

[External] [In-Progress] Database Migration PowerShell Commands

A. Initial Setup

	To-do	PS Command
1	Enable remoting	<code>Enable-PSRemoting</code>
2	One-to-one remote	<code>Enter-PSSession -computerName <Server-R2></code>

B. Auditing Server Properties

	Server Property	PS Command
1	Localhost	<code>\$env:computername</code>
2	IP address	<code>Get-NetIPAddress -AddressFamily IPv4</code>
3	PowerShell version	<code>Get-Host Select-Object -Property Version</code> <code>\$PSVersionTable</code>

C. dbatools


 dbatools require a minimum PowerShell version of 3.0: <https://www.powershellgallery.com/packages/dbatools/1.0.135>

See command B3 for checking the version, as well as the following articles on updating PowerShell:


- <https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/install/installing-windows-powershell?view=powershell-7.1#upgrading-existing-windows-powershell>
- <https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/wmf/setup/install-configure?view=powershell-7.1>

If PowerShell is greater or equal to 3.0 but less than 5.0, dbatools will have to be installed with an offline method:

<https://dbatools.io/offline/>

 If step 3 returns an error like "PowerShell Gallery is currently unavailable," try running the commands here and restart PowerShell:

<https://dev.to/darksmile92/powershell-disabled-support-for-tls-1-0-for-the-gallery-update-module-and-install-module-broken-1oii>

 Newer versions of PowerShell may throw an error like "The certificate chain was issued by an authority that is not trusted" when commands are run. As a workaround, Connect-DbalInstance can be used as such:

```
$server = Connect-DbalInstance -SqlInstance sql2016 -TrustServerCertificate
```

```
Find-DbaAgentJob -SqlInstance $server -JobName *backup*
```

 [Can we change the parameter of Find-DbaAgentJob?](#)

	To-do	PS Command	Example
--	-------	------------	---------

1	Get list of installed modules and versions	Get-InstalledModule	Get-InstalledModule dbatools
2	Execution Policy	Set-ExecutionPolicy - ExecutionPolicy RemoteSigned -Scope CurrentUser	
3	Trust PowerShell Gallery	Set-PSRepository -Name PSGallery - InstallationPolicy Trusted	
4	Install dbatools	Install-Module -Name dbatools	
5	Update dbatools	Update-Module dbatools	
6	Test DB connection	Connect-DbaInstance	Connect-DbaInstance -SqlInstance EBENGUNADI255E\MSSQLSERVER01
7	Disable SQL Agent Jobs	Set-DbaAgentJob -Disable	1 Get-DbaAgentJob -SqlInstance localhost 2 Set-DbaAgentJob -SqlInstance localhost -Disable
8	Enable running SQL Agent Jobs	Start-DbaAgentJob - Enable	1 Get-DbaAgentJob -SqlInstance localhost 2 Set-DbaAgentJob -SqlInstance localhost -Enable
9	Create Credential object	Get-Credential	\$cred = Get-Credential sa
10	Copy all non-system Logins	Copy-DbaLogin - ExcludeSystemLogins	1 Copy-DbaLogin -Source \$replica -Destination \$replicatoc 2 -SourceSqlCredential \$cred -DestinationSqlCredential \$ 3 -ExcludeLogin realcajun -ExcludeSystemLogins -EnableExc
11	Copy SQL Jobs <div>⚠ Some jobs have hard-coded IPs, hostnames, database names, or directories. To check for these, it's safer to recreate the jobs by scripting them out.</div>	Copy-DbaAgentJob	1 Copy-DbaAgentJob -Source sqlserver2014a -Destination sq 2 -SourceSqlCredential \$cred -DestinationSqlCredential \$
12	Visualize SQL Job schedules	Find-DbaAgentJob Get-DbaAgentJobHistory ConvertTo-DbaTimeline	1 \$oneweekago = [datetime]::Today.AddDays(-7) 2 Find-DbaAgentJob -SqlInstance localhost 3 Get-DbaAgentJobHistory -StartDate \$oneweekago 4 ConvertTo-DbaTimeline 5 Out-File -FilePath c:\temp\jobs.html -Encoding ASCII

13	Check for active connections to the database	Get-DbProcess	<pre> 1 Get-DbProcess -SqlInstance SourceServer -Database Migr 2 Select Host, login, Program </pre>
14	Remove connections to the database	Stop-DbProcess	<pre> 1 Get-DbProcess -SqlInstance SourceServer -Database Migr 2 Stop-DbProcess </pre>
15	Copy databases	Copy-DbDatabase	<pre> 1 Copy-DbDatabase -Source sql2014a -Destination sql2014b </pre>
16	Restore databases	Restore-DbDatabase	<pre> 1 Restore-DbDatabase -SqlInstance dbatoolslab\sql2017 -P </pre>
17	Install Standard Maintenance Plan	Install-DbMaintenanceSolution	<pre> 1 \$params = @{ 2 SqlInstance = 'SQLSERVER01' 3 Database = 'medical' 4 ReplaceExisting = \$true 5 InstallJobs = \$true 6 CleanupTime = 72 7 Verbose = \$true 8 } 9 Install-DbMaintenanceSolution @params </pre>
18	Automatically calculate and set maximum memory	Set-DbMaxMemory	<pre> 1 Set-DbMaxMemory -SqlInstance \$SQLInstance </pre>