Advanced Automated Administration with Windows PowerShell - MOC On Demand (MS-10962)

Modality: Self-Paced Learning
Duration: 2 Days
SATV Value: 2
SUBSCRIPTION: Master, Master Plus

About this course:

This course is Microsoft Official On-Demand course accessible for 90 days from the date of course request, if you have annual subscription, or course purchase individually. Course access will expire after 90 days of course enrollment.

This course that will teach students how to automate administrative tasks using Windows PowerShell 5.1. Students will learn core scripting skills such as creating advanced functions, writing controller scripts, and handling script errors. Students will learn how to works with Windows PowerShell Workflow, the REST API and XML and JSON formatted data files, Students will also learn how to use new administration tools such Desired State Configuration (DSC) and Just Enough Administration (JEA) to configure and secure servers.

The average salary for a IT Administrator is $59,350 per year.

Course Objective:

After completing this course, students will be able to:

- Create advanced functions.
- Use Microsoft .NET Framework and REST API in Windows PowerShell.
- Write controller scripts.
- Handle script errors.
- Use XML, JSON, and custom formatted data.
- Enhance server management with Desired State Configuration and Just Enough Administration.
- Analyze and debug scripts
- Understand Windows PowerShell workflow.

Audience:

This course is intended for IT professionals who are experienced in general Windows Server and Windows Client administration. Students should have a foundational knowledge of Windows PowerShell. In addition, this course provides scripting guidance for developers who support development environments and deployment processes.

Prerequisites:
In addition to their professional experience, students who attend this training should already have the following technical knowledge:

- Experience with Windows networking technologies and implementation.
- Experience with Windows Server administration, maintenance, and troubleshooting.
- Experience with Windows Client administration, maintenance, and troubleshooting.
- Experience with using Windows PowerShell to run commands and to create basic non-modularized scripts.

Suggested prerequisites courses:

- [Automating Administration with Windows PowerShell - MOC On Demand (MS-10961)](https://www.quickstart.com/)

Course Outline:

**Module 1: Creating Advanced Functions**
In this module students will learn how to parameterize a command into an advanced function. It is designed to teach several key principles in a single logical sequence, by using frequent hands-on exercises to reinforce new skills.

**Lessons**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

**Lab: Converting a Command into an Advanced Function**

- Converting a Command into an Advanced Function

**Lab: Creating a Script Module**

- Creating a Script Module

**Lab: Defining Parameter Attributes and Input Validation**

- Defining Parameter Attributes and Input Validation

**Lab: Writing Functions that use Multiple Objects**

- Writing Functions that use Multiple Objects

**Lab: Writing Functions that Accept Pipeline Input**
After completing this module, students will be able to:

- Parameterize a command and create an advanced function.
- Convert a script and function into a script module.
- Define parameter attributes and input validation for a function.
- Enumerate objects by using scripting constructs.
- Modify a function to accept pipeline input.
- Produce complex pipeline output in a function.
- Document a function by using comment-based Help.
- Create functions that support ?WhatIf and ?Confirm.

Module 2: Using Cmdlets and Microsoft .NET Framework in Windows PowerShell

Windows PowerShell provides commands that accomplish many of the tasks that you will need in a production environment. Sometimes, a command is not available but the .NET Framework provides an alternate means of accomplishing a task. Because Windows PowerShell is built on the .NET Framework, it is able to access those alternate means. In this module, you will learn how to discover and run Windows PowerShell commands, and how to use .NET Framework components from inside Windows PowerShell. These two techniques will provide you with the most flexibility and capability for accomplishing tasks in a production environment.

Lessons

- Running Windows PowerShell Commands
- Using Microsoft .NET Framework in Windows PowerShell

Lab : Using .NET Framework in Windows PowerShell

- Using Static and Instance Members

After completing this module, students will be able to:

- Discover Windows PowerShell commands by using the Help system.
- Describe and use .NET Framework classes and instances in Windows PowerShell.

Module 3: Writing Controller Scripts
In this module, students will learn how to combine tools – advanced functions that perform a specific task – and a controller script that provides a user interface or automates a business process.

**Lessons**

- Understanding Controller Scripts
- Writing Controller Scripts that Show a User Interface
- Writing Controller Scripts That Produce Reports

**Lab : Writing Controller Scripts that Display a User Interface**

- Write Functions to be Used in the Controller Script
- Write a Controller Script that Implements a Text-Based Menu

**Lab : Writing Controller Scripts That Produce HTML Reports**

- Writing Functions To Be Used in the Controller Script
- Writing a Controller Script That Produces HTML Reports

After completing this module, students will be able to:

- Describe the difference between tools and controller scripts.
- Write controller scripts that present a user interface.
- Write controller scripts that automate a business process.

**Module 4: Handling Script Errors**

In this module, students will learn how to perform basic error handling in scripts. The focus will be about how to add error handling to existing tools, primarily as a time-saving mechanism (instead of having students write new tools). A side benefit of this approach is that it will help build the skills that you must have to analyze and reuse existing code written by someone else.

**Lessons**

- Understanding Error Handling
- Handling Errors in a Script

**Lab : Handling Errors in a Script**

- Handling Errors in a Script

After completing this module, students will be able to:

- Describe the shell’s default error response mechanisms.
- Add error handling code to existing tools.

**Module 5: Using XML Data Files**

In this module, students will learn how to read, manipulate, and write data in XML files. XML files provide a robust, yet straightforward way to store both flat and hierarchical data. XML files are more flexible than CSV, more accessible for small amounts of data than SQL Server, and easier to code
against that Excel automation.

Lessons

- Reading, Manipulating and Writing Data in XML

Lab : Reading, Manipulating and Writing Data in XML

- Testing the Provided Tools
- Updating an XML Inventory Document

After completing this module, students will be able to:

- Read, manipulate, and write data in XML.

Module 6: Managing Server Configurations by Using Desired State Configuration
In this module, students will learn how to write Desired State Configuration (DSC) configuration files, deploy those files to servers, and monitor servers’ configurations.

Lessons

- Understanding Desired State Configuration
- Creating and Deploying a DSC Configuration

Lab : Creating and Deploying a DSC Configuration

- Writing, Running and Pushing a DSC Configuration

After completing this module, students will be able to:

- Describe the architecture and deployment models of DSC.
- Write and deploy DSC configuration files.

Module 7: Analyzing and Debugging Scripts
In this module, students will learn how to use native Windows PowerShell features to analyze and debug existing scripts. These skills are also useful when students have to debug their own scripts.

Lessons

- Debugging in Windows PowerShell
- Analyzing and Debugging and Existing Script

Lab : Analyzing and Debugging and Existing Script

- Analyzing and Debugging and Existing Script

After completing this module, students will be able to:

- Describe the debugging features of Windows PowerShell.
- Analyze and debug an existing script.
Module 8: Understanding Windows PowerShell Workflow

In this module, students will learn about the features of the Windows PowerShell Workflow technology.

Lessons

- Understanding Windows PowerShell Workflow

After completing this module, students will be able to:

- Describe the Workflow feature of Windows PowerShell.