Avant-propos

1. Objectifs du livre 31
2. À qui s'adresse ce livre ? 31
3. Approche du livre 32
4. Connaissances préliminaires 32
5. Comment utiliser ce livre ? 33
6. Remerciements 33

Chapitre 1
Scripting et automatisation

1. Automatisation : principe et intérêt 35
   1.1 Qu'est-ce que l'automatisation ? 35
   1.2 Automatisation dans l'informatique 35
2. Scripting et programmation : les alliés de l'automatisation 37
   2.1 La programmation 37
   2.2 Le scripting 37
   2.3 Langage de script vs langage de programmation 38
      2.3.1 Langage de programmation (ou langage compilé) 38
      2.3.2 Langage de script (ou langage interprété) 39
3. Pourquoi choisir PowerShell ? 39

Chapitre 2
Histoire de PowerShell

1. Introduction 41
2. Qu'est-ce qu'un shell ? 41
3. DOS, MS-DOS, CMD 42
4. VBScript (VBS) 42
5. Monad (MSH) : prémices de PowerShell 42
   5.1 Origine du nom de code Monad 42
   5.2 Pourquoi Monad ? 43
6. De PowerShell 1.0 à PowerShell 5 44

7. La suite de PowerShell 48
7.1 PowerShell 6.0 à 6.2 48
7.2 PowerShell 7 48

Chapitre 3
Comment apprendre PowerShell ?

1. Par où commencer ? 51
2. Transformez vos vieux scripts 54
3. La clé de la réussite : l'autoformation 54
4. Maîtrisez l'anglais technique 55
5. Automatisez votre quotidien 55
6. Ne renoncez jamais 56
7. Aidez-vous de l'intelligence artificielle 56
7.1 Générer un script complet 56
7.2 Apprendre PowerShell 58
7.3 Déboguer ou améliorer un script existant 59
7.4 Commenter un script 61

Chapitre 4
Comment s'améliorer ?

1. Testez, faites des erreurs, recommencez 63
2. Ressources sur Internet 64
3. Personnes à suivre pour apprendre 65

Chapitre 5
Quelques bonnes pratiques

1. Toujours vérifier l'aide proposée par PowerShell 67
2. Get-Member : comprendre ce que contient un objet 68
2.1 Fonctionnement 68
2.2 Lister uniquement les propriétés 69
2.3 Lister uniquement les méthodes 69

3. Get-Command pour trouver la bonne commande 69
4. Commenter son code 71
4.1 Que signifie commenter son code ? 71
4.2 Pourquoi est-ce important ? 72
4.3 Comment commenter son code ? 73
5. Indenter son code 74
5.1 Qu'est-ce que l'indentation ? 74
5.2 Indentation en exemple 74
5.3 Pourquoi indenter le code ? 75
5.4 Règles d'indentation 75
6. Utiliser la saisie semi-automatique ou l'autocomplétion 79
6.1 Pourquoi ? 79
6.2 Fonctionnement 79
6.3 Comment l'utiliser ? 79
7. Majuscules ou minuscules ? 80
8. Ne pas utiliser d'alias 81
8.1 Qu'est-ce qu'un alias ? 81
8.2 Pourquoi ne pas utiliser d'alias ? 82
9. Favoriser la simplicité à la complexité 83
10. Limiter la longueur des lignes 84
11. Éviter les points-virgules à la fin de vos lignes 84
11.1 À quoi sert le caractère point-virgule ? 84
11.2 Pourquoi éviter d'utiliser le point-virgule ? 85
12. Des espaces entre paramètres et opérateurs 85
13. Relire son code 85
14. PSScriptAnalyzer : un moyen d'analyser son code 86
14.1 Qu'est-ce que le module PSScriptAnalyzer ? 86
14.2 Comment l'installer ? 86
14.3 Les règles que le module va vérifier 86
14.4 Analyse d'un script 89
14.5 Analyse d'un code 91
14.6 Formater son code 92

Chapitre 6
Premiers pas avec PowerShell

1. Introduction 95
2. Prise en main de la console PowerShell 96
   2.1 PowerShell 32 ou 64 bits ? 96
   2.2 PowerShell 32 ou 64 bits, quelles sont les différences ? 97
   2.3 Démarrer la console 32 ou 64 bits 100
   2.4 Déterminer l'architecture utilisée 101
   2.5 Connaître la version de PowerShell 103
   2.6 PowerShell en utilisateur ou administrateur 104
   2.7 Configurer la console PowerShell 106
   2.8 Les raccourcis-clavier 112
3. Prise en main de PowerShell ISE 112
   3.1 Qu'est-ce que PowerShell ISE ? 112
   3.2 PowerShell ISE x64 ou x86 ? 113
   3.3 Raccourcis-clavier dans PowerShell ISE 114

Chapitre 7
Les outils pour éditer PowerShell

1. Introduction 117
2. Windows PowerShell ISE 117
   2.1 Barre de menu 119
   2.2 La partie script 120
   2.3 La partie commandes 122
   2.4 La partie console 123
3. Notepad++ 124
4. Visual Studio Code 125
5. PowerShell Studio 128

Chapitre 8
PowerShell, un langage de cmdlets

1. Qu'est-ce qu'une commande ? 131
2. Qu'est-ce qu'une cmdlet ? 132
3. Que se passe-t-il derrière ? 132
   3.1 Cmdlets et bibliothèques 132
   3.2 Cmdlets et bibliothèques en exemples 132
   3.3 Identifier la bibliothèque utilisée par une cmdlet 134
4. Syntaxe d'une cmdlet 135
   4.1 La syntaxe Verbe + Nom 135
   4.2 Qu'est-ce qu'un verbe ? 136
   4.3 Qu'est-ce qu'un nom ? 140
5. Get-command : lister les cmdlets 140
6. Les paramètres 142
   6.1 Qu'est-ce qu'un paramètre ? 142
   6.2 Syntaxe d'un paramètre 143
   6.3 Les types de paramètres 143
   6.4 Lister les paramètres disponibles 143
   6.5 Comportement d'un paramètre 145
7. Les alias de cmdlets 151
8. Show-Command : visualiser une cmdlet et ses paramètres 153
9. PowerShell et les modes d'analyse syntaxique 154
   9.1 Analyse rapide pour comprendre 154
   9.2 Le mode expression 155
   9.3 Le mode commande 155
   9.4 Mode expression vs mode commande en exemples 156

Chapitre 9
PowerShell et les scripts

1. Introduction 157
2. Qu'est-ce qu'un script ? 157
3. Format d'un script 158

4. Créer un script 160
4.1 Notre script 161
4.2 Création depuis Notepad++ 162
4.3 Création depuis PowerShell ISE 164

Chapitre 10
Personnalisation du profil

1. Qu'est-ce qu'un profil PowerShell ? 167
2. Les fichiers de profils 167
2.1 Localisation du profil 168
2.2 Ordre d'application des profils 169
2.3 Windows PowerShell 170
2.4 PowerShell 170
3. Création d'un profil 171
4. Modification d'un profil 172
4.1 Le prompt 172
4.2 Le profil 173
4.2.1 Modification des couleurs 173
4.2.2 PowerShell 7 175
4.3 Modification d'un profil à l'aide d'un module 176

Chapitre 11
Exécution de scripts

1. Introduction 181
2. Quel format de fichier ? 181
3. La stratégie d'exécution 183
3.1 Exécution par défaut 183
3.2 Qu'est-ce que la stratégie d'exécution ? 184
3.3 Stratégies d'exécution par défaut 185
3.4 Différentes stratégies d'exécution 185
3.5 Étendue de la stratégie d'exécution 187

3.6 Gérer la stratégie d'exécution avec PowerShell 187
3.6.1 Récupérer la stratégie appliquée actuellement 187
3.6.2 Modifier la configuration 189
4. Signer un script 190
4.1 Générer un certificat autosigné 191
4.2 Signer le script 195
4.3 Obtenir les informations d'un script signé 197
4.4 Sécurité des fichiers provenant d'Internet 199
4.4.1 Débloquer un fichier particulier 200
4.4.2 Débloquer le contenu d'un dossier 200

Chapitre 12
L'aide PowerShell, un précieux allié

1. Get-Help : pourquoi ? 201
2. Comment l'utiliser ? 202
3. Exemples d'utilisation 206
3.1 Obtenir de l'aide simple 206
3.2 Obtenir des exemples 207
3.3 Obtenir les détails d'une cmdlet 209
3.4 Obtenir l'aide en ligne 210
4. Télécharger la dernière mise à jour de l'aide 212

Chapitre 13
Le pipeline

1. Introduction 213
2. Principe et fonctionnement du pipeline 213
2.1 Qu'est-ce que le pipeline ? 213
2.2 Où se trouve le pipeline ? 214
2.3 L'enchaînement des commandes 214
3. Comprendre le pipeline en exemple 215
4. Schématiser un pipeline 218

5. La liaison des paramètres et objets 219
5.1 Principe 219
5.2 Connaître les paramètres utilisant le pipeline 219

Chapitre 14
La gestion des objets

1. Qu'est-ce qu'un objet ? 223
2. Récupérer la structure d'un objet 225
2.1 Comment faire ? 225
2.2 Les propriétés 227
2.3 Les méthodes 229
2.4 Les évènements 231
3. Quelles cmdlets utiliser pour gérer les objets ? 231
4. Filtrer les objets 234
4.1 Filtrer le contenu 234
4.2 Sélectionner les propriétés à afficher 235
5. Trier des objets 237
6. Créer un objet 238
6.1 Création d'un objet .NET ou COM 238
6.1.1 Objets .NET 239
6.1.2 Objets COM 240
6.2 Création d'objets personnalisés 244
6.2.1 Syntaxe 1 : New-Object 245
6.2.2 Syntaxe 2 : PSCustomObject 245
7. Ajouter ou supprimer des propriétés 246
7.1 Contenu de notre objet 246
7.2 Ajouter une propriété dans un objet 247
7.3 Supprimer une propriété d'un objet 248
8. Comparer des objets 249

Chapitre 15
Les opérateurs

1. Qu'est-ce qu'un opérateur ? 253
2. Les opérateurs arithmétiques 253
   2.1 L'addition. 254
   2.2 La soustraction 254
   2.3 La multiplication 255
   2.4 La division 255
   2.5 Le modulo ou reste 255
3. Les opérateurs de comparaison 256
   3.1 Les opérateurs d'égalité. 258
   3.2 Les opérateurs de correspondance 260
4. Les autres opérateurs 263
   4.1 Le pipeline 263
   4.2 L'affectation 263
   4.3 L'opérateur d'appel 263

Chapitre 16
Les variables

1. Qu'est-ce qu'une variable ? 265
   1.1 Comment déclarer une variable ? 265
   1.2 Afficher le contenu d'une variable 267
   1.3 Afficher le type d'une variable 269
   1.4 Définir le type d'une variable 270
2. Les variables d'environnement 271
3. Les variables automatiques 274
4. La portée des variables 279
   4.1 Qu'est-ce qu'une portée ? 279
   4.2 Comment définir la portée d'une variable ? 279
   4.3 Portée globale 280
      4.3.1 À quoi correspond-elle ? 280
      4.3.2 Déclaration de la portée 280
      4.3.3 Jusqu'où va cette portée ? 280

4.3.4 Comprendre son fonctionnement 280
4.4 Portée script 282
4.4.1 À quoi correspond-elle ? 282
4.4.2 Déclaration de la portée 282
4.4.3 Jusqu'où va cette portée ? 283
4.4.4 Comprendre son fonctionnement 283
4.5 Portée locale 284
4.5.1 À quoi correspond-elle ? 284
4.5.2 Déclaration de la portée 284
4.5.3 Jusqu'où va cette portée ? 285
4.5.4 Comprendre son fonctionnement 285

Chapitre 17
Les conditions

1. Qu'est-ce qu'une condition ? 287
2. Pourquoi les utiliser ? 288
3. Les différentes instructions 289
3.1 Condition simple : If 289
3.1.1 Fonctionnement 289
3.1.2 Structure 289
3.1.3 La condition If en exemple 289
3.2 Si/Sinon : If/Else 291
3.2.1 Fonctionnement 291
3.2.2 Structure 291
3.2.3 La condition If/Else en exemples 291
3.3 Si/Sinon si/Sinon : If/ElseIf/Else 293
3.3.1 Fonctionnement 293
3.3.2 Structure 293
3.4 Switch 294
3.4.1 Fonctionnement 294
3.4.2 Structure 294
3.4.3 L'instruction Switch en exemple 295

Chapitre 18
Les boucles

1. Qu'est-ce qu'une boucle ? 297
2. La boucle For 298
   2.1 Signification 298
   2.2 Utilisation 298
3. La boucle Foreach 299
   3.1 Signification 299
   3.2 Utilisation 300
      3.2.1 Première syntaxe 300
      3.2.2 Seconde syntaxe 301
4. La boucle While 302
   4.1 Signification 302
   4.2 Utilisation 302
5. La boucle Do-While 303
   5.1 Signification 303
   5.2 Utilisation 303
6. La boucle Do-Until 304
   6.1 Signification 304
   6.2 Utilisation 305
7. Comparaison des deux boucles 307
   7.1 Boucle traditionnelle ForEach 307
   7.2 Boucle ForEach parallélisée 308
   7.3 Optimisation des boucles avec la parallélisation 310
   7.4 Deux exemples au quotidien 312
      7.4.1 Espace disque libre 312
      7.4.2 Virtualisation 313

Chapitre 19
Entrées et sorties utilisateur

1. Introduction 315
2. Affichage d'informations 316
   - 2.1 Comment faire ? 316
   - 2.2 Write-Host 317
   - 2.3 Write-Output 318
   - 2.4 Write-Host ou Write-Output ? 319
     - 2.4.1 Envoi du texte 319
     - 2.4.2 Couleur de texte et fond 321
     - 2.4.3 L'avis du créateur de PowerShell 321
3. Affichage d'alertes 321
4. Saisies utilisateur 322
   - 4.1 Saisie et lecture d'un texte simple 323
   - 4.2 Saisie et lecture d'une valeur à sécuriser 324
5. Création d'un menu 325
   - 5.1 Afficher du contenu 325
   - 5.2 Récupérer la saisie 327
   - 5.3 Coder l'action exécutée en fonction du choix 328
6. Des menus en exemples 334
   - 6.1 Menu dépannage de poste 334
     - 6.1.1 Objectif 334
     - 6.1.2 Composition du menu 334
     - 6.1.3 Création du menu 335
   - 6.2 Menu gestion Active Directory 337
     - 6.2.1 Objectif 337
     - 6.2.2 Composition du menu 337
     - 6.2.3 Création du menu 338

Chapitre 20
Les cmdlets indispensables

1. Test-Path : tester l'existence d'un élément 345
   1.1 Que fait cette cmdlet ? 345
   1.2 Comment l'utiliser ? 346
   1.3 Exemples d'utilisation 346
      1.3.1 Tester l'existence d'un fichier ou d'un dossier 346
      1.3.2 Tester les éléments les plus récents 347
      1.3.3 Tester les éléments les plus anciens 347
      1.3.4 Tester l'existence d'une clé de registre 348
      1.3.5 Tester l'existence d'un certificat 348
2. New-Item : créer un nouvel élément 350
   2.1 Que fait cette cmdlet ? 350
   2.2 Comment l'utiliser ? 350
   2.3 Exemples d'utilisation 351
      2.3.1 Créer un dossier 351
      2.3.2 Créer un fichier 352
      2.3.3 Tester l'existence puis créer un élément 352
      2.3.4 Créer une clé de registre 354
3. Get-ChildItem : lister du contenu 355
   3.1 Que fait cette cmdlet ? 355
   3.2 Comment l'utiliser ? 355
   3.3 Exemples d'utilisation 356
      3.3.1 Lister un simple contenu 357
      3.3.2 Lister le contenu des dossiers et sous-dossiers 358
      3.3.3 Lister uniquement les fichiers ou dossiers 358
      3.3.4 Exclure certaines extensions de fichier 359
      3.3.5 Lister uniquement certains types de fichier 360
      3.3.6 Lister les éléments du registre 362
4. Remove-Item : supprimer un élément 363
   4.1 Que fait cette cmdlet ? 363
   4.2 Comment l'utiliser ? 364

4.3 Exemples d'utilisation 364
4.3.1 Suppression simple d'un fichier/dossier 364
4.3.2 Supprimer uniquement certains fichiers 366
4.3.3 Supprimer tout sauf certains fichiers 367
5. Rename-Item : renommer un élément 367
5.1 Que fait cette cmdlet ? 367
5.2 Comment l'utiliser ? 367
5.3 Exemples d'utilisation 368
5.3.1 Renommer un fichier ou un dossier 368
5.3.2 Renommer plusieurs éléments 368
6. Copy-Item 371
6.1 Que fait cette cmdlet ? 371
6.2 Comment l'utiliser ? 371
6.3 Exemples d'utilisation 372
6.3.1 Copier un fichier ou un dossier 372
6.3.2 Copier un dossier et tout son contenu 373
6.3.3 Copier certains fichiers uniquement 374
6.3.4 Copier tout sauf certains éléments 374
7. Get-Content 375
7.1 Que fait cette cmdlet ? 375
7.2 Comment l'utiliser ? 375
7.3 Exemples d'utilisation 375
7.3.1 Lister le contenu d'un fichier texte 375
7.3.2 Récupérer le nombre de lignes d'un fichier texte 376
7.3.3 Afficher la valeur de la première ligne 376
7.3.4 Afficher la valeur de la dernière ligne 377

Chapitre 21
De CMD à PowerShell

1. Introduction 379
2. Équivalences entre CMD et PowerShell 380
   2.1 Équivalence de commandes 380
      2.1.1 Commandes de gestion des fichiers/dossiers 380
      2.1.2 Commandes réseau. 381
      2.1.3 Commandes système 381
   2.2 Les variables 382
      2.2.1 En langage CMD/batch 382
      2.2.2 En PowerShell 382
   2.3 Test d'existence de contenu 383
      2.3.1 En batch 383
      2.3.2 En PowerShell 383
   2.4 Les conditions 383
      2.4.1 En batch 383
      2.4.2 En PowerShell 384
      2.4.3 Les opérateurs 385
   2.5 Les boucles. 385
      2.5.1 En batch 385
      2.5.2 En PowerShell 386
3. De CMD à PowerShell en exemple 386
   3.1 Notre script de test 386
   3.2 Le script en batch 387
   3.3 Le script en PowerShell 387

Chapitre 22
De VBS à PowerShell

1. Introduction 389
2. PowerShell vs VBScript 389

3. Équivalences entre VBS et PowerShell 390
3.1 Équivalence de commandes 390
3.1.1 Commandes de gestion des fichiers/dossiers 390
3.1.2 Commandes système 391
3.2 Les variables 391
3.2.1 En VBScript 391
3.2.2 En PowerShell 392
3.3 Test d'existence de contenu 392
3.3.1 En VBScript 392
3.3.2 En PowerShell 393
3.4 Les conditions 393
3.4.1 En VBScript 393
3.4.2 En PowerShell 394
3.4.3 Les opérateurs 395
3.5 Les boucles 396
3.5.1 En VBScript 396
3.5.2 En PowerShell 396
4. De VBScript à PowerShell en exemple 396
4.1 Notre script de test 396
4.2 Le script en VBScript 397
4.3 Le script en PowerShell 397

Chapitre 23

Les fonctions

1. Intérêt d'une fonction 399
2. Déclaration et appel d'une fonction 404
2.1 Déclaration d'une fonction 404
2.2 Appel d'une fonction 404
3. Fonctions et paramètres 405
3.1 Qu'est-ce qu'un paramètre de fonction ? 405
3.2 Déclaration d'un paramètre 405
3.3 Appel d'une fonction avec paramètres 406

3.4 Contrôle des paramètres 409
3.4.1 Mandatory : paramètre obligatoire 409
3.4.2 ValidateSet : valider un jeu de valeurs précis 410
3.4.3 ValidateNotNullOrEmpty : paramètre non nul ou vide 413
3.4.4 ValidateLength : longueur du paramètre 414
3.4.5 Position de paramètre. 415
3.4.6 Accepter une valeur dans le pipe 416
3.5 Typer nos paramètres. 417
3.5.1 Que signifie typer un paramètre ?. 417
3.5.2 Chaîne de caractères. 417
3.5.3 Entier 418
3.5.4 Switch 418
3.6 Les paramètres communs. 420
4. Pour aller plus loin : les modules 422

Chapitre 24
Les modules

1. Qu'est-ce qu'un module ?. 425
2. Lister les modules existants 426
2.1 Lister les modules chargés dans la session active 426
2.2 Lister tous les modules disponibles 427
2.3 Lister les modules installés au travers de la Gallery 427
2.4 Tester la présence d'un module 428
3. Lister le contenu d'un module 429
4. La PowerShell Gallery. 430
4.1 Qu'est-ce que c'est ? 430
4.2 Installer un module/script 432
4.2.1 Installer dans la dernière version 432
4.2.2 Installer une version précise. 434
4.3 Mettre à jour un module 435
5. Supprimer un module. 435

6. Créer un module 436
6.1 Fonction de base 436
6.2 Création du module 439
6.3 Import du module depuis un autre poste 440
6.4 Publication du module 442
6.5 Installation du module 443

Chapitre 25

Manipuler les chaînes de caractères

1. Les bases 445
1.1 Qu'est-ce qu'une chaîne de caractères ? 445
1.2 Qu'est-ce que le code ASCII ? 446
1.3 Chaîne de caractères en exemple 446
1.4 Distinguer une chaîne de caractères avec PowerShell 447
2. Méthodes disponibles pour gérer les chaînes 448
3. Manipulation des chaînes avec les méthodes 450
3.1 Compter le nombre de caractères 450
3.2 Passer le contenu en majuscules 450
3.3 Passer le contenu en minuscules 451
3.4 Remplacer des caractères 451
3.5 Supprimer des caractères 452
3.5.1 Trim : supprimer les espaces en début et fin de chaîne . 453
3.5.2 TrimStart : supprimer des caractères en début de chaîne 454
3.5.3 TrimEnd : supprimer des caractères en fin de chaîne. . . 455
3.6 Split : séparer une chaîne de caractères 456
3.7 Contains : vérifier si une chaîne contient un caractère 458
3.8 Vérifier si une chaîne commence par quelque chose 458
3.9 Vérifier si une chaîne se termine par quelque chose. 459
3.10 Here-String : un bloc de texte en tant que chaîne 459
3.11 Convertir une chaîne en chaîne de caractères sécurisée. 462

Chapitre 26
Quelques astuces

1. Ouvrir rapidement un dossier dans PowerShell 465
2. Récupérer le chemin complet d'un élément depuis l'explorateur Windows 467
3. Historique des commandes passées dans la console. 469
   3.1 Contexte 469
   3.2 Rechercher l'historique dans le fichier texte 469
   3.3 Rechercher dans l'historique avec une cmdlet 471
   3.4 Rechercher dans l'historique avec un raccourci-clavier 473
   3.5 Rechercher dans l'historique en utilisant un script 474
4. Lister les combinaisons de touches disponibles 477
5. Éditer une commande dans la console 481
6. Lister les cmdlets comportant un certain mot 481
7. Lister les paramètres pour une cmdlet dans un menu 482
8. Commenter un bloc de code 483
9. Ajouter un menu contextuel Exécuter en tant qu'administrateur . 485
10. Exécuter automatiquement un script dans Windows Sandbox . . . 489
    10.1 Qu'est-ce que Windows Sandbox ? 489
    10.2 Installation de Windows Sandbox 489
    10.3 Pourquoi exécuter un script dans Windows Sandbox ? 490
    10.4 Installation de l'outil 491
    10.5 Utilisation de l'outil 492
    10.6 Explorer une assembly, library, dll 493

Chapitre 27
La gestion des erreurs

1. Comment gérer les erreurs en PowerShell ? 497
2. Try/Catch et/ou Finally. 497
   2.1 Try/Catch 497
      2.1.1 Explications. 497
      2.1.2 Structure 498

2.1.3 Explication en exemple 498
2.2 Try/Catch/Finally 503
2.2.1 Explication 503
2.2.2 Structure 503
2.2.3 Explication en exemple 503
3. ErrorAction et ErrorActionPreference 504
3.1 Erreur bloquante ou non bloquante 504
3.2 Changer le comportement d'exécution 506
3.2.1 Paramètre ErrorAction 506
3.2.2 Variable $ErrorActionPreference 513
4. Récupérer le message d'erreur 514

Chapitre 28
La gestion des fichiers et des dossiers

1. Tester l'existence d'un fichier ou d'un dossier. 517
1.1 Test d'existence d'un dossier 517
1.2 Test d'existence d'un fichier. 518
1.3 Test d'existence inversé 518
2. Création de dossiers ou de fichiers 518
2.1 Création de dossiers 518
2.2 Création de fichiers 519
3. Suppression de dossiers ou de fichiers. 519

Chapitre 29
Mise en forme de la sortie des objets

1. Qu'est-ce que la mise en forme des objets ? 521
2. Mise en forme basique 522
3. Out-GridView : un tableau interactif et puissant 524
4. Format-Table : mise en forme dans un tableau 526
4.1 Affichage par défaut 526
4.2 Filtrer les colonnes 527
4.3 Masquer l'en-tête des colonnes 527

4.4 Renommer une colonne 528
5. Format-List : affichage sous forme d'une liste 529
6. Sort-Object : trier la liste en fonction d'une propriété 530
7. Group-Object : grouper les objets par propriété 530

Chapitre 30
Manipuler les fichiers avec PowerShell

1. Introduction 533
2. Encodage de caractères et fichiers 533
2.1 Qu'est-ce que l'encodage ? 533
2.2 Pourquoi l'encodage ? 534
2.3 PowerShell et l'encodage 536
3. Manipuler des fichiers texte. 538
3.1 Gestion des fichiers texte 538
3.2 Exporter le contenu d'un objet vers un fichier 539
3.3 Lire le contenu d'un fichier. 540
3.4 Lister le nombre total de lignes d'un fichier 541
3.5 Lire un contenu spécifique 542
3.6 Récupérer le numéro d'une ligne 543
3.7 Afficher une ou des premières lignes spécifiques 544
3.7.1 Afficher la première ligne 544
3.7.2 Afficher la dernière ligne 545
3.8 Ajouter une nouvelle ligne 545
3.9 Modifier ou remplacer une ligne 546
3.10 Modifier des lignes comportant un mot spécifique 547
3.11 Supprimer une ligne comportant un mot spécifique 547
4. Manipuler des fichiers XML. 549
4.1 Qu'est-ce que le XML ?. 549
4.2 Pourquoi utiliser le format XML ?. 549
4.3 Structure d'un fichier XML 551
4.3.1 Le prologue ou en-tête 551
4.3.2 Les balises 551
4.3.3 Les éléments ou nœuds 552

4.3.4 Les attributs 553
4.4 Export de contenu vers un fichier XML 554
4.5 Lire le contenu 555
4.6 Lister les méthodes 557
4.7 Accéder aux différents nœuds 559
4.8 Lister une application par son nom 565
4.9 Ajouter du contenu 567
4.10 Modifier du contenu 570
4.11 Supprimer du contenu 572
5. Manipuler des fichiers CSV 572
5.1 CSV et les délimiteurs 572
5.1.1 Qu'est-ce qu'un délimiteur ? 572
5.1.2 Détecter le délimiteur par défaut sur mon poste 573
5.1.3 Détecter le délimiteur utilisé dans un fichier 573
5.2 Export du contenu vers un fichier CSV 574
5.3 Importer/lire un fichier CSV 578

Chapitre 31

La gestion des dates

1. Introduction 581
2. Cmdlet Get-Date 581
2.1 Vérifier l'aide 582
2.2 Lister les propriétés 584
2.3 Lister les méthodes 589
3. Formatage des dates 592
3.1 Formatage simple 592
3.2 Formatage personnalisé 593
4. Conversion de dates 597
4.1 Convertir une chaîne de caractères en date 597
4.2 Convertir une date en chaîne de caractères 598
4.3 Conversion de dates depuis le registre ou WMI 599
4.3.1 Date d'installation de Windows 599
4.3.2 Date d'installation d'applications 602

5. Comparaison de dates 602
5.1 Calcul d'intervalle entre deux dates. 602
5.2 Comparer deux dates : antérieure ou postérieure. 605

Chapitre 32
PowerShell et WMI

1. Introduction 607
2. WBEM, DMTF, CIM, WMI. 608
2.1 WBEM 608
2.2 DMTF 608
2.3 CIM 608
2.4 WMI 609
3. WMI et PowerShell 610
3.1 Exécution d'une requête 610
3.2 PowerShell et les commandes WMI 611
3.2.1 Fonctionnement des cmdlets WMI 611
3.2.2 Lister les espaces de noms disponibles 614
3.2.3 Lister les classes disponibles 616
3.2.4 Composition d'une classe 618
3.2.5 Cas d'utilisation 620
3.3 PowerShell et les commandes CIM 633
3.3.1 Qu'est-ce que CIM ? 633
3.3.2 Fonctionnement des cmdlets 633
3.3.3 Lister les espaces de noms disponibles 635
3.3.4 Lister les classes disponibles 635

Chapitre 33
PowerShell et les classes

1. Introduction 639
2. Définition d'une classe 639
2.1 Une classe, qu'est-ce que c'est ? 639

2.2 Pourquoi des classes ? 640
2.2.1 Meilleure organisation du code 640
2.2.2 Facilité de réutilisation et de maintenance 640
2.2.3 Meilleure gestion du .NET 640
2.2.4 Utilisation des classes dans nos exemples 641
3. Contenu et création d'une classe 641
3.1 Déclaration d'une classe : Class 641
3.2 Les propriétés 642
3.3 Les méthodes 644
3.3.1 Qu'est-ce qu'une méthode ? 644
3.3.2 Création d'une première méthode 645
3.3.3 Création d'une seconde méthode 647
4. Ajout d'éléments dans une classe 650
4.1 Ce que nous souhaitons 650
4.2 Ajout d'éléments ou instanciation de la classe 650
4.2.1 Que signifie instancier une classe ? 650
4.2.2 Comment instancier une classe ? 650
4.2.3 Paramétrage de l'élément ajouté 651
4.2.4 Utilisation des méthodes 652
5. Visibilité 653
5.1 Qu'est-ce que la visibilité ? 653
5.2 Comment cacher un élément ? 654

Chapitre 34

PowerShell et le registre

1. Introduction 657
2. Le registre Windows 657
2.1 Qu'est-ce que c'est ? 657
2.2 Composition du registre 658
2.3 Comment accéder au registre et le modifier ? 659
3. PowerShell et la base de registre Windows 660
3.1 Accéder à la base de registre 660
3.2 Naviguer dans la base de registre 663

3.3 Lister les éléments du registre 665
3.3.1 Lister les clés 665
3.3.2 Récupérer les valeurs disponibles. 670
3.4 Tester l'existence d'un élément 673
3.5 Créer un élément 674
3.5.1 Créer une nouvelle clé 674
3.5.2 Créer une valeur 676
3.6 Modifier un élément de registre. 678
3.7 Supprimer un élément 680
3.7.1 Supprimer une clé. 680
3.7.2 Supprimer une valeur 680

Chapitre 35
PowerShell et Base64

1. Qu'est-ce que Base64 ? 681
1.1 Définition 681
1.2 Avantages 682
1.3 Inconvénients 683
2. Encoder/décoder en Base64 684
2.1 Encodage depuis le site. 684
2.1.1 Encoder du texte. 684
2.1.2 Encoder un fichier PNG 685
2.1.3 Encoder un exécutable 687
2.2 Décodage depuis le site. 689
2.2.1 Décoder du texte. 689
2.2.2 Décoder un fichier PNG 690
2.2.3 Décoder un exécutable 691
3. Encoder et décoder avec PowerShell 692
3.1 Encodage en Base64 avec PowerShell 692
3.1.1 Encodage de texte. 692
3.1.2 Encodage d'une image 693

3.2 Décodage en Base64 avec PowerShell 694
3.2.1 Décodage d'un texte 694
3.2.2 Décodage d'un fichier : image, exécutable 694

Chapitre 36

Gérer votre système

1. PowerShell et les services Windows 695
1.1 Get-Service : récupérer les informations du service 697
1.2 Restart-Service : redémarrer un service 698
1.3 Stop-Service : arrêter un service 699
1.4 Start-Service : démarrer un service 700
1.5 Suspend-Service : mettre en pause un service 701
2. PowerShell et les processus 701
2.1 Get-Process : lister les processus en cours 702
2.2 Stop-Process : stopper un processus 706
3. PowerShell et le gestionnaire d'évènements 708
3.1 Lister les différentes catégories d'évènements 708
3.2 Exemples d'utilisation 710
4. PowerShell et les tâches planifiées 713
5. PowerShell et le presse-papiers 717
5.1 Get-Clipboard : récupérer le contenu copié 718
5.1.1 Presse-papiers et texte 719
5.1.2 Presse-papiers et fichiers 719
5.1.3 Presse-papiers et image 721
5.2 Set-Clipboard : copier du contenu 725
5.2.1 Copie de texte 725
5.2.2 Copie de dossiers et de fichiers 726
6. Gérer les utilisateurs et groupes locaux 726
6.1 Comment ? 726
6.2 Gérer les utilisateurs 728
6.3 Gérer les groupes 731

7. PowerShell et Windows Update 733
7.1 Comment le faire avec PowerShell ? 733
7.2 Cas d'usage 734
8. PowerShell et les fonctionnalités Windows 737
8.1 Avec quelles commandes PowerShell ? 737
8.2 Cas d'usage 738

Chapitre 37
Débuter avec RegEx

1. Introduction 741
2. Démystifier RegEx, ou tout du moins essayer 742
2.1 Qu'est-ce que RegEx ? 742
2.2 RegEx et syntaxe 742
2.2.1 Les caractères d'échappement 743
2.2.2 Les métacaractères 744
2.2.3 Les quantificateurs 744
2.2.4 Classes et intervalles. 745
3. Exemples rapides. 747
4. Les ressources intéressantes 748
5. Quelques exemples du quotidien. 749
5.1 Validation d'une adresse e-mail 749
5.2 Saisie de nom d'ordinateur. 750
5.3 Vérification d'une adresse IP valide (IPv4) 751
5.4 Validation d'un numéro de téléphone portable 752
5.5 Validation d'un numéro de sécurité sociale 752
5.6 Validation d'une plaque d'immatriculation française. 753
6. PowerShell et RegEx. 754
6.1 Utiliser RegEx avec PowerShell 754
6.1.1 -Match et -NotMatch. 754
6.1.2 Classe [regex] 755
6.2 Nos exemples avec PowerShell 757
6.2.1 Opérateur -Match. 757
6.2.2 Classe [regex] 758

6.3 Travailler sur des chaînes de caractères. 761
6.3.1 Récupérer la valeur d'une chaîne 761
6.3.2 Extraire du contenu entre deux mots 763

Chapitre 38
PowerShell et Windows Forms

1. Introduction 767
2. Forcer l'aspect visuel moderne 768
3. Windows Forms 769
3.1 De quoi s'agit-il ? 769
3.2 Comment utiliser Windows Forms ? 769
3.3 Boîte de dialogue simple. 770
3.4 Boîte de dialogue de sélection 775
3.4.1 Sélection de dossier. 775
3.4.2 Sélection de fichier 777
4. Notification toast 780

Chapitre 39
Des exemples pour le quotidien

1. Introduction 783
2. Génération de fichier de log 783
3. Fonction de création de clé de registre 784
4. Vérifier la validité d'un lien web 786
5. Télécharger un fichier depuis Internet 787
6. Faire des captures d'écran avec PowerShell. 788
7. Zipper et dézipper 789
8. Créer un journal d'évènements et ajouter un évènement 789
9. Exporter un journal d'évènements. 792
10. Générer un GUID 793
11. Générer un mot de passe 794
12. Chercher une chaîne dans l'observateur d'évènements 798

13. Identifier les comptes administrateurs locaux 799
14. Obtenir la date de dernier redémarrage 800
15. Vérifier qu'un script s'exécute en administrateur 803
16. Redémarrer ou arrêter un poste de travail 804
16.1 Redémarrer un poste 804
16.2 Arrêter un ordinateur 806

Chapitre 40

PowerShell pour gérer vos environnements

1. Introduction 807
2. Les risques 808
3. PowerShell et Active Directory 808
3.1 Active Directory, qu'est-ce que c'est ? 808
3.2 Comment interagir avec PowerShell ? 809
3.3 Installer le module approprié 809
3.3.1 Sur Windows Server 809
3.3.2 Sur Windows 10 809
3.4 Que contient ce module ? 810
3.5 Cas d'usage 811
4. PowerShell et MECM (Configuration Manager) 819
4.1 MECM, qu'est-ce que c'est ? 819
4.2 Comment interagir avec PowerShell ? 819
4.3 Installer le module approprié 819
4.3.1 Utiliser le module depuis la console MECM 819
4.3.2 Importer directement le module 820
4.4 Que contient ce module ? 820
4.5 Cas d'usage 822

5. PowerShell et Entra ID 824
5.1 Entra ID, qu'est-ce que c'est ? 824
5.2 Comment interagir avec PowerShell ? 825
5.3 Installer le module approprié 825
5.4 Que contient ce module ? 825
5.5 S'authentifier. 828
5.6 Cas d'usage 829
6. PowerShell et Intune 832
6.1 Intune, c'est quoi ? 832
6.2 Comment interagir avec PowerShell ? 832
6.3 Installer le module approprié 832
6.4 Que contient ce module ? 833
6.5 S'authentifier. 834
6.6 Cas d'usage 836

Chapitre 41
S'amuser avec PowerShell

1. PowerShell en musique 839
2. Faire parler PowerShell. 841
3. Récupérer vos coordonnées géographiques 843
4. Afficher l'adresse de localisation du poste. 845
5. Afficher les coordonnées GPS dans une carte 846

Bibliographie

1. Les livres pour approfondir ses connaissances 849
1.1 Pour débuter et approfondir. 849
1.2 Approfondir sur d'autres domaines 850

Index 853

Table des matières

Les éléments à télécharger sont disponibles à l'adresse suivante :
http://www.editions-eni.fr
Saisissez la référence ENI de l'ouvrage **EI2POWFA** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

1. À propos de PowerShell. 17
2. À propos du livre 17

Introduction

1. Pourquoi automatiser ? 21
2. Infrastructure as Code et cloud 22
3. L'entrée dans l'open source 22
4. Le poids d'une communauté 23
5. À propos de l'auteur 23

Chapitre 1
PowerShell et la gestion des versions

1. Introduction. 25
   - 1.1 Les versions en détail 26
   - 1.2 GitHub 27
   - 1.3 .NET Core 27
2. Installation de PowerShell Core 28
   - 2.1 Téléchargement des sources depuis GitHub 28
   - 2.2 Installation sous Windows. 29
   - 2.3 Installation sous Linux 31
3. Gestion des mises à jour 32
   - 3.1 Sous Windows 33
   - 3.2 Sous Linux. 33

4. Les nouveautés, dépréciation et limitations de PowerShell 7 33
4.1 Nouveautés 33
4.2 Dépréciation 37
4.3 Limitations 37
5. Développer en PowerShell avec Visual Studio Code 38
5.1 Installation de Visual Studio Code 38
5.2 L'interface de Visual Studio Code 39
5.3 Extension PowerShell 40
5.4 Complément d'extension 41
5.5 Le débogueur 42
5.6 Utilisation de Git 47
6. Scripter en fonction de l'environnement 49
7. Fonctionnalités expérimentales 50

Chapitre 2
Fonctions avancées

1. Introduction 53
2. Structure 54
2.1 Param 54
2.1.1 Attribut Parameter 55
2.1.2 Attributs 59
2.1.3 Attributs de validation 60
2.2 CmdletBinding() 73
2.3 Type de sortie (OutputType) 82
2.4 Begin/Process/End/Clean 83
3. Mise en situation 84
3.1 Jeux de paramètres 84
3.2 Pipeline 87
3.2.1 Déterminer les commandes qui acceptent le pipeline 87
3.2.2 Quand une commande n'accepte pas le pipeline 91
3.2.3 L'implémentation du pipeline 92
3.2.4 ValueFromPipeline 93
3.2.5 ValueFromPipelineByPropertyName 97

4. Paramètre Dynamique 99
4.1 Déclaration 100
4.2 Nouvelle collection d'attributs 100
4.3 Définition et publication du paramètre dynamique 101
4.4 En résumé 102
4.5 Aller plus loin 103
4.5.1 Logique d'affichage 103
4.5.2 Ajout d'attribut de validation 104
4.6 Limitations et cas d'utilisation 106
5. ArgumentCompleter et Dynamic ValidateSet 106
5.1 ArgumentCompletions 106
5.2 ArgumentCompleter 107
5.2.1 Structure 107
5.2.2 Ajout de la saisie utilisateur 110
5.2.3 Utilisation de PSBoundParameter 111
5.2.4 Utiliser une fonction 114
5.3 Dynamic ValidateSet 117
6. Register-ArgumentCompleter 117
7. Splatting 120

Chapitre 3
Création de modules

1. Introduction 123
2. Module dynamique 124
3. Module binaire 127
4. Module script 128
4.1 Conversion d'un script en module 130
4.2 Convention de nommage 130
4.3 Internationalisation et partage du module 132
4.3.1 Structure et nommage des dossiers 132
4.3.2 Format des données et contenu des fichiers 133
4.3.3 La commande Import-LocalizedData 134
4.3.4 Gestion de l'erreur en cas de fichier absent 136
4.3.5 Section Data 136

5. Manifeste de module 137
5.1 Création du manifeste 138
5.2 Gestion des versions 145
5.3 Gestion des versions du framework .NET 147
5.4 Gestion du contenu additionnel 148
5.5 Test de conformité d'un manifeste 148
6. Exportation des membres d'un module 150
6.1 La commande Export-ModuleMember 150
6.2 Comportement particulier 151
7. Structure multifichier 154
8. Passer au Build avec Sampler 156
8.1 Création d'un module simple 156
8.2 Ajout d'une fonction 158
8.3 Build du module 159
8.4 Pour la suite 160

Chapitre 4
Gestion des modules et des packages

1. Introduction 161
2. Module PackageManagement 163
2.1 Ajout du premier fournisseur 164
2.2 Recherche d'un package 165
2.3 Installation d'un package 168
2.3.1 Depuis Chocolatey 168
2.3.2 Depuis un MSI ou en MSU 169
2.4 Listing des packages installés 169
2.5 Mise à jour d'un package 170
2.6 Suppression d'un package 170
3. Module PowerShellGet 171
3.1 Mise à jour de PowerShellGet pour le support du TLS 171
3.2 Recherche d'une ressource 172
3.2.1 Versioning 173
3.2.2 Contenu d'une ressource 174
3.3 Installation d'une ressource 176

3.4 Mise à jour d'une ressource 177
3.5 Suppression d'une ressource 177
4. Création d'un dépôt 178
4.1 Via un simple partage réseau 178
4.1.1 PackageManagement 179
4.1.2 PowerShellGet 180
4.1.3 Organisation du partage 183
4.2 Via un site web 183
4.2.1 Installation 183
4.2.2 Création utilisateur 189
4.2.3 Création du premier feed (dépôt) 192
4.2.4 Enregistrement en tant que source et publication d'un module 194

Chapitre 5
Objets personnalisés et types standards

1. Introduction 197
2. Objets personnalisés 198
2.1 Création d'un objet personnalisé 198
2.2 Ajout d'un membre 199
2.2.1 Ajout d'une propriété 200
2.2.2 Ajout d'une méthode 200
2.3 Création d'une collection d'objets 201
2.4 Transformation d'un objet existant en objet personnalisé 202
3. Types standards 204
3.1 Ajout de membre via un fichier .types.ps1xml 205
3.1.1 Fichier .types.ps1xml 206
3.1.2 Ajout d'une propriété 207
3.1.3 Ajout d'une méthode 209
3.2 Ajout de membre seulement avec Update-TypeData 210
3.2.1 Ajout d'une propriété 211
3.2.2 Ajout d'une méthode 211
3.3 Pour aller plus loin 211

4. Formatage 213
4.1 Fichiers de formatage par défaut 214
4.2 Objet personnalisé 216
4.2.1 Personnalisation du type 216
4.2.2 Application d'un fichier de formatage 218
4.3 Type standard 220
4.3.1 Modification du fichier de formatage par défaut 222
4.3.2 Application du fichier de formatage 226

Chapitre 6
Classes

1. Introduction 229
2. Création d'une classe 230
2.1 Mot-clé Class 230
2.2 Propriété 231
2.3 Méthode 232
2.4 Constructeur 233
2.5 Visibilité 235
3. Surcharge de méthode (overload) 236
4. Statique 238
5. Portée des variables 241
5.1 Propriété 242
5.2 Méthode 244
5.2.1 Globale 244
5.2.2 Script 245
6. Héritage 247
7. Création enum 249
7.1 Mot-clé enum 249
7.2 Manipulation des valeurs 250
8. Formatage 251
9. Intégration dans un module 251
10. Mot-clé Using 252

11. Utilisation des classes sur la validation de paramètre 253
11.1 ValidateSet Dynamique 253
11.2 ArgumentCompleter 255

Chapitre 7
Jobs et parallélisation

1. Introduction. 259
2. Travaux en arrière-plan locaux (BackgroundJob) 260
2.1 Création d'un premier job 261
2.2 Récupération du résultat d'un job 264
2.3 Arrêt d'un job 265
2.4 Suppression d'un job. 267
2.5 Erreurs et jobs 268
3. Travaux en arrière-plan basés sur les threads (ThreadJob) 271
3.1 Utilisation des ThreadJob 271
3.2 Comparatif de performance avec les BackgroundJob 273
4. Travaux en arrière-plan distants (RemoteJob) 274
4.1 Lancement d'un job distant 274
4.2 Multiples destinations 275
4.3 Attendre la fin de l'exécution des jobs. 276
5. Travaux planifiés (PSScheduledJob). 277
5.1 Création d'un travail planifié 278
5.2 Mise en place d'un déclencheur (trigger). 280
5.3 Modification des options 281
6. Workflows (PSWorkflowJob) 282
7. ForEach-Object 283
8. PowerShell API 283

Chapitre 8
Workflows

1. Introduction 287
   1.1 Qu'est-ce qu'un workflow ? 288
   1.2 Comment fonctionne-t-il ? 288
2. Création d'un workflow 289
   2.1 Mot-clé workflow 289
   2.2 Gestion des paramètres 290
      2.2.1 Paramètres communs 290
      2.2.2 Bloc param 291
      2.2.3 Paramètres propres aux workflows 292
   2.3 Exécution à travers une session distante 295
   2.4 Exécution en tant que job 296
3. Restrictions 297
   3.1 Restrictions de commande 297
   3.2 Restrictions sur les objets 299
   3.3 Restrictions de langage 299
   3.4 Mot-clé InlineScript 301
      3.4.1 Syntaxe 301
      3.4.2 Portée des variables 303
4. Exécution en parallèle 304
   4.1 Bloc parallel 304
   4.2 Bloc sequence 306
   4.3 Mot-clé Foreach 309
5. Points de synchronisation 310
6. Suspension d'un workflow 313
7. Relance d'un workflow (suite à une suspension ou à un crash) 315
8. Observation du contenu d'un workflow 316
9. Intégration dans un module 317
10. Workflow : pour qui ? pour quoi ? 318

Chapitre 9
Desired State Configuration (DSC)

1. Introduction 319
2. Versions et plateformes supportées par DSC 320
3. Vue d'ensemble de la solution 321
   - 3.1 Terminologie 321
   - 3.2 Considérations générales 322
   - 3.3 Deux modes : PUSH et PULL 322
     - 3.3.1 PUSH 322
     - 3.3.2 PULL 323
   - 3.4 Configuration 323
   - 3.5 Présentation des ressources DSC 325
   - 3.6 Gestionnaire de configuration local (LCM) 326
4. Ressources DSC 326
   - 4.1 Lister des ressources présentes 327
   - 4.2 Ressources intégrées 328
     - 4.2.1 Utilisation générale d'une ressource 328
     - 4.2.2 Cas particulier avec la ressource Script 330
   - 4.3 Extension des possibilités avec la DSC Community 332
     - 4.3.1 Suivi des nouveautés 332
     - 4.3.2 Interrogation de la PowerShell Gallery 333
     - 4.3.3 Installation depuis la PowerShell Gallery 334
     - 4.3.4 Politique de nommage des ressources DSC 335
     - 4.3.5 Ressources personnalisées 336
     - 4.3.6 Utilisation des ressources 336
5. Création d'une première configuration 336
   - 5.1 Création rapide avec ISE 336
   - 5.2 Application d'une configuration 339
   - 5.3 Utilisation de paramètres 340
   - 5.4 Importation d'une ressource 341
   - 5.5 Données de configuration (Configuration Data) 342
     - 5.5.1 Définition d'une configuration environnementale 342
     - 5.5.2 Variables spéciales $ConfigurationData, $AllNodes, $Nodes 344
     - 5.5.3 Prise en compte de la configuration environnementale 347
     - 5.5.4 Résumé 347

6. Configuration du LCM 349
6.1 Configuration courante 349
6.2 Configuration du LCM 350
6.3 Application d'une métaconfiguration 356
7. Fichiers MOF 357
7.1 Fichier MOF de configuration 358
7.2 Fichier MOF de métaconfiguration 359
8. Sécurité et contexte d'exécution (credentials) 360
9. Sécurisation des credentials dans une configuration 361
9.1 Prérequis 361
9.2 Étapes de réalisation 362
9.3 Données de configuration (Configuration Data) 362
9.4 Configuration 363
9.4.1 Écriture des configurations 363
9.4.2 Génération des fichiers MOF 364
9.4.3 Configuration du LCM 366
9.4.4 Application de la configuration 366
9.5 Obtenir l'empreinte d'un certificat de chiffrement 366
10. Configuration d'un PULL Server 367
10.1 Évolution du PULL Server 367
10.2 PULL Server sous partage SMB 367
10.3 PULL Server Web 368
10.3.1 Prérequis 368
10.3.2 Écriture de la configuration du PULL Server 368
10.3.3 Deux modes de configuration pour le LCM 372
10.3.4 Configuration du LCM en fonction de l'ID 373
10.3.5 Configuration du LCM en fonction du nom 376
10.3.6 Forcer une mise à jour de configuration 378
10.4 Déploiement des ressources 378
11. Test de conformité d'un nœud 380
12. Application de la dernière bonne configuration connue 380
13. Principe avancé 381
13.1 Ressource composite 381
13.2 Configuration partielle 383
13.2.1 Conception des configurations partielles 384

13.2.2 configuration du LCM. 386
13.2.3 PUSH d'une configuration partielle 387
13.2.4 PULL d'une configuration partielle. 388
13.2.5 Allier PUSH et PULL 391
14. Écriture d'une ressource personnalisée 392
14.1 Ressource personnalisée avec un fichier MOF 392
14.1.1 Constitution d'une ressource basée sur un schéma MOF 392
14.1.2 Aide à la création de ressources (xDSCResourceDesigner) . . . 393
14.1.3 Génération de la structure du module 394
14.1.4 Écriture du fichier PSM1 396
14.1.5 Déploiement de la ressource 399
14.1.6 Vérification de la prise en compte de la ressource 400
14.2 Ressource personnalisée avec les classes PowerShell 400
14.2.1 Constitution d'une ressource basée sur une classe PowerShell 400
14.2.2 PowerShell ISE à la rescousse 401
14.2.3 Rédaction du fichier PSM1 401
14.2.4 Génération et modification du manifeste 404
14.2.5 Déploiement de la ressource 404
14.3 Utilisation de la ressource personnalisée dans une configuration . . . 405
15. DSC sous Linux 405
15.1 Prérequis 405
15.1.1 Installation du package OMI 406
15.1.2 Installation du package DSC 406
15.2 Ressources DSC pour Linux. 406
15.3 Mise en place d'une session CIM. 407
15.4 Création de la configuration 408
15.5 Application de la configuration 408
15.6 Récupération de la configuration du LCM 409
16. Maintenance et dépannage 410
16.1 Forcer la réapplication d'une configuration. 410
16.2 État d'une configuration DSC 410
16.3 Remise d'un système à son état d'origine 411
16.4 Activation du mode Verbose 412

Chapitre 10
Gestion avancée des fichiers

1. Introduction 413
2. Gestion de l'archivage 414
   2.1 Compress-Archive 414
   2.2 Expand-Archive 415
   2.3 Avant PowerShell 5.1 416
      2.3.1 Compression d'un dossier 416
      2.3.2 Décompression d'une archive 417
3. Lien symbolique 417
   3.1 Fichiers 417
   3.2 Dossiers 418
4. Fichier temporaire 421
5. Édition de fichiers distants 422
6. Copie de fichiers à travers une session PowerShell 427
7. Conversion d'un fichier plat en objet 429
   7.1 Mode basique 429
   7.2 Mode avancé 431

Chapitre 11
Communication avec d'autres applications

1. Introduction 437
2. Les technologies COM. 437
3. Manipulation des objets COM 438
   3.1 Recherche d'objets. 438
   3.2 Création/instanciation d'un objet 439
   3.3 Utilisation d'une application COM. 441
      3.3.1 Internet Explorer 442
      3.3.2 Microsoft Excel 2016 445
      3.3.3 Windows Script Host (WSH) 449

Chapitre 12
Gestion à distance avancée

1. Introduction 453
2. JEA (Just Enough Administration) 453
   - 2.1 Pourquoi JEA ? 454
   - 2.2 Prérequis 454
   - 2.3 Rappel 455
     - 2.3.1 Une session, c'est quoi ? 455
     - 2.3.2 Configuration d'une session (PowerShell 4.0 et versions antérieures) 457
     - 2.3.3 Ce qui change avec PowerShell 5.0 461
   - 2.4 Capacité de rôle 462
     - 2.4.1 Préparation 462
     - 2.4.2 Fichier de capacité de rôle 463
     - 2.4.3 Intégration dans un module 466
     - 2.4.4 Fusion des capacités de rôle 468
   - 2.5 Sessions 470
     - 2.5.1 Fichier de configuration de session 470
     - 2.5.2 Enregistrement d'une session 475
     - 2.5.3 Modification d'une session 477
     - 2.5.4 Utilisation des sessions 478
   - 2.6 Audit 479
     - 2.6.1 Conformité des configurations de session 479
     - 2.6.2 Transcription 481
   - 2.7 Configuration via DSC 483
3. PowerShell Web Access 485
   - 3.1 Prérequis 485
   - 3.2 Installation 486
   - 3.3 Configuration 488
     - 3.3.1 Installation de l'application web 488
     - 3.3.2 Règles d'accès 490
   - 3.4 Utilisations 491
   - 3.5 Déconnexion d'une session 494
   - 3.6 Cas d'usage 499
   - 3.7 Limitations et futur de la fonctionnalité 499

4. Session distante avec SSH 500
4.1 Installation sous Windows 500
4.2 Installation sous Linux 502
4.3 Établissement de la connexion 502

Chapitre 13
Création d'interfaces graphiques

1. Introduction 505
1.1 Pourquoi une interface graphique ? 505
1.2 Vers quoi se tourner ? 506
2. Windows Forms 507
2.1 Présentation des Windows Forms 507
2.2 Réalisation d'une première Form 508
2.3 Ajout d'une action à une Form 511
2.3.1 Événements 511
2.3.2 Menus 513
2.4 Exemple de contrôle 516
2.5 Création de graphiques avec MS Chart Controls 521
3. Windows Presentation Foundation 525
3.1 Création d'une interface sur Visual Studio 526
3.2 Gestion des contrôles et de leurs événements 530
3.3 Création de graphiques évolués avec LiveCharts 534
3.4 Changement de thème avec MahApps 539
4. Mise à jour d'une GUI via des runspaces 545

Chapitre 14
Collecte de données de performances

1. Pourquoi et comment collecter des données de performances ? 551
2. Windows Performance Logs and Alerts (PLA) 552
3. Jeu de commandes PowerShell 553
3.1 Listing des catégories de compteurs de performances 553
3.2 Détermination du bon compteur 555
3.3 Accès aux données de compteurs 560

3.3.1 Contrôle du nombre de jeux de données et de la fréquence . . 561
3.3.2 Collecte des données sur des ordinateurs distants 563
3.4 Utilisation des jobs pour les tâches longues 565
3.5 Sauvegarde des données de performances sur disque 565
3.5.1 Exportation des données de performances au format binaire (BLG) 566
3.5.2 Exportation des données de performances au format Excel (CSV) 568
3.6 Manipulation d'un fichier d'export de données de performances . . . 569
4. Collecte de données à l'échelle de l'entreprise 576
4.1 Création du fichier de configuration XML 578
4.2 Création et déploiement d'un ensemble de collecteurs de données . . 585
4.2.1 Réalisation via logman 585
4.2.2 Réalisation via l'objet COM PLA 586
4.3 Gestion des ensembles de collecteurs de données 587
4.3.1 Interrogation des ensembles de collecteurs de données existants 587
4.3.2 Démarrage d'un ensemble de collecteurs de données 589
4.3.3 Arrêt d'un ensemble de collecteurs de données 589
4.3.4 Suppression d'un ensemble de collecteurs de données 590

Chapitre 15
Manipulation d'annuaires

1. Introduction 591
2. Module Active Directory 592
2.1 Prérequis 592
2.2 Recherche d'objets 593
2.2.1 Recherche basée sur l'identité 593
2.2.2 Filtre générique 594
2.2.3 Filtre LDAP 596
2.3 Gestion des unités d'organisation 599
2.3.1 Création d'une unité d'organisation 599
2.3.2 Modification d'une unité d'organisation 600
2.3.3 Suppression d'une unité d'organisation 600

2.4 Gestion des utilisateurs. 600
2.4.1 Affichage de la liste des utilisateurs et de leurs propriétés . . . 601
2.4.2 Création d'un utilisateur 603
2.4.3 Modification du mot de passe et activation d'un objet utilisateur existant 604
2.4.4 Modification d'un attribut 605
2.4.5 Suppression d'un utilisateur 607
2.5 Gestion des groupes 607
2.5.1 Affichage de la liste des groupes 608
2.5.2 Création d'un groupe 610
2.5.3 Gestion des membres 611
2.5.4 Suppression d'un groupe 614
2.6 Utilisation du provider Active Directory. 614
2.6.1 Exploration du provider. 615
2.6.2 Déplacement d'objets. 617
2.6.3 Modification d'objets 618
2.6.4 Utilisation des commandes Active Directory 618
2.6.5 Création d'un PSDrive vers un autre domaine. 618
3. ADSI 619
3.1 Présentation d'ADSI 619
3.2 Confusion avec Active Directory. 619
3.3 Encore utile aujourd'hui ?. 619
3.4 Interaction avec une base locale SAM 620
3.4.1 Généralités 621
3.4.2 Groupes locaux. 622
3.4.3 Utilisateurs locaux 624
3.5 Active Directory Domain Services. 624
3.5.1 Connexion à l'annuaire 624
3.5.2 Connexion à un domaine avec un utilisateur différent. 625
3.5.3 Recherche d'objets 627
4. ADSIPS 633

Index 635