Advanced Automated Administration with Windows PowerShell

Length: 3 Days

About this course: Learn how to automate and streamline day to day management and administration tasks and functions in your Windows Server Infrastructure.

This three-day course is built on Windows Server 2012 R2 and Windows 8.1 and while it is specifically focused on Windows PowerShell v3.0, is also relevant in v2.0 and v3.0 Windows PowerShell environments.

Expand and build upon the knowledge already acquired and focus on building more scalable and usable Windows PowerShell scripts for use in your organization by building your own Windows PowerShell tools. Learn about areas such as the creation of advanced functions, script modules, advanced parameters attributes and controller scripts. Also learn how to make your scripts more robust by learning about handling script errors and the analysis and debugging Windows PowerShell scripts. The course will also cover the use of Windows PowerShell cmdlets with .NET Framework as well as teaching how to configure your Windows Servers using Desired State Configuration and providing an understanding of Windows PowerShell workflow.

The detailed hands on labs and in depth content and learning will help remove manual tasks that you may currently have to perform as an Administrator, allowing you to make your own Windows PowerShell tools for automated, repeated, accurate management and provisioning of your Windows Server infrastructure.

Audience profile: This course is intended for IT Professionals already experienced in general Windows Server and Windows Client administration or already experienced in administering and supporting Application servers and services including applications such as Exchange, SharePoint, and SQL. System, Infrastructure and Application Administrators working in a Windows or Windows hybrid environment will all find this course relevant to their day to day jobs and future career and skills development.

The course is also intended for IT Professionals who want to build upon existing Windows PowerShell knowledge and skill to learn how to build their own tools for broader general use in their organization, using any Microsoft or independent software vendor (ISV) product that supports Windows PowerShell manageability.

At course completion: After completing this course, students will be able to:

- Create Advanced Functions
- Use Cmdlets and Microsoft .NET Framework in Windows PowerShell
- Write Controller Scripts
- Handle Script Errors
- Use XML Data Files
- Manage Server Configurations by Using Desired State Configuration
- Analyze and Debugging Scripts
- Understand Windows PowerShell Workflow

Prerequisites: Before attending this course, students must have:

- Knowledge and experience working with Windows PowerShell or knowledge equivalent to the content covered in course Automating Administration with Windows PowerShell
- Experienced in general Windows Server and Windows Client administration or already have experience in administering and supporting Application servers and services including applications like Exchange, SharePoint, and SQL

COURSE CONTENT

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 FunctionLab : Creating a Script ModuleLab : Defining Parameter Attributes and Input ValidationLab : Writing Functions that use Multiple ObjectsLab : Writing Functions that Accept Pipeline InputLab : Producing Complex Function OutputLab : Documenting Functions by using Content-Based HelpLab : 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 InterfaceLab : 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.