MS-AZ040T00: AUTOMATING ADMINISTRATION WITH POWERSHELL

DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
5 Days	Advanced	Azure	Instructor-led	NA

INTRODUCTION

This course provides students with the fundamental knowledge and skills to use PowerShell for administering and automating administration of Windows servers. This course provides students the skills to identify and build the command they require to perform a specific task. In addition, students learn how to build scripts to accomplish advanced tasks such as automating repetitive tasks and generating reports. This course provides prerequisite skills supporting a broad range of Microsoft products, including Windows Server, Windows Client, Microsoft Azure, and Microsoft 365. In keeping with that goal, this course will not focus on any one of those products, although Windows Server, which is the common platform for all those products, will serve as the example for the techniques this course teaches.

AUDIENCE PROFILE

This course is intended for IT Professionals who are already experienced in general Windows Server, Windows client, Azure, and Microsoft 365 administration, and who want to learn more about using Windows PowerShell for administration. This course is also suitable for IT Professionals already experienced in server administration, including Microsoft Exchange Server, Microsoft SharePoint Server, and Microsoft SQL Server.

PREREQUISITES

Before attending this course, delegates must have:

- Experience with Windows networking technologies and implementation.
- Experience with Windows Server administration, maintenance, and troubleshooting.

COURSE OBJECTIVES

After completing this course, students will be able to:

- Describe the functionality of Windows PowerShell and use it to run and find basic commands.
- Identify and run cmdlets for local system administration.
- Work with the Windows PowerShell pipeline.

COURSE CONTENT

Module 1: Review Windows PowerShell

This module introduces you to Windows PowerShell and its versions. It discusses commonly used host applications. It explains how to use Microsoft Visual Studio Code (VS Code) to develop PowerShell scripts

Lessons

- Introduction
- Learn about Windows PowerShell
- Get familiar with Windows PowerShell applications
- Identify factors to install and use Windows PowerShell
- Configure the Windows PowerShell console
- Configure the Windows PowerShell Integrated Scripting Environment (ISE)
- Use Visual Studio Code with PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe Windows PowerShell and its major versions.
- Identify the common Windows PowerShell hosting applications.
- Describe points to consider when using PowerShell.
- Explain how to configure the Windows PowerShell console host.
- Explain how to configure the Windows PowerShell ISE host.
- Describe how to use VS Code for PowerShell scripting.

Module 2: Understand the command syntax in Windows PowerShell

This module covers the cmdlet structure and parameters for using Windows PowerShell cmdlets. It also explains how to use tab completion and how to display About files content.

Lessons

- Introduction
- Discover the structure of PowerShell cmdlets
- Discover the parameters for using PowerShell cmdlets

- Review the tab completion feature in PowerShell
- Display the About files content in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe cmdlet structure.
- Identify how to use Windows PowerShell parameters.
- Explain how to use tab completion.
- Explain how to display the About files content.
- Use About files.

Module 3: Find commands and Get-Help in Windows PowerShell

This module explains how to find Windows PowerShell cmdlets that you can use to perform specific tasks. It also covers how to use Get-Help to retrieve detailed information about a cmdlet and its parameters.

Lessons

- Introduction
- Define modules in PowerShell
- Find cmdlets in PowerShell
- Use command aliases in PowerShell
- Use Show-Command and Get-Help in PowerShell
- Interpret the help file contents and update the local help content in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the relationship between modules and cmdlets.
- Identify options for finding cmdlets.
- Describe aliases.
- Use aliases.
- Explain how to use Show-Command.
- Explain how to use Get-Help.
- Review help.
- Explain how to interpret the help file contents.
- Explain how to update the local help content.

Module 4: Manage Active Directory Domain Services using PowerShell cmdlets

This module covers the cmdlets that are used to administer Active Directory Domain Services.

Lessons

- Introduction
- Manage user accounts in PowerShell
- Manage groups and group memberships in PowerShell
- Manage computer accounts in PowerShell
- Manage organizational units and Active Directory objects in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Identify user management cmdlets.
- List group management cmdlets.
- Manage users and groups.
- Describe the cmdlets for managing computer objects.
- Describe the cmdlets for managing organizational units (OUs).
- Describe the cmdlets for managing Active Directory objects.
- Manage Active Directory objects.

Module 5: Manage network service settings for Windows devices using PowerShell cmdlets

This module covers the PowerShell modules and cmdlets that are used to configure network settings for Windows devices.

Lessons

- Introduction
- Manage IP addresses in PowerShell
- Manage IP routing in PowerShell
- Manage DNS clients in PowerShell
- Manage Windows Firewall settings in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Identify cmdlets for managing TCP/IP settings.
- Describe how to manage local routing table settings.
- Describe how to modify DNS client configuration.
- List cmdlets for managing Windows Firewall.
- Configure network settings.

Module 6: Manage Windows Server settings using PowerShell cmdlets

This module covers cmdlets that are used to configure settings related to Group Policy, Server Manager, Hyper-V, and Internet Information Services (IIS).

Lessons

- Introduction
- Automate management tasks using the Group Policy management cmdlets
- Manage server roles and services using PowerShell cmdlets
- Manage Hyper-V Virtual Machines using PowerShell cmdlets
- Manage Internet Information Services using PowerShell cmdlets
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the cmdlets for managing Group Policy Objects (GPOs).
- Describe the cmdlets for managing server features, roles, and services.
- Describe the cmdlets for managing Hyper-V and virtual machines (VMs).
- Describe the cmdlets for managing and administering Internet Information Services (IIS).

Module 7: Manage settings for a local Windows machine using PowerShell cmdlets

This module covers common PowerShell cmdlets that can be used to perform tasks on a local Windows 10 computer.

Lessons

- Introduction
- Manage Windows 10 using PowerShell
- Manage permissions with PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the cmdlets for managing Windows 10 devices.
- Describe the cmdlets for managing local permissions with Windows PowerShell.

Module 8: Understand the Windows PowerShell pipeline

This module explains about the Windows PowerShell pipeline and some basic techniques for running multiple commands in it.

Lessons

- Introduction
- Review Windows PowerShell pipeline and its output
- Discover object members in PowerShel
- Control the formatting of pipeline output

- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the features and functionalities of the pipeline.
- Use the appropriate terminology to describe the pipeline output and pipeline objects.
- Explain how to discover and display object members.
- Review object members.
- Describe the cmdlets used to format the pipeline output for display.
- Format pipeline output.

Module 9: Select, sort, and measure objects using the pipeline

This module explains how to manipulate objects in the pipeline by using commands that sort, select, and measure objects.

Lessons

- Introduction
- Sort and group objects by property in the pipeline
- Measure objects in the pipeline
- Select a set of objects in the pipeline
- Select object properties in the pipeline
- Create and format calculated properties in the pipeline
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain how to sort objects by a specified property.
- Sort objects by using the Sort-Object command.
- Explain how to measure objects' numeric properties.
- Measure objects by using the Measure-Object command.
- Explain how to display a subset of objects in a collection.
- Explain how to display a customized list of objects' properties.
- Select objects by using the Select-Object command.
- Explain how to create calculated properties.
- Create custom-calculated properties for display.

Module 10: Filter objects out of the pipeline

This module explains how to filter objects out of the pipeline by using the Where-Object cmdlet to specify various criteria.

Lessons

- Introduction
- Learn about the comparison operators in PowerShell
- Review basic filter syntax in the pipeline
- Review advanced filter syntax in the pipeline
- Optimize the filter performance in the pipeline
- Knowledge check
- Summary

After completing this module, students will be able to:

- List the major PowerShell comparison operators.
- Explain how to filter objects by using basic syntax.
- Explain how to filter objects by using advanced syntax.
- Filter objects.
- Explain how to optimize filtering performance in the pipeline.

Module 11: Enumerate objects in the pipeline

This module explains how to enumerate objects in the pipeline so that you can work with one object at a time during automation.

Lessons

- Introduction
- Learn about enumerations in the pipeline
- Review basic syntax to enumerate objects in the pipeline
- Review advanced syntax to enumerate objects in the pipeline
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain the purpose of enumeration.
- Explain how to enumerate objects by using basic syntax.
- Perform basic enumerations.
- Explain how to enumerate objects by using advanced syntax.
- Perform advanced enumeration.

Module 12: Send and pass data as output from the pipeline

This module explains how to send pipeline data to files and in various output formats.

Lessons

- Introduction
- Write pipeline data to a file
- Convert pipeline objects to other forms of data representation in PowerShell
- Control additional output options in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain how to write pipeline data to a file.
- Explain how to convert pipeline data to the comma-separated values (CSV) format.
- Explain how to convert pipeline data to the XML format.
- Explain how to convert pipeline data to the JavaScript Object Notation (JSON) format.
- Explain how to convert pipeline data to the HTML format.
- Export data.
- Explain how to send pipeline data to other locations.
- Describe how PowerShell matches incoming pipeline data to the parameters of a cmdlet.

Module 13: Pass pipeline objects

This module explains how the Windows PowerShell command-line interface passes objects from one command to another in the pipeline.

Lessons

- Introduction
- Pipeline parameter binding
- Identify ByValue parameters
- Pass data by using ByValue
- Pass data by using ByPropertyName
- Identify ByPropertyName parameters
- Use manual parameters to override the pipeline
- Use parenthetical commands
- Expand property values
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe pipeline parameter binding.
- Identify ByValue parameters.
- Pass data by using ByValue.
- Identify ByPropertyName parameters.
- Pass data ByPropertyName.
- Pass pipeline data ByPropertyName.
- Use manual parameters to override the pipeline.
- Use parenthetical commands.
- Expand property values.

Module 14: Connect with data stores using PowerShell providers

This module covers PowerShell providers that connect Windows PowerShell to data stores. They offer an easier-to-understand and consistent interface for working with data stores.

Lessons

- Introduction
- Define Windows PowerShell providers
- Review the built-in providers in PowerShell
- Access provider help in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Use PowerShell to manage users, groups, and licenses in Azure AD
- Manage Exchange Online with PowerShell.
- Manage SharePoint Online with PowerShell.
- Manage Microsoft Teams with PowerShell.

Module 15: Use PowerShell drives in PowerShell

This module explains how to work with PowerShell drives.

Lessons

- Introduction
- Explain PowerShell drives in PowerShell
- Use PowerShell drive cmdlets in PowerShell
- Manage the file system in PowerShell
- Manage the registry in PowerShell
- Work with certificates in PowerShell
- Work with other PowerShell drives in PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain the purpose and use of PowerShell drives.
- Identify the cmdlets for using PowerShell drives.
- Explain how to find, delete, and create files and directories.
- Explain how to use Windows PowerShell to manage the file system.
- Explain how to work with the registry.
- Explain how to use Windows PowerShell to manage the registry.
- Explain how to work with certificates.
- Explain how to work with other PowerShell drives.

Module 16: Review CIM and WMI

This module covers CIM and WMI technologies to connect to a common information repository that contains management information that you can query and manipulate.

Lessons

- Introduction
- Review architecture of CIM and WMI
- Review repositories in CIM and WMI
- Locate online class documentation by using CIM and WMI cmdlets
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the architecture of CIM and WMI.
- Explain the purpose of the repository.
- Explain how to locate online documentation for repository classes.
- Locate online class documentation.

Module 17: Query configuration information by using CIM and WMI

This module explains the structure of the namespaces that contain classes and also how to query instances of a class. It covers how to query remote computers by using ad-hoc connections and CIM sessions.

Lessons

- Introduction
- List local repository namespaces and classes by using CIM and WMI
- Query instances by using commands and WMI Query Language
- Connect to remote computers by using CIM and WMI cmdlets
- Query repository classes from remote computers by using CIMSession objects
- Knowledge check
- Summary

After completing this module, students will be able to:

- List the available namespaces.
- List local repository namespaces.
- Retrieve a list of classes from a namespace.
- Retrieve a list of classes from the root\CIMv2 namespace and sort them.
- Query instances of a specified class.
- Query instances of a specified class by using WMI, CIM, and WMI Query Language (WQL).
- Connect to remote computers by using CIM or WMI.
- Create and manage CIM sessions.
- Query repository classes from remote computers by using CIM sessions objects.

Module 18: Query and manipulate repository objects by using CIM and WMI methods

This module explains how to use CIM and WMI to make changes by using methods. Discovering and understanding these methods is an important step in querying and manipulating the repository information.

Lessons

- Introduction
- Discover methods of repository objects by using CIM and WMI
- Locate class methods and documentation by using CIM and WMI
- Invoke methods of repository objects by using CIM and WMI
- Knowledge check
- Summary

After completing this module, students will be able to:

- Discover the methods of repository objects.
- Locate online documentation for methods.
- Locate the methods of the Win32_Service class and their documentation.
- Explain how to invoke methods of repository objects.
- Use methods for the Win32_OperatingSystem and Win32_Process classes.

Module 19: Manage variables in Windows PowerShell scripts

This module explains how to create variables, name them correctly, and assign the correct data type, while ensuring the data you store in variables is in the correct format and easily accessible.

Lessons

- Introduction
- Define variables in Windows PowerShell scripts
- Create variable names in Windows PowerShell scripts
- Assign values and types to variables in Windows PowerShell scripts
- Identify the methods and properties of variables in Windows PowerShell scripts

- Use string variables and methods in Windows PowerShell scripts
- Use date variables and methods in Windows PowerShell scripts
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain the purpose of variables.
- Describe the naming rules for using variables.
- Explain how to assign a value to a variable.
- Describe variable types.
- Explain how to assign a variable type.

Module 20: Work with arrays and hash tables in Windows PowerShell scripts

This module explains how to use arrays and hash tables in Windows PowerShell scripts.

Lessons

- Introduction
- Define an array in Windows PowerShell scripts
- Work with array lists in Windows PowerShell scripts
- Define hash tables in Windows PowerShell Scripts
- Work with hash tables in Windows PowerShell scripts
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain the purpose of an array.
- Work with arrays and their contents.
- Work with array lists and their contents.
- Manipulate arrays and array lists.
- Explain the purpose of a hash table.
- Work with hash tables and their contents.
- Manipulate hash tables.

Module 21: Create and run scripts by using Windows PowerShell

This module explains how to create and run scripts

Lessons

- Introduction
- Review Windows PowerShell scripts
- Modify scripts in the PowerShell Gallery
- Create scripts using Windows PowerShell
- Review the PowerShellGet module in Windows PowerShell
- Run scripts and set the execution policy in Windows PowerShell
- Review Windows PowerShell and AppLocker
- Sign the scripts digitally in Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe Windows PowerShell scripts.
- Explain how to find scripts and modify them.
- Describe how to create scripts.
- Describe the PowerShellGet module.
- Explain how to run Windows PowerShell scripts.
- Describe the script execution policy.
- Set the script execution policy.
- Explain how to use AppLocker to help secure Windows PowerShell scripts.
- Explain how to digitally sign scripts.
- Digitally sign a Windows PowerShell script.

Module 22: Work with scripting constructs in Windows PowerShell

This module explains how to use scripting constructs for iteration and decision making in Windows PowerShell.

Lessons

- Introduction
- Review and use the ForEach loop in Windows PowerShell scripts
- Review and use the If construct in Windows PowerShell scripts
- Review and use the Switch construct in Windows PowerShell scripts
- Review the For construct in Windows PowerShell scripts
- Review other loop constructs in Windows PowerShell scripts
- Review Break and Continue in Windows PowerShell scripts
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the syntax of the ForEach construct.
- Use the ForEach construct.
- Describe the syntax of the If construct.
- Use the **If **construct.
- Describe the syntax of the Switch construct.
- Use the Switch construct.
- Describe how to use the for construct.
- Describe the other loop constructs.
- Explain how to use Break and Continue.

Module 23: Import data in different formats for use in scripts by using Windows PowerShell cmdlets

This module explains how to import data from a text file, CSV file, XML file, and JavaScript Object Notation (JSON) file.

Lessons

- Introduction
- Use the Get-Content command in Windows PowerShell scripts
- Use the Import-Csv cmdlet in Windows PowerShell scripts
- Use the Import-Clixml cmdlet in Windows PowerShell scripts
- Use the ConvertFrom-Json cmdlet in Windows PowerShell scripts
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe how to use Get-Content to review file data.
- Describe how to use Import-Csv to retrieve data.
- Describe how to use Import-Clixml to import XML data.
- Describe how to use ConvertFrom-Json to work with JSON data.
- Import data from text, CSV, and XML files.

Module 24: Use methods to accept user inputs in Windows PowerShell scripts

This method explains multiple methods for accepting user input in a Windows PowerShell script.

Lessons

- Introduction
- Identify values that might change in Windows PowerShell scripts
- Use the Read-Host cmdlet in Windows PowerShell scripts
- Use the Get-Credential cmdlet in Windows PowerShell scripts
- Use the Out-GridView cmdlet in Windows PowerShell scripts
- Pass parameters to a Windows PowerShell script
- Knowledge check
- Summary

After completing this module, students will be able to:

- Identify values in a script that are likely to change.
- Explain how to use Read-Host to accept user input.
- Explain how to use Get-Credential to accept user credentials.
- Explain how to use Out-GridView to obtain user input.
- Obtain user input by using Read-Host, Get-Credential, and Out-GridView.
- Explain how to pass parameters to a script.
- Obtain user input by using parameters.

Module 25: Troubleshoot scripts and handle errors in Windows PowerShell

This module covers troubleshooting PowerShell scripts and understanding error messages.

Lessons

- Introduction
- Interpret error messages generated for Windows PowerShell commands
- Add output to Windows PowerShell scripts
- Use breakpoints in Windows PowerShell scripts
- Interpret error actions for Windows PowerShell commands
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe how error messages are stored.
- Explain how to add more troubleshooting information to scripts.
- Describe how to configure breakpoints for troubleshooting.
- Explain how to troubleshoot a script.
- Describe error actions in Windows PowerShell.

Module 26: **Use functions and modules in Windows PowerShell scripts**

This module covers how to create modules and functions in Windows PowerShell scripts.

Lessons

- Introduction
- Review functions in Windows PowerShell scripts
- Use variable scope in Windows PowerShell scripts
- Create modules in Windows PowerShell scripts
- Use the dot sourcing feature in Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe functions.
- Describe the implications of variable scope.
- Explain how to use dot sourcing.
- Create a function in a script.
- Explain how to create a module.
- Create a module.

Module 27: Manage single and multiple computers by using Windows PowerShell remoting

This module explains how to use remoting to perform administration on remote computers.

Lessons

- Introduction
- Review the remoting feature of Windows PowerShell
- Compare remoting with remote connectivity
- Review the remoting security feature of Windows PowerShell
- Enable remoting by using Windows PowerShell
- Use one-to-one remoting by using Windows PowerShell
- Use one-to-many remoting by using Windows PowerShell
- Compare remoting output with local output
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the Windows PowerShell remoting architecture.
- Explain the difference between Windows PowerShell remoting and other forms of remote administration.
- Describe Windows PowerShell remoting security and privacy features.
- Enable remoting on a computer.
- Use Windows PowerShell remoting for single-computer management.
- Use Windows PowerShell remoting for multiple-computer management.
- Use Windows PowerShell remoting.
- Explain the difference between local output and remoting output.

Module 28: Use advanced Windows PowerShell remoting techniques

This module covers some useful advanced techniques that help overcome the limitations of basic Windows PowerShell remoting.

Lessons

- Introduction
- Review common remoting techniques of Windows PowerShell
- Send parameters to remote computers in Windows PowerShell
- Set access protection to variables, aliases, and functions by using the scope modifier
- Enable multi-hop remoting in Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Configure common remoting options.
- Send parameters and local variables to remote computers.
- Describe the use of Windows PowerShell scopes.
- Send local variables to a remote computer.
- Configure multi-hop remoting authentication.

Module 29: Manage persistent connections to remote computers by using Windows PowerShell sessions

This module explains how to establish and manage persistent connections to remote computers, known as Windows PowerShell sessions or Sessions.

Lessons

- Introduction
- Review persistent connections in Windows PowerShell
- Create and manage persistent PSSessions by using Windows PowerShell
- Disconnect PSSessions by using Windows PowerShell
- Review the feature of implicit remoting in Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain the purpose of persistent connections.
- Create and use a PSSession.
- Transmit commands by using a PSSession.
- Explain how to disconnect from PSSessions.
- Disconnect and reconnect to PSSessions.
- Explain the concept of implicit remoting.

Module 30: Review Azure PowerShell module

This module explains the Azure PowerShell environment and the Az module for Windows PowerShell. It also explains how to manage Microsoft Entra ID by using PowerShell modules.

Lessons

- Introduction
- Review Azure PowerShell
- Review the benefits of the Azure PowerShell module
- Install the Azure PowerShell module
- Migrate Azure PowerShell from AzureRM to Azure
- Review Microsoft Azure Active Directory module for Windows PowerShell and Azure Active Directory PowerShell for Graph modules
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe Azure PowerShell.
- Describe the Azure Az PowerShell module.
- Install the Azure Az PowerShell module.
- Migrate Azure PowerShell from AzureRM to Az.
- Describe the Azure Active Directory module for Windows PowerShell and Azure Active Directory PowerShell for Graph modules.

Module 31: Review the features and tools for Azure Cloud Shell

This module covers Azure Cloud Shell and its features.

Lessons

- Introduction
- Review the characteristics of Azure Cloud Shell
- Review the features and tools of Azure Cloud Shell
- Configure and experiment with Azure Cloud Shell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe Azure Cloud Shell.
- Describe features and tools for Azure Cloud Shell.
- Use Azure Cloud Shell.

Module 32: Manage Azure resources with Windows PowerShell

This module explains how to install necessary modules for cloud services management. It also explains how to use PowerShell commands to perform administrative tasks on cloud resources like Azure virtual machines, subscriptions, and storage accounts.

Lessons

- Introduction
- Create a new Azure virtual machine by using Windows PowerShell commands
- Manage Azure virtual machines by using Windows PowerShell commands
- Manage Azure related storage by using Azure PowerShell
- Manage Azure subscriptions by using Azure PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe, install, and use the Azure PowerShell environment.
- Describe and use Azure Cloud Shell.
- Manage Azure VMs with Azure PowerShell.
- Manage Azure storage accounts and subscriptions with Azure PowerShell.

Module 33: Manage users, groups, and licenses in Microsoft Entra ID by using Windows PowerShell

This module covers how to give access to the services in Microsoft 365, so that you can create user accounts and then assign licenses that provide access to the services.

Lessons

- Introduction
- Review benefits to manage Microsoft 365 services by using Windows PowerShell
- Connect to the Microsoft 365 tenant by using Windows PowerShell
- Manage users in Microsoft 365 by using Windows PowerShell
- Manage groups in Microsoft 365 by using Windows PowerShell
- Manage roles in Microsoft 365 by using Windows PowerShell
- Manage licenses in Microsoft 365 by using Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the benefits of using PowerShell for Microsoft 365.
- Explain how to connect to a Microsoft 365 tenant with PowerShell.
- Explain how to create and manage users in Microsoft 365 with PowerShell.
- Explain how to create and manage groups in Microsoft 365 with PowerShell.
- Explain how to manage roles in Microsoft 365 with PowerShell.
- Explain how to manage licenses in Microsoft 365 with PowerShell.

Module 34: Manage Exchange Online by using Windows PowerShell

This module covers managing mailboxes, resources, and admin roles in Exchange Online with PowerShell.

Lessons

- Introduction
- Connect to Exchange Online by using Windows PowerShell
- Manage mailboxes in Exchange Online by using Windows PowerShell
- Manage resources in Exchange Online by using Windows PowerShell
- Manage admin roles in Exchange Online by using Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain how to connect to Exchange Online by using PowerShell.
- Explain how to manage mailboxes in Exchange Online.
- Explain how to manage resources in Exchange Online.
- Explain how to manage admin roles in Exchange Online.

Module 35: Manage SharePoint Online by using Windows PowerShell

This module describes SharePoint Online, which is a collaboration service that allows to store and share information through a web-based interface. The module explains how to manage SharePoint Online users and groups with PowerShell.

Lessons

- Introduction
- Install and connect to SharePoint Online Management Shell by using Windows PowerShell
- Manage SharePoint Online users and groups by using Windows PowerShell
- Manage SharePoint sites by using Windows PowerShell
- Manage SharePoint Online external user sharing by using Windows PowerShell
- Knowledge check

- Summary

After completing this module, students will be able to:

- Explain how to connect to SharePoint Online by using PowerShell.
- Explain how to manage SharePoint Online users and groups with PowerShell.
- Explain how to manage sites with PowerShell.
- Explain how to manage external user sharing with PowerShell.

Module 36: Manage Microsoft Teams by using Windows PowerShell

This module explains how to use Microsoft Teams PowerShell module to perform tasks such as creating teams and managing user permissions.

Lessons

- Introduction
- Review Microsoft Teams PowerShell module
- Install the Microsoft Teams PowerShell module
- Manage teams with Microsoft Teams PowerShell module
- Knowledge check
- Summary

After completing this module, students will be able to:

- Describe the Microsoft Teams PowerShell module.
- Explain how to connect to Microsoft Teams by using PowerShell.
- Manage Microsoft Teams with the Microsoft Teams PowerShell module.

Module 37: Create and manage background jobs using Windows PowerShell

This module explains the three types of jobs: local jobs, Windows PowerShell remote jobs, and Common Information Model (CIM)/Windows Management Instrumentation (WMI) jobs. These job types form the basis of the Windows PowerShell job system.

Lessons

- Introduction
- Define the types of background jobs in Windows PowerShell
- Start remote jobs and CIM/WMI jobs in Windows PowerShell
- Monitor jobs in Windows PowerShell
- Retrieve results for running jobs in Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain the purpose and functionality of background jobs.
- Start jobs.
- Manage jobs.
- Retrieve job results.
- Use background jobs.

Module 38: Create and manage scheduled jobs using Windows PowerShell

This module covers how to use scheduled jobs. In Windows PowerShell, scheduled jobs are essentially scheduled tasks. They follow the same rules for actions, triggers, and other features, and run Windows PowerShell scripts by design.

Lessons

- Introduction
- Create and run Windows PowerShell scripts as scheduled tasks
- Define scheduled jobs in Windows PowerShell
- Create job option and job trigger objects in Windows PowerShell
- Create and register a scheduled job in Windows PowerShell
- Retrieve the results from a scheduled job in Windows PowerShell
- Knowledge check
- Summary

After completing this module, students will be able to:

- Explain how to run Windows PowerShell scripts as scheduled tasks.
- Create and run a Windows PowerShell script as a scheduled task.
- Explain the purpose and use of scheduled jobs.
- Create job options and triggers.
- Create scheduled jobs.
- Retrieve scheduled job results.
- Use scheduled jobs.

ASSOCIATED CERTIFICATIONS & EXAM

There is no associated Certification or Exam for this course.