迁移过程而且不会向用户报警。因此结果就会导致虚拟网卡失去连接。

因此，每个 vSwitch 上必须保留足够的端口来满足可能会从其它主机上发生故障切换迁移过来的虚拟机的额外需求。为了检查每个 vSwitch 使用的端口数，可以运行这个 PowerCLI 脚本，然后对虚拟交换机进行适当地调整。

四、Report into Microsoft Word *Alan Renouf 制作*

该脚本不仅可以生成整个虚拟环境的报告，而且还会把结果写成微软的 Word 格式，并生成彩色的曲线图和饼状图。更进一步，该 PowerCLI 脚本很容易进行定制化，用户可以根据需求进行信息裁剪。这个脚本用于生成一个直观地、易于理解的报告给老板汇报也非常方便。

五、Customizing Site Recovery Manager plans *Mike Laverick 制作*

对于虚拟化管理员而言，结合 PowerCLI 和 SRM (Site Recovery Manager)，可以在灾难恢复的过程中提供更高级别的自动化。

在这篇文章中，Mike Laverick 揭示了如何在 SRM 服务器上运行 PowerCLI，并且给出一个脚本可以用于减少虚拟机在恢复过程中所需的内存数量。

关于 VMware vSphere PowerCLI 的更多内容，参考如下外部参考资源：

- vSphere PowerCLI Blog (VMware 社区博客)
- Managing VMware with PowerShell (VMware 社区 FAQ)
- PowerCLI Quick Reference Guide (来自 virtu-al.net)

(作者: Keith Kessinger 译者: 李哲贤 来源: TechTarget 中国)

原文标题: 五大必备 vSphere PowerCLI 脚本

原文链接: http://www.searchvirtual.com.cn/showcontent_42980.htm

掌控 PowerCLI：使用 Get-VM 来管理虚拟机

如果您刚刚开始应用 PowerCLI 管理 VMware 环境，需要学习的内容很多。下面，我们学习掌握一个最常见和最重要的 cmdlets。

在上一篇文章中，TechTarget 中国特约作者 Hal Rottenberg 推荐您使用 PowerCLI，希望可以激发您进一步研究的兴趣。今天将继续讲述工具箱中最重要的一个 cmdlet：Get-VM。PowerShell 自带有非常全面的帮助系统，因此让我们借助它来对 Get-VM 进行整体了解。如果您希望和我一起开展，现在请从开始菜单中打开 PowerCLI，如截图所示：



在 PowerCLI 窗口中，我从帮助文件中提炼了一些关键字段。

（编者注：为便于读者定位，我们从帮助文件中提取了相关代码并整理到一个 txt 文件中，从这里您可以访问。它被分成了[四个部分](#)，因此您可以理解作者所讲的是哪一段代码。）

注：“PS>”代表了命令提示符，之后紧跟的文本是我所输入的。其它的都属于由之前输入的命令产生的输出结果。

Get-VM 和 “get-cmdlets” 相似，用于查找一个或多个对象。你可以对这些对象进行的操作最简单的莫过于把它们显示到控制终端上。换句话说就是：生成报告。在上一篇文章中我已经向您演示过，在已经通过 Connect-VIServer cmdlet 建立到 ESX 或 vCenter 的连接后，单独运行 Get-VM 产生的结果。您会看到一个包含当前所有虚拟机在内的，分为四行内容的一个表格：Name、PowerState、Num CPUs 和 Memory (MB)。那么，接下来会发生什么呢？其实在这背后还隐藏着更多的信息，请参考我的一台虚拟机的情况（参照[摘录 2](#)）。

我希望您注意几点事情：

1. Get-VM 带有 Name 变量，可以指定一台或多台虚拟机的名称（或通配符）来确定执行范围。这是一个位置参数，而且位置为零，也就是说实际上您可以保留 “-name” 文本，而 PowerShell 可以指出您所需内容。
2. 我使用了 Format-List cmdlets，因此属性的显示方式是垂直分行的。而且在这个例子中，我通过加入通配符的方式让 Get-VM 命令显示所有的参数。这样的话，我们就获得了所有相关参数显示到屏幕上，而不仅仅是默认的四项。（注：如果您发现出现的属性项要比我的少，请检查是否使用了最新版的 PowerCLI。截止本文，已经发布的最新版是 4.0 U1，但是它跟老的 VI 环境还无法很好地兼容。）
3. 在冒号左侧的所有内容都是属性的名称，而位于冒号右侧的就是各个属性所对应的具体值。
4. 您看到属性的值位于大括号中，代表了查看了一组不同的值。换句话说，您的虚拟机可能有多个软驱、CD 驱动器、硬盘、网卡适配器等等。

理解 PowerShell 的属性

如果您对 vSphere 客户端的方式还算了解，那么 PowerState、Description 或 Name 这样的属性含义非常明显，所以我不会过多解释。而且还有一些属性的名称是以 “ID” 结尾的，如 “HostId” 。这表示字段中包含了到其它类型对象的链接。还有少量的字段用来表示一些更加高级的 VMware 功能，如 HARestartPriority、HAIsoilationResponse 和 DrsAutomationLevel 等。我现在不会对这部分内容作太深入的探讨，这只是表示它们指向位于 vSphere 客户端中相同的名称字段。

然而，还有一个特殊的字段我想可以对它作进一步的解释，那就是 Guest 属性。乍一看，Guest 属性的值是 VMware.VimAutomation.Client20.VMGuestImpl。实际上，这是拥有该属性的对象的完整 .NET 类名。

什么意思？

好吧，让我换一种说法。PowerShell 中的所有内容都可以称为对象。每个对象又有成员，而成员中包含了属性和方法（以及其它的一些东西）。如果您从未接触过面向对象的程序设计，让我们从一组基础的定义开始：

属性——某个对象的属性。属性会有一些跟它相关的值。例如，“汽车”是一个对象，“车体颜色”是它对应的属性，而“白色”就是这个属性的值。

方法——可以对某个对象进行的动作。通过执行方法可以达到期望的动作。例如，对象“汽车”的方法之一是“加速”。执行这个方法可以使汽车向前移动。

在 PowerShell (和 PowerCLI) 中，属性的值就是对象本身。很常见的一种情况就是属性就是本体，也就是说它仅仅是一个简单的字符串 (如“vm1”)，或者是一个整数 (如“32”)。现在明白我所说的了吗？

在例子中的 Guest 属性，并不是一个简单的字符串，例如我们取对象 VMGuestImpl，(取自于上一个例子中对象名称的最后一部分)。让我们使用 Get-Member cmdlet 来查看一下它的内容。Get-Member 来自于 PowerShell 而不是 PowerCLI，非常适合于检查对象的属性，如[摘录 3](#) 中所看到的。

在上面的第一行中，我指定了变量名为“vm”的虚拟机对象，这是从 Get-VM 中返回的结果看到的。这么做可以使接下来的代码更方便阅读，同时由于把所有的内容都提取到内存中执行，对于同一个对象执行连续的操作也可以加快执行的速度。

在代码的第二行中，我在希望检查的属性的名称后加了一个圆点。然后，我把该属性交给 Get-Member cmdlet 去处理。同时，我告诉 Get-Member cmdlet 只想查看该属性。这样做在本例中极大简化了输出。

Get-Member 的输出结果分为三行：Name、MemberType 和 Definition。头两项的含义非常直白，第三项 Definition 稍微复杂一点。我向您出示这张表的目的是为了解释 PowerShell 中属性和对象之间的关系。Definition 的第一部分告诉用户属性所属对象的类型。如您看到的，部分显示为“System.String.”，表明该对象仅为一组字母。其它的一些名称长一些，用“VMware”开头，您可以猜到这些对象要复杂一些 (跟 VMGuestImpl 对象类似)，其内部还包含有更多的信息。

讲到这里：如何显示属性和它们各自的值 ([摘录 4](#))，本文就结束了。请继续关注下篇文章，我讲教您如何使用 Set-VM cmdlet 来完成对虚拟机的设置。

(作者：Hal Rottenberg 译者：李哲贤 来源：TechTarget 中国)

原文标题：掌控 PowerCLI：使用 Get-VM 来管理虚拟机

原文链接：http://www.searchvirtual.com.cn/showcontent_43002.htm

使用 PowerShell 与 PowerCLI 自动化主机服务器任务

这是关于 PowerCLI 的另一篇技巧，你还记得 PowerCLI 吗？它是为 Windows PowerShell 脚本语言而创建的嵌入单元。这个嵌入单元能让你快速轻松地将 vSphere 环境的所有方面进行自动化。今天的文章介绍如何使用 PowerShell 和 PowerCLI 对主机服务器进行自动化操作。

VMware 说法中的主机服务器是个提供虚拟化服务的系统。也就是说的 hypervisor，当然这种情况下就是 VMware 的产品 ESX Server。PowerCLI 能管理 ESX Server 的各种版本，从 3.0 开始，包括“瘦”版本 ESXi。以前你可以使用 PowerCLI 管理 VMware Server（非裸金属 hypervisor），但那个功能处于某种原因消失了。

那么我们来看看使用 PowerCLI 能对主机服务器做什么。写这篇文章的时候，装载的是 PowerCLI 4.0 Update 1。这个版本是对先前版本的重大更新，因为它所包含的 cmdlets 清单比之前的多。（PowerShell cmdlets 类似于如 cmd. exe 或 bash 等 command-shells 中的内部命令。）

有个关于 PowerShell 的笑话，它只列出了 PowerCLI cmdlets，在名称中含有“host”。

(Note that "PS >" indicates the shell prompt, and the text that immediately follows is what I typed. Everything else is output from a previously typed command.)

```
PS > get-command *host* -pssnapin vmware* | format-wide
Add-VmHostNtpServer          Add-VmHostNtpServerApply-VMHostProfile          Export-
VMHostProfileGet-VMHost       VMHostProfileGet-VMHostAdvancedConfiguration          Get-
VMHostAccountGet-VMHostAdvancedConfiguration          Get-
VMHostAvailableTimeZoneGet-VMHostDiagnosticPartition          Get-
VMHostFirewallDefaultPolicyGet-VMHostFirewallException          Get-
VMHostFirmwareGet-VMHostHba          Get-
VMHostModuleGet-VMHostNetwork          Get-
VMHostNetworkAdapterGet-VMHostNtpServer          Get-
VMHostProfileGet-VMHostService          Get-VMHostSnmpGet-
VMHostStartPolicy          Get-VMHostStorageGet-
VMHostSysLogServer          Import-VMHostProfileInstall-
VMHostPatch          Move-VMHostNew-VMHostAccount
New-VMHostNetworkAdapterNew-VMHostProfile          Remove-
VMHostRemove-VMHostAccount          Remove-
VMHostNetworkAdapterRemove-VMHostNtpServer          Remove-
```

VMHostProfileRestart-VMHost	Restart-
VMHostServiceSet-VMHost	Set-
VMHostAccountSet-VMHostAdvancedConfiguration	Set-
VMHostDiagnosticPartitionSet-VMHostFirewallDefaultPolicy	Set-
VMHostFirewallExceptionSet-VMHostFirmware	Set-
VMHostHbaSet-VMHostModule	Set-VMHostNetworkSet-
VMHostNetworkAdapter	Set-VMHostProfileSet-VMHostService
Set-VMHostSnmpSet-VMHostStartPolicy	Set-
VMHostStorageSet-VMHostSysLogServer	Start-VMHostStart-
VMHostService	Stop-VMHostStop-VMHostService
Suspend-VMHostTest-VMHostProfileCompliance	Test-VMHostSnmp

那是 60 个 cmdlets。不过我不能对每个 cmdlet 的细节加以描述，但我们能重点介绍一些 cmdlets。同样，不要忘记 PowerShell 里的每个 cmdlet 都有内置帮助。需要的话输入“get-help”加上 cmdlet 名称，或者 cmdlet 名称加上“-?”。

首先我想介绍的是 Get-VMHost。与所有 get-cmdlets 一样，着重在于从容器检索对象。如果使用 Connect-VIServer cmdlet 连接到 vCenter 服务器，然后自己运行 Get-VMHost，那么结果可能如下：

```
PS > Connect-VIServer vlab.halr9000.comName Port
User---- -----
vlab.halr9000.com 443 halPS > Get-
VMHostName State PowerState Id CpuUsage CpuTotal Memory
Memory ----- Mhz Mhz UsageMB
TotalMB----- -----
4094atlesx01.hal... Connected PoweredOn ...-232 168 8000 2295
4094atlesx02.hal... Connected PoweredOn ...-225 366 8000 2645
4094atlesx03.hal... Connected PoweredOn ...t-10 350 8000 3264
4091
```

如果你没有 vCenter 备份，不用担心，可以用 PowerCLI 所有的信息。事实上，PowerCLI 在没有 vCenter 的情况下能一次管理多台 ESX 服务器。

Connect-VIServer 命令似乎不同，但最终结果是一样的。（没有 vMotion 或模板，你在使用某些功能时仍然需要 vCenter。）

现在来分析上面命令的输出结果，然后我们将用一些其他方式使用这个 cmdlet。这里执行了两个命令。第一个是建立到 vCenter 服务器的连接。第二个是不通过参数自己运行 Get-VMHost。这样，你可以看见所有连接到 vCenter 的主机服务器清单（ESX or ESXi）。这个输出表分为几列，包括主机名称、连接状态、电源状态。从 informational ID 查看，有四个参数能让你快速诊断系统健康状况。

如同 PowerShell 里的其他对象，我们不能立即看到所有东西。所以，我们来看看其他一些有用的信息：

```
PS > get-vmhost | format-table name, manufacturer, model, numcpu, version, build -autosize
```

Name	Manufacturer	Model	Num Cpu	Version
Build				
atlesx01.halr9000.com	Dell Inc.	PowerEdge		
SC1435	4	4.0.0	208167	
atlesx02.halr9000.com	Dell Inc.	PowerEdge		
SC1435	4	4.0.0	219382	
atlesx03.halr9000.com	Dell Inc.	PowerEdge		
SC1435	4	4.0.0	219382	

因此问题在于：你如何知道对象里的哪个领域可用？首先我会说在 PowerShell 中，每样事物都是个对象。对象包含成员，成员能包含属性和方式。属性相当于上面表格中输出信息栏所见的域名。方法我会在另一篇文章中介绍，但总之，它们能定义对对象所做的事情，就是行动。

回到属性。有种方法能列出所有属性。注意，使用 Format-Wide cmdlet 出来的信息横跨两个输出栏。

```
PS > get-vmhost | get-member -MemberType property | format-wide
```

Build	ConnectionString
CpuTotalMhz	CpuUsageMhz
CustomFields	HyperthreadingActive
Id	Manufacturer
MemoryTotalMB	MemoryUsageMB
Model	Name
NumCpu	ParentId
PowerState	ProcessorType
State	TimeZone
Version	VMSwapfileDatastoreId
VMSwapfilePolicy	

你可使用 Get-VMHost cmdlet 上的 Name 属性，只对那些匹配的服务器进行输出，你可以使用通用符。由于 PowerShell 管道工作的方式，你能将 VMHost 对象作为输入传送到 Get-VM cmdlet。最终结果将会显示宿主在某台主机服务器上的虚拟机列表。例如：

```
PS > get-vmhost -name atlesx01* | get-vm
```

Name	PowerState	Num CPUs	Memory (MB)
ELGFILO1	PoweredOn	1	384
VMGVIC01	PoweredOn	1	512
VMGADC01	PoweredOn	1	384
VMGADC02	PoweredOn	1	384
ELGADC01	PoweredOn	1	512

在文章最后，我将介绍如何使用主机服务器防火墙。使用 Get-VMHostFirewallException cmdlet 列出防火墙允许的活动。如果输入没有参数的 cmdlet，会反馈错误信息，因为你必须制定查询的具体主机服务器。下面是步骤：

```
PS > $vmhost = get-vmhost atlesx01*
PS > Get-VMHostFirewallException -Name 'FTP Server' -VMHost $vmhost
```

Name	Enabled	IncomingPorts	OutgoingPorts	Protocols	ServiceRunning
FTP Server	False	21		TCP	

在这个例子中，我也将通过指定防火墙规则名称过滤输出。

一旦你明白如何进行防火墙配置，使用 Set-VMHostFirewallException cmdlet 进行更改就很是容易的事。这个 cmdlet 有两个参数：

- **Exception:** 这个相当于需要更改的防火墙规则。参数能在管道上具体说明。
- **Enabled:** 这是个 Boolean 旗帜，如果真实，将启用规则（因为已经防火墙允许）。设置不正确将禁用规则并关闭任何响应的端口。

下图是启用 FTP 服务器防火墙规则的例子：

```
PS > $ftp = Get-VMHostFirewallException -Name 'FTP Server' -VMHost $vmhost
PS > $ftp | Set-VMHostFirewallException -Enabled:$true
```

Name	Enabled	IncomingPorts	OutgoingPorts	Protocols	ServiceRunning
FTP Server	True	21		TCP	

(作者: Hal Rottenberg 译者: 唐琼瑶 来源: TechTarget 中国)

原文标题: 使用 PowerShell 与 PowerCLI 自动化主机服务器任务

原文链接: http://www.searchvirtual.com.cn/showcontent_43019.htm

跳出框外 巧解 PowerShell 与 PowerCLI 难题

在这篇文章中，我想着重讲解一下在我浏览 PowerCLI 论坛时，看到的一些很有用的技巧。下面我将向你介绍这些技巧，会谈到 PowerCLI 是什么，以及我发现的一些特点。我还将提供涉及到的技术的一些背景资料。

PowerShell 以外的思路

首先介绍的一个帖子是，“[在 ESX 中，需要借助 PowerShell 脚本来找到某个虚拟机 \(VM\) 的进程 ID \(PID\)](#)。”此贴的前提是，作者需要找到一些信息，但他唯一知道获取到这些信息的途径就是登陆到 ESX 操作系统控制台 (COS)。这就是为什么当前需求与他需要进程 ID 不符的原因了，但这足以证明，有时你为了完成某些诊断任务而必须登录到控制台。

这个帖子的特点在于，它似乎暴露了 vSphere 应用程序编程接口 (API) 上的某些不足，并需要以 PowerCLI 为辅助。就如所有的 APIs 一样，功能都是创建了一个抽象层，可以对某一产品或某一产品套件进行尽可能灵活的一致性编程，而且可以引入新的功能来降低软件开发任务的难度。问题是，与 ESX 虚拟化层进行沟通工作的操作系统控制台 (COS) 实际上也是一个操作系统，而操作系统往往都相当复杂，并且在某些情况下，用户接口也非常分散。

回到 PowerCLI 和 vSphere API 上来。架构在 VMware 最底层架构之上的 vSphere API 层是非常有层次架构，并提供了最外层的功能。操作系统却有着不一样的架构！很不幸的是，vSphere API 并不能构建一张完美的流程图以使你可以登录到操作系统控制台 (COS)。然而，PowerCLI 也是基于某些 API 之上，这意味着你不能在操作系统 (OS) 控制台之内使用 PowerCLI 来执行相关任务——至少不是直接。这正是发布到论坛上的解决方案的用武之地。

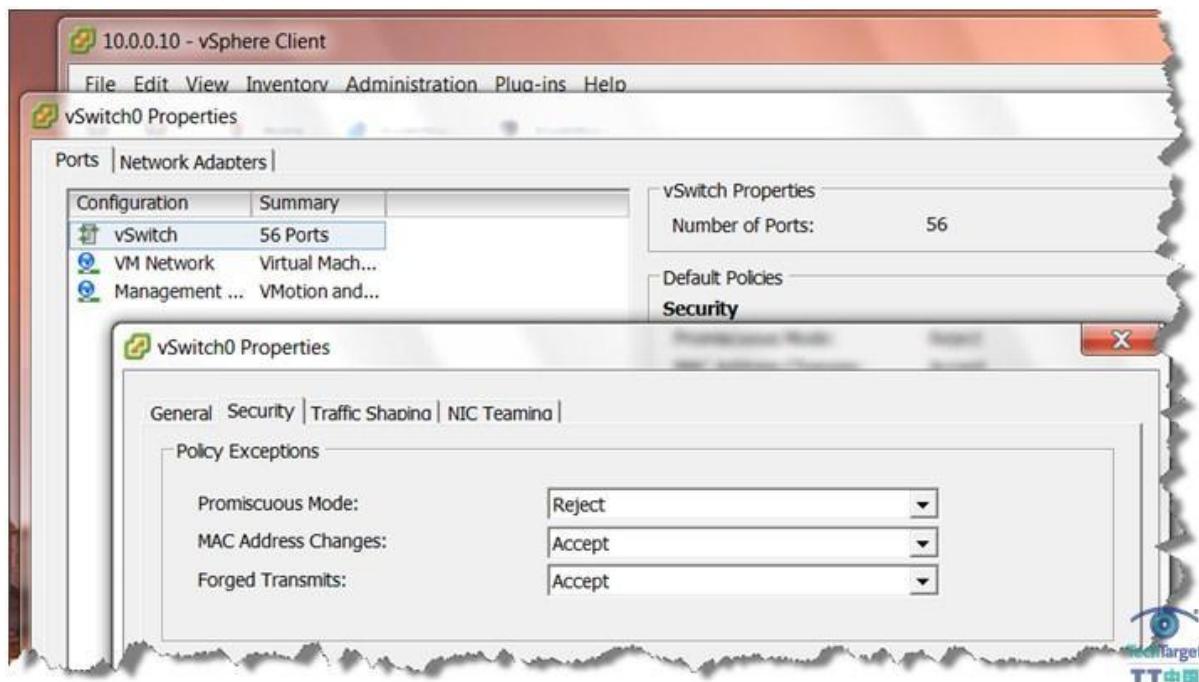
对我们和原始帖子的作者来说幸运的事情是，PowerCLI 是建立在 PowerShell 内，并且 PowerShell 可以调用一个外部进程作为脚本的一部分。该解决方案是在 LucD 的回贴中提到的。在这种情况下，外部进程是 plink.exe，它是一个非常流行的安全命令行 shell (SSH) 客户端—PuTTY 中的一部分。该脚本实现使用 plink 登录到 ESX 主机管理控制台，然后执行远程控制命令 “vm-support” 。此实用工具是一种收集信息的方法，正如原帖中提到的。

你可能会说这种行为有点像一个黑客，因为你不是通过“被认准的接口”来访问数据，但有些时候，你不得不去找到一个方法。这个特别帖子的寓意是：不要害怕，要跳出思想的框框。虽然 vSphere API 是一个相当大的框框，但它的目的不是为了解决你可能遇到的每一个 vSphere 系统的问题。事实上，对于它准确的理解是，它确实不适合用于诊断。

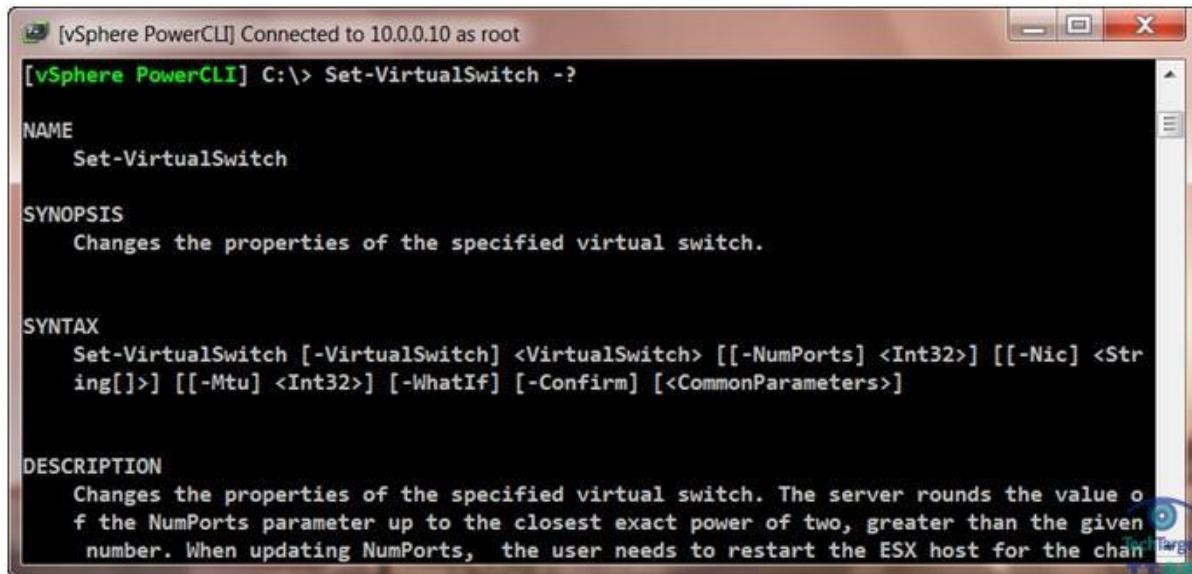
当 PowerCLI 的内置命令不起作用

这第二个帖子标题是，“[使用脚本来配置 vSwitch 上的安全参数](#)。”虽然在这个帖子中的脚本非常高深，但是我认为这将是一个很好的例子，因为我认为它描述了一种常见的场景，当你没有使用 PowerCLI 内置命令而使用到 vSphere API 的功能。其基本前提是内置命令（在以前的文章我也谈到过）与 API 的功能没有 100% 的对等。然而，通过使用 Get-View 内置命令，再加上其他一些技巧，你就可以完成很多的需要做的事情。

此贴的作者想要说的是关于设置 vSwitch 安全方面的设置。我打开安全选项卡中的一个选项页面，当你打开一台虚拟交换机的性能属性时就可以看到，如下图：



不幸的是，你不能通过内置命令设置这些属性，下图为 Set-VirtualSwitch 命令的帮助文档：



```
[vSphere PowerCLI] Connected to 10.0.0.10 as root
[vSphere PowerCLI] C:\> Set-VirtualSwitch -?

NAME
  Set-VirtualSwitch

SYNOPSIS
  Changes the properties of the specified virtual switch.

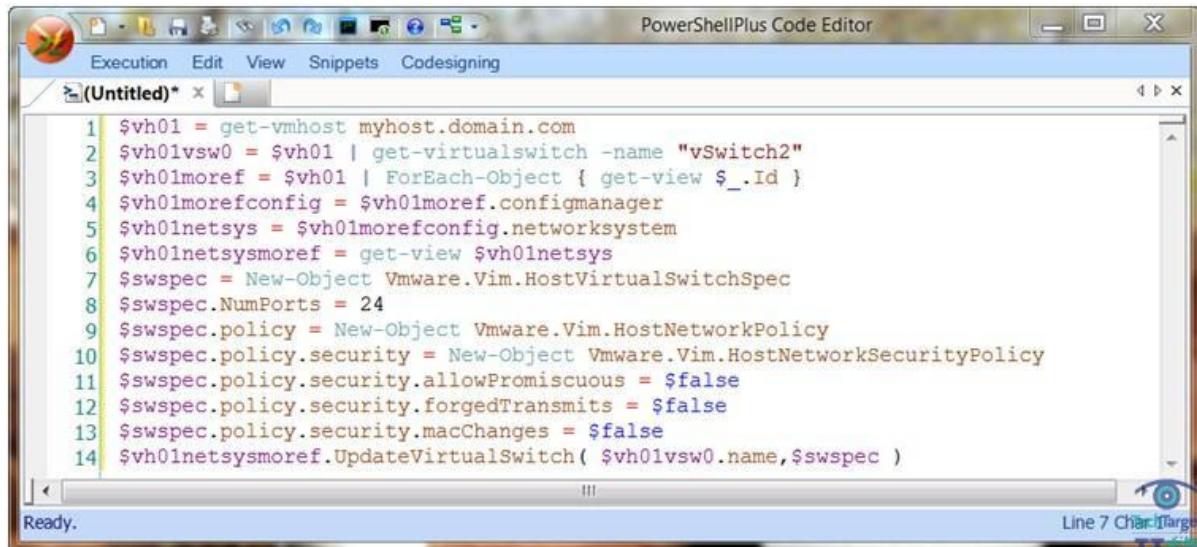
SYNTAX
  Set-VirtualSwitch [-VirtualSwitch] <VirtualSwitch> [[-NumPorts] <Int32>] [[-Nic] <String[]>] [[-Mtu] <Int32>] [-WhatIf] [-Confirm] [<CommonParameters>]

DESCRIPTION
  Changes the properties of the specified virtual switch. The server rounds the value of the NumPorts parameter up to the closest exact power of two, greater than the given number. When updating NumPorts, the user needs to restart the ESX host for the chan
```

该内置命令缺乏安全设置参数。也有可能，VMware 正在并将继续缩短内置命令和 API 之间的差距，但今天它确实是帮不了我们了。

为了弥补暂时的不足，你必须使用 Get-View 的内置命令。Get-View 是非常有用的——它就像释放出一种技能以让你闯过难关。我不打算详细介绍有关的内置命令，但你需要知道的是，目前 PowerCLI 内置命令给虚拟化管理员开发了一个可以与 PowerShell 协调工作的可定制接口。当它起作用时，这对于管理员来说是好消息，但是当你的操作受到了内置命令的制约，那么你必需采用底层 API 对象。换句话说，你需要理解一个完全不同的接口——这与程序员为 vSphere 开发软件时看到的一样。

回到帖子，帖子中有段代码，如下所示（为便于阅读，已稍微编辑）：



```
PowerShellPlus Code Editor
Execution Edit View Snippets Codesigning
(Untitled)*
1 $vh01 = get-vmhost myhost.domain.com
2 $vh01vsw0 = $vh01 | get-virtualswitch -name "vSwitch2"
3 $vh01moref = $vh01 | ForEach-Object { get-view $_.Id }
4 $vh01morefconfig = $vh01moref.configmanager
5 $vh01netsys = $vh01morefconfig.networksystem
6 $vh01netsysmoref = get-view $vh01netsys
7 $swspec = New-Object Vmware.Vim.HostVirtualSwitchSpec
8 $swspec.NumPorts = 24
9 $swspec.policy = New-Object Vmware.Vim.HostNetworkPolicy
10 $swspec.policy.security = New-Object Vmware.Vim.HostNetworkSecurityPolicy
11 $swspec.policy.security.allowPromiscuous = $false
12 $swspec.policy.security.forgedTransmits = $false
13 $swspec.policy.security.macChanges = $false
14 $vh01netsysmoref.UpdateVirtualSwitch( $vh01vsw0.name,$swspec )

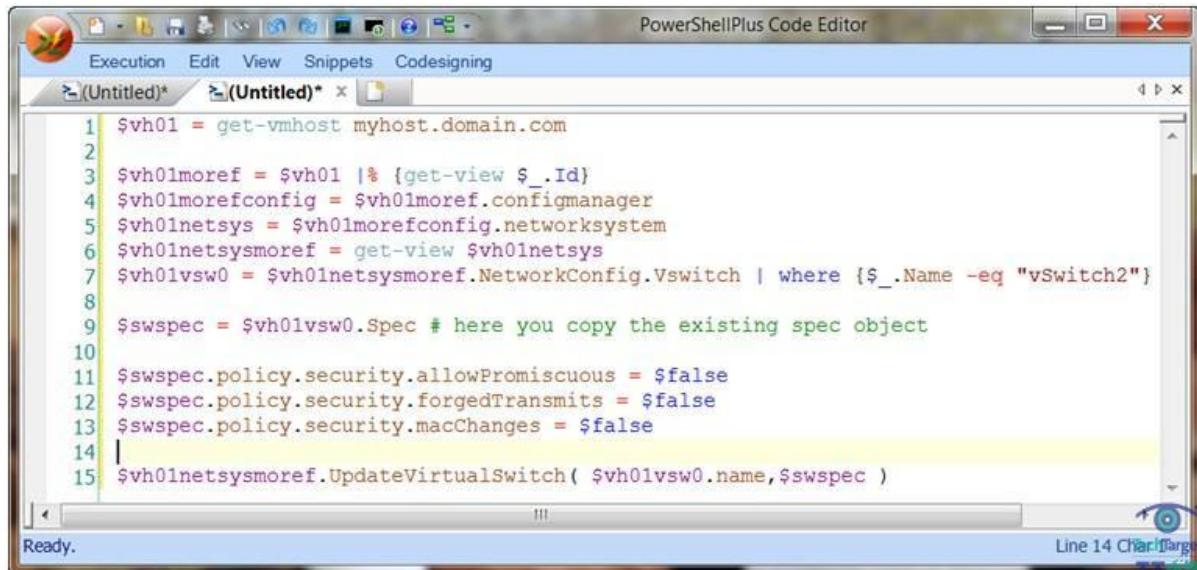
Ready.
Line 7 Char 1 Target
```

第 1 行检索了主机列表并将它们分配给一个变量。第 2 行遍历该变量中包含的每个主机服务器，并找到名为“vSwitch2”的交换机。第 3 行在这里你可以看到 Get-View 内置命令的使用以获取原始对象。

4-6 行，你会得到一个以一定方式命名的 vSphere API，这些将为更新虚拟交换机做好准备——“艰难的方式”。

在第 7 行会报错。为了达到编辑虚拟交换机的目的，作者给出了非常有效的假设，所以你将使用到一个新的对象（这些通常是 vSphere API 使用来描述一个正在创建或修改的对象）。

但问题是，就像 LucD 在回复中做出的非常有用的解释，该 API 有时是相当挑剔。希望看到这个规范的对象是，它其中的每一个参数字段都被明确定义并赋予了默认值。在这一点上，你可以找出这些默认值应该是什么，也可以简单地复制一个已有的，之后再修改成你的内容。更新后的代码如下所示：



The screenshot shows a Windows application window titled "PowerShellPlus Code Editor". The menu bar includes "Execution", "Edit", "View", "Snippets", and "Codesigning". The main window displays a PowerShell script in a code editor. The script is as follows:

```
1 $vh01 = get-vmhost myhost.domain.com
2
3 $vh01moref = $vh01 | % {get-view $_.Id}
4 $vh01morefconfig = $vh01moref.configmanager
5 $vh01netsys = $vh01morefconfig.networksystem
6 $vh01netsysmoref = get-view $vh01netsys
7 $vh01vsw0 = $vh01netsysmoref.NetworkConfig.Vswitch | where {$_.Name -eq "vSwitch2"}
8
9 $swspec = $vh01vsw0.Spec # here you copy the existing spec object
10
11 $swspec.policy.security.allowPromiscuous = $false
12 $swspec.policy.security.forgedTransmits = $false
13 $swspec.policy.security.macChanges = $false
14
15 $vh01netsysmoref.UpdateVirtualSwitch( $vh01vsw0.name,$swspec )
```

The status bar at the bottom left says "Ready." and the bottom right says "Line 14 Char 1 Target".

我想强调这个论坛帖子的目的是想强调什么方式才是将 PowerShell 和 vSphere API 对象协调工作的最佳方式。当你修改一个现有对象的时候，复制该对象的属性，然后再修改对象的属性。这将为你节省不必再通过 API 文档找到对象的正确默认值，因为那可能是一件头痛的事情，尤其当你去尝试使用对象的新属性的时候。但是，要记住，当你花时间来从头创建新对象的时候，那么这种便捷的方式对你不会有任何帮助。

(作者: Hal Rottenberg 译者: 于富春 来源: TechTarget 中国)

原文标题: 跳出框外 巧解 PowerShell 与 PowerCLI 难题

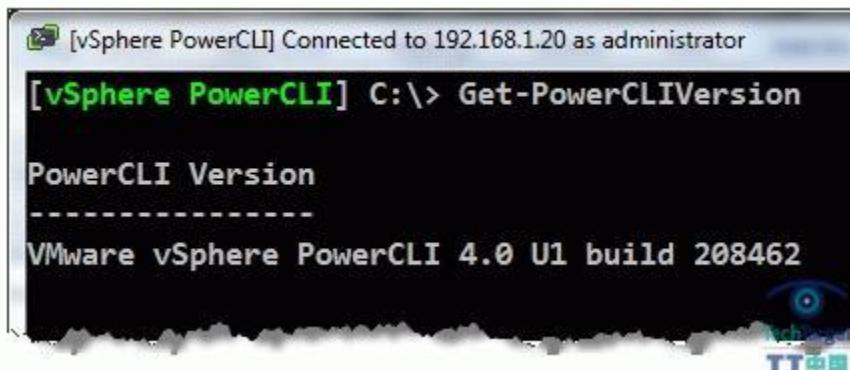
原文链接: http://www.searchvirtual.com.cn/showcontent_36581.htm

如何使用 vSphere PowerCLI 创建 host profiles?

VMware 的 host profiles 可以协助完成一些自动化的、具备重复性的以及紧急的配置管理任务。而这些方面的改善可以极大地提高数据中心的成熟度。伴随着虚拟架构规模的增长，如何有效利用 host profiles 等提供的一些功能属性成为越来越关键的课题。

本文，TechTarget 中国的特约作者 Hal Rottenberg 将阐述如何通过使用 vSphere PowerCLI 结合 host profiles 来完成对新主机的分发，以及完成对现有主机的一致性相关检查工作。

注：您必须首先拥有 PowerCLI 4.0 或者是更高版本、vSphere 4.0 以及可以正常运行 VMware host profiles 所需的软件许可。如果想验证您现有产品的版本，可以在 PowerCLI 提示符后输入 Get-PowerCLIVersion。



```
[vSphere PowerCLI] Connected to 192.168.1.20 as administrator
[vSphere PowerCLI] C:\> Get-PowerCLIVersion

PowerCLI Version
-----
VMware vSphere PowerCLI 4.0 U1 build 208462
```

Host profile cmdlets

首先，也是最重要的一点，就是了解哪些 cmdlets 可以和 host profiles 一起工作。找到这个答案最快捷的办法就是在 vSphere PowerCLI 命令提示行之后输入 Get-Help。如果我们同时再加入“*hostprofile*”的辅助字符串，就可以得到如下图所示的结果：



```
[vSphere PowerCLI] C:\> get-help *hostprofile*

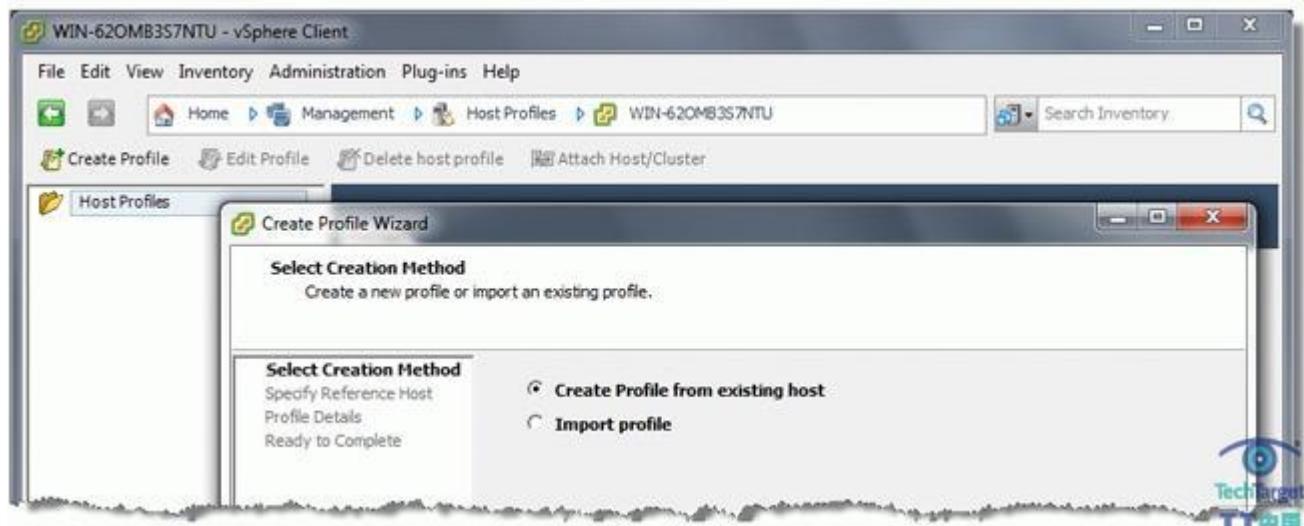
```

Name	Category	Synopsis
Apply-VMHostProfile	Cmdlet	Applies a host profile to the specified ho...
Export-VMHostProfile	Cmdlet	Exports the specified host profile to a file.
Get-VMHostProfile	Cmdlet	Retrieves the available host profiles.
Import-VMHostProfile	Cmdlet	Imports a host profile from a file. The fi...
New-VMHostProfile	Cmdlet	Creates a new host profile based on a ref...
Remove-VMHostProfile	Cmdlet	Removes the specified host profiles.
Set-VMHostProfile	Cmdlet	Modifies the specified host profile.
Test-VMHostProfileCompliance	Cmdlet	Tests hosts for profile compliance.

上图中仅显示出 cmdlets 命令的纲要，不过您可以通过在 Get-Help 命令后加入某个 cmdlets 的名称来了解更多信息，从而获悉它们可以完成哪些具体功能。

如果您对于 host profiles 如何在 vSphere Client 端工作的原理了如指掌，那您一定已经可以理解使用每个相关 cmdlets 的方法和目的。对于不太了解的人我简单解释一下：host profiles 背后代表的含义是如果我们选择把某个宿主机作为“reference host”。那么当使用 host profiles 的时候，所有基于这个“reference host”而创建的主机都会拥有和它完全相同的配置。

第一步是在 vSphere 客户端中，基于某个“reference host”创建一个 host profiles。



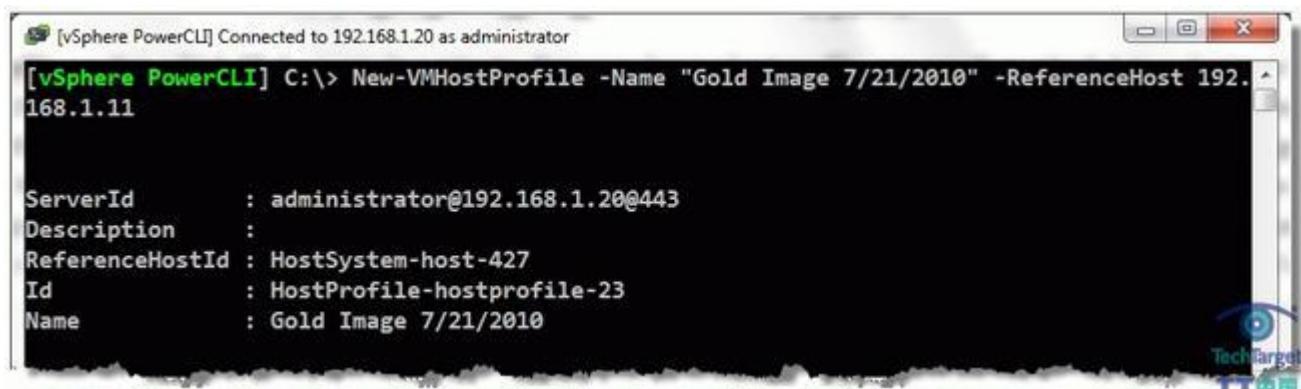
还有一种方式，通过使用 New-VMHostProfile cmdlets（用法我稍后会讲到）。一旦您拥有了 host profiles，就可以把它链接到一个集群或者是某台独立的主机上。这样，就可以通过和 host profiles 中主机配置的比对，来检查某台或者多台，甚至是整个集群系统配置的一致性。

在 PowerCLI 中，可以通过 Apply-VMHostProfile cmdlet 来执行相应改变，以及通过 Test-VMHostProfileCompliance cmdlet 来检查一致性。

最后一点，也是 host profiles 中任务量最大的一点就是如何把一个新 profile 的配置应用到一台全新的主机上。通过这一步您可以把所做的配置更改应用到目标主机系统中。很明显的一点，改变某个宿主机服务器的主要配置信息，会直接干扰到它提供虚拟化服务的能力。因此，主机需要被设置为维护模式，以方便应用所做的配置更改。当您已经做好触发必要的配置更改所需的准备工作后，就可以再一次地执行 Apply-VMHostProfile cmdlet 应用这些更改。

通过 PowerCLI 创建一个全新的 host profiles

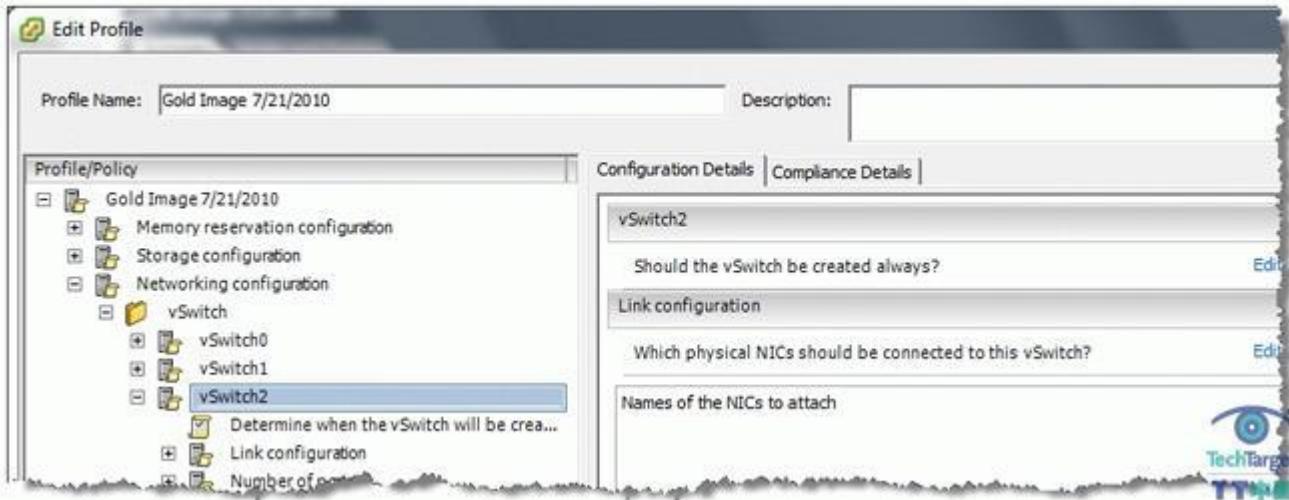
借助 PowerCLI 可以很轻易地创建一个新的 host profiles。您仅需要提供一个名称以及目标服务器就可以了，您可以看到下图所示内容：



```
[vSphere PowerCLI] Connected to 192.168.1.20 as administrator
[vSphere PowerCLI] C:\> New-VMHostProfile -Name "Gold Image 7/21/2010" -ReferenceHost 192.168.1.11

ServerId      : administrator@192.168.1.20@443
Description    :
ReferenceHostId: HostSystem-host-427
Id            : HostProfile-hostprofile-23
Name          : Gold Image 7/21/2010
```

很简单，对不对？实际操作仅会花费几秒钟的时间。而真正让人惊奇的事情稍后还会有很多，当我们开始把这些更改的配置信息应用于整个集群系统时。下面是在 vSphere Client 中看到的截图的内容，从中可以看出我的新 profile 服务器的部分配置信息。



接下来让我们开始进入下一步：挂载主机。

(作者: Hal Rottenberg 译者: 李哲贤 来源: TechTarget 中国)

原文标题: 如何使用 vSphere PowerCLI 创建 host profiles?

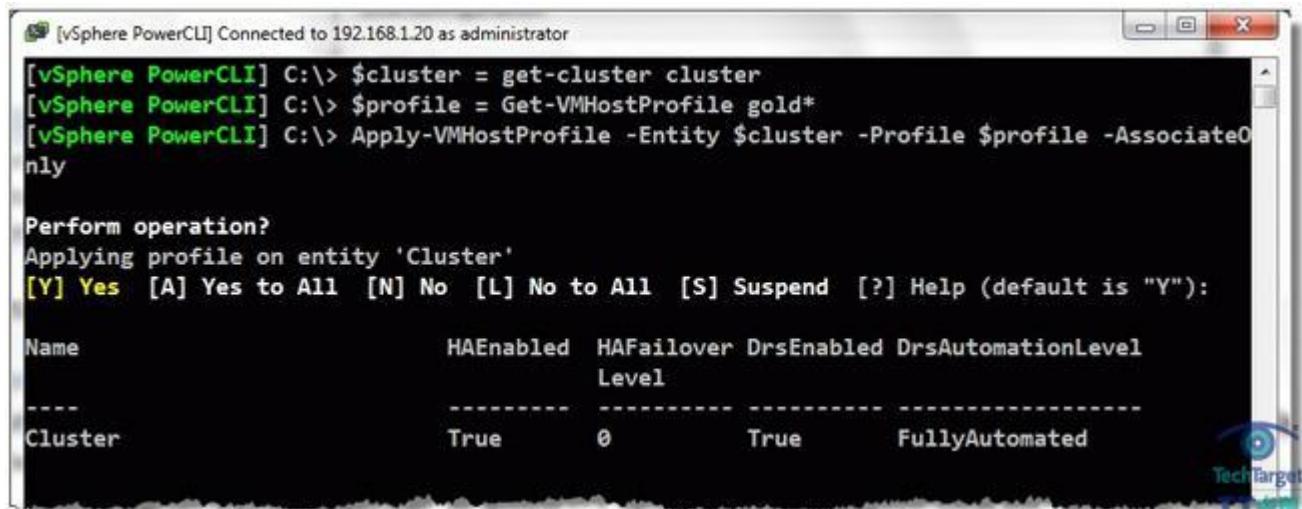
原文链接: http://www.searchvirtual.com.cn/showcontent_38829.htm

在 vSphere PowerCLI PowerShell 界面使用 host profiles

在本文的上半部分中，我们介绍了 [VMware host profiles 的概念以及使用 PowerCLI 创建一个全新的 host profiles 的方法](#)。下面我们将挂载主机并检查 host profiles 的一致性。

把某个主机或集群系统挂载到 host profiles

如我之前所讲，把主机挂载到 host profiles 并且应用这些更改的方法非常简单。只需在 PowerCLI 中运行一个简单的 cmdlets: `Apply-VMHostProfile`。在我们开始举例之前，先来简单解释一些这个 cmdlets 中各个参数所代表的含义。



```
[vSphere PowerCLI] Connected to 192.168.1.20 as administrator
[vSphere PowerCLI] C:\> $cluster = get-cluster cluster
[vSphere PowerCLI] C:\> $profile = Get-VMHostProfile gold*
[vSphere PowerCLI] C:\> Apply-VMHostProfile -Entity $cluster -Profile $profile -AssociateOnly

Perform operation?
Applying profile on entity 'Cluster'
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):

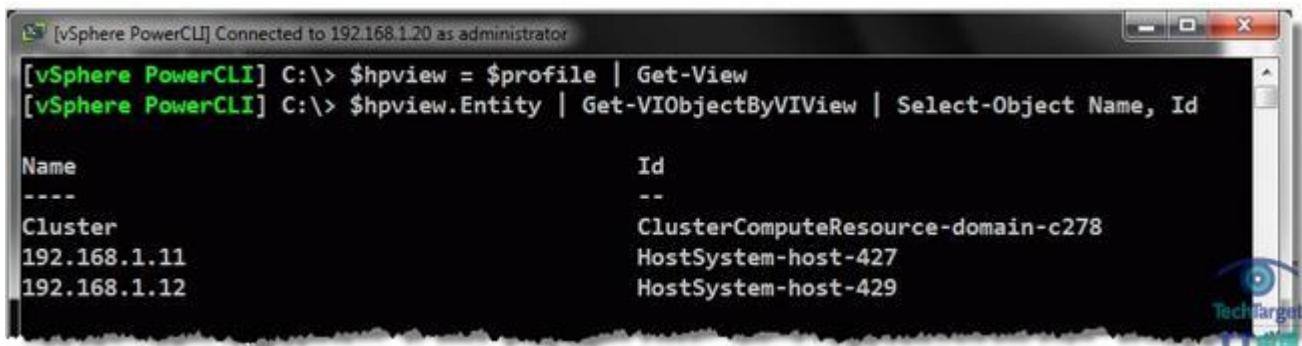
Name          HAEEnabled  HAFailover DrsEnabled DrsAutomationLevel
Level
-----
Cluster       True        0          True      FullyAutomated
```

名称	描述
Entity	将要涉及的主机或集群目录。
Profile	将要被采用的 host profiles 名称或者是目标地。
Variable	接受哈希算法，可以附加变量条件。例如把 vSwitch 的 IP 地址作为条件。
AssociateOnly	通过该命令把 profile 跟某个实体挂载，但是不应用更改。
ApplyOnly	通过该命令把 profile 更改应用到某个实体而不执行挂载。
RunAsync	通过该命令，允许在 vCenter 在后台启用某些特殊操作的时候，脚本程序可以迅速释放对资源的控制权。
Server	选择性地建立到一台或多台 vCenter 服务器的连接。如果您没有指定服务器，该操作将会应用到所有活动的 vCenter 服务器。

注：在使用 Apply-VMHostProfile 时还有很多需要特别注意的地方。请一定要注意参考 PowerCLI 中的 help 相关内容来获取更多信息。

让我们继续实现 profile 跟集群的连接：

如果连接成功了，您将会重新获得对目标实体的控制权。而在现有的 Get-VMHostProfile cmdlet 版本下，您无法轻易地看到哪些实体已经被挂载，但是接下来我将会向您展示如何借助 Get-View 和 Get-ManagedObjectByView cmdlets 来获得这些详细信息。

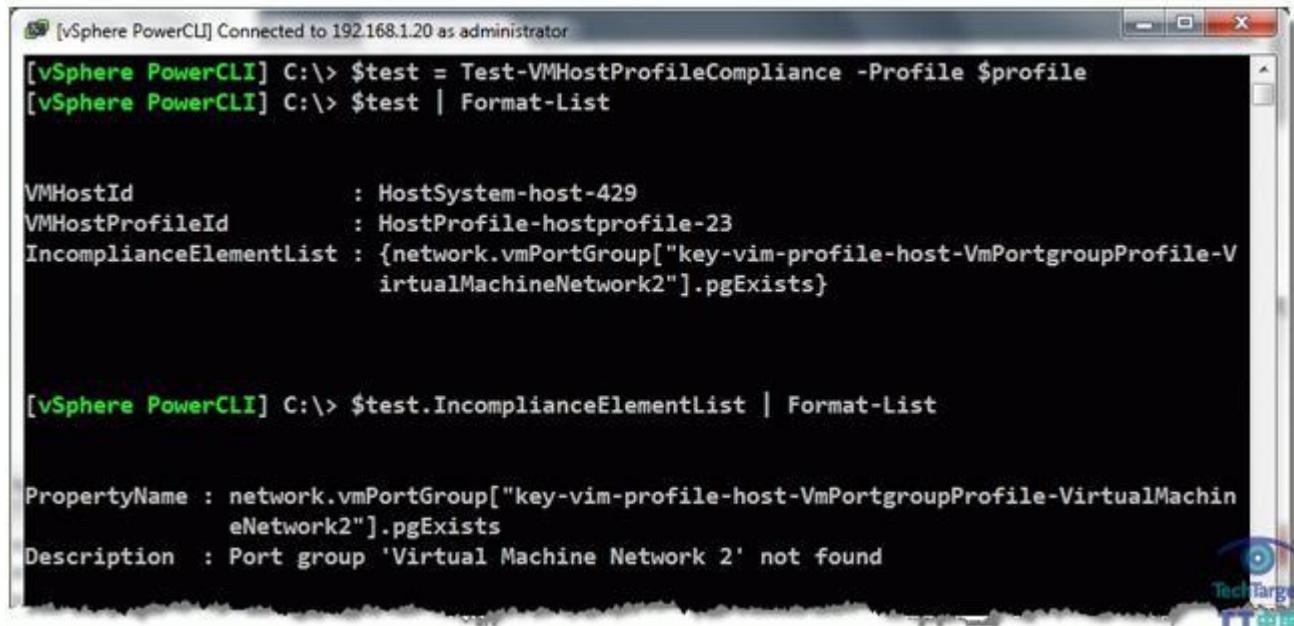


```
[vSphere PowerCLI] Connected to 192.168.1.20 as administrator
[vSphere PowerCLI] C:\> $hpview = $profile | Get-View
[vSphere PowerCLI] C:\> $hpview.Entity | Get-ManagedObjectByView | Select-Object Name, Id
Name                                Id
---                                --
Cluster                            ClusterComputeResource-domain-c278
192.168.1.11                      HostSystem-host-427
192.168.1.12                      HostSystem-host-429
```

检查 host profiles 的一致性

接下来，让我们一起来检查\$profile 变量指代的所有 host profiles 的一致性情况。这个操作是通过 Test-VMHostProfileCompliance cmdlet 实现的。

Test cmdlet 可以返回很多的相关信息，因此我强烈建议您通过变量来指定所需的结果。下面是如何进行操作的演示：



```
[vSphere PowerCLI] Connected to 192.168.1.20 as administrator
[vSphere PowerCLI] C:\> $test = Test-VMHostProfileCompliance -Profile $profile
[vSphere PowerCLI] C:\> $test | Format-List

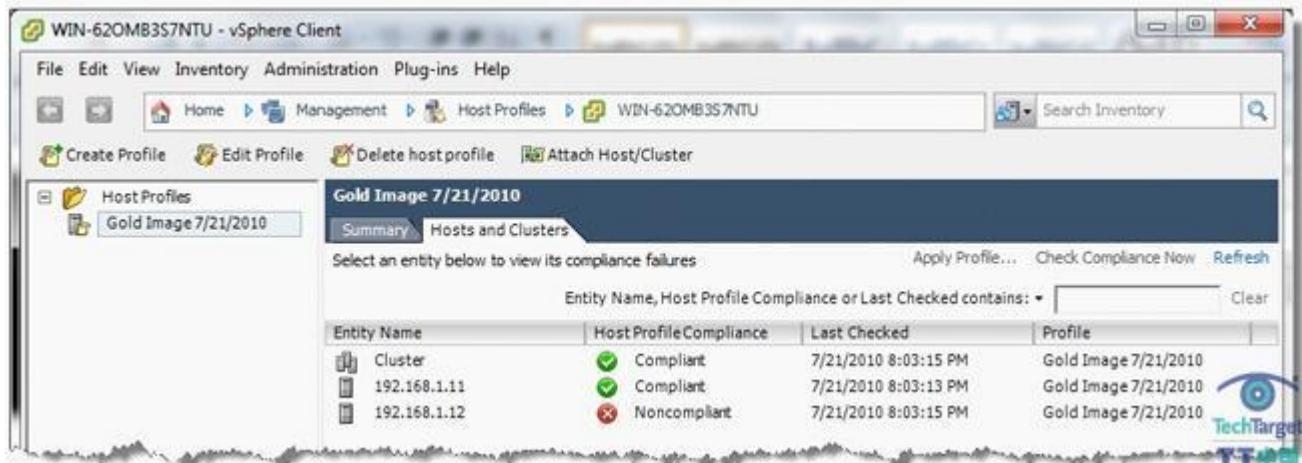
VMHostId          : HostSystem-host-429
VMHostProfileId  : HostProfile-hostprofile-23
IncomplianceElementList : {network.vmPortGroup["key-vim-profile-host-VmPortgroupProfile-V
                           virtualMachineNetwork2"].pgExists}

[vSphere PowerCLI] C:\> $test.IncomplianceElementList | Format-List

PropertyName : network.vmPortGroup["key-vim-profile-host-VmPortgroupProfile-VirtualMachin
                  eNetwork2"].pgExists
Description  : Port group 'Virtual Machine Network 2' not found
```

在第二行中，我在 Format-List cmdlet 中加入了 \$test 变量。这么做的目的是因为在默认的结果列表中，很多相关的信息被隐藏了。而很多有价值的信息都是来自于对 \$test 变量 IncomplianceElementList 属性的核对过程。在截图的最下方，您可以看到很简单地英文提示说明了兼容性检查失败的原因。对于实现 vSphere 环境中配置变化的检查而言，这是一个非常实用的工具。

请参考如下我在使用 vSphere Client 时看到的内容：



WIN-620MB3S7NTU - vSphere Client

File Edit View Inventory Administration Plug-ins Help

Home Management Host Profiles WIN-620MB3S7NTU Search Inventory

Create Profile Edit Profile Delete host profile Attach Host/Cluster

Host Profiles

Gold Image 7/21/2010

Summary Hosts and Clusters

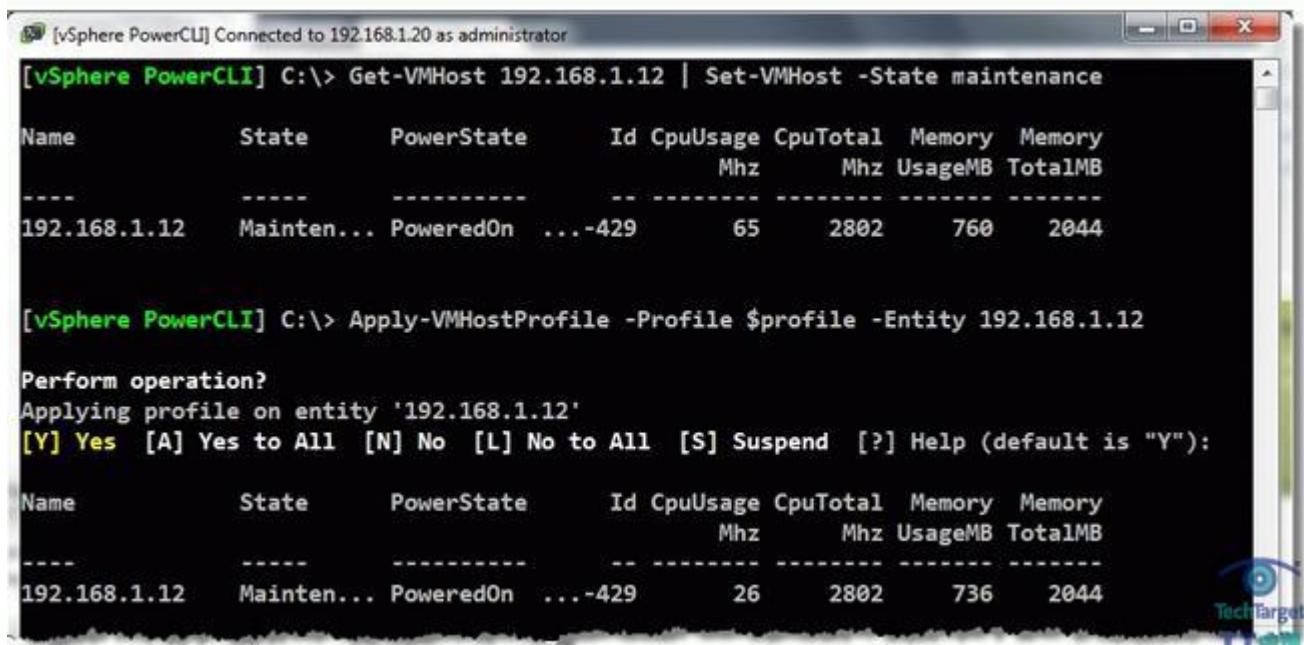
Select an entity below to view its compliance failures

Entity Name, Host Profile Compliance or Last Checked contains: Clear

Entity Name	Host Profile Compliance	Last Checked	Profile
Cluster	Compliant	7/21/2010 8:03:15 PM	Gold Image 7/21/2010
192.168.1.11	Compliant	7/21/2010 8:03:13 PM	Gold Image 7/21/2010
192.168.1.12	Noncompliant	7/21/2010 8:03:15 PM	Gold Image 7/21/2010

应用 host profiles 设置

接下来，让我们开始对 ESX 主机做一致性检查。如上所提到的，主机必须首先被设置为维护模式。您可以通过 PowerCLI 中的 Set-VMHost 以及状态参数 cmdlets 实现这个操作。在开始前，您还需要做好把某台主机从集群中断开的准备，正如我们在使用 vSphere client 时所做的那样。



[vSphere PowerCLI] Connected to 192.168.1.20 as administrator

```
[vSphere PowerCLI] C:\> Get-VMHost 192.168.1.12 | Set-VMHost -State maintenance
```

Name	State	PowerState	Id	CpuUsage	CpuTotal	Memory	Memory
				Mhz	Mhz	UsageMB	TotalMB
192.168.1.12	Mainten...	PoweredOn	...-429	65	2802	760	2044

```
[vSphere PowerCLI] C:\> Apply-VMHostProfile -Profile $profile -Entity 192.168.1.12
```

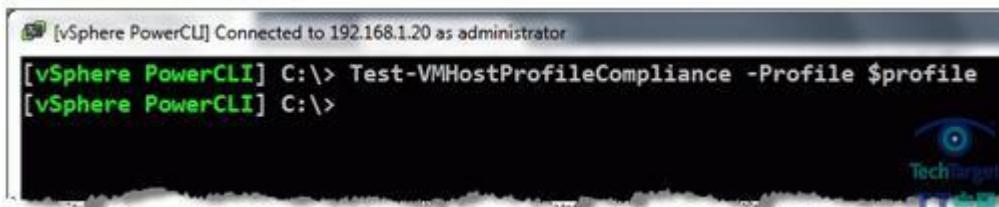
Perform operation?

```
Applying profile on entity '192.168.1.12'
```

```
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

Name	State	PowerState	Id	CpuUsage	CpuTotal	Memory	Memory
				Mhz	Mhz	UsageMB	TotalMB
192.168.1.12	Mainten...	PoweredOn	...-429	26	2802	736	2044

在第二行命令中，我应用了 profile。换句换说，vSphere 创建了丢失的 portgroup。接下来让我们重新检查一次一致性。



```
[vSphere PowerCLI] Connected to 192.168.1.20 as administrator
```

```
[vSphere PowerCLI] C:\> Test-VMHostProfileCompliance -Profile $profile
```

```
[vSphere PowerCLI] C:\>
```

如果此时您在 vSphere Client 中点击刷新，就可以看到所有的项目都已经处于绿色状态。

看过本文之后，您应该已经对于如何配合 host profiles 使用 PowerCLI 有了一定的了解。现在让我们抛开这些理论的东西，开始对系统的一致性检查吧！

(作者: Hal Rottenberg 译者: 李哲贤 来源: TechTarget 中国)

原文标题: 在 vSphere PowerCLI PowerShell 界面使用 host profiles

原文链接: http://www.searchvirtual.com.cn/showcontent_38831.htm

如何用 VMware PowerCLI 脚本来定制 SRM 恢复计划

在本篇快速提示中，我将演示如何把 VMware PowerCLI 脚本功能应用到 VMware SRM (Site Recovery Manager) 恢复计划中。本篇取自于我即将出版的一本关于 VMware SRM4.0 的书。

虽然您还可以采用传统的 (.bat) 文件方式在 SRM 中执行命令行步骤，但是微软提供的 .bat 文件并不是专门创建的，为操作和调整 vSphere 平台而提供的扩展程序接口 (API-advanced programming interface)。如果您的目的是为了制作更为精细的脚本，那么我们确实需要一种更为全面的脚本引擎工具。幸运地是，VMware 即将整合微软的 PowerShell 环境，并推出专用于管理 VMware 虚拟环境的 cmdlets 命令接口。

首先下载并在恢复站点的 SRM 服务器上安装微软的 PowerShell，然后下载和安装 VMware PowerCLI 工具。

一旦我们完成了在恢复站点 SRM 服务器上 PowerCLI 的安装，就可以开始创建 .PS 脚本。并且在此之前，我们要通过 PowerCLI 配置相应的安全设置，并且确保可以通过 PowerCLI 登陆恢复站点的 vCenter，这么做是非常有意义的。

在 SRM 论坛上，最常见的问题之一就是如何减少恢复进程中虚拟机对内存的占用量。这是一个实际上很常见的问题，因为通常情况下恢复站点的 ESX 服务器要相对差一些。例如，或许恢复站点的服务器物理内存数量要少于被保护的站点。通过使用 PowerCLI，我们可以通过 .PS 脚本，在启动虚拟机之前自动调用减少虚拟机内存分配的进程。

通过使用 PowerCLI，可以有很多种实现方法。您可以为每台虚拟机都创建一个 .PS 脚本来减少其内存占用。如下是一个实例，通过这个 .PS 脚本可以完成对我命名为 ctx01 的虚拟机的处理。在脚本中，我使用了 set-vm cmdlet 把恢复站点虚拟机的内存分配减少为 1024MB。另外，-confirm:\$false 命令 用来在需要人工确认更改时，完成操作而无须等待。

例 1

```
connect-viserver vc4nj.corp.com --user corp\administrator --password vmware
Set-VM ctx01 -MemoryMB "1024" -Confirm:$FALSE
Disconnect-VIServer -Server vc4nj.corp.com -Confirm:$FALSE
```

当然，为每台虚拟机都创建一个 .PS 脚本对管理员来说是非常沉重的负担，所以您可能更喜欢通过虚拟机名称来搜索和定位，并对多个虚拟机同时完成修改的操作方式。例如，在如下的 .PS 脚本中，get-vm cmdlet 是用来查找所有以 “ctx” 文本开头的虚拟机，

并把找到的虚拟机传递给 set-vm 命令。这个操作将对那些命名为 ctx01, ctx02, ……等等的虚拟机内存分配做调整。

例 2

```
connect-viserver vc4nj.corp.com --user corp\administrator --password vmware
get-vm ctx* | Set-VM -MemoryMB "1024" -Confirm:$FALSE
Disconnect-VIServer -Server vc4nj.corp.com -Confirm:$FALSE
```

更加精简的脚本方式是，并不指定特定的内存数量，而是通过检查虚拟机内存数量并设置在特定的条件下减少内存分配数量。

例如，假设我需要把分配给恢复的虚拟机的内存数量全部减少为原来的一半。如下的脚本就可以自动查询现在分配给虚拟机的内存数量，然后减少为 50%。通过查找虚拟机名字中的 ctx*字符串来定位并查询现在为虚拟机分配的内存数量，然后使用 set-vm cmdlet 实现正确的设置。

例 3

```
connect-viserver vc4nj.corp.com --user corp\administrator --password vmware
Foreach ($VM in Get-VM ctx*) {
    $NewMemAmount = $VM.MemoryMB / 2
    Set-VM $VM.MemoryMB $NewMemAmount -Confirm:$FALSE
}
Disconnect-VIServer -Server vc4nj.corp.com -Confirm:$FALSE
```

在我的环境中，最终采用了最后一种方法来控制分配给 CTX 虚拟机的内存。这里我想感谢来自英国的 Al Renouf，是他帮助我编写了最后一个例子中的脚本。或许您不知道 Al 是谁，他对 PowerShell 非常的熟悉，而且他的 Virtu-Al blo 博客也非常值得一读。

在下一步中要完成的是，让这些 PS 文件可以被 SRM 调用。相比 SRM 直接调用这些. PS 脚本的方式，我更喜欢创建一个. cmd/. bat 文件，然后通过该文件在合适的时机调用。这种方式可以减少在编写命令脚本程序时的代码行。通过在. cmd/. bat 文件中使用变量，我可以多次使用它来调用保存在 SRM 服务器上的. PS 文件。

步骤 1：创建一个可重定向的. bat 文件

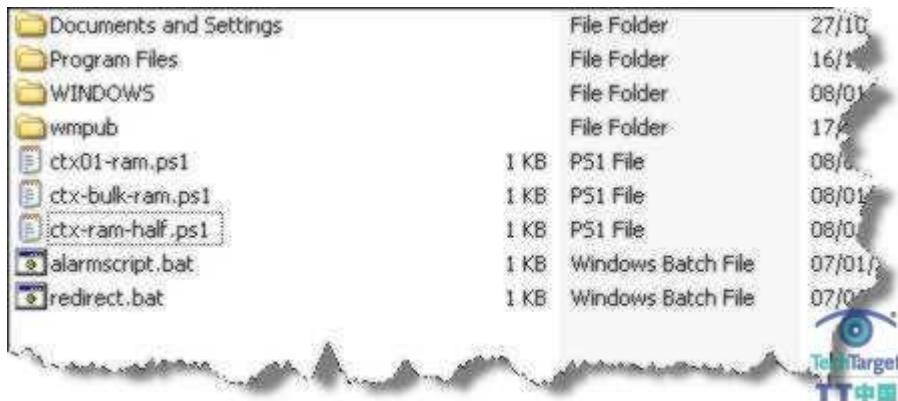
我第一次关注可重定向的. bat 方式是在阅读一个博客 (Carter Shaklin's PowerCLI blog) 中，那里讨论了如何在 vCenter Alarms 中使用. PS 脚本。

在 Virtu-AL 主页的帮助下，我创建了一个可以引用我的. PS1 脚本的. bat 文件。该脚本会加载微软的 Powershell 环境，同时启动 PowerShell 控制台文件 (.psc1，该文件使得 VMware PowerCLI 可以正常工作)。通过末尾的变量字符串 “%1”，可以在同一个重定向的. bat 文件中实现对所有. PS1 文件的调用。

```
@echo off
C:\WINDOWS\system32\windowspowershell\v1.0\powershell.exe -psc "C:\Program
Files\VMware\Infrastructure\vSphere PowerCLI\vim.psc1" "& '%1'"
```

步骤二：把. bat 重定向文件和 powercli.ps 脚本拷贝到恢复站点的 SRM 服务器上

下一步是要完成把 redirect.bat 和 .PS 文件拷贝到恢复站点 SRM 服务器上。相对来说，您把这些文件拷贝到什么地方并不重要，只要您在脚本中正确地键入了文件路径，那么当您在恢复计划中键入命令并执行后，它们就会毫无差错地运行起来。



Documents and Settings	File Folder	27/10
Program Files	File Folder	16/1
WINDOWS	File Folder	08/01
wmpub	File Folder	17/4
ctx01-ram.ps1	1 KB PS1 File	08/0
ctx-bulk-ram.ps1	1 KB PS1 File	08/01
ctx-ram-half.ps1	1 KB PS1 File	08/0
alarmscript.bat	1 KB Windows Batch File	07/01/
redirect.bat	1 KB Windows Batch File	07/0

在这个演示中，ctx01-ram.ps1，ctx-bulk-ram.ps1 和 ctx-ram-half.ps1 分别代表了如上讲过的三种不同处理方式。

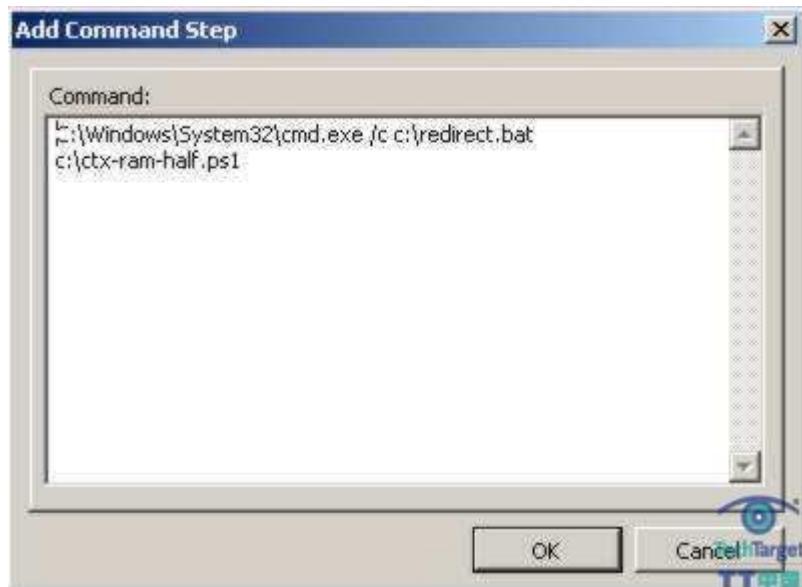
步骤三：在恢复计划中加入命令

1. 在 Recovery Steps 页的恢复计划中，选择 Recovery High Priority Virtual Machines
2. 点击 Add Command Step 按钮



点击看大图

3. 在翻译器中键入您希望执行的. bat 和. ps 文件的完整路径。



注：在本例中，因为对话框比较小而文件的路径非常的长，所以文件部分被覆盖了。
完整命令应该是：c:\windows\system32\cmd.exe /c c:\redirect.bat c:\ctx-ram-half.ps1，这在恢复计划中也可以看到。



点击看大图

.PS 脚本的位置是非常重要的，它必须在对高、中或低优先级虚拟机恢复之前被调用。否则的话，.PS 脚本将会对占位符.vmx 文件做修改，而不是修改原始的虚拟机.vmx 文件。请牢记，在“Prepare Storage”步骤里，占位符.vmx 文件是无法在 vCenter 中注册的，原始的虚拟机.vmx 文件会占据该位置。因此，所有对占位符.vmx 文件所做的修改都会被忽略和抛弃。

或许您对 SRM 自动运行这些脚本感到不太放心。另外一种方式是在恢复计划中输入这些信息并手动执行。

另外，您可能希望了解如何在 vCenter 中实现 PowerShell .PS 文件的验证。在演示中，我把问题简单化了：通过把用户名和密码以纯文本的形式保存在.PS 文件里。其实 PowerShell 有很多种实现验证的方法，但是这些都不是必要的。（Carter Shanklin 的博客讨论了如何使用这种方式来存储经过加密的认证信息，这样的话用户名和密码就不用直接出现在.PS 文件中了。）

警告：最后，请考虑使用 PowerCLI 调整虚拟机在故障回切时所造成的影响。当您决定切回到被保护站点的时候，所做的这些改变也被同时复制到被保护站点。请牢记，我们为了改变虚拟机内存分配而对 VMX 文件做了修改。作为故障回切操作的一部分，我们需要考虑到所有在恢复站点所做的修改在回切时也被反向复制回了被保护站点。为了防止这种情况出现，您需要一个 PS 脚本文件来完成对恢复计划中所做修改的“释放”过程。

结论：

您已经看到，在执行恢复计划时，通过结合 VMware SRM，PowerCLI 可以实现对虚拟机和 vSphere4 平台的各种定制和修改方式。而且，它的作用还不限于此。在 PowerCLI Update1，加入了称为 invoke-vmscript 的新 cmdlet。它可以允许我们在已经完成恢复的子系统虚拟机上调用 PowerShell 和其他脚本程序。这将是我们在下一篇中讨论的问题！

(作者: Mike Laverick 译者: 李哲贤 来源: TechTarget 中国)

原文标题: 如何用 VMware PowerCLI 脚本来定制 SRM 恢复计划

原文链接: http://www.searchvirtual.com.cn/showcontent_32266.htm

VMware 管理员必备：借助 PowerCLI 配置标准交换机

为了说明本文的主旨，最佳的办法是给出一个实例，接下来，TechTarget 特约作者 Mike Laverick 通过例子看看问题如何产生的。在周一的早上，老板通知你由于新项目的需求将要创建一个新的虚拟 LAN(VLAN)。而在我们的 VMware 集群中共有 32 台 ESX 主机，而且是可以支持分布式资源调度 (DRS) 和高可用 (HA) 的 VMware 集群，那么您需要跨所有的 32 台 ESX 主机上创建新的 VLAN，同时创建指定的端口组 VLAN20。

当然，如果您使用了 VMware vSphere4 和新的 vNetwork Distributed Switches，给分布式交换机增加端口组仅仅是一项琐碎的小工作，只需要稍等一会，它就会自动为每台 ESX 主机添加。但是，即使您使用的是 vSphere 4，也可能无法实现自动的过程，因为分布式交换机只有在最昂贵的 vSphere 版本中才会激活，即 Enterprise Plus 版本。

如果没有 Enterprise Plus 版本，您被迫需要使用标准的交换机技术。那么请遵循以下的步骤，在 VMware Infrastructure Client 或 vSphere Client 中操作：

1. 选择 ESX 主机；
2. 选择 Configuration 页；
3. 在 Hardware Pane 中选中 Network 选项；
4. 选择 Standard vSwitch 的参数；
5. 通过运行向导来增加端口组（在这些步骤之外，该向导还有额外的三步操作）；
6. 花费整个早上的时间用于不断地重复步骤 1-5 在其余的 31 台 ESX 主机上。当然，执行一个重复的工作进程是非常枯燥的事情，而且还带有极大的风险可能会发生人为错误。当人们不断地去重复某项工作时，出错误的概率会极大地上升。

PowerCLI 如何拯救这个过程？

VMware 在客户体验上很糟糕的一点就是总让管理员不断地面临这种状况。但是这并不是完全无能为力——通过使用一组 PowerCLI 代码，管理员可以在几分钟的时间内完成这个工作。

该过程的第一步就是学习如何在已存在的 vSwitch 中，为某台 ESX 主机添加一个端口组。接下来，就是把这个过程重复到集群中的每台 ESX 主机上，通过使用 PowerCLI 中的一个非常方便的“for-each”命令就可以把该配置改变应用到所有的 ESX 主机。为了达到这个目的，通过使用 Get-VirtualSwitch cmdlet 命令来选择希望调整的目标 vSwitch，然后配合使用 New-VirtualPortGroup 命令来创建端口组。

接下来要做的就是确保这个过程已经在指定集群的每台 ESX 主机上都正确运行。为了满足安全性和可靠性的需求，您需要采用必要的过滤机制来保证其它的无关 ESX 主机不会

执行该命令。该过滤机制主要用于通知到 PowerCLI，只对数据中心里指定的 VMware HA/DRS 集群中的主机做操作。这五行命令如下：

```
connect-viserver vc4nyc.corp.com -username corp\administrator -password vmware

Foreach ($vmhost in (get-cluster -name "AMD Cluster1" | get-vmhost))
{
    $vs = Get-VirtualSwitch -VMHost $vmHost -Name "vSwitch0"
    $vlan = New-VirtualPortGroup -VirtualSwitch $vs -Name "vlan20" -vlanid 20
}
```

如上的脚本程序会查找名为“AMD Cluster1”中的每台 ESX 主机，然后通过 Get-VirtualSwitch cmdlets 命令来标示出哪台 vSwitch 需要被调整，紧接着通过 New-VirtualSwitch cmdlets 命令来创建端口组。最后脚本程序会重复执行直到队列中的所有 ESX 主机都执行了这些命令。

如果某些情况下确实需要对每个 ESX 集群中的所有 ESX 主机做更改。很简单，您只需把“Get-Cluster -name “AMD Cluster1”这行删掉。然而，虽然说编写一个 PowerCLI 脚本可以对数据中心的每台 ESX 主机更改网络参数固然是一件很不错的事情，然而这样做的同时也为数据中心带来了非常大风险，一旦脚本没有获得很好的测试就会带来大的问题。请牢记：权力越大，责任也相应地越大。

(作者: Mike Laverick 译者: 李哲贤 来源: TechTarget 中国)

原文标题: VMware 管理员必备: 借助 PowerCLI 配置标准交换机

原文链接: http://www.searchvirtual.com.cn/showcontent_36690.htm

最佳免费的 VMware 脚本和命令行工具介绍

无论是使用 VMware ESX 还是 ESXi，VMware 命令行和脚本——如 vMA、Power CLI 和 vSphere Health Check——都可以辅助配置虚拟机、更好地管理 VMware vSphere 基础架构和通过命令行自动完成管理工作。VMware 为命令行和脚本提供三种免费工具，并且这些工具可以共享相同点。

在五种最佳虚拟化工具指南这部分，TechTarget 中国的特约虚拟化专家 David Davis 将介绍 VMware vMA、vCLI、PowerCLI 和 vSphere Health Check 提供的管理和虚拟机自动处理能力。

冬 1

```
vi@vi:/usr/bin$ ls vi*  
viadm-advcfg.pl    viadm-nas.pl    viadm-syslog.pl    viadm.pl  
viadm-efghacking.pl viadm-nics.pl    viadm-user.pl    vihostupdate.pl  
viadm-dns.pl       viadm-ntp.pl    viadm-vmhbaudev.pl vilogger  
viadm-dumpport.pl viadm-rescan.pl  viadm-wknnic.pl    vimaclean  
viadm-module.pl   viadm-route.pl  viadm-vswitch.pl  viptcl-support  
viadm-mpath.pl    viadm-zmp.pl    viip  
vi@vi:/usr/bin$ ls ex*  
excfg-advcfg    excfg-mpath    excfg-route    excfg-wknic  
excfg-efgbackp  excfg-nas      excfg-snap     excfg-vswitch  
excfg-dns       excfg-nics     excfg-syslog  
excfg-dumpport  excfg-ntp      excfg-user  
excfg-module    excfg-rescan  excfg-vmhbaudev  
vi@vi:/usr/bin$ _
```

点击图片本身就能放大

VMware vMA: vSphere 管理助手

VMware 发布 ESXi Hypervisor 时，并不支持命令行接口。虚拟化技术提供商不得不为习惯使用 CLI 的管理员和其他用户（这些用户拥有运行在 ESX 服务器控制台内的代理）提供一种解决方案。VMware 发布了虚拟基础架构管理助手（VIMA：Virtual Infrastructure Management Assistant）作为辅助设备，该设备可以运行 Linux，也有和 ESX 控制台相同的命令行选项。除了使用 VMware 命令行工具，VIMA 还提供 Perl 脚本，并且为这些脚本提供中心化储存池。

当 vSphere 4 发布时, VIMA 重新命名为 vMA, 仍然有同样的功用, 也仍然是 VMware 虚拟基础架构的免费 CLI 管理设备。

更多地了解 [vMA](#)，请阅读我的另外一篇文章：[使用 VMware vSphere 管理助手 \(vMA\)](#)。

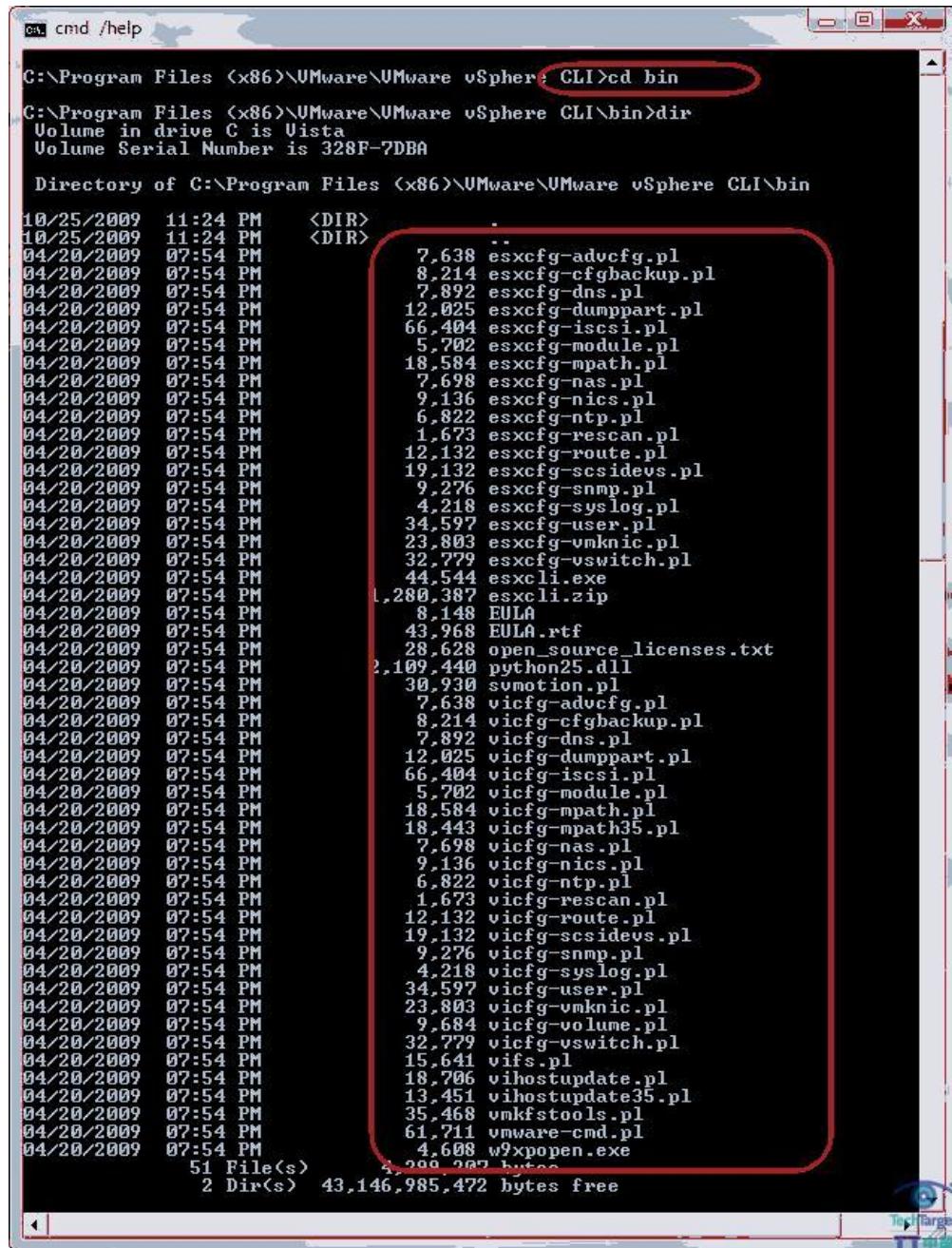
VMware vCLI: vSphere 命令行接口

同样的 VMware 命令行工具可以安装在 Windows 或者 Linux 工作站上，VMware 称这些工具为 vSphere 命令行接口（vCLI: vSphere Command Line Interface）。vCLI 允许用户在 PC 机上运行传统的 VMware 工具，如 esxcfg-nics 或者 vicfg-nics。

仅使用 vMA，就可以有大量 vCLI 可供使用的贡献型脚本（接下来详细介绍）。

访问 [VMware 的 vCLI 主页](#)，可以更多了解 vCLI，并且可以免费下载。

图 2



```

cmd /help

C:\>Program Files (<x86>)\VMware\VMware vSphere CLI>cd bin

C:\>Program Files (<x86>)\VMware\VMware vSphere CLI\bin>dir
 Volume in drive C is Vista
 Volume Serial Number is 328F-7DBA

 Directory of C:\>Program Files (<x86>)\VMware\VMware vSphere CLI\bin

10/25/2009  11:24 PM    <DIR>.
10/25/2009  11:24 PM    <DIR>..
04/20/2009  07:54 PM    7,638 esxcfg-advcfg.pl
04/20/2009  07:54 PM    8,214 esxcfg-cfgbackup.pl
04/20/2009  07:54 PM    7,892 esxcfg-dns.pl
04/20/2009  07:54 PM    12,025 esxcfg-dumppart.pl
04/20/2009  07:54 PM    66,404 esxcfg-iscsi.pl
04/20/2009  07:54 PM    5,702 esxcfg-module.pl
04/20/2009  07:54 PM    18,584 esxcfg-mpath.pl
04/20/2009  07:54 PM    7,698 esxcfg-nas.pl
04/20/2009  07:54 PM    9,136 esxcfg-nics.pl
04/20/2009  07:54 PM    6,822 esxcfg-ntp.pl
04/20/2009  07:54 PM    1,673 esxcfg-rescan.pl
04/20/2009  07:54 PM    12,132 esxcfg-route.pl
04/20/2009  07:54 PM    19,132 esxcfg-scsidevs.pl
04/20/2009  07:54 PM    9,276 esxcfg-snmp.pl
04/20/2009  07:54 PM    4,218 esxcfg-syslog.pl
04/20/2009  07:54 PM    34,597 esxcfg-user.pl
04/20/2009  07:54 PM    23,803 esxcfg-vmknic.pl
04/20/2009  07:54 PM    32,779 esxcfg-vswitch.pl
04/20/2009  07:54 PM    44,544 esxcli.exe
04/20/2009  07:54 PM    1,280,387 esxcli.zip
04/20/2009  07:54 PM    8,148 EULA
04/20/2009  07:54 PM    43,968 EULA.rtf
04/20/2009  07:54 PM    28,628 open_source_licenses.txt
04/20/2009  07:54 PM    2,109,440 python25.dll
04/20/2009  07:54 PM    30,930 svmotion.pl
04/20/2009  07:54 PM    7,638 vicfg-advcfg.pl
04/20/2009  07:54 PM    8,214 vicfg-cfgbackup.pl
04/20/2009  07:54 PM    7,892 vicfg-dns.pl
04/20/2009  07:54 PM    12,025 vicfg-dumppart.pl
04/20/2009  07:54 PM    66,404 vicfg-iscsi.pl
04/20/2009  07:54 PM    5,702 vicfg-module.pl
04/20/2009  07:54 PM    18,584 vicfg-mpath.pl
04/20/2009  07:54 PM    18,443 vicfg-mpath35.pl
04/20/2009  07:54 PM    7,698 vicfg-nas.pl
04/20/2009  07:54 PM    9,136 vicfg-nics.pl
04/20/2009  07:54 PM    6,822 vicfg-ntp.pl
04/20/2009  07:54 PM    1,673 vicfg-rescan.pl
04/20/2009  07:54 PM    12,132 vicfg-route.pl
04/20/2009  07:54 PM    19,132 vicfg-scsidevs.pl
04/20/2009  07:54 PM    9,276 vicfg-snmp.pl
04/20/2009  07:54 PM    4,218 vicfg-syslog.pl
04/20/2009  07:54 PM    34,597 vicfg-user.pl
04/20/2009  07:54 PM    23,803 vicfg-vmknic.pl
04/20/2009  07:54 PM    9,684 vicfg-volume.pl
04/20/2009  07:54 PM    32,779 vicfg-vswitch.pl
04/20/2009  07:54 PM    15,641 vifstools.pl
04/20/2009  07:54 PM    18,706 vihostupdate.pl
04/20/2009  07:54 PM    13,451 vihostupdate35.pl
04/20/2009  07:54 PM    35,468 vmkfstools.pl
04/20/2009  07:54 PM    61,711 vmware-cmd.pl
04/20/2009  07:54 PM    4,608 w9xpopen.exe

51 File(s)    4,299,287 bytes
2 Dir(s)   43,146,985,472 bytes free

```

VMware PowerCLI: PowerShell 命令行接口

如果不使用 Perl 作为 VMware 工具脚本引擎，对于那些熟悉 Microsoft PowerShell 的用户来讲，可以使用 VMware 的产品——PowerCLI。PowerCLI 是基于 PowerShell 的工具，用来管理 VMware 基础架构。

同 vCLI 和 vMA 一样，有大量 PowerCLI 可供使用的贡献型脚本。

访问 [VMware 的 PowerCLI 主页](#)，更多了解 PowerCLI，也可以免费下载。

VMware 的贡献型脚本存储池和 vSphere Health Check 脚本

拥有 Perl 和 PowerShell 脚本选项非常不错，但是如果在命令行中有有用的脚本库就更好了。值得感谢的是，很多 VMware 管理员在 VMware 社区贡献出所创建的脚本程序。例如，[vGhetto script repository](#) 是最好、最大的 VMware 脚本存储池之一。尤其是该存储池里面有我见过最好的脚本：[vSphereHealthCheck 脚本](#)。这个脚本可以查询 ESX 主机、vCenter 服务器和集群，然后返回大量相关信息——提供一个对 vSphere 基础架构清晰、有条理的报告。但是需要有 vMA、vCLI 或者 PowerCLI（使用的应该是 vSphereHealthCheck PowerShell 版本）才可以运行该脚本。

返回指南主页，了解[更多最佳免费虚拟化工具](#)。

(作者: David Davis 译者: 王越 来源: TechTarget 中国)

原文标题: 最佳免费的 VMware 脚本和命令行工具介绍

原文链接: http://www.searchvirtual.com.cn/showcontent_29056.htm

ESX 与 ESXi 管理员必备 25 个命令

正如所有的虚拟化管理员都知道，要应对 VMware 基础设施上需要的更多虚拟化管理，关键在于任务的自动化。虽然 VMware ESX 主机可以管理与 vSphere 客户端用户界面，数据中心管理员往往喜欢登录到 VMware 服务控制台和使用 ESX 命令行来解决，如网络配置问题或重新配置主机。有几个 VMware 的命令，可以帮助这些任务自动化，解决虚拟化基础设施方面的问题，调整您的（虚拟机）和更多的虚拟机的性能。

在本指南中的 VMware 命令行，概述了 25 个最有用 ESX 命令和 ESXi 命令。其中包括 Linux 和 ESX 的特定命令，很多可以用于远程命令行界面（RCLI），vSphere 已改名为 vSphere CLI，可在 VMware ESX 和 ESXi 中使用。

Linux 命令

ESX 服务控制台是基于 Linux 的，因此许多 Linux 命令都可以在它里面使用。下面是常见的 Linux 命令和一些 VMware 命令的详细版本。

- **Find/cat/grep** 命令查找，显示和搜索文件。Find 找出具体的文件，cat 显示文件内容，并把文件拼接在一起，同时 grep 负责在拼接的所有文件中找到所需文本。这些命令可以帮助管理员发现，如基础设施快照文件的具体内容，也显示日志和配置文件。他们还可以搜索内部文件的信息。
- **Tail** 显示一个文本的最后部分的文件，也可以输出到监控实时文件。此命令有助于实时监控日志文件。
- **Service** 可以启动、停止和重新启动主机服务器上的服务（或程序）。一些常见的 ESX 服务包括 MGMT 的和 VMware、vmware-vpxa、firewall、vmware-hostd 和 vmware Web Access。此命令可以重新启动服务，挂起或进行配置更改。
- **Nano 和 vi** 编辑文本文件。Nano 是一个比 Vi 更简单且易于使用的编辑器，但 Vi 具有更强大的编辑功能。在 ESX 主机上文本编辑器帮助编辑配置文件。
- **Su 和 sudo** 命令帮助控制和防止 root 帐户被使用。Su 提升了较差的用户帐户到超级用户的权限。sudo 运行另一个在 sudoers 配置文件中指定的用户命令。
- **ls** 列出文件和目录信息。通过使用某些交换机，可以显示详细的文件资料，包括所有者、大小、权限和最后修改日期。
- **Df 和 vdf** 显示文件系统（分区）的资料，包括自由空间。Df 命令不会显示虚拟机文件系统（VMFS）卷，因为它无法阅读。vdf 是该命令的 VMware 的版本，这也将显示 VMFS 卷的信息。两个命令都可以使用 H 变换，可用易读的形式显示。（即 2 GB，而不是 2016044）
- **Ps 和 kill** 命令可以强行终止 VM。Ps 可以显示主机状态或进程。它可以使用很多开关，但最常见的 EF，它可以显示每个进程运行的所有信息。kill 命令经常与 PS 命令一起用来终止某个正在运行的进程。
- **Ping 和 vmkping** 是最基本的网络故障排除命令。Ping 通过发送 Internet 控制消息协议数据包测试与其他主机和网络设备的网络连接。Vmckping 是 VMware 的 Ping 命令的特定版本。它使用的 IP VMkernel 去 Ping 另一个 ESX 主机的 VMkernel 端口。此命令有助于解决 VMotion 和网络存储的问题。

VMware ESX 和 ESXi 命令

这些 VMware ESX 和 ESXi 的命令可以运行在 ESX 服务控制台（本地或远程使用 Secure Shell）或 RCLI（在 VMware Infrastructure 3）和 vSphere CLI 的（在 vSphere）。在 RCLI 和 vSphere CLI 中，注意到很多命令已更名为 vicfg，而不是 esxcfg -（即 esxcfg - nics.pl 和 vicfg - nics.pl）。两个命令都执行相同的功能，但 VMware 试图从 esxcfg 迁移到 vicfg。

- **vmkfstools** 被比做虚拟磁盘中的瑞士军刀，可用于复制、转换、重命名、输入、输出和调整虚拟磁盘文件的大小。
- **Esxtop** 故障排除。它提供实时的 CPU、内存、硬盘和网络使用的历史表现的统计数字。
- **Esxcfg-nics** 观察和配置物理网络接口卡（NIC）。它显示网卡状态和配置的速度和全双工网卡。
- **Esxcfg-vswitch** 显示和配置虚拟交换机。它是在 vSphere 不能使用网络中有用的配置客户端。该命令用于配置端口组和连接物理网卡（称为上行）配置虚拟局域网 ID，思科协议（CDP）和 vswitch 中的 MTU。
- **Esxcfg-vswif** 和 **esxcfg-vmknic** 允许您查看和配置 vSwitches 特殊的端口组。Esxcfg - vswif 配置的 ESX 服务控制台网络接口，它也被称为 vswif 口。Esxcfg - vmknic 配置 VMkernel 网络接口，这是 VMotion 和连接到 iSCSI 和网络文件系统的网络存储设备所必要的。
- **Vmware-cmd** 是一个复合管理命令，负责管理和检索虚拟机信息。它可以改变虚拟机电源状态、管理快照、注册和注销的用户，并检索和设置各种虚拟机的信息。
- **Vimsh** 和 **vmware-vim-cmd** 是复杂的命令，只有完全了解才能使用。Vimsh 是一个强大的交互式框架，有很多允许执行的命令，以及具备显示和配置能力。VMware 的 vim - cmd 是一种逻辑的 vimsh，能够简化 vimsh，无需知道很多前端交换命令。
- **Vihostupdate** 和 **esxupdate** 更新 ESX 和 ESXi 主机和打补丁。Esxupdate 用于 ESX 服务控制台和 vihostupdate，通过 RCLI / vSphere CLI 使用。此外，vihostupdate35 是用来修补 ESX 和 ESXi 3.5 版主机。
- **Svmotion** 是 **RCLI/vSphere CLI** 命令，用于发起 Storage VMotion 的迁移虚拟机虚拟磁盘到另一个数据存储空间。此命令的 ESX 3.5 版本是唯一启动 SVMotion 的方法，加上 vSphere 客户端的 GUI，vSphere 能做到这一点。
- **Esxcfg-mpath** 显示和设置一台主机从所有路径到达它的存储设备。
- **Esxcfg-rescan** 让主机产生一个特定的存储适配器，用来发现新的存储设备。这是非常有用的工具，存储设备已被添加，删除或从存储网络改变。
- **Esxcfg-scsidevs** 和 **esxcfg-vmhbaidevs** 显示连接到主机的存储设备的资料。Esxcfg - vmhbaidevs 用于 ESX 3.5，在 vSphere 中被 esxcfg-scsidevs 取代。
- **Esxcfg-firewall** 显示信息和配置内置的防火墙保护 ESX 服务控制台。它允许和阻止特定的 TCP /IP 服务之间的控制台和其他网络设备端口。
- **Esxcfg-info** 命令提供了有关运行中的主机信息。它可以重新定向到一个文本文件记录主机配置。
- **Esxcfg-auth** 在 ESX 主机上配置服务控制台验证。它可以配置第三方 LDAP 或 Active Directory 服务器的身份验证并设置多个本地安全选项。
- **Vm-support** 是一个强大的信息收集工具，常用于故障排除。该命令收集大量信息、日志文件，并把很多命令以单一的 tgz 存档文件方式输出。它也可以用来显示 VM 的信息以及停止没有响应的虚拟机。

上述命令有很多不同的语法、选择和变化。有了它们，你的工作将事半功倍。欲了解更多有关这 25 个 ESX 和 ESXi 的命令的信息，请查看以下文件：

-
- [VSphere 命令行界面安装和参考指南](#)
 - [ESX 配置指南（附录）](#)
 - [Linux 命令目录](#)

(作者: Eric Siebert 译者: 李可嘉 来源: TechTarget 中国)

原文标题: ESX 与 ESXi 管理员必备 25 个命令

原文链接: http://www.searchvirtual.com.cn/showcontent_31129.htm