Course 10962B:

Advanced Automated Administration with Windows PowerShell

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

Lab : Creating a Script Module

Lab : Defining Parameter Attributes and Input Validation

Lab : Writing Functions that use Multiple Objects

Lab : Writing Functions that Accept Pipeline Input

Lab : Producing Complex Function Output

Lab : Documenting Functions by using Content-Based Help

Lab : Supporting -Whatif and -Confirm

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

- 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

Lab : Writing Controller Scripts That Produce 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

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

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

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 an Existing Script

Lab : Analyzing and Debugging an 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.