

# PowerShell for Sysadmins

## Workflow Automation Made Easy

by Adam Bertram

errata updated to print 8

Page	Error	Correction	Print corrected
15	For more information about strict mode, run <code>Get Help Set-StrictMode Examples</code> .	For more information about strict mode, run <code>Get-Help Set-StrictMode -Examples</code> .	Print 2
31	<pre>PS&gt; \$users = @{     abertram = 'Adam Bertram';     raquelcer = 'Raquel Cerillo'</pre>	<pre>PS&gt; \$users = @{     abertram = 'Adam Bertram';     raquelcer = 'Raquel Cerillo';</pre>	Print 3
41	Because parameters passed via <code>ByValue</code> depend on the type of input, each parameter passed via <code>ByValue</code> can be <b>passed by ByValue</b> .	Because parameters passed via <code>ByValue</code> depend on the type of input, each parameter passed via <code>ByValue</code> can be <b>one type only</b> .	Print 4
49	Returns <code>True</code> if the second value is “in” the <b>second</b> . You can use this to determine whether a value is inside an array.	Returns <code>True</code> if the second value is “in” the <b>first</b> . You can use this to determine whether a value is inside an array.	Print 4
52	<pre>if (-not (Test-Connection -ComputerName \$servers[0] -Quiet -Count 1)) { ❶     Write-Error -Message "The server \$servers[0] is not responding!" } elseif (\$servers[0] -eq \$problemServer) ❷     Write-Error -Message "The server \$servers[0] does not have the right file!"</pre>	<pre>if (-not (Test-Connection -ComputerName \$servers[0] -Quiet -Count 1)) { ❶     Write-Error -Message "The server \$servers[0] is not responding!" } elseif (\$servers[0] -eq \$problemServer) { ❷     Write-Error -Message "The server \$servers[0] does not have the right file!"</pre>	Print 3
57	<pre>\$servers = @('SERVER1','SERVER2','SERVER3','SERVER4','SERVER5') for (\$i = 0; \$i -lt \$servers.Length; \$i++) {     \$servers[\$i] = "new \$server" } \$servers</pre>	<pre>\$servers = @('SERVER1','SERVER2','SERVER3','SERVER4','SERVER5') for (\$i = 0; \$i -lt \$servers.Length; \$i++) {     \$servers[\$i] = "new \$(\$servers[\$i])" } \$servers</pre>	Print 3
74	In that case, the function will fail to find the <b>appropriate</b> folder because it doesn't exist.	In that case, the function will fail to find the <b>version's</b> folder because it doesn't exist.	Print 4

Page	Error	Correction	Print corrected
76	<pre>function Install-Software {     param(         [Parameter(Mandatory)]         [string]\$Version         [ValidateSet('1','2')],     ) }</pre>	<pre>function Install-Software {     param(         [Parameter(Mandatory)]         [ValidateSet('1','2')]         [string]\$Version,     ) }</pre>	Print 4
80	<p>You'll begin this chapter by looking at the <b>models</b> that are already installed in your system. Then, you'll take apart a <b>model</b> to see its different parts before finally looking at how to download and install PowerShell modules from the PowerShell Gallery.</p>	<p>You'll begin this chapter by looking at the <b>modules</b> that are already installed in your system. Then, you'll take apart a <b>module</b> to see its different parts before finally looking at how to download and install PowerShell modules from the PowerShell Gallery.</p>	Print 8
99	<p>To properly disconnect from a session, you pass your remote <b>computer's</b> name to the Session parameter by either calling it explicitly via Disconnect-PSSession -Session <i>session object</i></p>	<p>To properly disconnect from a session, you pass your remote <b>session</b> name to the Session parameter by either calling it explicitly via Disconnect-PSSession -Session <i>session name</i></p>	Print 4
103	<pre>PS&gt; Enable-WSManCredSSP ① -Role ② Client ③ -DelegateComputer WEBSRV1</pre>	<p><b>NOTE</b></p> <p>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.</p> <pre>PS&gt; Enable-WSManCredSSP ① -Role ② Client ③ -DelegateComputer WEBSRV1 -Force</pre>	Print 4
134	Insertion	<p>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.</p>	Print 4
155	<pre>## Find all of the CSV &lt;--&gt; AD user account matches \$positiveMatches = (Find-UserMatch).where({ \$_.CSVProperties -ne 'NoMatch' })</pre>	<pre>## Find all of the CSV &lt;--&gt; AD user account matches \$positiveMatches = (Find-UserMatch -SyncFieldMap \$syncFieldMap -FieldMatchIds \$fieldMatchIds).where({ \$_.CSVProperties -ne 'NoMatch' })</pre>	Print 4

Page	Error	Correction	Print corrected
166	<pre>PS&gt; \$vm = Set-AzVMOSDisk -Name OSDisk -CreateOption 'fromImage' -VM \$vm -VhdUri \$osDiskUri</pre>	<pre>PS&gt; \$vm = Set-AzVMOSDisk -Name \$osDiskName -CreateOption 'fromImage' -VM \$vm -VhdUri \$osDiskUri</pre>	Print 4
169	<pre>ServerName = 'PowerShellForSysAdmins-SQLSrv'</pre>	<pre>ServerName = 'powershellforsysadmins-sqlsrv'</pre>	Print 4
173	<p>You can install <code>AWSPowerShell</code> from the PowerShell Gallery the same way you installed the <code>AzureRM</code> module, by calling <code>Install-Module AWSPowerShell</code>. Once this module is downloaded and installed, you're ready to go.</p>	<p>You can install <code>AWSPowerShell</code> from the PowerShell Gallery the same way you installed the <code>Az</code> module, by calling <code>Install-Module AWSPowerShell</code>. Once this module is downloaded and installed, you're ready to go.</p>	Print 4
175	Insertion	<p><b>Notice the <code>Arn</code> property in Listing 13-2. You'll need this value coming up when you create the IAM role.</b></p>	Print 4
176	<p>Listing 13-3 is an example of a trust relationship policy document.</p> <pre>{   "Version": "2019-10-17",   "Statement": [     {       "Effect": "Allow",       "Principal": { "AWS": "arn:aws:iam::013223035658:user/Automator" },</pre>	<p>Listing 13-3 is an example of a trust relationship policy document. <b>Important: Notice the <code>xxxxxx</code> on the Principal line. Be sure to replace the ARN of the IAM user you just created there.</b></p> <pre>{   "Version": "2012-10-17",   "Statement": [     {       "Effect": "Allow",       "Principal": { "AWS": "arn:aws:iam::xxxxxx:user/Automator" },</pre>	Print 4
176	<pre>PS&gt; New-IAMRole -AssumeRolePolicyDocument \$json -RoleName 'AllAccess' Path      RoleName      RoleId      CreateDate ----      - /         AllAccess     AROAJ2B7YC3HH6M6F2WOM  9/16/2019 6:05:37 PM</pre>	<pre>PS&gt; New-IAMRole -AssumeRolePolicyDocument \$json -RoleName 'AllAccess' Path      RoleName      RoleId      CreateDate ----      - /         AllAccess     &lt;Your Specific Role ID&gt;  &lt;Date created&gt;</pre>	Print 4
184	<pre>PS&gt; \$ebApp = New-EBAApplication -ApplicationName 'AutomateWorkflow' PS&gt; \$ebSApp</pre>	<pre>PS&gt; \$ebApp = New-EBAApplication -ApplicationName 'AutomateWorkflow' PS&gt; \$ebSApp</pre>	Print 4

Page	Error	Correction	Print corrected
185-186	<pre>&gt;&gt; \$parameters = @{ &gt;&gt;     ApplicationName = 'AutomateWorkflow' &gt;&gt;     EnvironmentName = 'Testing' &gt;&gt;     SolutionStackName = '64bit Windows Server Core 2012 R2 running IIS 8.5' &gt;&gt;     Tier_Type = 'Standard' &gt;&gt;     Tier_Name = 'WebServer' &gt;&gt; } PS&gt; New-EBEnvironment @parameters  AbortableOperationInProgress : False ApplicationName               : AutomateWorkflow CNAME                         : DateCreated                   : 9/19/2019 12:19:36 PM DateUpdated                   : 9/19/2019 12:19:36 PM Description                   : EndpointURL                   : EnvironmentArn                : arn:aws:elasticbeanstalk:... EnvironmentId                 : e-wkba2k4kcf EnvironmentLinks              : {} EnvironmentName               : Testing Health                        : Grey HealthStatus                  : PlatformArn                   : arn:aws:elasticbeanstalk:... Resources                     : SolutionStackName             : 64bit Windows Server Core 2012 R2 running IIS 8.5 Status                        : Launching TemplateName                  : Tier                          : Amazon.ElasticBeanstalk.Model.EnvironmentTier VersionLabel                  :</pre>	<pre>PS&gt; \$instanceProfileOptionSetting = New-Object Amazon.ElasticBeanstalk.Model .ConfigurationOptionSetting -ArgumentList aws:autoscaling:launchconfiguration, IamInstanceProfile, 'aws-elasticbeanstalk-ec2-role'  &gt;&gt; \$parameters = @{ &gt;&gt;     ApplicationName = 'AutomateWorkflow' &gt;&gt;     EnvironmentName = 'Testing' &gt;&gt;     SolutionStackName = '64bit Windows Server Core 2019 v2.5.9 running IIS 10.0' &gt;&gt;     Tier_Type = 'Standard' &gt;&gt;     Tier_Name = 'WebServer' &gt;&gt;     OptionSetting = \$instanceProfileOptionSetting &gt;&gt; } PS&gt; New-EBEnvironment @parameters  AbortableOperationInProgress : False ApplicationName               : AutomateWorkflow CNAME                         : DateCreated                   : 10/3/2020 9:31:49 AM DateUpdated                   : 10/3/2020 9:31:49 AM Description                   : EndpointURL                   : EnvironmentArn                : arn:aws:elasticbeanstalk:us-east-1:054715970076: environment/AutomateWorkflow/Testing EnvironmentId                 : e-f3pfgxhrzf EnvironmentLinks              : {} EnvironmentName               : Testing Health                        : Grey HealthStatus                  : OperationsRole                 : PlatformArn                   : arn:aws:elasticbeanstalk:useast-1::platform/IIS 10.0 running on 64bit Windows Server Core Resources                     : SolutionStackName             : 64bit Windows Server Core 2019 v2.5.9 running IIS 10.0 Status                        : Launching TemplateName                  : Tier                          : Amazon.ElasticBeanstalk.Model.EnvironmentTier VersionLabel                  :</pre>	Print 4
200	<pre>PS&gt; Get-ChildItem -Path '\\WEBSRV1\c\$\Users\' -File   Measure-Object -Property Length -Sum</pre>	<pre>PS&gt; Get-ChildItem -Path '\\WEBSRV1\c\$\Users\' -File -Recurse   Measure-Object -Property Length -Sum</pre>	Print 4

Page	Error	Correction	Print corrected
200	<pre>\$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\\$server\c\$\Users\" -File   Measure-Object -Property Length -Sum).Sum</pre>	<pre>\$output.'UserProfileSize (MB)' = (Get-ChildItem -Path '\\WEBSRV1\c\$\Users\' -File -Recurse   Measure-Object -Property Length -Sum).Sum</pre>	Print 4
201	<pre>\$userProfileSize = (Get-ChildItem -Path "\\\$server\c\$\Users\" -File   Measure-Object -Property Length -Sum).Sum</pre>	<pre>\$output.'UserProfileSize (MB)' = (Get-ChildItem -Path "\\\$server\c\$\Users\" -File -Recurse   Measure-Object -Property Length -Sum).Sum</pre>	Print 5
202	Insertion	<p><b>NOTE</b></p> <p><i>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.</i></p>	Print 5
203, 205, 208, 209, 211, 212	<pre>\$output.'UserProfilesSize (MB)' = (Get-ChildItem -Path "\\\$server\c\$\Users\" -File   Measure-Object -Property Length -Sum).Sum / 1MB</pre>	<pre>\$userProfileSize = (Get-ChildItem -Path "\\\$server\c\$\Users\" -File -Recurse   Measure-Object -Property Length -Sum).Sum \$output.'User ProfileSize (MB)' = [int](\$userProfileSize / 1MB)</pre>	Print 5
203, 205, 208, 209, 210	<pre>ServerName UserProfilesSize (MB) ----- SQLSRV1          636245 WEBSRV1          603942</pre>	<pre>ServerName UserProfilesSize (MB) ----- SQLSRV1          1 WEBSRV1          1</pre>	Print 5
210	<pre>PS&gt; C:\Get-ServerInformation.ps1   Format-Table -AutoSize</pre>	<pre>PS&gt; C:\Get-ServerInformation.ps1</pre>	Print 4
212	<pre>Remove-CimSession -CimSession \$cimSession</pre>	<pre>Remove-CimSession -CimSession \$getCimInstParams.CimSession</pre>	Print 4
219	Since you're logged in via the local administrator account and may one day allow others to use your PowerLab module, create the module in the <i>All Users location of C:\Files</i> .	Since you're logged in via the local administrator account and may one day allow others to use your PowerLab module, create the module in <i>C:\ProgramFiles\WindowsPowerShell\Modules</i> .	Print 4
222	Deletion	<i>Because you previously imported the module, PowerShell hadn't loaded any functions into the session.</i>	Print 4

Page	Error	Correction	Print corrected
225– 226	<pre>-Path 'C:\PowerLab\VHDs\LABDC.vhdx'</pre>	<pre>-Path 'C:\PowerLab\VHDs\MYVM.vhdx'</pre>	Print 4
229	<pre>[+] created a virtual machine called LABDC 62ms</pre>	<pre>[+] created a virtual machine called MYVM 62ms</pre>	Print 4
232	<p>And finally, you need the unattended XML answer file (also available via the chapter's downloadable resources) called <i>unattend.xml</i> in the PowerLab module folder.</p>	<p>And finally, you need the unattended XML answer file (also available via the chapter's downloadable resources) called <i>LABDC.xml</i> in the PowerLab module folder.</p>	Print 4
242	<pre>Get-Credential   Export-CliXml -Path C:\Files.xml</pre>	<pre>Get-Credential   Export-CliXml -Path C:\PowerLab\VMCredential.xml</pre>	Print 4
248	<pre>New-ADGroup -Name \$GroupName -GroupScope \$GroupScope -Path "OU=\$OUName,DC=powerlab,DC=local" ④</pre>	<pre>New-ADGroup -Name \$group.GroupName -GroupScope \$GroupScope -Path "OU=\$group.OUName,DC=powerlab,DC=local" ④</pre>	Print 4