800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

Windows PowerShell Scripting and Toolmaking Training

Applied Technology Academy

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.

LEVEL	DURATION	EXPERIENCE	AVERAGE SALARY	LABS
Beginner	**5 Days**	**1 year: Windows PowerShell**	**$75,500**	**Yes**

Course Overview

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

Course Outline

- **Module 1: Tool Design**
  - This module explains how to design tools and units of automation that comply with native PowerShell usage patterns.
  - Lessons
  - Tools do one thing
  - Tools are flexible
  - Tools look native
  - Lab: Designing a Tool
  - Design a tool
- **Module 2: Start with a Command**
  - This module explains how to start the scripting process by beginning in the interactive shell console.
  - Lessons
  - Why start with a command?

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - Discovery and experimentation
  - Lab: Starting with a Command
  - Start with a command
- **Module 3: 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.
  - Lessons
  - Start with a basic function
  - Create a script module
  - Check prerequisites
  - Run the new command
  - Lab: Building a Basic Function and Module
  - Build a basic function and module
- **Module 4: Adding CmdletBinding and Parameterizing**
  - This module explains how to extend the functionality of a tool, parameterize input values, and use CmdletBinding.
  - Lessons
  - About CmdletBinding and common parameters
  - Accepting pipeline input
  - Mandatory-ness
  - Parameter validation
  - Parameter aliases
  - Lab: Adding CmdletBinding and Parameterizing
  - Adding CmdletBinding and parameterizing
- **Module 5: Emitting Objects as Output**
  - This module explains how to create tools that produce custom objects as output.
  - Lessons
  - Assembling information
  - Constructing and emitting output
  - Quick tests
  - Lab: Emitting Objects as Output
  - Emit objects as output
- **Module 6: An Interlude: Changing Your Approach**
  - This module explains how to re-think tool design, using concrete examples of how it's often done wrong.
  - Lessons
  - Examining a script

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - Critiquing a script
  - Revising the script

- **Module 7: Using Verbose, Warning, and Informational Output**
  - This module explains how to use additional output pipelines for better script behaviors.
  - Lessons
  - Knowing the six channels
  - Adding verbose and warning output
  - Doing more with verbose output
  - Informational output
  - Lab: Using Verbose, Warning, and Informational Output
  - Using verbose, warning, and informational output
- **Module 8: Comment-Based Help**
  - This module explains how to add comment-based help to tools.
  - Lessons
  - Where to put your help
  - Getting started
  - Going further with comment-based help
  - Broken help
  - Lab: Adding Comment-Based Help
  - Add comment-based help
- **Module 9: Handling Errors**
  - This module explains how to create tools that deal with anticipated errors.
  - Lessons
  - 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
  - Lab: Handling Errors
  - Handle errors
- **Module 10: Basic Debugging**
  - This module explains how to use native PowerShell script debugging tools.
  - Lessons
  - Two kinds of bugs

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - The ultimate goal of debugging
  - Developing assumptions
  - Write-Debug
  - Set-PSBreakpoint
  - The PowerShell ISE
  - Lab: Basic Debugging
  - Perform basic debugging
- **Module 11: Going Deeper with Parameters**
  - This module explains how to further define parameter attributes in a PowerShell command.
  - Lessons
  - Parameter positions
  - Validation
  - Multiple parameter sets
  - Value from remaining arguments
  - Help messages
  - Aliases
  - More CmdletBinding
- **Module 12: Writing Full Help**
  - This module explains how to create external help for a command.
  - Lessons
  - External help
  - Using PlatyPs
  - Supporting online help
  - “About” topics
  - Making your help updatable
  - Lab: Writing Full Help
  - Write full help for a tool
- **Module 13: Unit Testing Your Code**
  - This module explains how to use Pester to perform basic unit testing.
  - Lessons
  - Sketching out the test
  - Making something to test
  - Expanding the test
  - Going further with Pester
  - Lab: Unit Testing Your Code
  - Unit test your code
- **Module 14: Extending Output Types**

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - This module explains how to extend objects with additional capabilities.
  - Lessons
  - Understanding types
  - The Extensible Type System
  - Extending an object
  - Using Update-TypeData
- **Module 15: Analyzing Your Script**
  - This module explains how to use Script Analyzer to support best practices and prevent common problems.
  - Lessons
  - Performing a basic analysis
  - Analyzing the analysis
  - Lab: Analyzing Your Script
  - Analyze your script
- **Module 16: Publishing Your Tools**
  - This module explains how to publish tools to public and private repositories.
  - Lessons
  - Begin with a manifest
  - Publishing to PowerShell Gallery
  - Publishing to private repositories
  - Lab: Publishing Your Tools
  - Publish your tools
- **Module 17: Basic Controllers: Automation Scripts and Menus**
  - This module explains how to create controller scripts that put tools to use.
  - Lessons
  - Building a menu
  - Using UIChoice
  - Writing a process controller
  - Lab: Basic Controllers
  - Create basic controllers
- **Module 18: Proxy Functions**
  - This module explains how to create and use proxy functions.
  - Lessons
  - A proxy example
  - Creating the proxy base
  - Modifying the proxy
  - Adding or removing parameters

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - Lab: Proxy Functions
  - Create and modify proxy functions
- **Module 19: Working with XML Data**
  - This module explains how to work with XML data in PowerShell.
  - Lessons
  - Simple: CliXML
  - Importing native XML
  - ConvertTo-XML
  - Creating native XML from scratch
  - Lab: Working with XML Data
  - Work with XML data
- **Module 20: Working with JSON Data**
  - This module explains how to use JSON data in PowerShell.
  - Lessons
  - Converting to JSON
  - Converting from JSON
  - Lab: Working with JSON Data
  - Work with JSON data
- **Module 21: Working with SQL Server Data**
  - This module explains how to use SQL Server from within a PowerShell script.
  - Lessons
  - SQL Server terminology and facts
  - Connecting to the server and database
  - Writing a query
  - Running a query
  - Invoke-SqlCmd
  - Thinking about tool design patterns

Intended Audience

Administrators that have little or no programming experience but who have a working knowledge of Windows PowerShell and who are able to use Windows PowerShell to run complex, interactive commands. Students of this course may administer a wide variety of server and client products and technologies that offer Windows PowerShell integration, including Microsoft Exchange Server, Microsoft Windows Active Directory Domain Services, Microsoft SharePoint Server, and more.

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

Follow-On Courses

- Microsoft Automating Administration with PowerShell (AZ-040)
- Microsoft Automating Administration with Windows PowerShell (M10961)
- Configuring Windows Server Hybrid Advanced Services (AZ-801)

Ready to enroll? Call 800.674.3550 or email Info@AppliedTechAc.com — private team cohorts, GSA MAS purchasing, military credentialing funding, VA GI Bill and VR&E and student financing available.