Microsoft System Center PowerShell Essentials

Microsoft PowerShell as a scripting language has been growing strongly over the last couple of years. It has given administrators and IT professionals much more control when managing and implementing tasks within System Center environments. System Center facilitates the configuring, monitoring, and managing of the components of a private cloud. It covers products such as SCCM, SCOM, SCIM, SCVMM, and so on.

Starting with an introduction to PowerShell, this quick reference guide will enable you to get the most out of the latest Microsoft PowerShell techniques to manage the suite of System Center products. You will also get acquainted with the enhancements in the latest version of System Center automation with real-time examples.

By the end of this book, you will have the confidence to create a variety of PowerShell scripts and efficiently administer and maintain your System Center environment with PowerShell.

Who this book is written for
If you are a Microsoft System Center administrator who manages System Center environments and utilizes the console for management, then this book is ideal for you. This book is also for System Center users who now want to learn to manage systems using PowerShell.

What you will learn from this book
- Set up an environment to effectively use PowerShell for SCCM, SCOM, and SCIM
- Understand the advanced usage of all the available cmdlets with real-time examples
- Administer and maintain a System Center environment with PowerShell
- Create both simple and advanced functions for different cmdlets with PowerShell
- Get an insight into real-time System Center applications that you can automate using PowerShell
- Get to grips with scenario-based scripting for efficient System Center administration and maintenance
- Discover PowerShell tricks and best practices


Guruprasad HP
Harshul Patel

要及时而有效率地管理、自动化和维护System Center环境。System Center使IT管理员和专业人士在管理任务时拥有更多控制。它覆盖了SCCM、SCOM、SCIM、SCVMM等产品。

本书从PowerShell的介绍开始，快速指南将帮助你掌握最新Microsoft PowerShell技术，用于管理System Center产品的套件。你还将熟悉最新版本的System Center自动化增强功能，以实时示例。

在读完本书后，你将有信心创建各种PowerShell脚本，并有效地使用PowerShell管理System Center环境。

本书适合的对象
如果你是管理System Center环境的Microsoft System Center管理员，或者使用控制台进行管理，那么本书非常适合你。本书也适合希望使用PowerShell管理系统的System Center用户。

你会从本书中学到的内容包括:
- 设置环境以有效使用PowerShell管理SCCM、SCOM和SCIM
- 理解所有可用cmdlet的高级使用，包括实时示例
- 使用PowerShell管理并维护System Center环境
- 创建简单和高级的函数，针对不同cmdlet
- 了解你可以自动化使用PowerShell应用的实时System Center应用
- 掌握基于场景的脚本编写，以高效地管理与维护System Center
- 发现PowerShell技巧和最佳实践

访问www.PacktPub.com获取书籍、电子书、代码、下载和PacktLib。

Guruprasad HP
Harshul Patel

Microsoft System Center PowerShell Essentials

Efficiently administer, automate, and manage System Center environments using Windows PowerShell
In this package, you will find:

- The author's biography
- A preview chapter from the book, Chapter 1 'Setting up the Environment to Use PowerShell'
- A synopsis of the book’s content
- More information on Microsoft System Center PowerShell Essentials

About the Authors

Guruprasad HP is a technical practitioner and consultant. His technical areas of interest include System Center Configuration Manager, System Center Operation Manager, and automation using PowerShell scripting. He works with Microsoft and is a Microsoft Certified Technology Specialist in SCCM and SCOM.

He has a lot of experience in creating automation frameworks for various activities in SCCM, SCOM, and custom reporting by extending the SCCM inventory. He also has good knowledge of all Microsoft technologies.

Currently, Guruprasad is working with Microsoft as a consultant and is involved in various projects with different Microsoft technologies.

Harshul Patel is a technology enthusiast from India; he is thoroughly knowledgeable in virtualization and cloud computing techniques. He works for Microsoft. Harshul holds multiple Microsoft certifications, including Microsoft Certified Solutions Associate (Windows Server 2012 and Windows 8) and Microsoft Certified Solutions Expert (private cloud). Additionally, he holds a number of non-Microsoft certifications, such as Citrix Certified Administrator (XenApp 6.5, XenDesktop 5.6, and XenServer 6.0) and Citrix Certified Advanced Administrator (XenApp 6.5). He has also recently achieved an ITIL certification.
Harshul was one of the early Indian adopters of Windows PowerShell. He frequently lectures on Windows PowerShell in user group gatherings and delivers training (mostly on PowerShell) across various organizations. He is also a proud recipient of multiple faculty awards and has received an innovation award from his employer. He is a core member of the PowerShell Bangalore User Group (http://powershellgroup.org/bangalore.india) and a member of the New Delhi PowerShell User Group (http://powershellgroup.org/NewDelhi). He can be contacted at http://harshulpatel.com/.

Microsoft System Center PowerShell Essentials

_Microsoft System Center PowerShell Essentials_ mainly focuses on efficiently administering, automating, and managing System Center environments using Windows PowerShell. This book will help you to create powerful automation scripts for System Center products using PowerShell; PowerShell techniques efficiently handle SCCM, SCOM, and SCSM with real-time examples and sample codes. It is a step-by-step guide with practical examples and best practices that teaches you how to effectively use PowerShell in a System Center environment.

Microsoft PowerShell as a scripting language has been growing strongly over the last couple of years. It has given administrators and IT professionals much more control over managing and implementing tasks within System Center environments. It provides vast support for a wide range of vendor products and provides a standardized platform for automation and administration. System Center facilitates the configuration, monitoring, and management of the components of private cloud. It covers products such as SCCM, SCOM, SCSM, SCVMM, and so on.

Starting with an introduction to PowerShell, this quick reference guide will enable you to get the most out of the latest Microsoft PowerShell techniques to manage System Center products. You will get acquainted with the enhancements in the latest version of System Center automation through real-time examples.

By the end of this book, you will have the confidence to create a variety of PowerShell scripts and efficiently administer and maintain your System Center environment with PowerShell.
What This Book Covers

Chapter 1, Setting up the Environment to Use PowerShell, gives you an idea about the purpose of this book, and how to set up the environment with required modules for three products, SCCM, SCOM, and SCSM.

Chapter 2, Administration of Configuration Manager through PowerShell, focuses on administration activities for Configuration Manager, such as inventory, discovery, alert management, and so on.

Chapter 3, Scenario-based Scripting for SCCM Administration, gives you an insight into SCCM real-time applications by giving various scenarios, which are explained with the help of the required code blocks.

Chapter 4, Administration of Operations Manager through PowerShell, focuses on administration activities for Operations Manager, such as monitoring, authoring, basic administration, and so on.

Chapter 5, Scenario-based Scripting for SCOM Administration, gives you an insight into SCOM real-time applications by providing various scenarios, which are explained with the help of the required code blocks.

Chapter 6, Administration of Service Manager through PowerShell, focuses on the administration activities of Service Manager, such as the use of SMlets, incident reporting, managing service requests, and so on.

Chapter 7, Scenario-based Scripting for SCSM Administration, gives you more insight into SCSM real-time applications by providing various scenarios, which are explained with the help of the required code blocks.

Chapter 8, Best Practices, focuses on real-time applications, which can be used to derive best practices for these three products.
Setting up the Environment to Use PowerShell

Welcome! In this introductory chapter, we will throw some light on how the idea for this book came into our minds. Here, we will cover topics that can help users perform various routine tasks in the System Center environment by using legacy consoles.
A decade back, an administrator had to go with legacy Microsoft Management Consoles, broadly known as MMC, for most of the Microsoft products. Now, with the changes in the architecture of the Microsoft products and the birth of automation engines such as Windows PowerShell, automation has become easy; however, many of us are not fully aware of it. Let’s start with setting up the environment.

In this chapter, we will cover:

• The purpose of this book
• The target audience
• Why use PowerShell?
• PowerShell version references
• Setting up the System Center Configuration Manager environment
• Setting up the System Center Operations Manager environment
• Setting up the System Center Service Manager environment
The purpose of this book
This book will help you to achieve the idea of automation, especially in the System Center environment using Windows PowerShell. The purpose of this book is to provide you with an insight of various PowerShell techniques that can be applied to the following three System Center products:

- **System Center Configuration Manager (SCCM)**
- **System Center Operations Manager (SCOM)**
- **System Center Service Manager (SCSM)**

We will also highlight how to use the various PowerShell cmdlets available with these three product SDKs, along with their key tips and tricks. All guidance and assistance will be provided to you on a high-level basis. Further exploration and hands-on experience for these three products is required, so that you gain the most out of this book.

The target audience
This book is aimed mainly at IT professionals who maintain or perform routine activities in the System Center environment focusing on SCCM, SCOM, and SCSM products. This book will be very useful for people who seek out-of-the-box automation for their System Center infrastructure, using Windows PowerShell. You will find real time use of Windows PowerShell with these System Center products.

Why use PowerShell?
In the last few years, the scripting world has witnessed a number of changes. We can hardly recall the time when people used ancient mainframe machines with green-colored text and dark, black-screen backgrounds. Times have changed and we are living in a world where technological adoption is quicker than ever.

Nowadays, an ample number of scripting languages exist, which fulfill the needs of an administrator. One of the questions that arise in one's mind is: why should we go with Windows PowerShell? There are reasons why we prefer Windows PowerShell over other scripting languages. To answer the preceding question precisely, we would rather put a counter question in front of you: give us a valid reason why we shouldn't go with Windows PowerShell.
There are other examples of strong scripting languages, such as VBScript, Ruby, Python, Perl, and so on, and administrators have adopted them too. VBScript became popular because of the automation of routine, local administrator tasks, but the code was a bit complex and hard to understand for novice users. Looking at Windows PowerShell, we feel that the Microsoft team has worked hard to give us a powerful, interactive scripting shell with an object-driven approach.

The important and exciting thing about this language is that it's a spitted object-based output, which can be reused easily. It has pipeline and PSRemoting as its crucial features, which put this language as the first priority while comparing it with other scripting languages. Moreover, by following the Common Engineering Criteria (CEC), Microsoft has decided that all future Microsoft products will come with extensive Windows PowerShell support. This is also a good reason to learn and choose Windows PowerShell. Additionally, PowerShell can be leveraged to use the massive .Net Framework class functionality with most of the Microsoft products. We can also achieve inventory and reporting by efficiently using the WMI functionality that lies within PowerShell. A few Microsoft products support extensive functionality when used with PowerShell; the best example is Exchange Server.

**PowerShell version references**

In this section, we shall talk about the various versions of Windows PowerShell that are available and we will share a few notes on the latest versions v3.0 and v4.0, along with their preinstallation requirements and dependencies.

So far, we have had four stable versions available for Windows PowerShell. Windows PowerShell v1.0 was an extension of Command Prompt with a limited number of cmdlets. In the second version, the team introduced pipeline and PSRemoting concepts, which made Windows PowerShell a popular scripting shell. Furthermore, with the release of Windows Server 2012 and Windows 8, Windows PowerShell version 3.0 was a drastic improvement in terms of the number of cmdlets and modules. They have also introduced the Windows PowerShell Web Access (PWA), PowerShell Workflows, and Scheduled Jobs concept in this version. The exciting part is that while we were drafting this book, the Microsoft team was coming up with its next release of operating systems, named Windows Server 2012 R2 and Windows 8.1. In this release, they have introduced Windows PowerShell v4.0 embedded with extensive functionality, such as Desired State Configuration (DSC) and so on.

While we are in the process of publishing this book, the PowerShell team has already come up with the preview release of Windows PowerShell 5.0 with some extensive functionality.
By default, Windows PowerShell 3.0 comes up with Windows Server 2012 and Windows 8. There are a number of default modules present in this version. If you are running an operating system lower than the ones specified in the preceding section, you need to manually install Windows Management Framework 3.0, which is also known as WMF 3.0.

If you have installed any previous releases of Windows Management Framework, you must uninstall them before installing Windows Management Framework 3.0.

Windows Management Framework 3.0 can be installed only on the following operating system versions:

- Windows 7 SP1
- Windows Server 2008 R2 SP1 (WMF 3.0 is also supported if you are running Windows Server 2008 R2 as the server core)
- Windows Server 2008 SP2

Windows PowerShell 2.0 is embedded in the Windows Server 2008 R2 and Windows 7 operating system. You don't need to separately install it on these operating systems.

The contentions written here use the latest version of PowerShell (v 4.0). However, most of the cmdlets are also supported in the legacy version, as well. As a minimum, you need to have PowerShell 2.0 in your machine; however, it would be best to have the latest version of PowerShell. You can refer to the TechNet link (https://technet.microsoft.com/en-us/library/hh847769.aspx) for detailed information on the prerequisites for different versions of PowerShell.

Windows Management Framework 3.0 is available for all supported versions of Windows in the following languages: English, Chinese (simplified), Chinese (traditional), French, German, Italian, Japanese, Korean, Portuguese (Brazil), Russian, and Spanish.

Windows Management Framework 3.0 contains:

- Windows PowerShell 3.0
- **Windows Remote Management (WinRM) 3.0**
- **Windows Management Instrumentation (WMI)**
- Management OData IIS Extensions
- Server Manager CIM Provider
Windows Management Framework 3.0 requires the following software to be installed prior to the WMF 3.0 installation:

- **Windows 7 Service Pack 1 on computers running Windows 7**: To install SP1, go to http://www.microsoft.com/en-in/download/details.aspx?id=5842

In addition to the preceding software, you will need to meet the following requirements:

- To install Windows PowerShell Integrated Scripting Environment (ISE) for Windows PowerShell 3.0 on computers running Windows Server 2008 R2 with Service Pack 1, use Server Manager to add the optional Windows PowerShell ISE feature to Windows PowerShell before installing WMF 3.0.
- Install the latest updates before installing WMF 3.0.

WMF 4.0 has the same set of OS requirement, but it needs Microsoft .Net Framework 4.5 as a prerequisite.

### Setting up the System Center Configuration Manager environment

This section talks about how to setup your Windows PowerShell console to start with the SCCM activities. The traditional method of importing the SCCM module in Windows PowerShell is supported by SCCM 2007 and its later versions.

The prerequisites to set up SCCM are as follows:

- SCCM 2007 or its later version infrastructure
- Windows PowerShell 2.0 or its later version
Connecting to Windows PowerShell for SCCM

The steps for connecting to Windows PowerShell for SCCM are as follows:

1. Start the 32-bit Windows PowerShell console from your operating system box, as the SCCM infrastructure is only supported with the 32-bit PowerShell architecture.

2. If you are using Windows Server 2008 R2 or a similar operating system, then you can click on Start, search for Windows PowerShell (x86), and launch the console.

   If you are using Windows Server 2012 or a similar operating system, then you can press the Windows key + F, search for Windows PowerShell, and choose Apps in the console. From the search list, select Windows PowerShell (x86) and launch the console.

3. To import the Configuration Manager PowerShell module, we need to change the console location to the Configuration Manager Installation folder. For example, we will refer to the parent installation folder as C:\Program Files(x86).

4. Type the following lines into the PowerShell console:

   ```
   PS C:\> cd "C:\Program Files(x86)\Microsoft Configuration Manager\AdminConsole\bin"
   ```

   This will set the console location to the bin subfolder in the Configuration Manager Installation folder.
5. Now, import the ConfigurationManager.psd1 module file by using the Import-Module cmdlet:

```bash
PS C:\> Import-Module .\ConfigurationManager.psd1
```

To confirm the successful import of the module, you can type `Get-Module` CMDLET in the PowerShell console. Now you will be able to see the new module added to the ConfigurationManager list.

6. After successfully importing the module file, set the console location to your site location by using your site code. For example, we have taken ABC site code in the following command statement:

```bash
PS C:\> Set-Location ABC:
```

The Configuration Manager PowerShell module also includes PowerShell Driver Provider for Configuration Manager Sites. For example, if you have a central site administration, site ABC and two primary sites PS1 and PS2, then you can change the connection context like this:

```bash
PS C:\> Set-Location ABC:
PS C:\> Set-Location PS1:
PS C:\> Set-Location PS2:
```

If you don't change the connection context, then you can't manage the Configuration Manager Site as well.

7. Now you are ready to manage your Configuration Manager infrastructure using Windows PowerShell.

**Downloading the example code**

You can download the example code files from your account at http://www.packtpub.com for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.
There is also another simple method available to connect SCCM using PowerShell with the latest releases of SCCM 2012 and so on. The prerequisites for that are as follows:

- System Center Configuration Manager 2012 SP1 RTM or a later version infrastructure
- Windows Server 2012 or Windows Server 2008 R2 with WMF 3.0

## Connecting to Windows PowerShell from the SCCM console

The steps for connecting to Windows PowerShell from the SCCM console are as follows:

1. Press the Windows key + F, search for **Configuration Manager**, and choose **Apps**. From the search list, select **Configuration Manager Console** and launch the console.
2. In the **Configuration Manager Console**, click on the upper-left corner of the console and select **Connect via Windows PowerShell**.

3. The **Configuration Manager** then imports the **PowerShell** module automatically.

4. Now you are ready to manage your **Configuration Manager** infrastructure using the **Windows PowerShell** console.

### Setting up the System Center Operations Manager environment

This section discusses how to set up your PowerShell console to start with the **SCOM** activities. The traditional method of importing the **SCOM** module in Windows PowerShell is supported by **SCOM 2012** and its later versions.

The prerequisites for this are as follows:

- SCOM 2012 or the later version infrastructure
- Windows PowerShell 2.0 or its later version
Connecting to Windows PowerShell for SCOM

The steps for connecting to Windows PowerShell for SCOM are as follows:

1. Start the 32-bit Windows PowerShell console from your operating system box.

2. If you are using Windows Server 2008 R2 or a similar operating system, then you can click on Start, search for Windows PowerShell (x86), and launch the console.

3. If you are using Windows Server 2012 or a similar operating system, then you can press the Windows key + F, search for Windows PowerShell, and choose Apps. From the search list, select Windows PowerShell (x86) and launch the console.

4. To import the Operations Manager PowerShell module, we need to change the console location to the Operations Manager Console installation folder. For example, we will refer to the parent installation folder as C:\Program Files(x86).
   Type the following lines into the PowerShell console:
   ```powershell
   PS C :\> cd 'C:\Program Files\System Center 2012\Operations Manager\PowerShell\'
   ```
   This will set the console location to the PowerShell subfolder in the Operations Manager Console installation folder.

5. Now, import the OperationsManager.psd1 module file by using the Import-Module cmdlet:
   ```powershell
   PS C :\> Import-Module .\OperationsManager.psd1
   ```
   To confirm the successful import of the module, type the Get-Module cmdlet on the PowerShell console. Now you will be able to see the new module added to the OperationsManager list.

6. Now you are ready to manage your Operations Manager infrastructure, using the Windows PowerShell console.
Setting up the System Center Service Manager environment

In this section, we will talk about how to set up your PowerShell console to start with the SCSM activities. The traditional method of importing the SCSM module in Windows PowerShell is supported by SCSM 2010 and its later versions.

Here are the prerequisites to set up the SCSM environment:

- SCSM 2010 or its later version infrastructure
- Windows PowerShell 2.0 or its later version

Connecting to Windows PowerShell for SCSM

1. Start the Windows PowerShell console from your operating system box.
2. If you are using Windows Server 2008 R2 or a similar operating system, then you can click on Start, search for Windows PowerShell (x86), and launch the console.
3. If you are using Windows Server 2012 or a similar operating system, then you can press the Windows key + F, search for Windows PowerShell, and choose Apps. From the search list, select Windows PowerShell (x86) to launch the console.
4. To import the Service Manager PowerShell module, we need to change the console location to the Service Manager Console installation folder. For example, we will refer to the parent installation folder as C:\Program Files(x86).
5. Type the following lines into the PowerShell console:
   ```powershell
   PS C:\> cd 'C:\Program Files\Microsoft System Center 2012\Service Manager\'
   This will set the console location to the Service Manager subfolder in the Service Manager Console installation folder.
   ```
6. Import the System.Center.Service.Manager.psd1 module file for SCSM Management Servers by using the Import-Module cmdlet:
   ```powershell
   PS C:\> Import-Module .\System.Center.Service.Manager.psd1
   ```
7. Now you are ready to manage your Service Manager infrastructure for SCSM Management Servers using Windows PowerShell.

8. Import the `Microsoft.EnterpriseManagement.Warehouse.Cmdlets.psd1` module file for Data Warehouse Management Servers by using the `Import-Module` cmdlet:

   ```powershell
   PS C:\> Import-Module .\Microsoft.EnterpriseManagement.Warehouse.Cmdlets.psd1
   ```

   To confirm the successful import of the module, you can type the `Get-Module` cmdlet on the PowerShell console. Now you will be able to see the new module added to the `System.Center.Service.Manager` and `Microsoft.EnterpriseManagement.Warehouse.Cmdlets` lists.

9. Now you are ready to manage your Service Manager infrastructure for Data Warehouse Management Servers using Windows PowerShell.

**Summary**

By end of this introductory chapter, you should be able to understand the basic terminology and setup requirement to use Windows PowerShell with several System Center products.

Going ahead, we will specifically look at each of these products and try to explore more functionalities that we can achieve using Windows PowerShell.
Where to buy this book

You can buy Microsoft System Center PowerShell Essentials from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.