# **Module 2**

1. **Configuring Windows Server 2019**

|  |
| --- |
| **At a Glance** |

#### Instructor’s Manual Table of Contents

* Overview
* Objectives
* Teaching Tips
* Quick Quizzes
* Class Discussion Topics
* Additional Projects
* Additional Resources
* Key Terms

|  |
| --- |
| Lecture Notes |

# **Overview**

In this module, your students will follow-up on the installation material covered in Module 1 and learn about tools an administrator can use to monitor and manage Windows Server systems. The first tool covered in this module is the Server Manager. It is installed by default, provides the capability to combine several different administrative tools and functions together, and can install roles from a single console. After discussing installing roles within Server Manager, this module introduces the Best Practices Analyzer tool, and explains how your students can use it to verify that configurations follow Microsoft best practices. Next, the Windows Admin Center tool is discussed. Details are given to explain how to provide remote Web-based management for several Windows Server systems. There are two additional tools your students should be aware of that help configure devices. These two tools are the Devices and Printers tool and the Device Manager tool. Both are examined to illustrate the importance of Windows Server 2019 device management. A part of device management is verifying driver and file integrity. For these tasks, this module introduces the Sigverif and the System File Checker tools. With roles, management tools, and device installations complete, your students will learn that it is important to ensure that performance options are configured properly to ensure a smoothly running system. To wind down this module, the importance of the Windows Registry is discussed in some detail. Finally, the notion of cmdlets is introduced and the powerful PowerShell utility and its features are explained in detail. Together, these tools provide an arsenal of power at your students’ disposal so they can confidently configure a Windows Server 2019 system.

# **Module Objectives**

* Use Server Manager to monitor and manage Windows Server systems
* Install and use the Windows Admin Center to monitor and manage Windows Server systems
* Configure server hardware devices
* Use the System File Checker and Sigverif to verify system files
* Configure key Windows Server components within Control Panel and Device Manager
* Explain the purpose and configuration of the Windows Registry
* Identify the components, features, and usage of Windows PowerShell
* Use Windows PowerShell to manage a server
* Create PowerShell scripts for systems administration

# **Teaching Tips**

**Working with Server Manager**

1. Introduce this topic by reviewing some advantage of using Server Manager.
* Starts by default
* Combines several different administrative tools and functions together
1. Discuss how Server Manager can be used to monitor and manage several different Windows Server systems on a network, as well as the roles they provide.
2. Point out that as an administrator adds more roles to their server, or more servers to the Server Manager console, he or she will notice more sections displayed in the navigation area.
3. Refer to Figure 2-1 to illustrate the Sever Manager Dashboard area sections.
4. Refer to Figure 2-2 to illustrate and explain why many administrators choose to hide the Welcome to Server Manager pane.
5. Explain the color system and icons implemented in the Sever Manager Dashboard.
* A green arrow is used next to the boxes within the Roles and Server Groups pane.
* Each box is color coded so an administrator can see right away if there is a problem on a server.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Mention that although an administrator can add other Windows Server 2008 and later servers to Server Manager in Windows Server 2019, not all monitoring and management features are available in Windows Server 2012 R2 and earlier. |

1. Explain that each section within the navigation area of Server Manager has additional panes that allow an administrator to monitor and manage the associated server, server group, or server role:
* Events pane
* Services pane
* Best Practices Analyzer pane
* Performance pane
* Roles and Features pane
1. Refer to Figure 2-3 through Figure 2-7 to illustrate properties and features of the additional panes.
2.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Mention that because results of a BPA Scan are often large, an administrator can use the Filter dialog box within this pane to display only certain results. |

1. Refer to Figure 2-8 to illustrate how the Server Manager Properties window can be used to set default values such as the refresh time interval.
2. Discuss how some server roles that were introduced starting with Windows Server 2012 have their configuration tools built into the Server Manager interface. Explain that Storage Spaces is an example of one such role. Refer to Figure 2-9 to illustrate how Storage Spaces can be configured on any server that has been added to the Server Manager console.
3. Discuss how some server roles have their own MMC tool for configuration. Refer to Figure 2-10 to illustrate how to easily access these tools by selecting them from the Tools menu—as opposed to selecting the tools from the Start menu on the local server.
4. Explain how to start an MMC tool that connects to another server that is managed by Server Manager. Refer to Figure 2-11 to illustrate the processes using the DHCP server role.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Explain that an administrator can install the Remote Server Administration Tools (RSAT) on a Windows 10 PC that is joined to an Active Directory domain within the organization. This allows the administrator to perform server administration remotely using Server Manager and a wide range of MMC tools from a Windows 10 PC within their IT office.Demonstrate how to locate, download, and install RSAT. Visit the following Web site to obtain the product: <https://www.microsoft.com/en-ca/download/details.aspx?id=45520> |

1. **Adding Roles and Features Using Server Manager**
2. Introduce this topic by explaining why it is important to know how to properly install and verify server roles on a Windows Server 2019 system.
3. Introduce the three different ways to add roles and features within Server Manager.
* Selecting Add roles and features from the Welcome to Server Manager pane within the Dashboard section
* Selecting Add Roles and Features from the Manage menu
* Selecting Add Roles and Features from the Tasks menu within the Roles and Features pane for a server or server role
1. Emphasize that regardless of which method an administrator selects, the Add Roles and Features Wizard will start to guide him or her through the installation of the role.
2. Refer to Figure 2-12 through Figure 2-17 to illustrate how to add a role.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Remember that in general, server services are displayed under the roles section, and supporting features and programs are displayed under the features section. However, an administrator will find some server services under the features section, such as the iSNS Server Service shown in Figure 2-16. |

1. **Using the BPA to Verify Server Roles**
2. Explain the purpose of a BPA scan.
3. Review the resulting levels of security.
4. Review the resulting categories an administrator will see for BPA recommendations.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | For more information on reading a BPA report, see the following Web site: <https://docs.microsoft.com/en-us/dynamics365/unified-service-desk/admin/read-best-practices-analyzer-report?view=dynamics-usd-4.1> |

1. **Working with the Windows Admin Center**
2. Introduce this topic by briefly reviewing a disadvantages of using Server Manager.
* An administrator often needs to connect to Windows Server system to run it, or install the RSAT on a Windows 10 PC.
1. Introduce the Windows Admin Center as a tool that can be used to remotely manage Windows Server 2019 from any computer that has a modern Web browser. Mention that this is often preferred in environments that host a large number of remote Windows Server systems that must be managed centrally, including cloud environments.
2. Point out that Windows Admin Center boasts a wide range of monitoring and management functionalities that Microsoft wants to expand to all areas of Windows Server administration.
3. Explain that when administrators first open Server Manager on Windows Server 2019, they are prompted to download the Windows Admin Center in an information dialog.
4. **Installing the Windows Admin Center**
5. Distinguish between the preview and regular versions of the Windows Admin Center.
6. Explain that the download process downloads an installer file called WindowsAdminCenter<version>.msi.
7. Explain that after starting the installer, an administrator will navigate through several screens, clicking Next each time.
8. Refer to Figure 2-18 through Figure 2-20 to illustrate how to install the Windows Admin Center. Discuss the details of these screens displayed as part of the installation process.
* Prompts to accept the license agreement
* Prompts to automatically update the Windows Admin Center with Windows Update (recommended)
* Screens to view information about how the Windows Admin Center functions in different scenarios
* Prompts to allow the Windows Admin Center to modify the local computer’s trusted host settings
* Prompts to either generate an encryption certificate for use with HTTPS (called a self-signed certificate), or supply the thumbprint for an existing HTTPS certificate that is already installed on the computer and signed by a public Certification Authority
1. Note that at the end of the installation, the URL that can be used to access it will be shown on the final screen.
2. **Using the Windows Admin Center**
3. Explain that when an administrator navigates to the Windows Admin Center website in a modern Web browser (e.g., Google Chrome) for the first time, the administrator is prompted to log in with valid credentials on the gateway server or Active Directory domain to which the gateway server is joined.
4. Refer to Figure 2-21 and Figure 2-22 to illustrate the events that take place when setting up and using the Windows Admin Center.
5. Review the similarities in functionality between the Windows Admin Center and Server Manager.
6. Note that an administrator can restart, shut down, or edit the computer name and domain membership (Edit computer ID), just as they can in the Local Server section of Server Manager. Emphasize that other server configurations, such as remote access, power options, and Windows Admin Center roles, can be handled by clicking the Settings icon below the navigation pane.
7. Explain that most of the tools within the navigation pane allow an administrator to configure additional areas of the system or access configuration functionality that would normally be provided by MMC tools. Review these tools.
8. Discuss the tools within the navigation pane for obtaining remote access to a system.
9. Discuss the tools within the Windows Admin Center that are focused on providing integration with the Microsoft Azure cloud.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Point out to your students that as the authors noted, an administrator could also manage clusters within the Windows Admin Center by selecting Failover Cluster Manager from the Server Manager drop-down menu shown in Figure 2-22. If you have this menu available, this might be a good class demonstration item. |

**Configuring Server Hardware Devices**

1. Introduce this topic by explaining why an administrator might need to replace existing hardware in a server.
2. List examples of hardware devices.
3. Define the term Plug and Play (PnP) and explain what is meant by having a PnP device.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Inform your students that another term they may run across, universal PnP (UPnP), refers to an open standard that is used in all types of systems and that enables connectivity through networks and network protocols. Explain that UPnP supports server-based networking, wireless networking, peer-to-peer networking, and other networking services. |

1. Review the simple steps to install a PnP device.
2. Discuss any additional concerns that might need to be addressed.

**Adding Hardware Using Control Panel**

1. Discuss why an administrator would use the Devices and Printers utility to manually launch PnP or to manually install a device without PnP.
2. Introduce the three broad tasks the Devices and Printers utility can be used to complete.
* Force the operating system to detect and install new PnP hardware
* Install non-PnP hardware
* Troubleshoot problems an administrator might be having with existing hardware
1. Refer to Figure 2-23 and explain how to start the Devices and Printers utility from Control Panel. Distinguish between Category view and Classic view.
2. Explain how to add a device using the Devices and Printers utility.
3. Explain how to display a device using the Devices and Printers utility.
4. Explain how to troubleshoot a device using the Devices and Printers utility.

**Using Device Manager**

1. Discuss why an administrator might need to use the Device Manager utility.
2. Explain how to open the Device Manager utility.
3. Refer to Figure 2-24 and explain how to update the device driver for a device within Device Manager.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Devices that show up with Generic or Unknown within Device Manager will require an updated driver to provide full functionality. Some hardware devices may require that an administrator first remove the existing generic device driver by choosing Uninstall device from the right-click menu shown in Figure 2-24. |

1. Discuss issues that must be considered when PnP hardware is installed and the resources that the hardware requires need to be coordinated with the system to ensure that there are no conflicts.
* Interrupt Request (IRQ) line
* Input/output (I/O) address
* Reserved memory range
1. Explain that resource conflicts can sometimes occur when a network interface, a new storage controller, or some other hardware is added to the system that does not communicate properly with PnP, or is not fully PnP compliant.
2. Refer to Figure 2-24 and explain how to use Device Manager to check for a resource conflict as well as examine other properties associated with a device.
3. Refer to Figure 2-25 and explain the contents and purpose of the four tabs in the Properties box for each device in Device Manager.
4. **Quick Quiz 1**
5. Which tool can be used to monitor and manage several different Windows Server systems on a network as well as the roles that they provide?
6. Server Manager
7. Windows Admin Center
8. Control Panel
9. Device Manager
10. Answer: a. Server Manager
11. When an administrator installs the Windows Admin Center on a Windows Server 2016 or 2019 system to provide remote Web access for administrations, it is said to function in \_\_\_\_\_\_\_\_\_\_ mode, as it provides the ability to manage other Windows Server systems on the network.
	1. gateway server
	2. routed
	3. matrixed
	4. open
12. Answer: a. gateway server
13. Which tool allows an administrator to scan servers and roles for configuration issues that do not follow Microsoft’s recommendations?
	1. Windows Registry editor scan
	2. Best Practices Analyzer (BPA)
	3. Windows Performance Monitor
	4. Microsoft Server Performance Advisor (SPA)
14. Answer: b. Best Practices Analyzer (BPA)
15. Which view in Control Panel displays utilities grouped by general function?
	1. Properties
	2. Expanded
	3. Classic
	4. Category
16. Answer: d. Category
17. **Verifying System Files**
18. Introduce this topic by explaining that device drivers are not the only files on a system that are signed. Note that many other files that comprise Windows Server 2019 require a signature.
19. List some ways system file signatures can become invalid.
* Overwritten
* Corrupted
* Modified by malware
1. Introduce the System File Checker as a Windows Server 2019 tool that can scan system files for integrity and replace damaged or overwritten files with the proper version.
2. Discuss how to start and run the System File Checker tool manually from a Command Prompt window (MS-DOS shell) or Windows PowerShell window by executing the sfc/scannow command.
3. Refer to Figure 2-26 to illustrate how the System File Checker scan will indicate whether issues were found and repaired after running for several minutes.
4. Emphasize that an administrator can use the sfc/scanfile:filename command to scan a single file that he or she believes is corrupted.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | For more information on the Windows Server 2019 System File Checker scan tool, see the following Web site: <https://support.microsoft.com/en-us/help/4026529/windows-10-using-system-file-checker>   |

1. Introduce the Windows Server 2019 File Signature Verification tool (Sigverif), which verifies system and critical files, including device drivers, to determine if they have a signature. Emphasize that this tool only scans files and does not overwrite inappropriate files, thereby allowing safe use of the tool while the server is active.
2. Point out that after the Sigverif scan is complete, the results are written to a log file, called sigverif.txt.
3. Mention that if the tool finds a file without a signature that an administrator believes needs to be replaced, he or she can replace the file using the System File Checker.
4. Explain that an administrator can run the System File Checker manually from a Command Prompt or Windows PowerShell window by executing the sigverif command.
5. Refer to Figure 2-27 and note that if an administrator runs the System File Checker tool from the Windows PowerShell window by executing the sigverif command, the Signature Verification window opens. Explain that an administrator can then click the Advanced button to change the location and name of the log file, or Start to start the scan process.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | For more information on the Windows Server 2019 File Signature Verification tool (Sigverif), see the following Web site: <https://www.processlibrary.com/en/directory/files/sigverif/28294/>   |

1. **Configuring Windows Settings**
2. Introduce this topic by listing different elements of the operating system that can be configured by an administrator using tools included with Windows Server 2019.
* Performance options
* Environment variables
* Startup and recovery options
* Power options

**Configuring Performance Options**

1. Introduce the main areas that an administrator can configure within the operating system to optimize performance.
* Processor scheduling and Data Execution Prevention
* Virtual memory
* File caching and flushing
1. Introduce and discuss the topic of configuring processor scheduling and Data Execution Prevention.
* Explain that processor scheduling allows an administrator to configure how processor resources are allocated to programs.
* Discuss how to access the processor scheduling option within Control Panel.
* Refer to Figure 2-28 to illustrate the Performance Options window and the Advanced tab used to modify processor scheduling options.
* Introduce and discuss the performance (and security) option known as Data Execution Prevention (DEP).
* Explain that when programs are running on the server, DEP monitors how they use memory to ensure they are not causing memory problems. Emphasize that this is intended to foil malware, such as computer viruses, Trojan horses, and worms.
* Mention that some types of applications might not work well with DEP.
* Refer to Figure 2-29 to illustrate how to configure DEP options within Windows Server 2019 by navigating to the same area where processor scheduling is configured.
1. Introduce and discuss the topic of configuring virtual memory.
* Discuss what is meant by the concept of virtual memory.
* Note that virtual memory uses a technique called paging, whereby blocks of information, called pages, are moved from physical memory into virtual memory on disk. Explain the concept of pages being allocated into blocks of 4 KB of data.
* Explain what a page file is and how it is allocated. Emphasize that a default amount of virtual memory is always established when Windows Server 2019 is installed, but an administrator should periodically check to ensure that the amount of virtual memory is appropriate for the needs of the server.
* Explain why the location of the paging file is important. Provide examples of where the page file can be located if multiple hard drives are available.
* Discuss how to tune the size of the paging file by setting two parameters: initial size and maximum size. Note that a general rule for configuring the initial size is to multiply the amount of installed RAM times 1.5.
* Refer back to Figure 2-28 to remind students how to get to the same area where they configured processor scheduling.
* Refer to Figure 2-30 to illustrate how an administrator would click the Change button in the Virtual memory section, and deselect Automatically manage paging file size for all drives.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | For more information on virtual memory, see the following Web site: <https://computer.howstuffworks.com/virtual-memory.htm>  |

1. Introduce and discuss the topic of configuring file caching.
* Define and describe what is meant by the term *file caching*.
* Define and describe what is meant by the term *flushing*.
* Discuss why an administrator might turn off caching and flushing. Point out some issues that might occur if caching is turned off.
	+ The server can seem slower to users, particularly during times of heavy disk read and write operations.
	+ When flushing is turned off, less memory is used for file operations, but there may be data loss when a disk drive is hot swapped while the server is in use.
* Note that in most cases, server performance is better and disk operations are safer when file caching and flushing are turned on.
* Refer to Figure 2-31 to illustrate how to configure file caching and flushing.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | For more information on file caching, see the following Web site: <https://docs.microsoft.com/en-us/windows/win32/fileio/file-caching>  |

1. **Configuring Environment Variables**
2. Define and introduce the concept of environment variables, noting that they are used to tell the operating system where to find certain programs and program-related information.
3. Define and describe the two types of environment variable categories: system environment variables and user environment variables.
4. Refer to Figure 2-32 to illustrate how to configure environment variables.

**Configuring Startup and Recovery**

1. Introduce this topic by mentioning that Windows Server 2019 enables an administrator to configure parameters that dictate the startup sequence and how the system recovers from errors.
2. Review the many startup options that can be configured.
* Which operating system to boot by default, if more than one operating system is installed
* How long to display a list of operating systems from which to boot
* How long to display a list of recovery options if the computer needs to go into recovery mode after a system failure
1. Review the options that can be configured in the event of a system failure.
* Writing information to the system log (mandatory in Windows Server 2019)
* Whether to start automatically after a system failure
* How and where to write debugging information
1. Refer to Figure 2-33 to illustrate how to set the appropriate startup options in the Startup and Recovery window.

**Configuring Power Options**

1. Introduce this topic by emphasizing that after installing a new Windows Server 2019 system, an administrator should check the power options to make sure that they are set appropriately for the computer and the way an administrator is using the computer on the network.
2. List the power options that an administrator can set.
* Select a power plan.
* Choose what the power button does.
* Create a power plan.
* Choose when to turn off the display.
1. Refer to Figure 2-34 to illustrate how an administrator can configure power options within Control Panel by navigating to Hardware and selecting Power Options from the Category view.
2. Introduce and explain the three power plans that are already created after an installation.
* Balanced
* Power saver
* High performance
1. Explain how to create a custom power plan.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | If an administrator installs Windows Server 2019 directly on the hardware, he or she will also see a power option that allows him or her to choose what happens when the power button is pressed (shut down, hibernate, or do nothing)! |

1. **The Windows Registry**
2. Define and describe the Windows Registry as a database containing all information the operating system needs about the entire system, including hardware and software.
3. Emphasize that this information is vital for the Windows operating system. Explain what can happen if the Windows Registry becomes corrupted.
4. List some examples of data contained in the Registry.
* Information about all hardware components, including the CPU, disk drives, network interface cards, optical drives, and more
* Information about Windows Server 2019 services that are installed, which services they depend on, and the order in which they are started
* Data about user profiles
* Data on the previous settings used to boot the computer
* Configuration information about all software in use
* Software licensing information
* Server Manager and Control Panel parameter configurations
1. Explain that to modify the Registry, an administrator can use the regedit command within the Start menu, Run dialog box, Command Prompt, or PowerShell window to launch the Registry Editor.
2. Refer to Figure 2-35 to illustrate the results of launching the Windows Registry Editor.
3. Mention that making incorrect changes to the Registry can have profound consequences and provide examples.

**Windows Registry Contents**

1. Refer back to Figure 2-35 to illustrate the how the Windows Registry is hierarchical in structure and is made up of keys, subkeys, and entries.
2. Define and describe a key, subkey, and entry.
3. List and explain the purpose of each of the five root keys in the Windows Registry.
* HKEY\_LOCAL\_MACHINE
* HKEY\_CURRENT\_USER
* HKEY\_USERS
* HKEY\_CLASSES\_ROOT
* HKEY\_CURRENT\_CONFIG
1. Point out that a root key is also called a subtree. Explain that the root key is primary, or the highest-level category of data contained in the Windows Registry.
2. **Using Windows PowerShell**
3. Describe Windows PowerShell as a modern replacement for the MS-DOS shell that provides advanced system configuration and scripting features.
4. Introduce this section by listing the topics that will be presented.
* How to work with Windows PowerShell
* Key commands an administrator can use to perform system administration
* How to query system information using WMI
* The creation and usage of PowerShell scripts
1. **Working with Windows PowerShell**
2. Explain what happens initially when an administrator starts Windows PowerShell.
3. Point out that if a user logs into a Windows Server 2019 system as Administrator, he or she will receive the following prompt: PS C:\Users\Administrator>.
4. Remind students that Windows PowerShell can be used to execute most MS-DOS commands, as well as many UNIX/Linux commands.
5. Explain how to use MS-DOS shell features such as output redirection (>>) and command chaining (;).
6. Emphasize that most commands that an administrator will run within PowerShell will consist of cmdlets that have a specific format describing the function of the cmdlet as well as the object or item that it operates on.
7. Review some of the common cmdlets used in PowerShell.
8. Explain what is meant by piping and describe how it can be accomplished. Emphasize that administrators use the pipe symbol extensively within Windows PowerShell, as it is one of the most useful ways of sending information between cmdlets to build more complex commands or filter output to display only the output that they wish to see.
9. Review some of the many cmdlet options PowerShell supports.
* –whatif, –confirm, -verbose, -debug, -erroraction
1. Explain how PowerShell supports special navigation keys and key combinations.
* Up Arrow/Down Arrow
* Home/End
* Q/q
* Ctrl1 + LeftArrow/Ctrl1 + RightArrow
* Tab–Auto
1. Explain that Windows PowerShell has several different features that provide for additional functionality and allow an administrator to be more efficient.
* Customize the Windows PowerShell session
* Use aliases and functions to simplify commands
* Use PowerShell profile scripts to automatically execute commands
* Modify command output to make results easier to read
* Use PowerShell objects to view and manipulate items
* Use PowerShell providers to work with different areas on a system
1. Describe two ways to customize Windows PowerShell sessions.
* If an administrator starts PowerShell using the PowerShell command, the administrator can provide specific options to the PowerShell executable to control its execution.
* An administrator can create a customized Windows PowerShell session by creating a PowerShell console file that has a .psc1 extension. The administrator can then double-click this PowerShell console file to open Windows PowerShell.
1. Describe how to use aliases and functions to simplify commands.
* Explain that aliases are essentially shortcuts to commands. Describe their characteristics. Mention that cmdlets may have several aliases attached to them.
* Explain that if an administrator would like to execute multiple cmdlets, the administrator can use functions.
1. Describe how to use PowerShell profile scripts to automatically execute commands.
* Introduce this topic by explaining that both aliases and functions are essentially variables, and when an administrator exits Windows PowerShell, the aliases and functions that have been created are destroyed.
* Explain that an administrator can create a PowerShell profile script to make aliases and functions load into memory each time the administrator starts a PowerShell session.
* Mention that they are stored within the Documents folder under a user’s Windows profile directory in a file called Microsoft.
* Explain why an administrator needs to first enable script execution before making the script.
* Explain how an administrator can edit their new PowerShell profile using the command notepad $profile within PowerShell and add any aliases, functions, or other commands that should run each time he or she opens PowerShell.
1. Describe how to modify command output to make results easier to read.
* Review the many ways to modify command output to expand, modify, or reduce information produced from cmdlets.
	+ Pipe output
	+ Use -recurse option
	+ Use Sort-Object
	+ Use Group-Object
	+ Use ConvertTo-HTML
	+ Use Export-CSV
1. Describe how to use PowerShell objects to view and manipulate items.
* Explain that everything in Windows PowerShell can be treated as an object that has attributes (properties that describe the object) and methods (things that the object can do).
* Discuss how to view attributes and methods.
* Discuss how to create objects.
* Describe how the exit status is used.
1. Describe how to use PowerShell providers to work with different areas of a system.
* Describe what PowerShell providers are and what they do for PowerShell.
* Describe the default provider and explain how it is used.
* Describe the other providers and discuss how they are used.
* Mention that many cmdlets are used with providers.
* Refer to Table 2-1 for a list of common cmdlets used by many providers.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | For more information regarding PowerShell providers, please see the following Web site: <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_providers?view=powershell-7>  |

1. **System Administration Commands**
2. Introduce this topic by mentioning that there are some key PowerShell cmdlets and commands that administrators often use to manage Windows Server 2019.
3. List some examples of commands administrators commonly use after opening a PowerShell window.
* Restart-Computer
* Stop-Computer
1. Remind students that many post-installation tasks can be performed easily using Windows PowerShell. Provide examples if necessary.
2. Mention that Windows PowerShell can also be used to view, install, and remove Windows roles and features.
3. Explain how Windows PowerShell can be used to configure and troubleshoot the network.
4. Explain how an administrator can use Windows PowerShell to configure firewall settings.
5. Explain how an administrator can use Windows PowerShell to manage services and processes. Emphasize that managing running processes is important when solving performance issues.
6. Explain how an administrator can use Windows PowerShell to perform remote administration of computers within a domain environment.
7. Point out that some cmdlets allow an administrator to specify the computer name that he or she wishes to perform a particular task on.
8. Explain that if an administrator wishes to run multiple Windows PowerShell commands on a remote system, the administrator can instead obtain a PowerShell session on the remote computer.
9. Mention that if an administrator creates a PowerShell script, the administrator can also execute that script on several computers within their domain.
10. **Using WMI within Windows PowerShell**
11. Introduce this topic by mentioning that starting with Windows NT 4.0, Microsoft introduced an interface called Windows Management Instrumentation (WMI) that allowed programs and system software to query the hardware and software on the Windows computer.
12. Define WMI consumers as the programs and system software that can query WMI.
13. Define the WMI infrastructure as the collective components that are built into the operating system that respond to WMI queries.
14. Point out that Windows PowerShell is a WMI consumer that has built-in support for querying WMI.
15. Inform your students that before they can learn how to query WMI, they must first learn about the structure and nomenclature used to navigate the WMI infrastructure components.
16. Mention that the WMI infrastructure consists of three components that work together:
* WMI namespaces
* WMI providers
* WMI classes
1. Explain that WMI namespaces represent/organize different types of WMI data in much the same way that a file cabinet organizes files. Note that there are different namespaces for different purposes, and each namespace holds different WMI providers.
2. Explain the purpose of the CIMv2 namespace (Common Information Model version 2).
3. Explain that WMI providers are like the drawers in a file cabinet. Note that each provider contains different WMI classes that can be used to obtain/modify different hardware and software information on a system.
4. Explain that WMI classes are individual types of data and are like the files in each drawer of a filing cabinet. Note that it is these classes that an administrator can use within Windows PowerShell to work with different pieces of hardware and software.
5. Explain that the real power of WMI is within the WMI classes that administrators can query and manipulate. Emphasize that the best way to do this is to run the Get-WmiObject (alias gwmi) cmdlet with the –list option (most of them start with Win32\_).
6. Point out that an administrator can also use Get-WmiObject to query and manipulate WMI classes.
7. Explain that Windows PowerShell has default values for the –computer and –namespace options if an administrator does not supply them. Point out that by default, PowerShell assumes the local computer and the CIMv2 namespace.
8. Explain that instead of specifying the WMI class alongside the Get-WmiObject cmdlet, an administrator can instead create a WMI query statement and specify it using the –query option. WMI query statements use a SQL database language called WQl (WMI Query Language).
9. **Creating PowerShell Scripts**
10. Mention that PowerShell scripts are simply text files with a .ps1 extension that can be executed within Windows PowerShell.
11. Point out that the commands are executed from top-to-bottom.
12. Explain that PowerShell scripts can contain both Windows commands and PowerShell cmdlets, as well as complex control structures.
13. Note that Windows PowerShell provides the ability to reuse Windows PowerShell code in different situations.
14. List the topics covered in this section.
* Executing PowerShell scripts
* Using Windows PowerShell ISE
* Variables and constants
* Protecting PowerShell metacharacters
* Coloring and formatting output
* Decision constructs
* Loop constructs
* Creating your own PowerShell scripts
* Finding PowerShell scripts on the Internet
1. Discuss how to execute PowerShell scripts.
* Mention that script support in Windows PowerShell must be enabled.
* Explain how to review current execution policy.
	+ Restricted
	+ AllSigned
	+ RemoteSigned
	+ Unrestricted
	+ Bypass
	+ Undefined
* Explain how to execute a PowerShell script in PowerShell.
* Explain how to execute a PowerShell script outside Windows PowerShell.
1. Discuss how to use the Windows PowerShell ISE.
* Mention that the Windows PowerShell ISE tool is useful for creating, testing, and executing scripts.
* Explain that when an administrator opens the Start menu in Windows Server 2019, Windows PowerShell ISE is listed next to Windows PowerShell.
* Refer to Figure 2-36 to illustrate a new PowerShell script in Windows PowerShell ISE in the upper pane called Untitled.ps1.
* Explain how to modify and run a script in the Windows PowerShell ISE.
* Explain how to save the script with a more descriptive file name.
* Explain how to create and test scripts on remote computers that have winRM started within Windows PowerShell ISE.
1. Discuss the concepts of variables and constants.
* Mention that variables and constants store a value in memory for later use and are often used within PowerShell scripts.
* Define variable values as those that can be changed after being set.
* Define constant values as those that cannot be changed after being set.
* Describe characteristics of variable names within Windows PowerShell.
* Mention that if an administrator types the name of a variable at the PowerShell prompt, it will expand the variable and print its contents.
* Explain how to create variables and assign values within Windows PowerShell.
* Define an array variable as a variable that contains multiple values.
* Explain how an administrator can ensure that the value of the variable cannot be changed after creation.
* Emphasize that by default, all information stored within variables and constants in Windows PowerShell is text strings.
* Explain how to use a type cast to change this behavior.
* Explain how the Read-Host cmdlet is quite useful in obtaining information from the user within a PowerShell script.
1. Discuss the concept of protecting PowerShell metacharacters.
* Define a metacharacter as a keyboard character that has a special meaning and provide examples.
* Explain how to protect metacharacters from being interpreted by PowerShell by using special quote characters.
* List and describe three of these special quote characters.
	+ Double-quotes (") around text tells Windows PowerShell to treat all characters within (except variables) as a single unit.
	+ Single-quotes (') can be used instead of double-quotes to protect everything within from special interpretation by Windows PowerShell.
	+ The back-quote (`) protects only the next character from special interpretation by Windows PowerShell.
1. Discuss the concept of coloring and formatting output.
* Explain that it is useful to add different colors to different parts of the output to allow the output to be read easily.
* Mention that there are many cmdlets that allow an administrator to modify the output color when writing output. Provide examples of some of these cmdlets.
* Mention that a separator character can be used when printing the results of several variables to the Windows PowerShell window to ease readability. Provide examples utilizing separator characters.
* Explain how to use special separator characters (called echo escape characters).
1. Discuss the concept of decision constructs.
* Explain that decision constructs allow an administrator to modify the flow of their PowerShell script.
* Explain that for a decision construct to work, there must be a condition that returns true or false.
* Explain that it is easy to compare data and return a true or false result within Windows PowerShell.
* Refer to Table 2-2 for examples of comparison operators that return true or false.
* Refer to Table 2-3 for examples of logical operators that return true or false.
* Explain how to use the if and switch constructs to generate a true or false result.
1. Discuss the concept of loop constructs.
* Point out that loop constructs allow an administrator to perform a task several times.
* Explain how to use and code five loop constructs.
	+ foreach
	+ for
	+ while
	+ do...while
	+ do...until
1. Explain how a student can create his or her own PowerShell scripts.
* Explain the advantages of writing PowerShell scripts: they can save time, prevent typographical errors, and be reused again and again; and an administrator can easily edit the script to change the contents.
* Note that the $args special array variable can be used to accept an argument when running the script.
* Explain that the first step in creating any PowerShell script is to start small.
* Explain the importance of ensuring that a PowerShell script allows the cmdlets to operate on different objects (e.g., different processes, files, registry keys, etc.) and contains comment lines that describe the function of the script.
1. Explain how to find PowerShell scripts on the Internet
* Provide a list of some Web sites on the Internet that have plenty of reusable PowerShell scripts organized by function.
* Explain that the most effective way to find a PowerShell script that performs a specific function is to search using a search engine such as Google. Mention that to do this, an administrator must ensure that a search begins with the word PowerShell and is followed by any major and minor tasks, in that order.

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Demonstrate how to navigate to one of the common Web sites that contain reusable PowerShell code. Demonstrate how to search for a script, download it, and run it. |

|  |  |
| --- | --- |
| ***Teaching*** ***Tip*** | Demonstrate how to find a PowerShell script that performs a specific function by using a search engine such as Google. |

1.
2. **Quick Quiz 2**
3. Which Windows Server 2019 tool scans system files for integrity issues and replaces damaged or overwritten files with the proper version?
	1. Device Manager
	2. Server Manager
	3. System File Checker
	4. File Signature Verification tool (Sigverif)
4. Answer: c. System File Checker
5. Which term refers to disk storage used to expand the capacity of the physical memory installed in the computer?
	1. virtual memory
	2. RAID
	3. namespace
	4. filesystem provider
6. Answer: a. virtual memory
7.
8. Which term refers to a database containing all information the operating system needs about the entire system, including hardware and software?
	1. Certificate Store
	2. CIMv2 namespace
	3. WMI namespace
	4. Windows Registry
9. Answer: d. Windows Registry
10. Which character defines a PowerShell variable?
	1. #
	2. $
	3. %
	4. &
11. Answer: b. $

# **Class Discussion Topics**

1. What are the main differences between a PowerShell variable and a constant?
2. Compare the System File Checker tool to the File Signature Verification tool (Sigverif). What are the benefits of having both of these tools available within PowerShell?
3. Why is virtual memory so important to running servers and operating systems today? What happens if virtual memory is configured incorrectly?

# **Additional Projects**

1. Ask your students to read the article about Data Execution Prevention (DEP) at <https://support.microsoft.com/en-us/help/912923/how-to-determine-that-hardware-dep-is-available-and-configured-on-your> and write a short report summarizing its most important points.
2. Ask your students to read more about virtual memory and write a report explaining how it works. You can use the following link as a starting point: <http://computer.howstuffworks.com/virtual-memory.htm>.
3. Have your students review some basic PowerShell commands and write a summary of the commands they feel would be critical to learn as a system administrator. You can find a list of basic PowerShell commands at <https://blogs.technet.microsoft.com/heyscriptingguy/2015/06/11/table-of-basic-powershell-commands/>.

# **Additional Resources**

1. Run Best Practices Analyzer Scans and Manage Scan Results
2. <https://docs.microsoft.com/en-us/windows-server/administration/server-manager/run-best-practices-analyzer-scans-and-manage-scan-results>
3. Virtual Memory in OS: What is, Demand Paging, Advantages

<https://www.guru99.com/virtual-memory-in-operating-system.html>

1. 10 Fundamental Concepts for PowerShell Scripting
2. <https://www.techrepublic.com/blog/10-things/10-fundamental-concepts-for-powershell-scripting/>
3. Structure of the Registry

<https://docs.microsoft.com/en-us/windows/win32/sysinfo/structure-of-the-registry>

1. About WMI
2. <https://docs.microsoft.com/en-us/windows/win32/wmisdk/about-wmi>

**Key Terms**

* **alias** A shortcut to a command.
* **alias provider** A provider used to view and manage aliases.
* **array variable** A variable construct that contains multiple variables as opposed to a single variable.
* **attribute** A property that describes the object.
* **Best Practices Analyzer (BPA)** A scan that allows an administrator to scan associated servers and roles for configuration issues that do not follow Microsoft’s recommendations.
* **certificate provider** A provider used to view and manage encryption certificates issued to user accounts on the system as well as on the local computer.
* **Certificate Store** A storage location on a local computer that contains one or more certificates. A certificate store often has numerous certificates, possibly issued from a number of different certification authorities (CAs).
* **Certification Authority** An entity that issues digital certificates.
* **command chaining** An MS-DOS shell feature that allows an administrator to run multiple lines on one line in PowerShell.
* **comment** Lines of code in a script that are effectively ignored during execution. However, they provide a means for an administrator to document the purpose of their script or any key areas that are difficult to trace.
* **Component Object Model (COM)** A platform-independent, distributed, object-oriented system for creating binary software components that can interact.
* **constant** A name or an identifier for a simple value. A constant value cannot change during the execution of a script.
* **Data Execution Prevention (DEP)** A security feature that can help prevent damage to a computer from viruses and other security threats.
* **decision construct** A PowerShell feature that allows a programmer to modify the flow of a PowerShell script.
* **Device Manager utility** An extension of the Microsoft Management Console that provides a central and organized view of all the Microsoft Windows recognized hardware installed in a computer. Device Manager is used to manage the hardware devices installed in a computer, such as hard disk drives, keyboards, sound cards, USB devices, and more.
* **Devices and Printers utility** A utility that can be used to monitor and manage devices and printers installed within a Windows server or computer.
* **driver signing** A two-part process where Microsoft software first examines a new devices’ driver to verify that it is secure and then, second, assigns a unique Microsoft digital signature that is incorporated into the driver.
* **entry** A component of the Windows Registry hierarchical structure. The entry component is an item that appears in the details pane and is the lowest level in the Windows Registry. An entry consists of an entry name, its data type, and its value.
* **environment provider** A provider used to view and manage variables.
* **environment variables** Variables that are normally set within the Windows operating system.
* **escape character** The PowerShell escape character is the grave-accent(`). When used at the end of a line, it is a continuation character—so the command will continue on the next line. Additionally, the escape character can be used to indicate that the next character following it should be passed without substitution. Finally, when used inside double quotation marks, the escape character indicates that the following character should be interpreted as a 'special' character.
* **exit status** A value generated by PowerShell upon a script’s completion to indicate the script’s execution status.
* **file caching** A performance option that can be configured within Windows Server 2019.
* **File Signature Verification tool (Sigverif)** A Windows Server 2019 tool that verifies system and critical files, including device drivers, to determine if they have a signature.
* **filesystem provider** The default provider. It is responsible for generating PS C:\Users\Administrator> as the prompt when a programmer starts PowerShell. A programmer can use the filesystem provider to view and manage the files on a filesystem.
* **flushing** A performance option that can be configured within Windows Server 2019.
* **function** A grouping of code that has an optional input and output. It is a way of collecting up a bunch of code to perform one or many different times by just pointing to it instead of duplicating that code repeatedly.
* **function provider** A provider that is used to view and manage functions within Windows PowerShell.
* **gateway server mode** A state where Windows Admin Center on a Windows Server 2016 or 2019 system is able to provide remote Web access for administrations.
* **Input/Output (I/O) address** An address that is used to communicate between devices and software.
* **Interrupt Request (IRQ) line** A channel for communication with the processor.
* **key** A component of the Windows Registry hierarchical structure. The key component is a folder that appears in the left pane of the Registry Editor and can contain subkeys and entries, for example, HKEY\_CURRENT\_USER.
* **loop construct** A programming construct that allows a programmer to perform a task several times. There are many different types of loop constructs available, including foreach, for, while, do...while, and do...until.
* **metacharacter** A Windows PowerShell feature that is frequently used within PowerShell scripts. A metacharacter is a special keyboard character that has special meaning within Windows PowerShell.
* **method** A property that describes things that an object can do.
* **object** A component of PowerShell that has attributes (properties that describe the object) and methods (things that the object can do).
* **output redirection** An MS-DOS shell feature that is implemented as an operator defined as a rightward pointing angular bracket (>). The operator is used in shells to redirect standard output to a file, where it is written and saved, or to a device (such as a printer, where it is printed).
* **page** Blocks of information used in a virtual memory system.
* **paging file** An area of memory allocated by the operating system that is moved from virtual memory back into physical memory when the operating system calls for the block of code.
* **piping** A technique for passing information from one program process to another.
* **Plug and Play (PnP)** A technique that allows an operating system to work with hardware devices to automatically detect and configure recently installed hardware to work with the operating system.
* **power plan** A combination of power options that define the amount of power used by the computer and the operating system. Options include how soon to turn off the display, whether to require a password on wakeup, how soon to turn off the storage devices, sleep/hibernate settings, USB settings, PCI card settings, and processor settings.
* **PowerShell console file** An XML file that has an extension .psc1. A PowerShell console file stores configuration settings for a particular console that leads to a customized Windows PowerShell session.
* **PowerShell profile script** A script that runs when a PowerShell console is started. The script is used to configure PowerShell. There are four profile scripts documented in the PowerShell documentation pack.
* **PowerShell provider** A PowerShell plug-in that provides functionality within Windows PowerShell (e.g., aliases, functions, variables) or allows PowerShell to interact with other parts of the system (e.g., registry, filesystem, environment variables, certificates). PowerShell providers provide useful ways of interacting with a system.
* **processor scheduling** A performance option that can be configured within Windows Server 2019. Processor scheduling allows an administrator to configure how processor resources are allocated to programs.
* **Registry Editor** A Windows Server 2019 tool that allows an administrator to search and edit the Windows Registry.
* **registry provider** A provider used to view and modify the Windows Registry keys HKEY\_LOCAL\_MACHINE and HKEY\_CURRENT\_USER.
* **Remote Desktop Protocol (RDP)** A proprietary protocol developed by Microsoft that provides a user with a graphical interface to connect to another computer over a network connection. The user employs RDP client software for this purpose, while the other computer must run RDP server software.
* **Remote Server Administration Tools (RSAT)** An optional tool that can be downloaded and installed on a Windows 10 PC. RSAT allows an administrator to perform server administration remotely using Server Manager and a wide range of MMC tools from a Windows 10 PC within an IT office.
* **self-signed certificate** An encryption certificate for use with HTTPS.
* **subkey** A component of the Windows Registry hierarchical structure. The subkey component is a part of the Windows Registry that is below a key. A subkey can contain entries or other subkeys.
* **system environment** **variables** A type of environment variable. System environment variables are defined by the operating system and apply to any user logged into the computer. Administrators can add new system environment variables or change the values of existing ones.
* **System File Checker** A tool that can be used to scan system files for integrity and replace them with the correct versions.
* **tracing** The process of reading and understanding PowerShell scripts on the Internet.
* **type cast** A process of converting one object type to another. By default, all information is stored within variables and constants in Windows PowerShell as text strings, but this behavior can be changed if an administrator prefixes the variable with a type cast.
* **Universal PnP (UPnP)** An open standard that is used in all types of systems and that enables connectivity through networks and network protocols. UPnP supports server-based networking, wireless networking, peer-to-peer networking, and other networking services.
* **user environment variables** A type of environment variable. User environment variables can be defined on a per-user basis and may be used to provide a wide variety of different information, such as specifying the path where application files are stored.
* **variable** A data item that may take on more than one value during the runtime of a program.
* **variable provider** A provider used to view and manage variables.
* **Virtual Desktop Infrastructure (VDI)** A tool that allows client computers to remotely connect to a central server to obtain their Windows desktop.
* **virtual memory** A performance option that can be configured within Windows Server 2019. It consists of disk storage used to expand the capacity of the physical memory installed in the computer.
* **Windows PowerShell Integrated Scripting Environment (ISE)** The ISE is a handy tool that allows an administrator to write scripts on the fly, and has a convenient look-up for all the PowerShell commands.
* **Windows Registry** A database containing all the information the operating system needs about the entire system, including hardware and software.
* **winRM** A component required for remote management with Windows PowerShell.
* **WMI class** Individual types of data. Classes can be used within Windows PowerShell to work with different pieces of hardware and software.
* **WMI consumers** The programs and system software that can query WMI.
* **WMI infrastructure** The collective components built into the operating system that respond to WMI queries.
* **WMI namespace** A component that represents/organizes different types of WMI data in much the same way that a file cabinet organizes files. There are different namespaces for different purposes, and each namespace holds different WMI providers.
* **WMI provider** A component that contains different WMI classes that can be used to obtain/modify different hardware and software information on a system.
* **WQL (WMI Query Language)** An SQL database language used by WMI query statements.