Contents

About the Author xv

About the Technical Reviewer xvii

Acknowledgments xix

Introduction xxi

Chapter 1: PowerShell Basics 1

What Is PowerShell? 1

Why Is PowerShell Important? 2

History of PowerShell Versions 3

Starting PowerShell 4

The Command Line Interface (CLI) 5

The Integrated Scripting Environment (ISE) 7

Enabling Scripting 8

Using the Integrated Scripting Environment 10

Encountering Script Errors 16

Summary 17

Chapter 2: The PowerShell Language 19

Challenges to Learning PowerShell 19

Thinking Differently 19

Just Got to Get the Job Done 21

Getting Good Examples 21

The Paradox of Power 22

The PowerShell Language 22
A Brief Introduction Using a Script 22
Operators and Operands 25
Cmdlets 26
Variables 27
Cmdlet Output 28
Strings 29
Files 33
Control Flow 35
Script Blocks 39
Arrays 39
Associative Arrays 41
Summary 42
Chapter 3: Advanced Programming 43
Passing Parameters 43
Functions 48
It's All about Objects 51
A Real-World Use of PowerShell Objects for ETL 55
Remember the Pipeline 60
Pipelining Example: SQL Server Deployment 61
Formatting Output 65
Creating and Using COM Objects 70
Creating and Using .Net Objects 71
Waiting for a File 73
Summary 74

Chapter 4: Writing Scripts 75

Strict Mode 75

Error Handling 76

Using the Try/Catch Block 76

The Trap Statement 78

Using Try/Catch and the Trap Statement Together 80

Setting the $ErrorActionPreference Preference Variable 85

Using the $Error Variable and the ErrorAction Parameter 85

More on the $Error Variable 87

Common cmdlet Parameters 88

Debugging 90

Events and Script Blocks: The Perfect Marriage 91

Using Windows Events 91

Using .NET Object Events 97

Using PowerShell Transactions 99

Conceptual Overview 99

An Example of How Transactions Should Work 100

Why Doesn't It Work? 103

Setting the Integrated Script Editor (ISE) Options 104

Summary 109

Chapter 5: Writing Reusable Code 111

CmdletBinding Arguments 111

SupportsShouldProcess 112

HelpURI 114

SupportsPaging 115

Validating Parameters 115

Validating a String 115

Validating a Number 116

Validating an Array 117

Using the ValidateScript Attribute 118

Default Parameter Values 122
Specifying a Default Parameter Value 122
Using $PSDefaultParameterValues 123
Default Parameter Values When No Default Is Specified 125
Distinguishing Parameter Default Values from Intended Values 127
Validating Parameters the Old-Fashioned Way 132
Using Parameter Sets 133
Using Switch Parameters with Parameter Sets 136
ValueFromPipeline and ValueFromPipelineByPropertyName Attributes 137
Internationalization with the Data Section 139
Summary 143
Chapter 6: Extending PowerShell 145
The Four Types of Modules 145
Script Modules 145
Flexibility in Modules 155
Module Manifest 159
Dynamic Modules 161
Popular Free PowerShell Modules 163
OData PowerShell Explorer 164
PowerShell Community Extensions (PSCX) 164
ShowUI 166
SQL Server PowerShell Extensions 166
SQLISE 167
Windows Automation Snap-In (WASP) 168
Summary 171
Chapter 7: Accessing SQL Server 173
About the Module 173
Using umd_database 173
How umd_database Works 176
Supporting Functions 180

Calling Stored Procedures 187
Summary 202
Chapter 8: Customizing the PowerShell Environment 203
PowerShell Variables 203
Automatic Variables 204
Preference Variables 220
Environment Variables 226
Using PS Drives 227
Aliases 229
Customizing the Console 230
Profiles 234
Summary 239
Chapter 9: Augmenting ETL Processes 241
The Functions 243
Wait for a File 243
Copy File 245
Unzip the File 246
Add the File Name to a Flat File 248
Move the Files 251
Send Email 252
Load the Files 255
Let's Process Some Files 258
More Useful ETL Functions 261
Data Validation 261
Combine Files 265
Add Column Headings 267
Securing Credentials 268
Send Secure Email 269
Summary 271

Chapter 10: Configurations, Best Practices, and Application Deployment 273

Configuring PowerShell Applications 273

Using Environment Variables 273

PowerShell Variables Scope 275

A Simple Configuration Approach 280

A More Advanced Configuration Approach 281

Taking Configuration a Step Further 284

Storing Configuration Values in a SQL Server Table 287

Using a Script Module to Support Configuration 291

PowerShell Development Best Practices 293

Function Naming 293

File Naming 294

Make Code Reusable 295

Use Source Code Control 296

Build In Configurations from the Start 296

PowerShell Application Deployment 297

Deploying Script Modules 297

Deploying Script Files 299

Deploying Jobs 300

Summary 302

Chapter 11: PowerShell Versus SSIS 303

Introduction 303

Advantages of PowerShell as an ETL Tool 303

Pre-Installed and Free 303

Supports Interaction 304

PowerShell Goes Where SSIS Can't 304

Reusability and Extensibility 304

Dynamic Code 304

Upgrades and Code Dependencies 305

Advantages of SSIS 305
Performance 305
Visual Development Interface 305
SQL Server Integration and Configuration 305
Built-In Components 306
The PowerShell ETL Framework 306
Mapping 307
The Northwind Company Use Case 321
The Products Table Load 323
The Order Table Load 328
The Customers Table 333
Packaging Up the ETL Job 337
Summary 338
Chapter 12: PowerShell Jobs 339
PowerShell Jobs 339
Running Background Jobs 341
Scheduled Jobs 346
Scripting with the Job Cmdlets 357
Using PowerShell with SQL Server Agent 360
Accessing SQL Server Agent Jobs from PowerShell 371
Summary 373
Chapter 13: PowerShell Workflows 375
PowerShell Remote Execution 375
Configuring Remote Execution 376
Using Remote Execution with Invoke-Command 377
Workflows 380
Parallel Execution 383
Sequence 385
Foreach –Parallel 386

ETL Workflow 387
The Workflow Code 388
Calling the Workflow 392
Register the Events 395
Sending Mail 395
Workflow Creation Steps 396
A Poor Man's Hadoop 397
Setting Up the Poor Man's Hadoop Workflow 397
Calling the Poor Man's Hadoop Workflow 399
Summary 400

Index 401