

Avant-propos

1. Objectifs du livre	29
2. À qui s'adresse ce livre ?	29
3. Approche du livre	30
4. Connaissances préliminaires	30
5. Comment utiliser ce livre ?	31
6. Remerciements	31

Chapitre 1

Scripting et automatisation

1. Automatisation : principe et intérêt	33
1.1 Qu'est-ce que l'automatisation ?	33
1.2 Automatisation dans l'informatique	33
2. Scripting et programmation : les alliés de l'automatisation	35
2.1 La programmation	35
2.2 Le scripting	35
2.3 Langage de script vs langage de programmation	36
2.3.1 Langage de programmation (ou langage compilé)	36
2.3.2 Langage de script (ou langage interprété)	37
3. Pourquoi choisir PowerShell ?	37

Chapitre 2

Histoire de PowerShell

1. Introduction	39
2. Qu'est-ce qu'un shell ?	39
3. DOS, MS-DOS, CMD	40
4. VBScript (VBS)	40

2 _____ Débuter avec PowerShell

5. Monad (MSH) : prémices de PowerShell	40
5.1 Origine du nom de code Monad	40
5.2 Pourquoi Monad ?	41
6. De PowerShell 1.0 à PowerShell 5	42
7. La suite de PowerShell	46
7.1 PowerShell 6.0 à 6.2	46
7.2 PowerShell 7	46

Chapitre 3

Comment apprendre PowerShell ?

1. Par où commencer ?	49
2. Transformez vos vieux scripts	52
3. La clé de la réussite : l'autoformation	52
4. Maîtrisez l'anglais technique	53
5. Automatisez votre quotidien	53
6. Ne renoncez jamais	54

Chapitre 4

Comment s'améliorer ?

1. Testez, faites des erreurs, recommencez	55
2. Ressources sur Internet	56
3. Personnes à suivre pour apprendre	57

Chapitre 5

Quelques bonnes pratiques

1. Toujours vérifier l'aide proposée par PowerShell	59
2. Get-Member : comprendre ce que contient un objet	60
2.1 Fonctionnement	60
2.2 Lister uniquement les propriétés	61
2.3 Lister uniquement les méthodes	61
3. Get-Command pour trouver la bonne commande	61
4. Commenter son code	63
4.1 Que signifie commenter son code ?	63
4.2 Pourquoi est-ce important ?	64
4.3 Comment commenter son code ?	65
5. Indenter son code	66
5.1 Qu'est-ce que l'indentation ?	66
5.2 Indentation en exemple	66
5.3 Pourquoi indenter le code ?	67
5.4 Règles d'indentation	67
6. Utiliser la saisie semi-automatique ou l'autocomplétion	71
6.1 Pourquoi ?	71
6.2 Fonctionnement	71
6.3 Comment l'utiliser ?	71
7. Majuscules ou minuscules ?	72
8. Ne pas utiliser d'alias	73
8.1 Qu'est-ce qu'un alias ?	73
8.2 Pourquoi ne pas utiliser d'alias ?	74
9. Favoriser la simplicité à la complexité	75
10. Limiter la longueur des lignes	76
11. Éviter les points-virgules à la fin de vos lignes	76
11.1 À quoi sert le caractère point-virgule ?	76
11.2 Pourquoi éviter d'utiliser le point-virgule ?	77

4 _____ Débuter avec PowerShell

12. Des espaces entre paramètres et opérateurs	77
13. Relire son code	77
14. PSScriptAnalyzer : un moyen d'analyser son code	78
14.1 Qu'est-ce que le module PSScriptAnalyzer ?	78
14.2 Comment l'installer ?	78
14.3 Les règles que le module va vérifier	78
14.4 Analyse d'un script	81
14.5 Analyse d'un code	83
14.6 Formater son code	84

Chapitre 6

Premiers pas avec PowerShell

1. Introduction	87
2. Prise en main de la console PowerShell	88
2.1 PowerShell 32 ou 64 bits ?	88
2.2 PowerShell 32 ou 64 bits, quelles sont les différences ?	89
2.3 Démarrer la console 32 ou 64 bits	92
2.4 Déterminer l'architecture utilisée	93
2.5 Connaître la version de PowerShell	95
2.6 PowerShell en utilisateur ou administrateur	96
2.7 Configurer la console PowerShell	98
2.8 Les raccourcis-clavier	104
3. Prise en main de PowerShell ISE	104
3.1 Qu'est-ce que PowerShell ISE ?	104
3.2 PowerShell ISE x64 ou x86 ?	105
3.3 Raccourcis clavier dans PowerShell ISE	106

Chapitre 7

Les outils pour éditer PowerShell

1. Introduction	109
2. Windows PowerShell ISE	109
2.1 Barre de menu	111
2.2 La partie script	112
2.3 La partie commandes	114
2.4 La partie console	115
3. Notepad++	116
4. Visual Studio Code	117
5. PowerShell Studio	120

Chapitre 8

PowerShell, un langage de cmdlets

1. Qu'est-ce qu'une commande ?	123
2. Qu'est-ce qu'une cmdlet ?	124
3. Que se passe-t-il derrière ?	124
3.1 Cmdlets et bibliothèques	124
3.2 Cmdlets et bibliothèques en exemples	124
3.3 Identifier la bibliothèque utilisée par une cmdlet	126
4. Syntaxe d'une cmdlet	127
4.1 La syntaxe Verbe + Nom	127
4.2 Qu'est-ce qu'un verbe ?	128
4.3 Qu'est-ce qu'un nom ?	132
5. Get-command : lister les cmdlets	132
6. Les paramètres	134
6.1 Qu'est-ce qu'un paramètre ?	134
6.2 Syntaxe d'un paramètre	135
6.3 Les types de paramètres	135
6.4 Lister les paramètres disponibles	135

6 _____ Débuter avec PowerShell

6.5 Comportement d'un paramètre.	137
7. Les alias de cmdlets.	143
8. Show-Command : visualiser une cmdlet et ses paramètres	145
9. PowerShell et les modes d'analyse syntaxique	146
9.1 Analyse rapide pour comprendre.	146
9.2 Le mode expression.	147
9.3 Le mode commande	147
9.4 Mode expression vs mode commande en exemples.	148

Chapitre 9

PowerShell et les scripts

1. Introduction	149
2. Qu'est-ce qu'un script ?	149
3. Format d'un script	150
4. Créer un script	152
4.1 Notre script.	153
4.2 Création depuis Notepad++	154
4.3 Création depuis PowerShell ISE	156

Chapitre 10

Exécution de scripts

1. Introduction	159
2. Quel format de fichier ?	159
3. La stratégie d'exécution	161
3.1 Exécution par défaut	161
3.2 Qu'est-ce que la stratégie d'exécution ?	162
3.3 Stratégies d'exécution par défaut.	163
3.4 Différentes stratégies d'exécution	163
3.5 Étendue de la stratégie d'exécution	165

3.6	Gérer la stratégie d'exécution avec PowerShell.	165
3.6.1	Récupérer la stratégie appliquée actuellement	165
3.6.2	Modifier la configuration.	167
4.	Signer un script.	168
4.1	Générer un certificat autosigné	169
4.2	Signer le script.	173
4.3	Obtenir les informations d'un script signé	175
4.4	Sécurité des fichiers provenant d'Internet	177
4.4.1	Débloquer un fichier particulier	178
4.4.2	Débloquer le contenu d'un dossier	178

Chapitre 11

L'aide PowerShell, un précieux allié

1.	Get-Help : pourquoi ?	179
2.	Comment l'utiliser ?	180
3.	Exemples d'utilisation	184
3.1	Obtenir de l'aide simple	184
3.2	Obtenir des exemples	185
3.3	Obtenir les détails d'une cmdlet	187
3.4	Obtenir l'aide en ligne	188
4.	Télécharger la dernière mise à jour de l'aide	190

Chapitre 12

Le pipeline

1.	Introduction	191
2.	Principe et fonctionnement du pipeline	191
2.1	Qu'est-ce que le pipeline ?	191
2.2	Où se trouve le pipeline ?	192
2.3	L'enchaînement des commandes	192
3.	Comprendre le pipeline en exemple	193

8 _____ Débuter avec PowerShell

4. Schématiser un pipeline	196
5. La liaison des paramètres et objets	197
5.1 Principe	197
5.2 Connaître les paramètres utilisant le pipeline	197

Chapitre 13

La gestion des objets

1. Qu'est-ce qu'un objet ?	201
2. Récupérer la structure d'un objet.	203
2.1 Comment faire ?	203
2.2 Les propriétés	205
2.3 Les méthodes.	207
2.4 Les évènements.	209
3. Quelles cmdlets utiliser pour gérer les objets ?	209
4. Filtrer les objets.	212
4.1 Filtrer le contenu	212
4.2 Sélectionner les propriétés à afficher.	213
5. Trier des objets	215
6. Créer un objet	216
6.1 Création d'un objet .NET ou COM.	216
6.1.1 Objets .NET	217
6.1.2 Objets COM	218
6.2 Création d'objets personnalisés	222
6.2.1 Syntaxe 1 : New-Object.	223
6.2.2 Syntaxe 2 : PSCustomObject	223
7. Ajouter ou supprimer des propriétés.	224
7.1 Contenu de notre objet	224
7.2 Ajouter une propriété dans un objet	225
7.3 Supprimer une propriété d'un objet	226
8. Comparer des objets.	227

Chapitre 14

Les opérateurs

1. Qu'est-ce qu'un opérateur ?	231
2. Les opérateurs arithmétiques	231
2.1 L'addition	232
2.2 La soustraction	232
2.3 La multiplication	233
2.4 La division	233
2.5 Le modulo ou reste	233
3. Les opérateurs de comparaison	234
3.1 Les opérateurs d'égalité	236
3.2 Les opérateurs de correspondance	238
4. Les autres opérateurs	241
4.1 Le pipeline	241
4.2 L'affectation	241
4.3 L'opérateur d'appel	241

Chapitre 15

Les variables

1. Qu'est-ce qu'une variable ?	243
1.1 Comment déclarer une variable ?	243
1.2 Afficher le contenu d'une variable	245
1.3 Afficher le type d'une variable	247
1.4 Définir le type d'une variable	248
2. Les variables d'environnement	249
3. Les variables automatiques	252
4. La portée des variables	257
4.1 Qu'est-ce qu'une portée ?	257
4.2 Comment définir la portée d'une variable ?	257

10 _____ Débuter avec PowerShell

4.3	Portée globale	258
4.3.1	À quoi correspond-elle ?	258
4.3.2	Déclaration de la portée	258
4.3.3	Jusqu'où va cette portée ?	258
4.3.4	Comprendre son fonctionnement	258
4.4	Portée script	260
4.4.1	À quoi correspond-elle ?	260
4.4.2	Déclaration de la portée	260
4.4.3	Jusqu'où va cette portée ?	261
4.4.4	Comprendre son fonctionnement	261
4.5	Portée locale	262
4.5.1	À quoi correspond-elle ?	262
4.5.2	Déclaration de la portée	262
4.5.3	Jusqu'où va cette portée ?	263
4.5.4	Comprendre son fonctionnement	263

Chapitre 16 Les conditions

1.	Qu'est-ce qu'une condition ?	265
2.	Pourquoi les utiliser ?	266
3.	Les différentes instructions	267
3.1	Condition simple : If	267
3.1.1	Fonctionnement	267
3.1.2	Structure	267
3.1.3	La condition If en exemple	267
3.2	Si/Sinon : If/Else	269
3.2.1	Fonctionnement	269
3.2.2	Structure	269
3.2.3	La condition If/Else en exemples	269

3.3	Si/Sinon si/Sinon : If/ElseIf/Else	271
3.3.1	Fonctionnement	271
3.3.2	Structure	271
3.4	Switch	272
3.4.1	Fonctionnement	272
3.4.2	Structure	272
3.4.3	L'instruction Switch en exemple	273

Chapitre 17 Les boucles

1.	Qu'est-ce qu'une boucle ?	275
2.	La boucle For	276
2.1	Signification	276
2.2	Utilisation	276
3.	La boucle Foreach	277
3.1	Signification	277
3.2	Utilisation	278
3.2.1	Première syntaxe	278
3.2.2	Seconde syntaxe	279
4.	La boucle While	280
4.1	Signification	280
4.2	Utilisation	280
5.	La boucle Do-While	281
5.1	Signification	281
5.2	Utilisation	281
6.	La boucle Do-Until	282
6.1	Signification	282
6.2	Utilisation	283

12 _____ Débuter avec PowerShell

Chapitre 18

Entrées et sorties utilisateur

1. Introduction	285
2. Affichage d'informations	286
2.1 Comment faire ?	286
2.2 Write-Host	287
2.3 Write-Output	288
2.4 Write-Host ou Write-Output ?	289
2.4.1 Envoi du texte	289
2.4.2 Couleur de texte et fond	291
2.4.3 L'avis du créateur de PowerShell	291
3. Affichage d'alertes	291
4. Saisies utilisateur	292
4.1 Saisie et lecture d'un texte simple	293
4.2 Saisie et lecture d'une valeur à sécuriser	294
5. Création d'un menu	295
5.1 Afficher du contenu	295
5.2 Récupérer la saisie	297
5.3 Coder l'action exécutée en fonction du choix	298
6. Des menus en exemples	304
6.1 Menu dépannage de poste	304
6.1.1 Objectif	304
6.1.2 Composition du menu	304
6.1.3 Création du menu	305
6.2 Menu gestion Active Directory	307
6.2.1 Objectif	307
6.2.2 Composition du menu	307
6.2.3 Création du menu	308

Chapitre 19

Les cmdlets indispensables

1. Test-Path : tester l'existence d'un élément	315
1.1 Que fait cette cmdlet ?	315
1.2 Comment l'utiliser ?	316
1.3 Exemples d'utilisation	316
1.3.1 Tester l'existence d'un fichier ou d'un dossier	316
1.3.2 Tester les éléments les plus récents	317
1.3.3 Tester les éléments les plus anciens	317
1.3.4 Tester l'existence d'une clé de registre	318
1.3.5 Tester l'existence d'un certificat	318
2. New-Item : créer un nouvel élément	320
2.1 Que fait cette cmdlet ?	320
2.2 Comment l'utiliser ?	320
2.3 Exemples d'utilisation	321
2.3.1 Créer un dossier	321
2.3.2 Créer un fichier	322
2.3.3 Tester l'existence puis créer un élément	322
2.3.4 Créer une clé de registre	324
3. Get-ChildItem : lister du contenu	325
3.1 Que fait cette cmdlet ?	325
3.2 Comment l'utiliser ?	325
3.3 Exemples d'utilisation	326
3.3.1 Lister un simple contenu	327
3.3.2 Lister le contenu des dossiers et sous-dossiers	328
3.3.3 Lister uniquement les fichiers ou dossiers	328
3.3.4 Exclure certaines extensions de fichier	329
3.3.5 Lister uniquement certains types de fichier	330
3.3.6 Lister les éléments du registre	332
4. Remove-Item : supprimer un élément	333
4.1 Que fait cette cmdlet ?	333
4.2 Comment l'utiliser ?	334

14 _____ Débuter avec PowerShell

4.3	Exemples d'utilisation	334
4.3.1	Suppression simple d'un fichier/dossier	334
4.3.2	Supprimer uniquement certains fichiers	336
4.3.3	Supprimer tout sauf certains fichiers	337
5.	Rename-Item : renommer un élément	337
5.1	Que fait cette cmdlet ?	337
5.2	Comment l'utiliser ?	337
5.3	Exemples d'utilisation	338
5.3.1	Renommer un fichier ou un dossier	338
5.3.2	Renommer plusieurs éléments	338
6.	Copy-Item	341
6.1	Que fait cette cmdlet ?	341
6.2	Comment l'utiliser ?	341
6.3	Exemples d'utilisation	342
6.3.1	Copier un fichier ou un dossier	342
6.3.2	Copier un dossier et tout son contenu	343
6.3.3	Copier certains fichiers uniquement	344
6.3.4	Copier tout sauf certains éléments	344
7.	Get-Content	345
7.1	Que fait cette cmdlet ?	345
7.2	Comment l'utiliser ?	345
7.3	Exemples d'utilisation	345
7.3.1	Lister le contenu d'un fichier texte	345
7.3.2	Récupérer le nombre de lignes d'un fichier texte	346
7.3.3	Afficher la valeur de la première ligne	346
7.3.4	Afficher la valeur de la dernière ligne	347

Chapitre 20

De CMD à PowerShell

1. Introduction	349
2. Équivalences entre CMD et PowerShell	350
2.1 Équivalence de commandes	350
2.1.1 Commandes de gestion des fichiers/dossiers	350
2.1.2 Commandes réseau	351
2.1.3 Commandes système	351
2.2 Les variables	352
2.2.1 En langage CMD/batch	352
2.2.2 En PowerShell	352
2.3 Test d'existence de contenu	353
2.3.1 En batch	353
2.3.2 En PowerShell	353
2.4 Les conditions	353
2.4.1 En batch	353
2.4.2 En PowerShell	354
2.4.3 Les opérateurs	355
2.5 Les boucles	355
2.5.1 En batch	355
2.5.2 En PowerShell	356
3. De CMD à PowerShell en exemple	356
3.1 Notre script de test	356
3.2 Le script en batch	357
3.3 Le script en PowerShell	357

16 _____ Débuter avec PowerShell

Chapitre 21

De VBS à PowerShell

1. Introduction	359
2. PowerShell vs VBScript	359
3. Équivalences entre VBS et PowerShell	360
3.1 Équivalence de commandes	360
3.1.1 Commandes de gestion des fichiers/dossiers	360
3.1.2 Commandes système	361
3.2 Les variables	361
3.2.1 En VBScript	361
3.2.2 En PowerShell	362
3.3 Test d'existence de contenu	362
3.3.1 En VBScript	362
3.3.2 En PowerShell	363
3.4 Les conditions	363
3.4.1 En VBScript	363
3.4.2 En PowerShell	364
3.4.3 Les opérateurs	365
3.5 Les boucles	366
3.5.1 En VBScript	366
3.5.2 En PowerShell	366
4. De VBScript à PowerShell en exemple	366
4.1 Notre script de test	366
4.2 Le script en VBScript	367
4.3 Le script en PowerShell	367

Chapitre 22

Les fonctions

1. Intérêt d'une fonction	369
2. Déclaration et appel d'une fonction	374
2.1 Déclaration d'une fonction	374
2.2 Appel d'une fonction	374
3. Fonctions et paramètres	375
3.1 Qu'est-ce qu'un paramètre de fonction ?	375
3.2 Déclaration d'un paramètre	375
3.3 Appel d'une fonction avec paramètres	376
3.4 Contrôle des paramètres	379
3.4.1 Mandatory : paramètre obligatoire	379
3.4.2 ValidateSet : valider un jeu de valeurs précis	380
3.4.3 ValidateNotNullOrEmpty : paramètre non nul ou vide	383
3.4.4 ValidateLength : longueur du paramètre	384
3.4.5 Position de paramètre	385
3.4.6 Accepter une valeur dans le pipe	386
3.5 Typer nos paramètres	387
3.5.1 Que signifie typer un paramètre ?	387
3.5.2 Chaîne de caractères	387
3.5.3 Entier	388
3.5.4 Switch	388
3.6 Les paramètres communs	390
4. Pour aller plus loin : les modules	392

Chapitre 23

Manipuler les chaînes de caractères

1. Les bases	395
1.1 Qu'est-ce qu'une chaîne de caractères ?	395
1.2 Qu'est-ce que le code ASCII ?	396
1.3 Chaîne de caractères en exemple	396

18 _____ Débuter avec PowerShell

1.4 Distinguer une chaîne de caractères avec PowerShell	397
2. Méthodes disponibles pour gérer les chaînes	398
3. Manipulation des chaînes avec les méthodes	400
3.1 Compter le nombre de caractères	400
3.2 Passer le contenu en majuscules	400
3.3 Passer le contenu en minuscules	401
3.4 Remplacer des caractères	401
3.5 Supprimer des caractères	402
3.5.1 Trim : supprimer les espaces en début et fin de chaîne .	403
3.5.2 TrimStart : supprimer des caractères en début de chaîne	404
3.5.3 TrimEnd : supprimer des caractères en fin de chaîne. . .	405
3.6 Split : Séparer une chaîne de caractères	406
3.7 Contains : Vérifier si une chaîne contient un caractère.	408
3.8 Vérifier si une chaîne commence par quelque chose	408
3.9 Vérifier si une chaîne se termine par quelque chose.	409
3.10 Here-String : un bloc de texte en tant que chaîne	409
3.11 Convertir une chaîne en chaîne de caractères sécurisée.	412

Chapitre 24

Quelques astuces

1. Ouvrir rapidement un dossier dans PowerShell	415
2. Récupérer le chemin complet d'un élément depuis l'explorateur Windows	417
3. Historique des commandes passées dans la console.	419
3.1 Contexte	419
3.2 Rechercher l'historique dans le fichier texte	419
3.3 Rechercher dans l'historique avec une cmdlet	421
3.4 Rechercher dans l'historique avec un raccourci-clavier	423
3.5 Rechercher dans l'historique en utilisant un script	424
4. Lister les combinaisons de touches disponibles	427

5. Éditer une commande dans la console	431
6. Lister les cmdlets comportant un certain mot	431
7. Lister les paramètres pour une cmdlet dans un menu	432
8. Commenter un bloc de code	433
9. Ajouter un menu contextuel Exécuter en tant qu'administrateur .	435
10. Exécuter automatiquement un script dans Windows Sandbox . .	439
10.1 Qu'est-ce que Windows Sandbox ?	439
10.2 Installation de Windows Sandbox	439
10.3 Pourquoi exécuter un script dans Windows Sandbox ?	440
10.4 Installation de l'outil ?	441
10.5 Utilisation de l'outil ?	442

Chapitre 25

La gestion des erreurs

1. Comment gérer les erreurs en PowerShell ?	443
2. Try/Catch et/ou Finally	443
2.1 Try/Catch	443
2.1.1 Explications	443
2.1.2 Structure	444
2.1.3 Explication en exemple	444
2.2 Try/Catch/Finally	449
2.2.1 Explication	449
2.2.2 Structure	449
2.2.3 Explication en exemple	449
3. ErrorAction et ErrorActionPreference	450
3.1 Erreur bloquante ou non bloquante	450
3.2 Changer le comportement d'exécution	452
3.2.1 Paramètre ErrorAction	452
3.2.2 Variable \$ErrorActionPreference	459
4. Récupérer le message d'erreur	460

Chapitre 26

La gestion des fichiers et des dossiers

1. Tester l'existence d'un fichier ou d'un dossier. 463
 - 1.1 Test d'existence d'un dossier 463
 - 1.2 Test d'existence d'un fichier. 464
 - 1.3 Test d'existence inversé 464
2. Création de dossiers ou de fichiers 464
 - 2.1 Création de dossiers 464
 - 2.2 Création de fichiers 465
3. Suppression de dossiers ou de fichiers. 465

Chapitre 27

Mise en forme de la sortie des objets

1. Qu'est-ce que la mise en forme des objets ? 467
2. Mise en forme basique 468
3. Out-GridView : un tableau interactif et puissant 470
4. Format-Table : mise en forme dans un tableau 472
 - 4.1 Affichage par défaut. 472
 - 4.2 Filtrer les colonnes 473
 - 4.3 Masquer l'en-tête des colonnes 473
 - 4.4 Renommer une colonne 474
5. Format-List : affichage sous forme d'une liste 475
6. Sort-Object : trier la liste en fonction d'une propriété. 476
7. Group-Object : grouper les objets par propriété. 476

Chapitre 28

Manipuler les fichiers

1. Introduction	479
2. Encodage de caractères et fichiers	479
2.1 Qu'est-ce que l'encodage ?	479
2.2 Pourquoi l'encodage ?	480
2.3 PowerShell et l'encodage	482
3. Manipuler des fichiers texte.	484
3.1 Gestion des fichiers texte	484
3.2 Exporter le contenu d'un objet vers un fichier	485
3.3 Lire le contenu d'un fichier.	486
3.4 Lister le nombre total de lignes d'un fichier	487
3.5 Lire un contenu spécifique	488
3.6 Récupérer le numéro d'une ligne	489
3.7 Afficher une ou des premières lignes spécifiques	490
3.7.1 Afficher la première ligne	490
3.7.2 Afficher la dernière ligne	491
3.8 Ajouter une nouvelle ligne	491
3.9 Modifier ou remplacer une ligne	492
3.10 Modifier des lignes comportant un mot spécifique	493
3.11 Supprimer une ligne comportant un mot spécifique	493
4. Manipuler des fichiers XML.	495
4.1 Qu'est-ce que le XML ?	495
4.2 Pourquoi utiliser le format XML ?	495
4.3 Structure d'un fichier XML	497
4.3.1 Le prologue ou en-tête	497
4.3.2 Les balises	497
4.3.3 Les éléments ou nœuds	498
4.3.4 Les attributs	499
4.4 Export de contenu vers un fichier XML	500
4.5 Lire le contenu	501
4.6 Lister les méthodes	503

22 _____ Débuter avec PowerShell

4.7	Accéder aux différents nœuds	505
4.8	Lister une application par son nom	511
4.9	Ajouter du contenu	513
4.10	Modifier du contenu	516
4.11	Supprimer du contenu	518
5.	Manipuler des fichiers CSV	518
5.1	CSV et les délimiteurs	518
5.1.1	Qu'est-ce qu'un délimiteur ?	518
5.1.2	Détecter le délimiteur par défaut sur mon poste	519
5.1.3	Détecter le délimiteur utilisé dans un fichier	519
5.2	Export du contenu vers un fichier CSV	520
5.3	Importer/lire un fichier CSV	524

Chapitre 29

La gestion des dates

1.	Introduction	527
2.	Cmdlet Get-Date	527
2.1	Vérifier l'aide	528
2.2	Lister les propriétés	530
2.3	Lister les méthodes	535
3.	Formatage des dates	538
3.1	Formatage simple	538
3.2	Formatage personnalisé	539
4.	Conversion de dates	543
4.1	Convertir une chaîne de caractères en date	543
4.2	Convertir une date en chaîne de caractères	544
4.3	Conversion de dates depuis le registre ou WMI	545
4.3.1	Date d'installation de Windows	545
4.3.2	Date d'installation d'applications	548

5. Comparaison de dates	548
5.1 Calcul d'intervalle entre deux dates.	548
5.2 Comparer deux dates : antérieure ou postérieure.	551

Chapitre 30

PowerShell et WMI

1. Introduction	553
2. WBEM, DMTF, CIM, WMI.	554
2.1 WBEM.	554
2.2 DMTF	554
2.3 CIM.	554
2.4 WMI	555
3. WMI et PowerShell	556
3.1 Exécution d'une requête.	556
3.2 PowerShell et les commandes WMI	557
3.2.1 Fonctionnement des cmdlets WMI.	557
3.2.2 Lister les espaces de noms disponibles	560
3.2.3 Lister les classes disponibles.	562
3.2.4 Composition d'une classe.	564
3.2.5 Cas d'utilisation	566
3.3 PowerShell et les commandes CIM.	579
3.3.1 Qu'est-ce que CIM ?	579
3.3.2 Fonctionnement des cmdlets.	579
3.3.3 Lister les espaces de noms disponibles	581
3.3.4 Lister les classes disponibles.	581

24 _____ Débuter avec PowerShell

Chapitre 31

PowerShell et le registre

1. Introduction	585
2. Le registre Windows	585
2.1 Qu'est-ce que c'est ?	585
2.2 Composition du registre.	586
2.3 Comment accéder au registre et le modifier ?	587
3. PowerShell et la base de registre Windows	588
3.1 Accéder à la base de registre.	588
3.2 Naviguer dans la base de registre.	591
3.3 Lister les éléments du registre	593
3.3.1 Lister les clés	593
3.3.2 Récupérer les valeurs disponibles.	598
3.4 Tester l'existence d'un élément	601
3.5 Créer un élément	602
3.5.1 Créer une nouvelle clé	602
3.5.2 Créer une valeur	604
3.6 Modifier un élément de registre	606
3.7 Supprimer un élément	608
3.7.1 Supprimer une clé.	608
3.7.2 Supprimer une valeur.	608

Chapitre 32

PowerShell et Base64

1. Qu'est-ce que Base64 ?	609
1.1 C'est quoi ?	609
1.2 Avantages	610
1.3 Inconvénients	611

2. Encoder/décoder en Base64	612
2.1 Encodage depuis le site	612
2.1.1 Encoder du texte	612
2.1.2 Encoder un fichier PNG	613
2.1.3 Encoder un exécutable	615
2.2 Décodage depuis le site	617
2.2.1 Décoder du texte	617
2.2.2 Décoder un fichier PNG	618
2.2.3 Décoder un exécutable	619
3. Encoder et décoder avec PowerShell	620
3.1 Encodage en Base64 avec PowerShell	620
3.1.1 Encodage de texte	620
3.1.2 Encodage d'une image	621
3.2 Décodage en Base64 avec PowerShell	622
3.2.1 Décodage d'un texte	622
3.2.2 Décodage d'un fichier : image, exécutable	622

Chapitre 33

Gérer votre système

1. PowerShell et les services Windows	623
1.1 Get-Service : récupérer les informations du service	625
1.2 Restart-Service : redémarrer un service	626
1.3 Stop-Service : arrêter un service	627
1.4 Start-Service : démarrer un service	628
1.5 Suspend-Service : mettre en pause un service	629
2. PowerShell et les processus	629
2.1 Get-Process : lister les processus en cours	630
2.2 Stop-Process : stopper un processus	634
3. PowerShell et le gestionnaire d'évènements	636
3.1 Lister les différentes catégories d'évènements	636
3.2 Exemples d'utilisation	638

26 _____ Débuter avec PowerShell

4. PowerShell et les tâches planifiées.	641
5. PowerShell et le presse-papiers	645
5.1 Get-Clipboard : récupérer le contenu copié	646
5.1.1 Presse-papiers et texte	647
5.1.2 Presse-papiers et fichiers.	647
5.1.3 Presse-papiers et image.	649
5.2 Set-Clipboard : copier du contenu.	653
5.2.1 Copie de texte.	653
5.2.2 Copie de dossiers et de fichiers	654

Chapitre 34

Débuter avec RegEx

1. Introduction	655
2. Démystifier RegEx, ou tout du moins essayer	656
2.1 Qu'est-ce que RegEx ?	656
2.2 RegEx et syntaxe	656
2.2.1 Les caractères d'échappement	657
2.2.2 Les métacaractères	658
2.2.3 Les quantificateurs	658
2.2.4 Classes et intervalles.	659
3. Exemples rapides	661
4. Les ressources intéressantes	662
5. Quelques exemples du quotidien.	663
5.1 Validation d'une adresse e-mail	663
5.2 Saisie de nom d'ordinateur.	664
5.3 Vérification d'une adresse IP valide (IPv4)	665
5.4 Validation d'un numéro de téléphone portable	666
5.5 Validation d'un numéro de sécurité sociale	666
5.6 Validation d'une plaque d'immatriculation française	667

6. PowerShell et RegEx	668
6.1 Utiliser RegEx avec PowerShell	668
6.1.1 -Match et -NotMatch.	668
6.1.2 Classe [regex]	669
6.2 Nos exemples avec PowerShell	671
6.2.1 Opérateur -Match.	671
6.2.2 Classe [regex]	672
6.3 Travailler sur des chaînes de caractères.	675
6.3.1 Récupérer la valeur d'une chaîne	675
6.3.2 Extraire du contenu entre deux mots	677

Chapitre 35 PowerShell et Windows Forms

1. Introduction	681
2. Forcer l'aspect visuel moderne	682
3. Windows Forms	683
3.1 De quoi s'agit-il ?	683
3.2 Comment utiliser Windows Forms ?	683
3.3 Boîte de dialogue simple.	684
3.4 Boîte de dialogue de sélection	689
3.4.1 Sélection de dossier.	689
3.4.2 Sélection de fichier	691
4. Notification toast	694

Chapitre 36 Des exemples pour le quotidien

1. Introduction	697
2. Génération de fichier de log	697
3. Fonction de création de clé de registre	698
4. Vérifier la validité d'un lien web	700

28 _____ Débuter avec PowerShell

5. Télécharger un fichier depuis Internet	701
6. Faire des captures d'écrans avec PowerShell	702
7. Zipper et dézipper	703
8. Créer un journal d'évènements et ajouter un évènement	703
9. Exporter un journal d'évènements.	706
10. Générer un GUID	707
11. Générer un mot de passe	708
12. Chercher une chaîne dans l'observateur d'évènements	712
13. Identifier les comptes administrateurs locaux	713
14. Obtenir la date de dernier redémarrage.	714
15. Vérifier qu'un script s'exécute en administrateur.	717

Chapitre 37

S'amuser avec PowerShell

1. PowerShell en musique	719
2. Faire parler PowerShell.	721
3. Récupérer vos coordonnées géographiques.	723
4. Afficher l'adresse de localisation du poste.	725
5. Afficher les coordonnées GPS dans une carte.	726

Chapitre 38

Bibliographie

1. Les livres pour approfondir ses connaissances	729
1.1 Commencer et approfondir	729
1.2 Approfondir sur d'autres domaines	730

Index	733
-----------------	-----