training code: MS 55039 / ENG DL 5d / EN

Windows PowerShell Scripting and Toolmaking

Training recipients

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.

Benefits

After completing 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

Training program

1. Module 1: Tool Design
    - Tools do one thing
    - Tools are flexible
    - Tools look native
2. Module 2: Start with a Command
    - Why start with a command?
    - Discovery and experimentation
3. Module 3: Build a Basic Function and Module
    - Start with a basic function

   - Create a script module
   - Check prerequisites
   - Run the new command

4. Module 4: Adding CmdletBinding and Parameterizing
   - About CmdletBinding and common parameters
   - Accepting pipeline input
   - Mandatory-ness
   - Parameter validation
   - Parmeter aliases
5. Module 5: Emitting Objects as Output
   - Assembling information
   - Constructing and emitting output
   - Quick tests
6. Module 6: An Interlude: Changing Your Approach
   - Examining a script
   - Critiquing a script
   - Revising the script
7. Module 7: Using Verbose, Warning, and Informational Output
   - Knowing the six channels
   - Adding verbose and warning output
   - Doing more with verbose output
   - Informational output
8. Module 8: Comment-Based Help
   - Where to put your help
   - Getting started
   - Going further with comment-based help
   - Broken help
9. Module 9: Handling Errors
   - Understanding errors and exceptions
   - Bad handling
   - Two reasons for exception handling
   - Handling exceptions in our tool
   - Capturing the actual exception
   - Handling exceptions for non-commands
   - Going further with exception handling
   - Deprecated exception handling
10. Module 10: Basic Debugging
    - Two kinds of bugs
    - The ultimate goal of debugging
    - Developing assumptions
    - Write-Debug

   - Set-PSBreakpoint
   - The PowerShell ISE
11. Module 11: Going Deeper with Parameters
   - Parameter positions
   - Validation
   - Multiple parameter sets
   - Value from remaining arguments
   - Help messages
   - Aliases
   - More CmdletBinding
12. Module 12: Writing Full Help
   - External help
   - Using PlatyPs
   - Supporting online help
   - “About” topics
   - Making your help updatable
13. Module 13: Unit Testing Your Code
   - Sketching out the test
   - Making something to test
   - Expanding the test
   - Going further with Pester
14. Module 14: Extending Output Types
   - Understanding types
   - The Extensible Type System
   - Extending an object
   - Using Update-TypeData
15. Module 15: Analyzing Your Script
   - Performing a basic analysis
   - Analyzing the analysis
16. Module 16: Publishing Your Tools
   - Begin with a manifest
   - Publishing to PowerShell Gallery
   - Publishing to private repositories
17. Module 17: Basic Controllers: Automation Scripts and Menus
   - Building a menu
   - Using UIChoice
   - Writing a process controller
18. Module 18: Proxy Functions
   - A proxy example
   - Creating the proxy base
   - Modifying the proxy

- Adding or removing parameters

19. Module 19: Working with XML Data
    - Simple: CliXML
    - Importing native XML
    - ConvertTo-XML
    - Creating native XML from scratch
20. Module 20: Working with JSON Data
    - Converting to JSON
    - Converting from JSON
21. Module 21: Working with SQL Server Data
    - SQL Server terminology and facts
    - Connecting to the server and database
    - Writing a query
    - Running a query
    - Invoke-SqlCmd
    - Thinking about tool design patterns

Expected preparation of the participant

Before attending this course, students must have:

- 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

5 days / 35 hours

Language

- Training: English
- Materials: English