

**Windows PowerShell Scripting and Toolmaking (MCC-55039)**

**Course Number:** MCC-55039
**Duration:** 5 days

**Overview**

This Microsoft Community Course 55039, Windows PowerShell Scripting and Toolmaking training, teaches students the correct patterns and practices for building reusable, tightly scoped units of automation.

**Prerequisites**

All students must have experience in:

* Basic Windows administration
* Windows PowerShell to query and modify system information
* Windows PowerShell to discover commands and their usage
* WMI and/or CIM to query system information

**Materials**

All Microsoft training students receive Microsoft official courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

**Objectives**

* 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
* Use the debugging tools

**Outline**

* introduction
* Tool Design
	+ Tools do one thing
	+ Tools are flexible
	+ Tools look native
* Start with a Command
	+ Why start with a command?
	+ Discovery and experimentation
* Build a Basic Function and Module
	+ Start with a basic function
	+ Create a script module
	+ Check prerequisites
	+ Run the new command
* Adding CmdletBinding and Parameterizing
	+ About CmdletBinding and common parameters
	+ Accepting pipeline input
	+ Mandatory-ness
	+ Parameter validation
	+ Parmeter aliases
* Emitting Objects as Output
	+ Assembling information
	+ Constructing and emitting output
	+ Quick tests
* An Interlude: Changing Your Approach
	+ Examining a script
	+ Critiquing a script
	+ Revising the script
* Using Verbose, Warning, and Informational Output
	+ Knowing the six channels
	+ Adding verbose and warning output
	+ Doing more with verbose output
	+ Informational output
* Comment-Based Help
	+ Where to put your help
	+ Getting started
	+ Going further with comment-based help
	+ Broken help
* 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
* Basic Debugging
	+ Two kinds of bugs
	+ The ultimate goal of debugging
	+ Developing assumptions
	+ Write-Debug
	+ Set-PSBreakpoint
	+ The PowerShell ISE
* Going Deeper with Parameters
	+ Parameter positions
	+ Validation
	+ Multiple parameter sets
	+ Value from remaining arguments
	+ Help messages
	+ Aliases
	+ More CmdletBinding
* Writing Full Help
	+ External help
	+ Using PlatyPs
	+ Supporting online help
	+ “About” topics
	+ Making your help updatable
* Unit Testing Your Code
	+ Sketching out the test
	+ Making something to test
	+ Expanding the test
	+ Going further with Pester
* Extending Output Types
	+ Understanding types
	+ The Extensible Type System
	+ Extending an object
	+ Using Update-TypeData
* Analyzing Your Script
	+ Performing a basic analysis
	+ Analyzing the analysis
* Publishing Your Tools
	+ Begin with a manifest
	+ Publishing to PowerShell Gallery
	+ Publishing to private repositories
* Basic Controllers: Automation Scripts and Menus
	+ Using UIChoice
	+ Writing a process controller
* Proxy Functions
	+ A proxy example
	+ Creating the proxy base
	+ Modifying the proxy
	+ Adding or removing parameters
* Working with XML Data
	+ Simple: CliXML
	+ Importing native XML
	+ ConvertTo-XML
	+ Creating native XML from scratch
* Working with JSON Data
	+ Converting to JSON
	+ Converting from JSON
* 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
* Final Exam
	+ Lab problem
	+ Break down the problem
	+ Do the design
	+ Test the commands
	+ Code the tool
* Conclusion