

# Advanced Automated Administration with Windows PowerShell (MS-10962)

**ID MS-10962** **Price 1,690.—** €(excl. tax) **Duration 3 days**

This was retired by Microsoft on 31 December 2021 and is now offered by Fast Lane as a workshop with the same curriculum. You will receive a textbook as supplemental material.

## Course Overview

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 a follow on course from the 10961 course. It is built on Windows Server 2012 R2 and Windows 8.1 and while it is specifically focused on Windows PowerShell v4.0, is also relevant in v2.0 and v3.0 Windows PowerShell environments. Expand and build upon the knowledge already acquired in course 10961B 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.

## Who should attend

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.

## Prerequisites

Before attending this course, students must have:

- Knowledge and experience working with Windows PowerShell or knowledge equivalent to the content covered in course 10961.
- Experienced in general Windows Server and Windows Client administration or already experience in administering and supporting Application servers and services including applications like Exchange, SharePoint, and SQL.

## Course Objectives

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

## Detailed Course Outline

### 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 Function

- 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: Creating a Script Module

- 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: Defining Parameter Attributes and Input Validation

- 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: Writing Functions that use Multiple Objects

- 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: Documenting Functions by using Content-Based Help

- 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: Supporting -Whatif and -Confirm

- 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: Writing Functions that Accept Pipeline Input

- 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: Producing Complex Function Output

- 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

After completing this module, students will be able to:

- Convert a Command into an Advanced Function

- Create a Script Module
- Define Parameter Attributes and Input Validation
- Write Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Produce Complex Function Output
- Document Functions by using Content-Based Help
- 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**

- Using Static and Instance Members

After completing this module, students will be able to:

- Run Windows PowerShell commands
- Use Microsoft .NET Framework 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

### **Lab: Writing Controller Scripts that Display a User Interface**

- Write Functions to be Used in the Controller Script

- Write a Controller Script that Implements a Text-Based Menu

After completing this module, students will be able to:

- Understand Controller Scripts
- Write Controller Scripts that show a User Interface

## **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**

- Handling Errors in a Script

After completing this module, students will be able to:

- Understand Error Handling
- Handle Errors in a Script

## **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 than Excel automation.

### **Lessons**

- Reading, Manipulating and Writing Data in XML

### **Lab: Reading, Manipulating and Writing Data in XML**

- Test the Provided Tools
- Update an XML Inventory Document

After completing this module, students will be able to:

- Reading, Manipulating and Writing Data in XML

After completing this module, students will be able to:

### **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**

- Write, Run and Push a DSC Configuration

After completing this module, students will be able to:

- Understand Desired State Configuration
- Create and Deploy a DSC Configuration

### **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 and Existing Script

#### **Lab: Analyzing and Debugging and Existing Script**

- Analyzing and Debugging and Existing Script

After completing this module, students will be able to:

- Analyze and Debug a 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

- Understanding Windows PowerShell Workflow

# About Fast Lane



Fast Lane is a global, award-winning specialist in technology and business training as well as consulting services for digital transformation. As the only global partner of the three cloud hyperscalers- Microsoft, AWS and Google- and partner of 30 other leading IT vendors, Fast Lane offers qualification solutions and professional services that can be scaled as needed. More than 4,000 experienced Fast Lane professionals train and advise customers in organizations of all sizes in 90 countries worldwide in the areas of cloud, artificial intelligence, cyber security, software development, wireless and mobility, modern workplace, as well as management and leadership skills, IT and project management.

## Fast Lane Services

- ✓ High End Technology Training
- ✓ Business & Soft Skill Training
- ✓ Consulting Services
- ✓ Managed Training Services
- ✓ Digital Learning Solutions
- ✓ Content Development
- ✓ Remote Labs
- ✓ Talent Programs
- ✓ Event Management Services

## Training Methods

- ✓ Classroom Training
- ✓ Instructor-Led Online Training
- ✓ FLEX Classroom – Classroom & Online Hybrid
- ✓ Onsite & Customized Training
- ✓ E-Learning
- ✓ Blended & Hybrid Learning
- ✓ Mobile Learning

## Technologies & Solutions

- ✓ Digital Transformation
- ✓ Artificial Intelligence
- ✓ Cloud
- ✓ Networking
- ✓ Cyber Security
- ✓ Wireless & Mobility
- ✓ Modern Workplace
- ✓ Data Center



**Worldwide Presence**  
with high-end training centers  
around the globe



**Multiple Awards**  
from vendors such as AWS,  
Microsoft, Cisco, Google, NetApp,  
VMware



**Experienced SMEs**  
with over 19.000 combined  
certifications

### Germany

**Fast Lane Institute for Knowledge  
Transfer GmbH**  
Tel. +49 40 25334610  
[info@flane.de](mailto:info@flane.de) / [www.flane.de](http://www.flane.de)

### Austria

**ITLS GmbH**  
(Partner of Fast Lane)  
Tel. +43 1 6000 8800  
[info@itls.at](mailto:info@itls.at) / [www.itls.at](http://www.itls.at)

### Switzerland

**Fast Lane Institute for Knowledge  
Transfer (Switzerland) AG**  
Tel. +41 44 8325080  
[info@flane.ch](mailto:info@flane.ch) / [www.flane.ch](http://www.flane.ch)