

PowerShell for Sys Admins – Errata (updated to 5th printing)

Page 15: In the box “Turning on Strict Mode”:

Get Help Set-StrictMode Examples

Should now read:

Get-Help Set-StrictMode -Examples

Page 31: In Listing 2-34:

```
PS> $users = @{  
abertram = 'Adam Bertram'  
raquelcer = 'Raquel Cerillo'
```

Should now read:

```
PS> $users = @{  
abertram = 'Adam Bertram';  
raquelcer = 'Raquel Cerillo';
```

Page 41: In the last sentence of the third paragraph:

Because parameters passed via ByValue depend on the type of input, only one parameter can be passed via ByValue.

Should now read:

Because parameters passed via ByValue depend on the type of input, each parameter passed via ByValue can be one type only.

Page 49: In the last entry in the list (-contains):

Returns True if the second value is “in” the second.

Should now read:

Returns True if the second value is “in” the first.

Page 52: On the third line of Listing 4-5, we inserted “{” before the <2> annotation.

Page 57: In the code block in the middle of the page, the third line:

```
$servers[$i] = “new $server”
```

Should now read:

```
$servers[$i] = "new $(servers[$i])"
```

Page 74: In the last sentence of the third paragraph, we replaced “appropriate folder” with “version’s folder”

Page 76: In the first code block, we swapped the fourth and fifth lines:

```
[string]$Version  
[ValidateSet('1','2')],
```

So it should now read:

```
[ValidateSet('1','2')],  
[string]$Version
```

Page 99: In the second to last paragraph, we changed “computer’s name” to “session name” and changed “-Session *session object*” to “-Session *session name*”

Page 103: In the first line of Listing 8-24:

```
PS> Enable-WsManCredSSP -Role Client -DelegateComputer WEBSRV1
```

Should now read:

```
PS> Enable-WsManCredSSP -Role Client -DelegateComputer WEBSRV1 -Force
```

And right before Listing 8-24, we inserted a new note that reads:

“To get CredSSP to work, you may have to relax a local policy. If you receive a permission error when trying to enable CredSSP, be sure you enable the *Allow Delegating Saved Credentials with NTLM-only Server Authentication* setting by running gpedit.msc and looking under Computer Configuration > Administrative Templates > System > Credentials Delegation. While in the policy, click on the **Show** button and enter **WSMAN/*** to allow delegation from any endpoint.”

Page 111: In Listing 9-5:

```
(Get-WindowsFeature @parameters).Installed | should be $true
```

Should now read:

```
(Get-WindowsFeature @parameters).Installed | should -Be $true
```

Page 134: We inserted a new paragraph right after the first code block that reads:

“In Windows PowerShell, Invoke-WebRequest relies on Internet Explorer. If you don't have Internet Explorer on your computer, you may have to use the -UseBasicParsing parameter to remove the dependency. “Advanced” parsing breaks down the resulting HTML output a bit more but it's not needed in all cases.”

Page 155: In the second line of Listing 11-11:

```
$positiveMatches = (Find-UserMatch).where({ $_.CSVProperties -ne 'NoMatch' })
```

Should now read:

```
$positiveMatches = (Find-UserMatch -SyncFieldMap $syncFieldMap -FieldMatchIds $fieldMatchIds).where({ $_.CSVProperties -ne 'NoMatch' })
```

Page 166: In the last line of Listing 12-14, we changed “OSDisk” to “\$osDiskName”

Page 169: In the third line of Listing 12-20:

```
ServerName = 'PowerShellForSysAdmins-SQLSrv'
```

Should now read:

```
ServerName = 'powershellforsysadmins-sqlsrv'
```

Page 173: In the third paragraph, we changed “AzureRM” to “Az”

Page 175: Right after Listing 13-2, we inserted a new sentence:

“Notice the Arn property in Listing 13-2. You'll need this value coming up when you create the IAM role.”

Page 176: Right before Listing 13-3, we inserted a new sentence: “Important: Notice the *XXXXXX* on the Principal line. Be sure to replace the ARN of the IAM user you just created there.”

In both Listing 13-3 and Listing 13-4, we replaced “2019-10-17” with “2012-10-17” and we replaced “arn:aws:iam::013223035658:user/Automator” with “arn:aws:iam::XXXXXX:user/Automator”

In Listing 13-4, we also replaced the RoleId “AROAJ2B7YC3HH6M6F2WOM” with “<Your Specific Role ID>” and we replaced the CreateDate “9/16/2019 6:05:37 PM” with “<Date created>”

Page 184: In the second line of Listing 13-19, we changed “PS> \$ebSApp” to “PS> \$ebApp”

Page 185: At the beginning of Listing 13-21, we inserted 3 new lines before “>>\$parameters = @{}” that read:

```
PS> $instanceProfileOptionSetting = New-Object Amazon.ElasticBeanstalk.Model.  
ConfigurationOptionSetting -ArgumentList aws:autoscaling:launchconfiguration,  
IamInstanceProfile,'aws-elasticbeanstalk-ec2-role'
```

And we changed “2012 R2 running IIS 8.5” to “2019 v2.5.9 running IIS 10.0”

And we replaced the “}” line that appears after “Tier_Name = ‘WebServer’” with the following two new lines:

```
>> OptionSetting = $instanceProfileOptionSetting  
>>}
```

And we changed both instances of “9/19/2019 12:19:36 PM” (after DateCreated and DateUpdated) to “10/3/2020 9:31:49 AM”

And we changed “arn:aws:elasticbeanstalk:...” (after EnvironmentArn) to “arn:aws:elasticbeanstalk:us-east-1:054715970076:environment/AutomateWorkflow/Testing”

And we changed “e-wkba2k4kcf” (after EnvironmentID) to “e-f3pfgxhrzf”

And we added a new blank row, OperationsRole, between HealthStatus and PlatformArn

And we changed “arn:aws:elasticbeanstalk...” (PlatformArn) to “arn:aws:elasticbeanstalk:us-east-1::platform/IIS 10.0 running on 64bit Windows Server Core”

And we changed “64bit Windows Server Core 2012 R2 running IIS 8.5” (SolutionStackName) to “64bit Windows Server Core 2019 v2.5.9 running IIS 10.0”

Page 200: In the first line of the first code block:

```
PS> Get-ChildItem -Path '\\WEBSRV1\c$\Users\' -File | Measure-Object -Property Length -Sum
```

Should now read:

```
PS> Get-ChildItem -Path '\\WEBSRV1\c$\Users\' -File -Recurse | Measure-Object -Property  
Length -Sum
```

And in Listing 14-4, the line:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-  
Object -Property Length -Sum).Sum
```

Should now read:

```
$output.'UserProfileSize (MB)' = (Get-ChildItem -Path "\\WEBSRV1\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum
```

Page 201: In the first line of the code block:

```
$UserProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File |
```

Should now read:

```
$output.'UserProfileSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse
```

Page 202: Under the second code block, we inserted a new note that reads:

The above command will only work if a computer only has a single disk. In my test environment, sqlsrv1 only has a C drive. If your server has more than one drive, you can combine the free space of all mounted drives by using the Measure-Object command like so: (Get-CimInstance -ComputerName sqlsrv1 -ClassName Win32_LogicalDisk | Measure-Object -Property FreeSpace -Sum).Sum. Subsequent free space code listings will assume your remote server has a single drive.

Page 203: In Listing 14-5:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-Object -Property Length -Sum).Sum / 1MB
```

Should now read:

```
$UserProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum  
$output.'User ProfileSize (MB)' = [int]($UserProfileSize / 1MB)
```

and in the last 2 lines of the next code block, both “636245” and “603942” should now read “1”

Page 205: In Listing 14-6:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-Object -Property Length -Sum).Sum / 1MB
```

Should now read:

```
$UserProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum  
$output.'User ProfileSize (MB)' = [int]($UserProfileSize / 1MB)
```

and in the last 2 lines of the next code block, both “636245” and “603942” should now read “1”

Page 208: In Listing 14-7:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-Object -Property Length -Sum).Sum / 1MB
```

Should now read:

```
$userProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum  
$output.'User ProfileSize (MB)' = [int]($userProfileSize / 1MB)
```

and in the last 2 lines of the next code block, both “636245” and “603942” should now read “1”

Page 209: In Listing 14-7:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-Object -Property Length -Sum).Sum / 1MB
```

Should now read:

```
$userProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum  
$output.'User ProfileSize (MB)' = [int]($userProfileSize / 1MB)
```

and in the last 2 lines of the next code block, both “636245” and “603942” should now read “1”

Page 210: In the first line of the first listing:

```
PS> C:\Get-ServerInformation.ps1 | Format-Table -AutoSize
```

Should now read:

```
PS> C:\Get-ServerInformation.ps1
```

and in the same listing, “636245” and “603942” should both read “1”

Page 211: In Listing 14-9:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-Object -Property Length -Sum).Sum / 1MB
```

Should now read:

```
$userProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum  
$output.'User ProfileSize (MB)' = [int]($userProfileSize / 1MB)
```

Page 212: In the third to last line of Listing 14-10:

```
Remove-CimSession -CimSession $cimSession
```

Should now read:

```
Remove-CimSession -CimSession $getCimInstParams.CimSession
```

and in the same listing:

```
$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\$server\c$\Users\" -File | Measure-Object -Property Length -Sum).Sum / 1MB
```

Should now read:

```
$UserProfileSize = (Get-ChildItem -Path "\\$server\c$\Users\" -File -Recurse | Measure-Object -Property Length -Sum).Sum  
$output.'User ProfileSize (MB)' = [int]($UserProfileSize / 1MB)
```

Page 219: In the first sentence of the fourth paragraph, we changed “in the *All Users* location of *C:\Files*” to “in *C:\ProgramFiles\WindowsPowerShell\Modules*”

Page 222: We deleted the first line of the last paragraph that read: “Because you previously imported the module, PowerShell hadn’t loaded any functions into the session.”

Page 225: In the second code block, we changed “*C:\PowerLab\VHDs\LABDC.vhdx*” to “*C:\PowerLab\VHDs\MYVM.vhdx*”

Page 226: In the last line of the second code block, we changed “*C:\PowerLab\VHDs\LABDC.vhdx*” to “*C:\PowerLab\VHDs\MYVM.vhdx*”

Page 229: In the sixth line of the code block, we changed “*LABDC*” to “*MYVM*”

Page 232: In the fourth paragraph, we changed “*unattend.xml*” to “*LABDC.xml*”

Page 242: In the first line of the code block, we changed “*C:\Files.xml*” to “*C:\PowerLab\VMCredential.xml*”

Page 248: In Listing 17-2, the lines:

```
New-ADGroup -Name GroupName -GroupScope GroupScope -Path  
"OU=OUName,DC=powerlab,DC=local" <4>
```

Should now read:

New-ADGroup -Name *\$group.GroupName* -GroupScope GroupScope -Path "OU=*\$group.OUName*,DC=powerlab,DC=local" <4>

Page 286: Under “cmdlets”:

Invoke-RestMethod, 134

Should now read:

Invoke-RestMethod, 135

And in the third entry under D, we added “-104” so it reads:

DelegateComputer parameter, 103–104

Page 288: The entry:

Invoke-RestMethod cmdlet, 134

Should now read:

Invoke-RestMethod cmdlet, 135

Page 289: Under “parameters”:

DelegateComputer, 103

Should now read:

DelegateComputer, 103–104