

# Contents

<b>Preface</b>	<b>xvii</b>
<b>Introduction</b>	<b>xix</b>
<b>Chapter 1: Introduction to PowerShell</b>	<b>1</b>
<b>Windows PowerShell Design Principles</b>	<b>1</b>
Preserve the Customer's Existing Investment	2
Provide a Powerful, Object-Oriented Shell	2
Extensibility, Extensibility, Extensibility	2
Tear Down the Barriers to Development	2
<b>A Quick Tour of Windows PowerShell</b>	<b>3</b>
Cmdlets	3
<b>High-Level Architecture of Windows PowerShell</b>	<b>9</b>
Host Application	9
Windows PowerShell Engine	10
Windows PowerShell Snap-ins	10
<b>Summary</b>	<b>11</b>
<b>Chapter 2: Extending Windows PowerShell</b>	<b>13</b>
<b>Types of PowerShell Snap-ins</b>	<b>13</b>
<b>Creating a Standard PowerShell Snap-in</b>	<b>14</b>
Writing a PowerShell Snap-in	14
Registering Your PowerShell Snap-in	17
Listing Available PowerShell Snap-ins	19
Loading a PowerShell Snap-in to a Running Shell	19
Removing a PowerShell Snap-in from a Running Shell	20
Unregistering a PowerShell Snap-in	20
Registering a PowerShell Snap-in without Implementing a Snap-in Clas	21
Saving Snap-in Configuration	22
Starting PowerShell with a Saved Snap-in Configuration	22
Using a Profile to Save a Snap-in Configuration	23
<b>Creating a Custom PowerShell Snap-in</b>	<b>23</b>
Writing a Custom PowerShell Snap-in	23
Using a Custom PowerShell Snap-in	25
<b>Summary</b>	<b>27</b>

## Contents

---

<b>Chapter 3: Understanding the Extended Type System</b>	<b>29</b>
<b>PSObject</b>	<b>29</b>
<b>Constructing a PSObject</b>	<b>30</b>
PSObject(Object)	31
PSObject()	31
PSObject.AsPSObject(someObject)	32
<b>ImmediateBaseObject and BaseObject</b>	<b>33</b>
<b>Members</b>	<b>34</b>
PSMemberInfoCollection	35
ReadOnlyPSMemberInfoCollection	36
Base, Adapted, and Extended Members	37
<b>Types of Members</b>	<b>37</b>
Properties	38
Methods	46
Sets	51
<b>TypeNames</b>	<b>53</b>
<b>Lookup Algorithm</b>	<b>54</b>
<b>Distance Algorithm</b>	<b>54</b>
<b>PSObject Intrinsic Members and MemberSets</b>	<b>55</b>
<b>Errors and Exceptions</b>	<b>55</b>
Runtime Errors	55
Initialization Errors	56
<b>Type Conversion</b>	<b>57</b>
Standard PS Language Conversion	57
Custom Converters	58
<b>ToString Mechanism</b>	<b>60</b>
<b>Type Configuration (TypeData)</b>	<b>60</b>
Well-Known Members	62
Script Access	62
<b>Summary</b>	<b>62</b>
<b>Chapter 4: Developing Cmdlets</b>	<b>63</b>
<b>Getting Started</b>	<b>63</b>
Command-Line Parsing	65
Command Discovery	65
Parameter Binding	66
Command Invocation	67
<b>Using Parameters</b>	<b>67</b>
Mandatory Parameters	67
Positional Parameters	68

## Contents

---

Parameter Sets	71
Parameter Validation	78
Parameter Transformation	80
<b>Processing Pipeline Input</b>	<b>84</b>
Pipeline Parameter Binding	87
<b>Generating Pipeline Output</b>	<b>91</b>
<b>Reporting Errors</b>	<b>92</b>
ErrorRecord	93
ErrorDetails	95
Non-terminating Errors and Terminating Errors	97
<b>Supporting ShouldProcess</b>	<b>98</b>
Confirming Impact Level	100
ShouldContinue()	101
<b>Working with the PowerShell Path</b>	<b>101</b>
<b>Documenting Cmdlet Help</b>	<b>106</b>
<b>Best Practices for Cmdlet Development</b>	<b>114</b>
Naming Conventions	114
Interactions with the Host	115
<b>Summary</b>	<b>116</b>
<b>Chapter 5: Providers</b>	<b>117</b>
<b>Why Implement a Provider?</b>	<b>118</b>
Providers versus Cmdlets	118
<b>Essential Concepts</b>	<b>119</b>
Paths	119
Drives	121
Error Handling	121
Capabilities	122
<b>Hello World Provider</b>	<b>123</b>
<b>Built-in Providers</b>	<b>125</b>
Alias Provider	125
Environment Provider	126
FileSystem Provider	126
Function Provider	126
Registry Provider	127
Variable Provider	128
Certificate Provider	128
<b>Base Provider Types</b>	<b>128</b>
CmdletProvider	129
DriveCmdletProvider	129
ItemCmdletProvider	129

## Contents

---

ContainerCmdletProvider	131
NavigationCmdletProvider	132
<b>Optional Provider Interfaces</b>	<b>132</b>
IContentCmdletProvider	132
IPropertyCmdletProvider	133
IDynamicPropertyCmdletProvider	134
ISecurityDescriptorCmdletProvider	134
<b>CmdletProvider</b>	<b>134</b>
Methods and Properties on CmdletProvider	136
DriveCmdletProvider	139
ItemCmdletProvider	141
ContainerCmdletProvider	147
NavigationCmdletProvider	153
<b>Design Guidelines and Tips</b>	<b>162</b>
<b>Summary</b>	<b>163</b>
<b>Chapter 6: Hosting the PowerShell Engine in Applications</b>	<b>165</b>
<b>Runspaces and Pipelines</b>	<b>165</b>
<b>Getting Started</b>	<b>166</b>
<b>Executing a Command Line</b>	<b>166</b>
Using RunspaceInvoke	166
Using Runspace and Pipeline	168
<b>Using the Output of a Pipeline</b>	<b>170</b>
The Return Value of Invoke()	170
Using PSObject Objects Returned from a Pipeline	170
Handling Terminating Errors	171
<b>Input, Output, and Errors for Synchronous Pipelines</b>	<b>172</b>
Passing Input to Your Pipeline	172
The Output Pipe in Synchronous Execution	173
Retrieving Non-Terminating Errors from the Error Pipe	173
The ErrorRecord Type	174
<b>Other Pipeline Tricks</b>	<b>174</b>
Nested Pipelines	174
Reusing Pipelines	175
Copying a Pipeline Between Runspaces	175
<b>Configuring Your Runspace</b>	<b>176</b>
Creating a Runspace with a Custom Configuration	176
Adding and Removing Snap-Ins	177
Creating RunspaceConfiguration from a Console File	177
Creating RunspaceConfiguration from an Assembly	177
Using SessionStateProxy to Set and Retrieve Variables	178
Fine-Tuning RunspaceConfiguration	179

## Contents

---

<b>Running a Pipeline Asynchronously</b>	<b>181</b>
Calling InvokeAsync()	181
Closing the Input Pipe	182
Reading Output and Error from an Asynchronous Pipeline	182
Monitoring a Pipeline's StateChanged Event	185
Reading Terminating Errors via PipelineStateInfo.Reason	186
Stopping a Running Pipeline	187
<b>Asynchronous Runspace Operations</b>	<b>187</b>
The OpenAsync() Method	187
Handling the Runspace's StateChanged Event	188
<b>Constructing Pipelines Programmatically</b>	<b>189</b>
Creating an Empty Pipeline	189
Creating a Command	189
Merging Command Results	190
Adding Command Parameters	191
Adding Commands to the Pipeline	192
<b>Cmdlets as an API Layer for GUI Applications</b>	<b>193</b>
High-Level Architecture	193
Keys to Successful GUI Integration	194
Providing a Custom Host	194
<b>Summary</b>	<b>195</b>
<b>Chapter 7: Hosts</b>	<b>197</b>
<b>Host-Windows PowerShell Engine Interaction</b>	<b>197</b>
<b>Built-In Cmdlets That Interact with the Host</b>	<b>199</b>
Write-Debug	199
Write-Verbose	200
Write-Warning	202
Write-Progress	203
Write-Host and Out-Host	203
Read-Host	204
<b>Cmdlet and Host Interaction</b>	<b>204</b>
<b>PSHost Class</b>	<b>207</b>
Instanceld	208
Name	209
Version	210
CurrentCulture	210
CurrentUICulture	210
PrivateData	211
EnterNestedPrompt	211
ExitNestedPrompt	212

## Contents

---

Application Notification Methods	214
SetShouldExit	214
<b>PSHostUserInterface Class</b>	<b>221</b>
WriteDebugLine	222
WriteVerboseLine	223
WriteWarningLine	223
WriteProgress	223
WriteErrorLine	224
Write Methods	224
Prompt Method	224
PromptForCredential	226
Read Methods	227
<b>PSHostRawUserInterface Class</b>	<b>227</b>
<b>Summary</b>	<b>231</b>
<b>Chapter 8: Formatting &amp; Output</b>	<b>233</b>
<b>The Four View Types</b>	<b>233</b>
Table: format-table	234
List: format-list	234
Custom: format-custom	235
Wide: format-wide	235
<b>Formatting without#.format.ps1xml</b>	<b>236</b>
<b>Format Configuration File Example</b>	<b>237</b>
<b>Loading Your Format File(s)</b>	<b>238</b>
Update-formatdata	239
Snap-ins	240
RunspaceConfiguration API	240
<b>Anatomy of a Format Configuration File</b>	<b>240</b>
View	241
Name	241
ViewSelectedBy	241
GroupBy	242
<b>TableControl</b>	<b>243</b>
TableHeaders	243
TableRowEntries	244
<b>ListControl</b>	<b>244</b>
ListEntries	245
<b>Wide Control</b>	<b>246</b>
WideEntries	246
<b>Custom Control</b>	<b>246</b>
CustomEntries	248

## Contents

---

<b>Miscellaneous Configuration Entries</b>	<b>248</b>
Wrap	248
AutoSize	248
<b>Scenarios</b>	<b>249</b>
Format Strings	249
Formatting Deserialized Objects	250
Class Inheritance	250
Selection Sets	253
Colors	253
<b>Summary</b>	<b>255</b>
<b>Appendix A: Cmdlet Verb Naming Guidelines</b>	<b>257</b>
<b>Common Verbs</b>	<b>257</b>
<b>Data Verbs</b>	<b>259</b>
<b>Communication Verbs</b>	<b>260</b>
<b>Diagnostic Verbs</b>	<b>260</b>
<b>Lifecycle Verbs</b>	<b>261</b>
<b>Security Verbs</b>	<b>261</b>
<b>Appendix B: Cmdlet Parameter Naming Guidelines</b>	<b>263</b>
<b>Ubiquitous Parameters</b>	<b>263</b>
<b>Activity Parameters</b>	<b>264</b>
<b>Date/Time Parameters</b>	<b>266</b>
<b>Format Parameters</b>	<b>266</b>
<b>Property Parameters</b>	<b>267</b>
<b>Quantity Parameters</b>	<b>268</b>
<b>Resource Parameters</b>	<b>268</b>
<b>Security Parameters</b>	<b>269</b>
<b>Appendix C: Metadata</b>	<b>271</b>
<b>CmdletAttribute</b>	<b>271</b>
Cmdlet Attribute Example	272
<b>ParameterAttribute</b>	<b>272</b>
ParameterAttribute Example	273
<b>AliasAttribute</b>	<b>273</b>
AliasAttribute Example	273
<b>Argument Validation Attributes</b>	<b>273</b>
ValidateSetAttribute	274
ValidatePatternAttribute	274
ValidateLengthAttribute	274

## Contents

---

ValidateCountAttribute	275
ValidateRangeAttribute	275
<b>Allow and Disallow Attributes</b>	<b>276</b>
AllowNullAttribute	276
AllowEmptyStringAttribute	276
AllowEmptyCollectionAttribute	277
ValidateNotNullAttribute	277
ValidateNotNullOrEmptyAttribute	277
CredentialAttribute	277
<b>Extending Parameter Metadata Attributes</b>	<b>278</b>
ValidateArgumentsAttribute	278
ValidateEnumeratedArgumentsAttribute	279
ArgumentTransformationAttribute	279
<b>Adding Attributes to Dynamic Parameters at Runtime</b>	<b>280</b>
<b>ValidateScriptAttribute</b>	<b>281</b>
<b>Appendix D: Provider Base Classes and Overrides/Interfaces</b>	<b>283</b>
<b>CmdletProvider</b>	<b>283</b>
<b>DriveCmdletProvider</b>	<b>287</b>
<b>ItemCmdletProvider</b>	<b>288</b>
<b>ContainerCmdletProvider</b>	<b>290</b>
<b>NavigationCmdletProvider</b>	<b>294</b>
<b>IContentCmdletProvider</b>	<b>295</b>
<b>IContentReader</b>	<b>296</b>
<b>IContentWriter</b>	<b>297</b>
<b>IPropertyCmdletProvider</b>	<b>297</b>
<b>IDynamicPropertyCmdletProvider</b>	<b>298</b>
<b>Appendix E: Core Cmdlets for Provider Interaction</b>	<b>303</b>
<b>Drive-Specific Cmdlets</b>	<b>303</b>
<b>Item-Specific Cmdlets</b>	<b>303</b>
<b>Container-Specific Cmdlets</b>	<b>304</b>
<b>Property-Specific Cmdlets</b>	<b>304</b>
<b>Dynamic Property Manipulation Cmdlets</b>	<b>305</b>
<b>Content-Related Cmdlets</b>	<b>305</b>
<b>Security Descriptor-Related Cmdlets</b>	<b>305</b>
<b>Index</b>	<b>307</b>