

Microsoft - Advanced Automated Administration With Windows PowerShell

Code:	10962
Length:	3 days
URL:	View Online

This course is intended for IT Professionals who are already experienced in general Windows Server and Windows Client administration. Students should have a foundational knowledge of Windows PowerShell, which they can obtain by taking course 10961C: Automating Administration with Windows PowerShell.

Skills Gained

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.

Who Can Benefit

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, which they can obtain by taking course 10961C: Automating Administration with 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.

Students who attend this training can meet the prerequisites by obtaining equivalent knowledge and skills through practical experience as a Windows System Administrator. Course 10961: Automating Administration with Windows PowerShell is strongly recommended to obtain prerequisite skills in Windows PowerShell.

Course Details

Outline

Module 1: Creating advanced functions

This module explains how to parameterize a command into an advanced function.

Lessons

- Converting a command into an advanced function
- Creating a script module
- Defining parameter attributes and input validation
- Writing functions that accept pipeline input
- Producing complex pipeline output
- Documenting functions by using comment-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 accept pipeline input

- Writing functions that accept pipeline input

Lab : Producing complex pipeline output

- Producing complex pipeline output

Lab : Documenting functions by using comment-based help

- Documenting functions by using comment-based help

Lab : Supporting WhatIf and -Confirm

- Supporting WhatIf and -Confirm

After completing this module, students will be able to:

- Convert a command into an advanced function.
- Create a script module.
- Define parameter attributes and input validation.
- Write functions that accept pipeline input.
- Produce complex pipeline output.
- Document functions by using comment-based help.
- Support -WhatIf and -Confirm.

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

This module explains how to use Microsoft .NET Framework and REST API to supplement functionality that Windows PowerShell commands provide.

Lessons

- Using Microsoft .NET Framework in Windows PowerShell
- Using REST API in Windows PowerShell

Lab : Using Microsoft .NET Framework in Windows PowerShell

- Using Microsoft .NET Framework in Windows PowerShell

Lab : Using REST API in Windows PowerShell

- Using REST API in Windows PowerShell

After completing this module, students will be able to:

- Use Microsoft .NET Framework in Windows PowerShell.
- Use REST API in Windows PowerShell.

Module 3: Writing controller scripts

This module explains how to combine advanced functions that perform a specific task and a controller script that provides a user interface or automates a business process.

Lessons

- Introducing controller scripts
- Writing controller scripts that display a user interface
- Writing controller scripts that produce reports

Lab : Writing controller scripts

- Writing controller scripts that display a user interface
- Writing controller scripts that implement 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 controller scripts.
- Write controller scripts that display a user interface.
- Write controller scripts that produce reports.

Module 4: Handling script errors

This module explains how to perform basic error handling in scripts.

Lessons

- Understanding error handling
- Handling errors and timeouts in a script

Lab : Handling errors in a script

- Handling errors in a script

After completing this module, students will be able to:

- Describe error handling.
- Handle errors and timeouts in a script.

Module 5: Using XML, JSON, and custom-formatted data

This module explains how to read, manipulate, and write data in XML and JSON format.

Lessons

- Reading, manipulating, and writing XML-formatted data
- Reading, manipulating, and writing JSON-formatted data
- Reading and manipulating custom-formatted data

Lab : Reading, manipulating, and writing XML-formatted data

- Testing the provided tools
- Updating an XML inventory document

After completing this module, students will be able to:

- Read, manipulate, and write XML-formatted data.
- Read, manipulate, and write JSON-formatted data.
- Read and manipulate custom-formatted data.

Module 6: Enhancing server management with Desired State Configuration and Just Enough Administration

This module explains how to write Desired State Configuration (DSC) configuration files, deploy those files to servers, and monitor servers configurations. This module also explains how to restrict administrative access with Just Enough Administration (JEA).

Lessons

- Understanding Desired State Configuration
- Creating and deploying a DSC configuration
- Implementing Just Enough Administration

Lab : Creating and deploying a DSC configuration

- Creating and deploying a DSC configuration

Lab : Configuring and using JEA

- Configuring and using JEA

After completing this module, students will be able to:

- Understand DSC.
- Create and deploy a DSC configuration.
- Implement JEA.

Module 7: Analyzing and debugging scripts

This module explains how to use native Windows PowerShell features to analyze and debug existing scripts.

Lessons

- Debugging in Windows PowerShell
- Analyzing and debugging an existing script

Lab : Analyzing and debugging an existing script

- Analyzing and debugging an existing script

After completing this module, students will be able to:

- Debug in Windows PowerShell.
- Analyze and debug an existing script.

Module 8: Understanding Windows PowerShell Workflow

This module explains the features of the Windows PowerShell Workflow technology.

Lessons

- Understanding Windows PowerShell Workflow

Lab : Creating and running a Windows PowerShell Workflow

- Creating and running a Windows PowerShell Workflow

After completing this module, students will be able to describe and implement the Workflow feature of Windows PowerShell.

Download Whitepaper: Transforming Software Development in the Enterprise: Agile, DevOps and Kubernetes

Get Your Free Copy Now

ExitCertified® Corporation and iMVP® are registered trademarks of ExitCertified ULC and ExitCertified Corporation and Tech Data Corporation, respectively
Copyright ©2021 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.
All Rights Reserved.

Generated 3