Course Outline

MOC 550309 B Windows PowerShell Scripting and Toolmaking

Course Summary

Description

This five-day instructor-led is intended for IT professionals who are interested in furthering their skills in Windows PowerShell and administrative automation. The course assumes a basic working knowledge of PowerShell as an interactive command-line shell, and teaches students the correct patterns and practices for building reusable, tightly scoped units of automation.

Objectives

At the end of this course, students will be able to:

- Describe the correct patterns for building modularized tools in Windows PowerShell
- Build highly modularized functions that comply with native PowerShell patterns
- Build controller scripts that expose user interfaces and automate business processes
- Manage data in a variety of formats
- Write automated tests for tools
- Debug tools

Topics

- Tool Design
- Start with a Command
- Build a Basic Function and Module
- Adding CmdletBinding and Parameterizing
- Emitting Objects as Output
- An Interlude: Changing Your Approach
- Using Verbose, Warning, and Informational Output
- Comment-Based Help
- Handling Errors
- Basic Debugging
- Going Deeper with Parameters
- Writing Full Help
- Unit Testing Your Code
- Extending Output Types
- Analyzing Your Script
- Publishing Your Tools
- Basic Controllers: Automation Scripts and Menus
- Proxy Functions
- Working with XML Data
- Working with JSON Data
- Working with SQL Server Data
- Final Exam

Audience

This course is intended for administrators in a Microsoft-centric environment who want to build reusable units of automation, automate business processes, and enable less-technical colleagues to accomplish administrative tasks.

Prerequisites

- Experience at basic Windows administration
- Experience using Windows PowerShell to query and modify system information
- Experience using Windows PowerShell to discover commands and their usage
- Experience using WMI and/or CIM to query system information

Duration

Five days

Course Outline

MOC 550309 B Windows PowerShell Scripting and Toolmaking

Course Outline

I. Tool Design

This module explains how to design tools and units of automation that comply with native PowerShell usage patterns.

A. Tools do one thing
B. Tools are flexible
C. Tools look native
D. Lab 1: Designing a Tool
E. Design a tool

II. Start with a Command

This module explains how to start the scripting process by beginning in the interactive shell console.

A. Why start with a command?
B. Discovery and experimentation
C. Lab 1: Designing a Tool
D. Start with a command

III. Build a Basic Function and Module

This module explains how to build a basic function and module, using commands already experimented with in the shell.

A. Start with a basic function
B. Create a script module
C. Check prerequisites
D. Run the new command
E. Lab 1: Designing a Tool
F. Build a basic function and module

IV. Adding CmdletBinding and Parameterizing

This module explains how to extend the functionality of a tool, parameterize input values, and use CmdletBinding.

A. About CmdletBinding and common parameters
B. Accepting pipeline input
C. Mandatory-ness
D. Parameter validation
E. Parmeter aliases
F. Lab 1: Designing a Tool
G. Adding CmdletBinding and Parameterizing

V. Emitting Objects as Output

This module explains how to create tools that produce custom objects as output.

A. Assembling information
B. Constructing and emitting output
C. Quick tests
D. Lab 1: Designing a Tool
E. Emitting objects as output

VI. An Interlude: Changing Your Approach

This module explains how to re-think tool design, using concrete examples of how it's often done wrong.

A. Examining a script
B. Critiquing a script
C. Revising the script
D. Lab 1: No lab

VII. Using Verbose, Warning, and Informational Output

This module explains how to use additional output pipelines for better script behaviors.

A. Knowing the six channels
B. Adding verbose and warning output
C. Doing more with verbose output
D. Informational output
E. Lab 1: Designing a Tool
F. Using Verbose, Warning, and Informational Output

VIII. Comment-Based Help

This module explains how to add comment-based help to tools.

A. Where to put your help
B. Getting started
C. Going further with comment-based help
D. Broken help
E. Lab 1: Designing a Tool
F. Comment-based help

IX. Handling Errors

This module explains how to create tools that deal with anticipated errors.

A. Understanding errors and exceptions
B. Bad handling
C. Two reasons for exception handling
D. Handling exceptions in our tool
E. Capturing the actual exception
F. Handling exceptions for non-commands
G. Going further with exception handling
H. Deprecated exception handling
I. Lab 1: Designing a Tool
J. Handling errors

Course Summary (cont'd)

X. Basic Debugging
This module explains how to use native PowerShell script debugging tools.

A. Two kinds of bugs
B. The ultimate goal of debugging
C. Developing assumptions
D. Write-Debug
E. Set-PSBreakpoint
F. The PowerShell ISE
G. Lab 1: Designing a Tool
H. Basic debugging

XI. Going Deeper with Parameters
This module explains how to further define parameter attributes in a PowerShell command.

A. Parameter positions
B. Validation
C. Multiple parameter sets
D. Value from remaining arguments
E. Help messages
F. Aliases
G. More CmdletBinding
H. Lab 1: No Lab
I. Click here to enter text.

XII. Writing Full Help
This module explains how to create external help for a command.

A. External help
B. Using PlatyPs
C. Supporting online help
D. "About" topics
E. Making your help updatable
F. Lab 1: Designing a Tool
G. Writing full help

XIII. Unit Testing Your Code
This module explains how to use Pester to perform basic unit testing.

A. Sketching out the test
B. Making something to test
C. Expanding the test
D. Going further with Pester
E. Lab 1: Designing a Tool
F. Unit testing your code

XIV. Extending Output Types
This module explains how to extend objects with additional capabilities.

A. Understanding types
B. The Extensible Type System
C. Extending an object
D. Using Update-TypeData
E. Lab 1: No Lab
F. Click here to enter text.

XV. Analyzing Your Script
This module explains how to use Script Analyzer to support best practices and prevent common problems.

A. Performing a basic analysis
B. Analyzing the analysis
C. Lab 1: Designing a Tool
D. Analyzing your script

XVI. Publishing Your Tools
This module explains how to publish tools to public and private repositories.

A. Begin with a manifest
B. Publishing to PowerShell Gallery
C. Publishing to private repositories
D. Lab 1: Designing a Tool
E. Publishing your tools

XVII. Basic Controllers: Automation Scripts and Menus
This module explains how to create controller scripts that put tools to use.

A. Building a menu
B. Using UIChoice
C. Writing a process controller
D. Lab 1: Designing a Tool
E. Basic controllers

XVIII. Proxy Functions
This module explains how to create and use proxy functions.

A. A proxy example
B. Creating the proxy base
C. Modifying the proxy
D. Adding or removing parameters
E. Lab 1: Designing a Tool
F. Proxy functions

Course Outline

MOC 550309 B Windows PowerShell Scripting and Toolmaking

Course Summary

XIX. Working with XML Data

This module explains how to work with XML data in PowerShell.

A. Simple: CliXML
B. Importing native XML
C. ConvertTo-XML
D. Creating native XML from scratch
E. Lab 1: Designing a Tool
F. Working with XML

XX. Working with JSON Data

This module explains how to using JSON data in PowerShell.

A. Converting to JSON
B. Converting from JSON
C. Lab 1: Designing a Tool
D. Working with JSON data

XXI. Working with SQL Server Data

This module explains how to use SQL Server from within a PowerShell script.

A. SQL Server terminology and facts
B. Connecting to the server and database
C. Writing a query
D. Running a query
E. Invoke-SqlCmd
F. Thinking about tool design patterns
G. Lab 1: No Lab
H. Click here to enter text.

XXII. Final Exam

This module provides a chance for students to use everything they have learned in this course within a practical example.

A. Lab problem
B. Break down the problem
C. Do the design
D. Test the commands
E. Code the tool
F. Lab 1: Final Exam
G. Lab one
H. Lab 2: Final Exam
I. Lab two