



# Practical PowerShell For SQL Server Developers and DBAs

Download the latest version of this PowerShell™ wallchart and read the accompanying in-depth article from Simple-Talk at <http://bit.ly/MZmOX9>

## Module Repositories

Store your modules—including sqlps—in one of these.

<b>system</b> -level	\$env:windir\System32\WindowsPowerShell\v1.0\Modules
<b>user</b> -level	\$HOME\Documents\WindowsPowerShell\Modules

## Modules vs. Snap-ins

Loading snap-ins for SQL support are deprecated, though available.

	SQL Server 2008 & 2008 R2	SQL Server 2012
Cmdlet support	SQLServerCmdletSnapin100	SQLServerCmdletSnapin110
Provider support	SQLServerProviderSnapin100	SQLServerProviderSnapin110

Rather, use the sqlps module instead.

	Load	View items loaded	View items available to load
Modules	Import-Module	Get-Module	Get-Module -ListAvailable
Snap-ins	Add-PSSnapin	Get-PSSnapin	Get-PSSnapin -Registered

Load [sqlps](#) module:

**Import-Module** [sqlps](#) *–DisableNameChecking*

## Executing Queries

Invoke-Sqlcmd

```
[-ServerInstance <PSObject>]
[-Database <String>]
[-EncryptConnection]
[-Username <String>]
[-Password <String>]
[[-Query] <String>]
[-QueryTimeout <Int32>]
[-ConnectionTimeout <Int32>]
[-ErrorLevel <Int32>]
[-SeverityLevel <Int32>]
[-MaxCharLength <Int32>]
[-MaxBinaryLength <Int32>]
[-AbortOnError]
[-DedicatedAdministratorConnection]
[-DisableVariables]
[-DisableCommands]
[-HostName <String>]
[-NewPassword <String>]
[-Variable <String[]>]
[-InputFile <String>]
[-OutputSqlErrors]
[-SuppressProviderConte xtWarning]
[-IgnoreProviderContext]
[<CommonParameters>]
```

Copyright © 2012 [Michael Sorens](#)  
2012.07.02, Version 1.0.1  
Published on Simple-Talk.com

## Nodes in SQL Server Space

Description	Node	Default Properties	Object Type (... prefix = Microsoft.SqlServer.Management.)
SQL Server data store root	\	Name, Root, Description	....PowerShell.Extensions.SqlServerProviderExtension
Network root	\SQL	MachineName	....PowerShell.Extensions.Machine
Instances on selected machine	\SQL\machine	InstanceName	....Smo.Server
Top-level instance objects	\SQL\machine\instance	<b>list of object names</b>	System.String
Databases in selected instance	\SQL\machine\instance\Databases	Name, Status, RecoveryModel, CompatLvl, Collation, Owner	....Smo.Database
Top-level DB objects	\SQL\machine\instance\Databases\database	<b>list of object names</b>	System.String
<b>- - - Selected database nodes - - -</b>			
Tables in selected database	\SQL\machine\instance\Databases\database\Tables	Schema, Name, Created	....Smo.Table
Views in selected database	\SQL\machine\instance\Databases\database\Views	Schema, Name, Created	....Smo.View
Roles in selected database	\SQL\machine\instance\Databases\database\Roles	Name	....Smo.DatabaseRole
Triggers in selected database	\SQL\machine\instance\Databases\database\Triggers	Name, Created	....Smo.Trigger

**Top-level nodes:** \SQL, \SQLPolicy, \SQLRegistration, \Utility, \DAC, \DataCollection, \IntegrationServices, \SQLAS

**Database nodes:** ApplicationRoles, Assemblies, AsymmetricKeys, Certificates, DatabaseAuditSpecifications, Defaults, ExtendedProperties, ExtendedStoredProcedures, FileGroups, FullTextCatalogs, FullTextStopLists, LogFiles, PartitionFunctions, PartitionSchemes, PlanGuides, **Roles**, Rules, Schemas, ServiceBroker, StoredProcedures, SymmetricKeys, Synonyms, **Tables**, **Triggers**, UserDefinedAggregates, UserDefinedDataTypes, UserDefinedFunctions, UserDefinedTableTypes, UserDefinedTypes, Users, **Views**, XmlSchemaCollections

## Cmdlets implemented by SQL Server Provider

Cmdlet	Canonical alias	Other aliases	Description
<b>Get-Location</b>	gl	pwd	Gets current node
<b>Set-Location</b>	sl	cd, chdir	Changes current node
<b>Get-ChildItem</b>	gci	dir, ls	Lists the objects at current node
<b>Get-Item</b>	gi		Properties of current node
<b>Rename-Item</b>	rni	ren	Renames an object
<b>Remove-Item</b>	ri	del, erase, rd, rm, rmdir	Removes an object

Source: [Navigate SQL Server PowerShell Paths](#) on MSDN

## SQL Server drives

Use **New-PSDrive** to create drive shortcuts.

**New-PSDrive** *–Name* DB *–PSProvider* SQLSERVER  
*–Root* SQLSERVER:\sql\localhost\SQLEXPRESS\Databases

**Get-PSDrive** | ? { \$\_.Provider.Name -eq "SqlServer" } | **select** name,root

Name	Root
----	----
DB	SQLSERVER:\sql\localhost\SQLEXPRESS\Databases
SQLSERVER	SQLSERVER:\

## References

[Learning PowerShell](#), PowerShell in SQL Server [2012](#) / [2008](#),  
[SQL Server PowerShell Help](#), [Using SQL Server Cmdlets](#)

## Key Terms

<a href="#">SMO</a>	.NET classes to create applications that manage SQL Server.
<a href="#">SQLPS</a>	Module providing SQL Server support in PowerShell in the form of new cmdlets (Invoke-Sqlcmd and others) and a SQL Server provider.
<a href="#">SQL Server Provider</a>	Allows you to interact with the hierarchy of SQL Server objects just as a native PowerShell file system provider allows you to interact with files. You can navigate through SQL Server space using paths to nodes analogously to paths to files.
<a href="#">SQLPSX</a>	CodePlex project introduced prior to sqlps to provide SQL Server support in PowerShell, SQLPSX still provides some functionality not covered by sqlps.