



source: <http://www.MSSQLTips.com/tip.asp?id=3407> -- printed: 11/21/2014 1:04:59 PM

# One Line PowerShell Solutions To Common SQL Server DBA Tasks

Written By: Jeffery Yao -- 11/20/2014

## Problem

What are the common DBA tasks suitable for SQL Server PowerShell? How can I start to learn about these tasks in SQL Server?

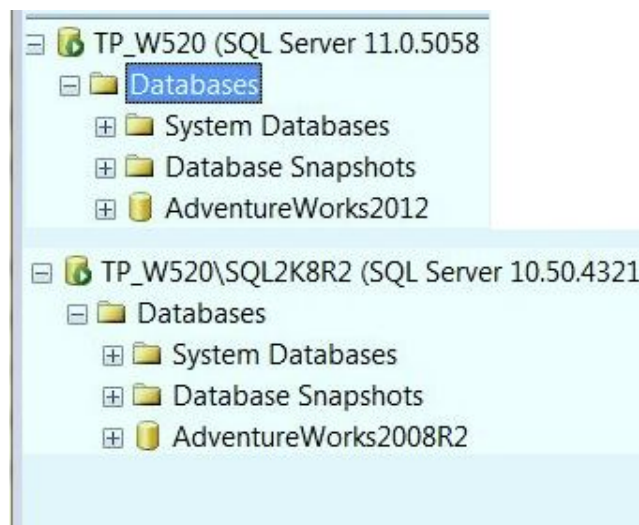
## Solution

These days, PowerShell is no doubt a hot skill for all Microsoft Server product Administrators. To DBAs, there is a special SQL Server module called SQLPS that has been available since SQL Server 2012. We can use it to do many DBA tasks on almost all SQL Server versions (from SQL Server 2005 on). For example, in the DBA community, how to script out a particular object type, such as tables, foreign keys, UDFs, stored procedures, SQL Server Agent Jobs, etc., seems to be a never-ending topic. This is probably because DBAs frequently need to do this task, like during a deployment, we will script out the affected objects as a backup before the deployment changes them.

There are many different ways using T-SQL to generate the creation or deletion script. However the issue with this T-SQL approach is that it is not intuitive and may need quite some time to come up with a complex T-SQL solution using many joins and even cursors.

With SQL Server PowerShell available, it is very intuitive to come up with a solution as simple as one line. The following examples are intended to illustrate such one-line solutions to some common DBA tasks.

All scripts are tested on my laptop which has SQL Server 2012 developer Edition, PowerShell V4 and Window 7 professional. For demo purpose, I will use my local SQL Server instances where possible.



## Quick Introduction of SQLPS

With SQLPS, a SQL Server environment is presented as a folder structure, with the SQL Server instance as root folder in the format of:

```
SQLServer:\SQL\[ComputerName]\[InstanceName]\
```

In my local computer case, my computer name is TP\_W520, and I have two SQL Server instances, one is the [default] instance and another is [SQL2K8R2] and as such, in SQLPS, I can explore the two sql instances via the following two folders:

- DIR SQLSERVER:\SQL\TP\_W520\DEFAULT\
- DIR SQLSERVER:\SQL\TP\_W520\SQL2K8R2\

## Example 1: Scripting SQL Server Database Objects

```
#first we need to import SQLPS module
import-module SQLPS -DisableNameChecking;

# script out all foreign keys in T-SQL
dir sqlserver:\sql\tp_w520\default\databases\AdventureWorks2012\tables | % {$_foreignkeys } | % {$_script()};

# if we need to save the script to a file, we just add out-file at the end of the code
dir sqlserver:\sql\tp_w520\default\databases\AdventureWorks2012\tables | % {$_foreignkeys } | % {$_script()} | out-file c:\temp\job.sql

# script out all foreign key deletion in T-SQL
dir sqlserver:\sql\tp_w520\default\databases\AdventureWorks2012\tables | % {$_foreignkeys } | % {"alter table $(($_foreignkeys).Name) drop constraint $(($_foreignkeys).FKName)}

#script out all stored procedures
dir sqlserver:\sql\tp_w520\SQL2K8R2\databases\AdventureWorks2008R2\StoredProcedures | % {$_script()+'go'};

#script out views with prefix as 'vEmployee'
dir sqlserver:\sql\tp_w520\SQL2K8R2\databases\AdventureWorks2008R2\Views | ? {$_name -like 'vEmployee*'} | % {$_script()+'go'};

#script out all DDL triggers
dir sqlserver:\sql\tp_w520\SQL2K8R2\databases\AdventureWorks2008R2\Triggers | % {$_script()+'go'};

#script out UDFs
dir sqlserver:\sql\tp_w520\SQL2K8R2\databases\AdventureWorks2008R2\UserDefinedFunctions | % {$_script()+'go'};

#script out SQL Server Agent Jobs whose name is 'ps test' and save it to a file at c:\temp\job.sql, if the file exists, overwrite it
dir sqlserver:\sql\tp_w520\SQL2K8R2\jobserver\jobs | ? {$_name -eq 'ps test'} | % {$_script()+'go'} | out-file c:\temp\job.sql -f
```

## Example 2: Collect SQL Server Information

```
#find the top 10 largest tables (in rows) in a database
dir sqlserver:\sql\tp_w520\default\databases\AdventureWorks2012\tables | sort rowcount -desc | select name, rowcount

#find out logins with sysadmin rights on multiple servers (assume the default sql instance only)
'ServerA', 'ServerB', 'ServerC' | % { dir "sqlserver:\sql\$_\default\logins" } | ? {$_ismember('sysadmin')} | select name, ismember

#find user database sizes across multi servers (assuming default sql instances)
'ServerA', 'ServerB', 'ServerC' | % { dir sqlserver:\sql\$_\default\databases } | select parent, name, size | ogv

#check whether a specific login name is on which servers (assume default sql instances)
'ServerA', 'ServerB', 'ServerC' | % { dir sqlserver:\sql\$_\default\logins } | ? {$_name -eq 'specific-loginname'} | select name, ismember

#check whether there is any non-simple recovery database which has not a had a transaction log backup in the last 7 days
'ServerA', 'ServerB', 'ServerC' | % {dir sqlserver:\sql\$_\default\databases} | ? {($_.RecoveryModel -ne 'Simple')} | select name, recoverymodel, lastbackup
```

## Example 3: More Complex PowerShell Examples for SQL Server

```
#find out the database user role membership
dir sqlserver:\sql\tp_w520\default\databases\MyTestDB\users | % -Begin {$a=@()} -process { $a += New-Object PSObject -Property ($_.roles)}

#find the last execution status of all SQL Server Agent Jobs on a SQL Server instance
dir sqlserver:\sql\tp_w520\default\jobserver\jobs | % {$_enumhistory()} | group jobname | % {$_group[0]} | select jobname, laststatus

#find the current failed SQL Server Agent Jobs
dir sqlserver:\sql\tp_w520\default\jobserver\jobs | % {$_enumhistory()} | group jobname | % {$_group[0]} | ? {$_laststatus -eq 'Failed'}

#get the reason of last server shutdown/reboot (note: the local language should be English - US, otherwise, [Message] will be in Chinese)
'ServerA', 'ServerB', 'ServerC' | % {Get-WinEvent -ComputerName $_ -filterhashtable @{logname='System'; id=1074; level=Error}}

#check when the multiple machines were last rebooted
gwmi -class win32_OperatingSystem -Computer ServerA, ServerB, ServerC | select @{label='Server'; e={$_.PSComputerName}}, lastreboot
```

## Next Steps

To use the script in this tip, please make sure the necessary changes for [MachineName], [InstanceName], [DatabaseName] and [LoginName] etc. There are many good resources over the internet about PowerShell, but the important thing is to practice it as much as possible, and the best way is to use it in daily DBA work. The following links have some relevant content so we can see the pros and cons of each different solution.

- [Drop and Re-Create All Foreign Key Constraints in SQL Server](#)
- [Generating SQL Scripts using Windows PowerShell](#)

SQL PowerShell shows its strength best in multiple server scenario, it is concise, elegant and intuitive. I especially enjoy using PowerShell script in some complex DBA tasks, such as Always-On HA, replication, mirroring setup and administration, because usually there are multiple servers involved in such tasks. I hope this tip may intrigue you to explore more about what SQLPS can do for you as a SQL Server DBA.

Follow	Learning	Resources	Search	Community	More Info
<a href="#">Get Free SQL Tips</a>	<a href="#">DBAs</a>	<a href="#">Tutorials</a>	<a href="#">Tip Categories</a>	<a href="#">First Timer?</a>	<a href="#">Join</a>
<a href="#">Twitter</a>	<a href="#">Developers</a>	<a href="#">Webcasts</a>	<a href="#">Search By TipID</a>	<a href="#">Pictures</a>	<a href="#">About</a>
<a href="#">LinkedIn</a>	<a href="#">BI Professionals</a>	<a href="#">Whitepapers</a>	<a href="#">Top Ten</a>	<a href="#">Free T-shirt</a>	<a href="#">Copyright</a>
<a href="#">Google+</a>	<a href="#">Careers</a>	<a href="#">Tools</a>	<a href="#">Authors</a>	<a href="#">Contribute</a>	<a href="#">Privacy</a>
<a href="#">Facebook</a>	<a href="#">Q and A</a>			<a href="#">Events</a>	<a href="#">Disclaimer</a>
<a href="#">Pinterest</a>	<a href="#">Today's Tip</a>			<a href="#">User Groups</a>	<a href="#">Feedback</a>
<a href="#">RSS</a>				<a href="#">Author of the Year</a>	<a href="#">Advertise</a>

---

Copyright (c) 2006-2014 [Edgewood Solutions, LLC](#) All rights reserved

Some names and products listed are the registered trademarks of their respective owners.