

FOUR-PHASE ORCHESTRATION



Engineering Idempotent PowerShell
for Windows Endpoint Automation

Four-Phase Orchestration

Engineering Idempotent PowerShell for Windows Endpoint Automation

Owen Heaume

This book is available at <https://leanpub.com/atomic-powershell>

This version was published on 2026-07-26



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Owen Heaume

Contents

Preface	1
Who This Book Is For	2
How to Use This Book	3
 Part I: The Case for Deterministic Orchestration	 5
Chapter 1: The Problem with Procedural Scripts	6
1.1 The Script That Worked, Until It Didn't	6
1.2 Four Symptoms of Procedural Rot	7
1.3 Why This Matters at Fleet Scale	10
1.4 Introducing Ashfield College	10
1.5 What Solving This Properly Requires	11
Chapter Summary	12
 Chapter 2: The Four-Phase Contract	 14
2.1 Four Questions, Not Four Steps	14
2.2 The Central Rule	14
2.3 One-Way Data Flow	14
2.4 Why Four, and Not Three or Five	14
2.5 Resolving Chapter 1, Symptom by Symptom	14
2.6 The Shape in 4 -phase.ps1	14
Chapter Summary	15
 Chapter 3: The Atomic Instrument	 16
3.1 The Instrument, Not the Orchestrator	16
3.2 The Discovery Contract	16
3.3 The Execution Contract	16
3.4 Execution Instruments Are Dumb On Purpose	16
3.5 Why Scalar, Not Iterative	16

3.6 The Naming Convention Is Part of the Contract	16
Chapter Summary	17
Part II: The Framework Chassis	18
Chapter 4: The Parameter Gate and the Metadata Block	19
4.1 A Parser-Level Guarantee, Not a Style Convention	19
4.2 Why This Bites Hardest at the Worst Possible Time	19
4.3 No CmdletBinding at the Top: A Deliberate Absence	19
4.4 The Metadata Block: Identity as a First-Class Citizen	19
4.5 Why Metadata Is Not a Parameter	19
Chapter Summary	19
Chapter 5: Transcription as a Forensic Requirement	21
5.1 Two Kinds of Audit Trail	21
5.2 The IntuneManagementExtension\Logs Convention	21
5.3 The %TEMP% Fallback, and What It Actually Costs	21
5.4 Defensive by Necessity: Stop-Transcript Before Start-Transcript .	21
5.5 Stop-fpTranscript and the Discipline of a finally Block	21
5.6 \$CurrentLogFile: Why a Side Channel Was Never Necessary	21
Chapter Summary	22
Chapter 6: Two Data Contracts, Config and Pipeline	23
6.1 Generated Output, Authored Input	23
6.2 New-fpPipelineObject: A Fixed Skeleton, Not a Growing Object . .	23
6.3 Get-fpStaticPipeline: Solving a Problem That No Longer Exists . .	23
6.4 \$config: Desired State, Authored by a Human	23
6.5 A First Pass at the Ashfield \$config	23
Chapter Summary	24
Chapter 7: The Visual Grammar of the Console	25
7.1 Why Console Output Deserves Its Own Contract	25
7.2 Write-fpConsoleLog: Seven Types, One Grammar	25
7.3 What Actually Survives Into the Transcript	25
7.4 Write-fpObjectToLog: Deconstruction as a Console Convenience .	25
7.5 Two Header Functions, Two Jobs	25
Chapter Summary	25

Part III: Phase 1, Discovery 27

Chapter 8: The Principle of Passive Interrogation 28

8.1 Read-Only Is Necessary. Passive Is the Actual Standard. 28

8.2 The Temptation to Test by Doing 28

8.3 Facts, Not Opinions: The Boundary Discovery Must Not Cross . . . 28

8.4 Test-Path: An Audit of Every Call Site 28

8.5 What This Buys You When the System Is Actually on Fire 28

Chapter Summary 29

Chapter 9: The Discovery Library, Read One Instrument at a Time . . . 30

9.1 Get-fpService: The Simplest Case 30

9.2 Get-fpRegistryValue: The One That Nearly Broke the Contract . . 30

9.3 Get-fpFile: The Reference Pattern 30

9.4 Get-fpFileEncoding: Reading Bytes Without Reading the File . . . 30

9.5 Get-fpShortcut: COM in Discovery, and Why That’s Fine Here . . . 30

9.6 Test-fpPathExists: One Instrument, Two Providers 31

Chapter Summary 31

Chapter 10: The Conductor and the Predicate 32

10.1 What a Conductor Actually Does 32

10.2 Which Facts Earn Their Own Instrument 32

10.3 Test-fpIsAdmin: Discovery’s Interrogation, Not Validation’s Verdict 32

10.4 Which Direction Elevation Actually Needs to Run 32

10.5 What Happens When Composition Itself Fails 32

10.6 An Aggregation Idiom Worth Retiring 32

Chapter Summary 33

Part IV: Phase 2, Validation 34

Chapter 11: Hard Stops vs Soft Exits 35

11.1 Two Ways For a Run To Stop 35

11.2 Why Elevation Never Gets to Be a Soft Exit 35

11.3 The General Principle, For Checks This Book Has Not Written Yet 35

Chapter Summary 35

Chapter 12: Building Get-fpValidation 36

12.1 What Runs Before Any Check Does 36

12.2 Why a Check Cannot Be Anything but a Predicate 36

12.3 Why the Function Returns a Dictionary, Not Just a Verdict	36
Chapter Summary	36
Part V: Phase 3, Logic	37
Chapter 13: The Brain of the Orchestrator	38
13.1 A Different Shape of Comparison	38
13.2 Reading Discovery, Never Re-Asking It	38
13.3 What Logic Decides, and What It Refuses to Touch	38
13.4 An Empty List Is Still a Complete Answer	38
13.5 Why Execution Never Has to Ask	38
Chapter Summary	38
Chapter 14: Constructing the Work Order	40
14.1 The Initialisation That Isn't Optional	40
14.2 Returning Without Losing the Shape	40
14.3 A Second Guard at the Call Site, and Why It Isn't the Same Mistake	40
14.4 Reason Exists for a Reader, Not for the Machine	40
Chapter Summary	40
Chapter 15: Case Study, the Ashfield Work Order Matrix	41
15.1 What Discovery Has to Have Gathered First	41
15.2 Extending Ashfield's Config	41
15.3 Six Requirements, Seven Comparisons, Two Prerequisites	41
15.4 Three Machines, One Function, Zero Code Changes	41
Chapter Summary	41
Part VI: Phase 4, Execution	42
Chapter 16: The Task Dispatcher Pattern	43
16.1 The One Loop in the Whole Framework	43
16.2 Why the Switch Lives in the Dispatcher, Not the Instruments . . .	43
16.3 The Default Case Is Not Decoration	43
16.4 The Same Lesson, a Second Time	43
16.5 WhatIf Passes Through, It Doesn't Originate Here	43
Chapter Summary	43
Chapter 17: Building the Execution Library	45
17.1 The Shape Every Instrument Shares	45

CONTENTS

17.2 The Refactor: Start-fpService and Set-fpService	45
17.3 Set-fpRegistryValue: No Exception Needed Anymore	45
17.4 Set-fpIniValue: The Same Simplification, Applied to a File	45
17.5 New-fpRegistryKey and New-fpFileStructure: Where the Excep- tion Went	45
17.6 Remove-fpIniValue: Removal Is Mechanical, Not Judgment	46
17.7 Set-fpShortcut and Set-fpFileEncoding	46
Chapter Summary	46
Chapter 18: WhatIf as a First-Class Citizen	47
18.1 The Automatic Way and the Manual Way	47
18.2 The Automatic Message Doesn't Speak This Transcript's Language	47
18.3 Every Instrument Has to Return a Real Result Object, Even in Simulation	47
18.4 Determinism Over Convention	47
18.5 What This Trade Gives Up	47
Chapter Summary	48
Chapter 19: The Result Object and Forensic Capture	49
19.1 Four Fields, Every Time	49
19.2 Why the Contract Doesn't Need to Say What Was Attempted	49
19.3 Turning a List of Results Into One Verdict	49
19.4 A Gap in the Contract's Own Enforcement	49
19.5 One Assignment, One Exit	49
Chapter Summary	49
Part VII: The Ashfield College Scenario, End to End	51
Chapter 20: Defining the Scenario	52
20.1 A Requirements Document Is Not a Config File	52
20.2 Why Now	52
20.3 The Ashfield Classroom PC: Six Requirements	52
20.4 What This Document Doesn't Decide	53
20.5 The Complete Config	53
Chapter Summary	53
Chapter 21: Discovery Implementation	54
21.1 Filling the Extension Point	54
21.2 Two Facts, One Node: What IniPolicy Reports	54

CONTENTS

21.3 Assembling the Extended Host Object	54
21.4 Extending MAIN's Reporting Block	54
Chapter Summary	54
Chapter 22: Validation Implementation	55
22.1 Why This Phase Needed No Extension Point	55
22.2 PRE-1: Administrative Context	55
22.3 PRE-2: Available Disk Space	55
22.4 What the Transcript Actually Shows	55
Chapter Summary	55
Chapter 23: Execution and Dispatch	56
23.1 Wiring Logic Into the Template	56
23.2 The Dispatcher Learns Nine New Words	56
23.3 Why Service State No Longer Composes Two Calls Into One . . .	56
23.4 A Dry Run for the Whole Estate	56
Chapter Summary	56
Chapter 24: Reading the Forensics	57
24.1 Two Artifacts, One Underlying Object	57
24.2 A Coarse Status Hides a Story Worth Reading Further	57
24.3 Matching a Result Back to the Item That Staged It	57
24.4 Reading a Run That Never Reached Execution	57
Chapter Summary	57
Part VIII: Hardening the Framework	58
Chapter 25: Unit Testing the Framework with Pester	59
25.1 The Tests Appendix E Already Asked For	59
25.2 Mocking Get-Service: What Discovery Needs That Logic Didn't	59
25.3 Testing the Post-Flight Check: Proving a Lying Cmdlet Gets Caught	59
25.4 What Mocking Can't Give You	59
Chapter Summary	59
Chapter 26: Extending the Instrument Library	60
26.1 Get-fpFirewallRule: Filling In the Discovery Template	60
26.2 Set-fpFirewallRule: Filling In the Execution Template, and Earning Its Post-Flight Check	60
26.3 A Short Note on Where Discovery Is Allowed to Look	60

26.4 Closing the Loop: What Comes After Writing the Pair	60
26.5 Looking Up What Already Exists	60
26.6 One Dependency Worth Knowing About Before You Test in Isolation	61
Chapter Summary	61
Chapter 27: Anti-Patterns and Common Failures	62
27.1 A Discovery Instrument That Mutates “Just This Once”	62
27.2 A Logic Comparison That Calls Get-Service Directly	62
27.3 An Execution Instrument That Re-Implements Discovery’s Own Check	62
27.4 An Execution Instrument Whose -WhatIf Check Arrives Too Late	62
27.5 A Pipeline Object Serialised Before EndTime Is Set	62
Chapter Summary	63
Appendix A: Full Annotated Listing	64
Appendix B: Atomic Instrument Library Reference	65
Discovery Instruments	65
Execution Instruments	66
The Template’s Own Placeholder	68
Appendix C: Troubleshooting Guide	69
The Transcript Won’t Start	69
JSON Serialisation: Slow, Truncated, or Silently Wrong, Not Actually Hanging	69
COM Object Leak: Get - fpShortcut, Not Set - fpShortcut	69
Appendix D: Pester Test Suite for the Ashfield Scenario	70
Logic: All Nine Comparisons	70
Discovery: Confirming the \$null-on-Absence Contract	70
Execution: Happy Path and the WhatIf Short-Circuit	70
What This Suite Does and Doesn’t Prove	70
Appendix E: Recommended Build Order for a New Orchestration	71

Preface

Every PowerShell engineer who has managed production systems for long enough has a version of the same story. A script that started as ten reasonable lines grows, over months of “just one more fix,” into three hundred lines nobody wants to touch. It works, mostly, until the day it doesn’t, and the only evidence of what went wrong is a scroll of console output that has already scrolled past.

This book exists because that story shouldn’t be inevitable. The four-phase architecture it teaches, Discovery, Validation, Logic, Execution, is not a new invention. Separating a program into clearly bounded concerns is old, well-understood engineering discipline. What’s missing from most PowerShell work isn’t the idea, it’s a concrete, worked example of what that discipline actually looks like in the specific, unglamorous reality of Windows endpoint automation: services that report success without the change actually landing, registry keys that need creating before they can be set, transcripts that have to survive a run even when the run itself fails partway through.

Every piece of code in this book was written the way it’s meant to be used: built, run, and checked against a real Windows 11 environment, not only reasoned about on paper. Several of the bugs described in these chapters are real, found while the surrounding text was being written, and they are left in deliberately, as the teaching moments they turned out to be, rather than quietly corrected and smoothed away afterward. A framework earns more trust for having been wrong once and caught, in front of the reader, than for arriving looking as though it was always right.

None of this promises automation without failure. It promises something more useful: automation that fails honestly, in a form a colleague, or a version of you six months from now, can actually read.

Who This Book Is For

This book is written for PowerShell practitioners who are already comfortable with functions, parameters, objects, and basic error handling, and who are responsible for automation that has to keep working after they've moved on to something else. If you maintain scripts that manage real Windows 10 or 11 endpoints, classroom fleets, departmental workstations, kiosk machines, anything under central management, and you're the person who gets the call when one of them breaks in a way nobody can immediately explain, this book is for you.

It assumes PowerShell 5.1 as the working environment, since that's still the version running on the majority of production Windows endpoints, and it assumes you've written scripts before. It does not cover PowerShell syntax from first principles, variables, loops, `if` statements, and basic cmdlet usage are all treated as already known.

This book is likely to disappoint you if you're looking for:

- **An introduction to PowerShell itself.** Start with fundamentals first; this book picks up well past that point.
- **Cloud or Azure Resource Manager automation.** Everything here targets on-premises and hybrid Windows endpoints specifically, not cloud infrastructure provisioning.
- **A deployment or CI/CD tutorial.** Getting a finished script onto a thousand machines, via Intune, SCCM, Group Policy, or anything else your organisation already uses, is left entirely to the reader. This book stops at producing a script that's safe to ship, however you ship it.
- **A grab-bag of unconnected tips and cmdlet tricks.** Every pattern here is built and demonstrated through one continuous, evolving case study, not a series of disconnected snippets.

If your team currently has five engineers writing PowerShell five different ways, and you want a shared architecture any of them could open, understand, and trust without asking the person who wrote it, this is precisely the gap this book is written to close.

How to Use This Book

Read the chapters in order, at least on your first pass. This isn't a reference manual organised by topic that rewards dipping in at random. It builds one continuous case study, a further-education college's classroom-fleet automation, from a blank file in Part I through to a complete, tested, forensically auditable script by the book's final chapters. Later chapters assume you remember decisions made in earlier ones. Chapter 15's six comparisons, for instance, get referenced directly across several of the chapters that follow it. Reading out of order will work against you here more than it would in a typical technical book.

Keep an editor open, and build your own copy of the template as you go, rather than only reading about it. Nearly every chapter quotes real, working PowerShell, and the strongest way to absorb it is to actually type or copy each piece into your own `4-phase.ps1` at the point the book introduces it, then run it, rather than treating the code blocks as illustrations to skim past. Appendix E lays out a recommended build order matching the book's own chapter sequence, if you want a concrete, step-by-step checklist to follow alongside your reading rather than working it out yourself as you go. You don't need an elaborate test setup to do this, any Windows 10 or 11 machine with PowerShell 5.1 will do, though testing what you build against a real machine rather than only in your head is very much the spirit of the book.

Treat each chapter's summary as a checkpoint, not a substitute. The bullet points at the end of every chapter confirm you've taken away the right conclusions. They aren't there to let you skip the reasoning that produced them. This book cares as much about why a design choice was made as what the choice was, and the summaries only ever capture the latter.

When a chapter names a trade-off, take it seriously rather than searching for the "right" answer underneath it. Several decisions in this book, `Get-fpRegistryValue`'s scalar return shape instead of a full snapshot, the deliberately coarse `Completed/Failed/Aborted` status model, the granularity lost when one dispatcher branch has to compose two instrument calls into a single result, are considered trades with honestly stated costs, not compromises

waiting to be perfected in a later chapter. Understanding why a reasonable-sounding alternative was rejected is usually more valuable than the pattern that was chosen instead.

Part I: The Case for Deterministic Orchestration

Chapter 1: The Problem with Procedural Scripts

1.1 The Script That Worked, Until It Didn't

Every PowerShell practitioner has written this script. It starts small, a fix for one machine, and it looks something like this.

```
1  # fix-lab-pc.ps1, "quick and dirty" version, do not use in production
2  param(
3      [string]$IniPath = 'C:\ProgramData\Ashfield\exam.ini'
4  )
5
6  Write-Host "Checking exam policy file..."
7  $existing = Get-Content -Path $IniPath -ErrorAction SilentlyContinue
8
9  if ($existing -match 'Lockdown=Enabled') {
10     Write-Host "Lockdown already enabled, skipping."
11 } else {
12     Write-Host "Writing lockdown policy..."
13     Add-Content -Path $IniPath -Value 'Lockdown=Enabled'
14 }
15
16 Write-Host "Checking exam monitor service..."
17 $svc = Get-Service -Name 'AshExamMonitor' -ErrorAction SilentlyContinue
18
19 if ($svc.Status -ne 'Running') {
20     Write-Host "Starting service..."
21     Start-Service -Name 'AshExamMonitor'
22     if ($svc.StartType -ne 'Automatic') {
23         Set-Service -Name 'AshExamMonitor' -StartupType Automatic
24     }
25 }
26
27 Write-Host "Done."
```

It is short, it is readable, and on the machine it was written for, it works. It gets deployed as an Intune remediation, pushed to forty classroom PCs, and for a while nobody thinks about it again.

Then the helpdesk starts getting tickets. On three machines, the script exits with a red error partway through and the policy is never written. On another two, the service starts but the startup type is never corrected, because `$svc` was captured before the service existed in a running state and its `StartType` property was stale by the time the second `if` block read it. On one machine, the check reports the policy as already enabled and skips the write entirely, because `-match` found the string sitting under an old section left over from a previous exam period rather than the live section the exam software actually reads. Nothing in the script distinguishes a match that matters from one that doesn't, so a leftover from months ago looks identical to a setting that is actually in force, and the policy the machine needs is never written at all. The script has no idea any of this happened. It just stops, or reports success while the real work was never done.

This is not a story about a badly written script. The script above is competent, defensive even, by the standards most of us were taught. The problem is the shape of it, not the content. This chapter is about naming that shape and why it fails, reliably, once you run the same script against more than one machine, more than once, over any meaningful stretch of time.

1.2 Four Symptoms of Procedural Rot

Look again at `fix-lab-pc.ps1`. Every one of its problems traces back to one of four structural symptoms.

Hidden state. The script's understanding of the machine is scattered across two `if` conditions, evaluated at two different moments, using two different queries. There is no single point in the script where you could ask "what does this machine currently look like" and get one answer. The state is implicit in the control flow itself.

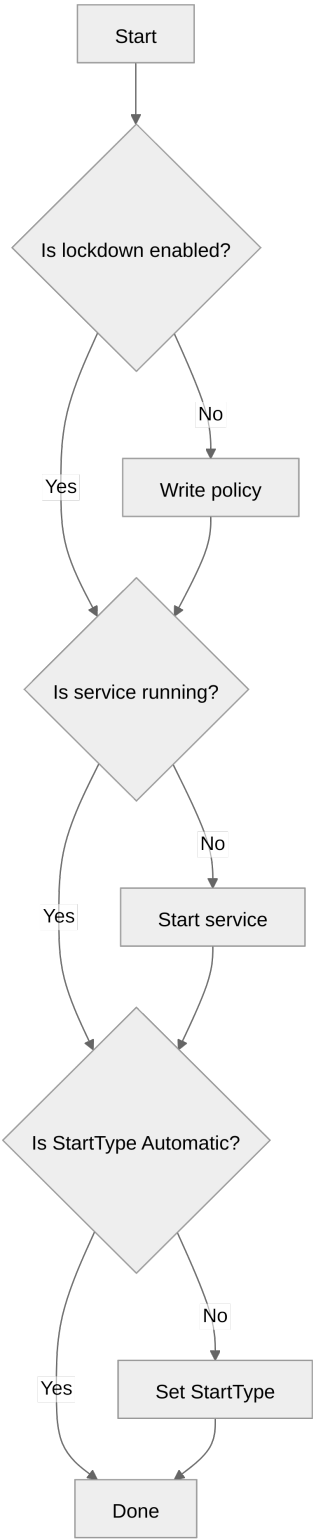
Non-idempotent side effects. Run the script twice in a row on a healthy machine and it should do nothing the second time. Run it twice in a row on a machine where the first run failed partway through, and you cannot predict what happens, because the second run's decisions depend on exactly how far the first run got before it died. A half-written config file is worse than none at all: a file left with a duplicated key, or a key appended under the wrong section, is a file the next run's crude `-match` check will misread again, compounding the damage. Idempotency is usually described as "safe to run more than once."

The sharper way to think about it is: the outcome should depend only on the current state of the machine, never on the history of previous attempts.

Unreviewable failure points. There is no way to ask this script what it is *about* to do without letting it do it. A change advisory board, a scheduled maintenance window, a nervous administrator running this by hand for the first time, none of them can get a straight answer to “what will this actually touch” short of reading the source and simulating it in their head.

The audit gap. When the script dies on line 14 of the deployed version, what survives? A transcript, if one was configured. A stream of `Write-Host` text, if the console was still attached. Nothing at all, if this ran unattended and the host closed. There is no structured record of what was found, what was decided, and what was attempted, only what happened to be printed before the failure. Six months later, when someone asks “why does this machine still not have the lockdown policy set,” the honest answer is that nobody can reconstruct it without physically visiting the machine again.

None of these four symptoms are visible from a single successful run on a single machine. That is exactly what makes them dangerous. The script demos perfectly. It is the fleet, and time, that expose it.



Notice what this diagram cannot show you: at no point is there a node that represents “the complete current state of the machine.” Detection and correction are fused together at every single decision point. There is no seam anywhere in this flow where you could insert a validation gate, generate a dry run report, or hand the “what needs to change” question to a different piece of code than the “make it change” question. The tangle is not a flaw in this particular script. It is the inevitable result of writing detection and correction in the same breath.

1.3 Why This Matters at Fleet Scale

A script that assumes a single starting state is a script that has only ever been tested against a single starting state, whether or not the author realised it.

Forty classroom PCs that were imaged from the same source, six months ago, are not forty identical machines. Some had the exam policy file survive a profile reset and some did not. Some had a technician manually fix the exam monitor service during an incident last term and forget to update the deployment baseline. One received a feature update slightly ahead of the rest of the fleet because it happened to be powered on during a maintenance window. None of this is exotic. It is the ordinary entropy of forty machines existing in the world for six months.

`fix-lab-pc.ps1` does not fail on all forty machines. It fails on the ones whose starting state was not the one its author was picturing at 4pm on a Friday. That is the trap: a script like this can pass every test the author runs, because the author only ever tests against machines that look like the one on their desk. It then meets the actual diversity of a real fleet in production, where the cost of a failure is a support ticket instead of a compile error.

This is the point at which “write a script that fixes the problem” quietly becomes a different, harder problem: “write a script that can be run against a machine in *any* state and reliably bring it to the *same* state, tell you exactly what it found and what it did, and do so safely whether it is the first time it has run on that machine or the fiftieth.” That is not a scripting problem anymore. It is an architecture problem.

1.4 Introducing Ashfield College

The rest of this book is built around a single running scenario, so that every architectural decision can be judged against a real requirement rather than a toy example.

Ashfield College is a further-education college with several computer labs used for teaching and formal examinations. The lab PCs are imaged once a year from a known-good baseline, then left to drift for the rest of the academic year: local fixes applied by technicians during incidents, group policy changes that only partially apply, ad hoc changes made under time pressure before an exam. By the time the IT team needs to run a compliance pass before the start of a new term, the estate is in a genuinely unknown, heterogeneous state, and nobody can say with confidence which of the forty machines need which fixes.

The compliance baseline the team needs every lab PC to reach, regardless of its starting condition, is this:

- The exam lockdown policy key in the machine-wide configuration file is present and set correctly.
- Any deprecated key left over from a previous configuration is removed.
- The exam monitoring service is running and set to start automatically.
- The desktop shortcut used to launch exam software points at the current, correct target.
- A specific registry value enforcing exam lockdown policy is present and correct.
- The logon script consumed by a legacy ingestion tool is saved as UTF-8 without a byte order mark, because the tool fails silently on files that have one.

We are not yet in a position to build this properly. That is deliberate. What this chapter has established is why a script written the way `fix-lab-pc.ps1` was written cannot be trusted to do it, no matter how carefully it is written, because the problem is structural, not a matter of missing error handling or an extra `-ErrorAction Stop`. Chapters 2 and 3 introduce the architecture that resolves each of the four symptoms named in this chapter directly, and Part VII returns to this exact list of six requirements and builds all of it for real.

1.5 What Solving This Properly Requires

Before the next chapter names the Four-Phase Contract, it is worth being precise about what any real solution has to guarantee, because these four properties are the yardstick the rest of the book is measured against.

Idempotency. Running the solution against a machine that is already compliant must be indistinguishable, in its effect on the system, from not running it at all.

Determinism. Given the same starting state, the solution must make the same decisions every time. No decision should depend on how far a previous, failed run happened to get.

Auditability. After the fact, without remoting into the machine and without reproducing the problem, it must be possible to reconstruct exactly what was found, what was decided, and what was attempted or achieved.

Separation of concerns. Changing how a problem is detected must never require touching the code that fixes it, and changing how a problem is fixed must never require touching the code that detects it. These are different questions, asked by different code, at different times.

`fix-lab-pc.ps1` fails all four. It is not a matter of degree, it fails each one completely, because detection and correction are the same statement. The rest of this book is the answer to a single question: what does a script have to look like, structurally, before it can honestly claim all four?

* * *

Chapter Summary

- A script that mixes detection and correction in the same statement cannot be idempotent, deterministic, auditable, or properly separated into reviewable concerns, no matter how carefully it is written.
- These failures are invisible on a single successful run against a single machine. They only surface once the same script meets a real fleet, in a real range of starting states, over real time.

- Ashfield College's six-point compliance baseline is the running requirement for the rest of the book: exam policy file key, deprecated key removal, exam service state, desktop shortcut correctness, a registry policy value, and BOM-free file encoding.
- Any real solution must guarantee idempotency, determinism, auditability, and separation of concerns. Chapter 2 introduces the architecture built to guarantee all four: the Four-Phase Contract.

Chapter 2: The Four-Phase Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

2.1 Four Questions, Not Four Steps

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

2.2 The Central Rule

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

2.3 One-Way Data Flow

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

2.4 Why Four, and Not Three or Five

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

2.5 Resolving Chapter 1, Symptom by Symptom

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

2.6 The Shape in 4-phase.ps1

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 3: The Atomic Instrument

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

3.1 The Instrument, Not the Orchestrator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

3.2 The Discovery Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

3.3 The Execution Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

3.4 Execution Instruments Are Dumb On Purpose

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

3.5 Why Scalar, Not Iterative

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

3.6 The Naming Convention Is Part of the Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part II: The Framework Chassis

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 4: The Parameter Gate and the Metadata Block

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

4.1 A Parser-Level Guarantee, Not a Style Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

4.2 Why This Bites Hardest at the Worst Possible Time

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

4.3 No CmdletBinding at the Top: A Deliberate Absence

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

4.4 The Metadata Block: Identity as a First-Class Citizen

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

4.5 Why Metadata Is Not a Parameter

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 5: Transcription as a Forensic Requirement

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

5.1 Two Kinds of Audit Trail

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

5.2 The IntuneManagementExtension\Logs Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

5.3 The %TEMP% Fallback, and What It Actually Costs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

5.4 Defensive by Necessity: Stop-Transcript Before Start-Transcript

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

5.5 Stop-fpTranscript and the Discipline of a finally Block

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

5.6 \$CurrentLogFile: Why a Side Channel Was Never Necessary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 6: Two Data Contracts, Config and Pipeline

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

6.1 Generated Output, Authored Input

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

6.2 New-fpPipelineObject: A Fixed Skeleton, Not a Growing Object

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

6.3 Get-fpStaticPipeline: Solving a Problem That No Longer Exists

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

6.4 \$config: Desired State, Authored by a Human

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

6.5 A First Pass at the Ashfield \$config

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 7: The Visual Grammar of the Console

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

7.1 Why Console Output Deserves Its Own Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

7.2 Write-fpConsoleLog: Seven Types, One Grammar

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

7.3 What Actually Survives Into the Transcript

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

7.4 Write-fpObjectToLog: Deconstruction as a Console Convenience

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

7.5 Two Header Functions, Two Jobs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part III: Phase 1, Discovery

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 8: The Principle of Passive Interrogation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

8.1 Read-Only Is Necessary. Passive Is the Actual Standard.

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

8.2 The Temptation to Test by Doing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

8.3 Facts, Not Opinions: The Boundary Discovery Must Not Cross

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

8.4 Test-Path: An Audit of Every Call Site

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

8.5 What This Buys You When the System Is Actually on Fire

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 9: The Discovery Library, Read One Instrument at a Time

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

9.1 Get-fpService: The Simplest Case

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

9.2 Get-fpRegistryValue: The One That Nearly Broke the Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

9.3 Get-fpFile: The Reference Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

9.4 Get-fpFileEncoding: Reading Bytes Without Reading the File

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

9.5 Get-fpShortcut: COM in Discovery, and Why That's Fine Here

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

9.6 Test-fpPathExists: One Instrument, Two Providers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 10: The Conductor and the Predicate

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

10.1 What a Conductor Actually Does

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

10.2 Which Facts Earn Their Own Instrument

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

10.3 Test-fplsAdmin: Discovery's Interrogation, Not Validation's Verdict

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

10.4 Which Direction Elevation Actually Needs to Run

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

10.5 What Happens When Composition Itself Fails

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

10.6 An Aggregation Idiom Worth Retiring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part IV: Phase 2, Validation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 11: Hard Stops vs Soft Exits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

11.1 Two Ways For a Run To Stop

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

11.2 Why Elevation Never Gets to Be a Soft Exit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

11.3 The General Principle, For Checks This Book Has Not Written Yet

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 12: Building Get-fpValidation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

12.1 What Runs Before Any Check Does

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

12.2 Why a Check Cannot Be Anything but a Predicate

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

12.3 Why the Function Returns a Dictionary, Not Just a Verdict

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part V: Phase 3, Logic

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 13: The Brain of the Orchestrator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

13.1 A Different Shape of Comparison

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

13.2 Reading Discovery, Never Re-Asking It

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

13.3 What Logic Decides, and What It Refuses to Touch

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

13.4 An Empty List Is Still a Complete Answer

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

13.5 Why Execution Never Has to Ask

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 14: Constructing the Work Order

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

14.1 The Initialisation That Isn't Optional

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

14.2 Returning Without Losing the Shape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

14.3 A Second Guard at the Call Site, and Why It Isn't the Same Mistake

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

14.4 Reason Exists for a Reader, Not for the Machine

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 15: Case Study, the Ashfield Work Order Matrix

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

15.1 What Discovery Has to Have Gathered First

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

15.2 Extending Ashfield's Config

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

15.3 Six Requirements, Seven Comparisons, Two Prerequisites

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

15.4 Three Machines, One Function, Zero Code Changes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part VI: Phase 4, Execution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 16: The Task Dispatcher Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

16.1 The One Loop in the Whole Framework

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

16.2 Why the Switch Lives in the Dispatcher, Not the Instruments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

16.3 The Default Case Is Not Decoration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

16.4 The Same Lesson, a Second Time

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

16.5 WhatIf Passes Through, It Doesn't Originate Here

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 17: Building the Execution Library

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.1 The Shape Every Instrument Shares

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.2 The Refactor: Start-fpService and Set-fpService

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.3 Set-fpRegistryValue: No Exception Needed Anymore

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.4 Set-fpIniValue: The Same Simplification, Applied to a File

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.5 New-fpRegistryKey and New-fpFileStructure: Where the Exception Went

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.6 Remove-fpIniValue: Removal Is Mechanical, Not Judgment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

17.7 Set-fpShortcut and Set-fpFileEncoding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 18: WhatIf as a First-Class Citizen

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

18.1 The Automatic Way and the Manual Way

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

18.2 The Automatic Message Doesn't Speak This Transcript's Language

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

18.3 Every Instrument Has to Return a Real Result Object, Even in Simulation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

18.4 Determinism Over Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

18.5 What This Trade Gives Up

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 19: The Result Object and Forensic Capture

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

19.1 Four Fields, Every Time

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

19.2 Why the Contract Doesn't Need to Say What Was Attempted

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

19.3 Turning a List of Results Into One Verdict

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

19.4 A Gap in the Contract's Own Enforcement

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

19.5 One Assignment, One Exit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part VII: The Ashfield College Scenario, End to End

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 20: Defining the Scenario

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

20.1 A Requirements Document Is Not a Config File

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

20.2 Why Now

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

20.3 The Ashfield Classroom PC: Six Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

REQ-1: Exam Policy Key

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

REQ-2: Deprecated Key Removal

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

REQ-3: Exam Service State

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

REQ-4: Desktop Shortcut Correctness

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

REQ-5: Registry Policy Value

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

REQ-6: Logon Script Encoding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

20.4 What This Document Doesn't Decide

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

20.5 The Complete Config

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 21: Discovery Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

21.1 Filling the Extension Point

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

21.2 Two Facts, One Node: What IniPolicy Reports

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

21.3 Assembling the Extended Host Object

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

21.4 Extending MAIN's Reporting Block

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 22: Validation Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

22.1 Why This Phase Needed No Extension Point

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

22.2 PRE-1: Administrative Context

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

22.3 PRE-2: Available Disk Space

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

22.4 What the Transcript Actually Shows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 23: Execution and Dispatch

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

23.1 Wiring Logic Into the Template

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

23.2 The Dispatcher Learns Nine New Words

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

23.3 Why Service State No Longer Composes Two Calls Into One

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

23.4 A Dry Run for the Whole Estate

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 24: Reading the Forensics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

24.1 Two Artifacts, One Underlying Object

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

24.2 A Coarse Status Hides a Story Worth Reading Further

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

24.3 Matching a Result Back to the Item That Staged It

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

24.4 Reading a Run That Never Reached Execution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Part VIII: Hardening the Framework

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 25: Unit Testing the Framework with Pester

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

25.1 The Tests Appendix E Already Asked For

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

25.2 Mocking Get-Service: What Discovery Needs That Logic Didn't

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

25.3 Testing the Post-Flight Check: Proving a Lying Cmdlet Gets Caught

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

25.4 What Mocking Can't Give You

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 26: Extending the Instrument Library

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

26.1 Get-fpFirewallRule: Filling In the Discovery Template

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

26.2 Set-fpFirewallRule: Filling In the Execution Template, and Earning Its Post-Flight Check

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

26.3 A Short Note on Where Discovery Is Allowed to Look

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

26.4 Closing the Loop: What Comes After Writing the Pair

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

26.5 Looking Up What Already Exists

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

26.6 One Dependency Worth Knowing About Before You Test in Isolation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter 27: Anti-Patterns and Common Failures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

27.1 A Discovery Instrument That Mutates “Just This Once”

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

27.2 A Logic Comparison That Calls Get-Service Directly

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

27.3 An Execution Instrument That Re-Implements Discovery’s Own Check

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

27.4 An Execution Instrument Whose -WhatIf Check Arrives Too Late

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

27.5 A Pipeline Object Serialised Before EndTime Is Set

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Chapter Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Appendix A: Full Annotated Listing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Appendix B: Atomic Instrument Library Reference

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Discovery Instruments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Test-fpIsAdmin

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpNetworkIdentity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpHardwareIdentity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpService

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpShortcut

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpRegistryValue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpFileEncoding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Get-fpIniValue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Test-fpPathExists

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Execution Instruments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

New-fpRegistryKey

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

New-fpFileStructure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Set-fpIniValue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Remove-fpIniValue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Start-fpService

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Set-fpService

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Set-fpShortcut

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Set-fpRegistryValue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Set - fpFileEncoding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

The Template's Own Placeholder

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Set - fpApp

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Appendix C: Troubleshooting Guide

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

The Transcript Won't Start

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

JSON Serialisation: Slow, Truncated, or Silently Wrong, Not Actually Hanging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

COM Object Leak: Get - fpShortcut, Not Set - fpShortcut

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Appendix D: Pester Test Suite for the Ashfield Scenario

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Logic: All Nine Comparisons

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Discovery: Confirming the \$null-on-Absence Contract

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Execution: Happy Path and the WhatIf Short-Circuit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

What This Suite Does and Doesn't Prove

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Appendix E: Recommended Build Order for a New Orchestration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 1: Write `$config` First, Even as a Rough Pass

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 2: Build Atomic Instrument Pairs, One Resource at a Time

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 3: Wire Discovery, Then Run It Alone

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 4: Write Validation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 5: Write Logic

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 6: Wire the Dispatcher Last

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.

Step 7: Run It End to End, Against More Than One State

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/atomic-powershell>.