

The purpose of the **SQL Server** sample scripts is to demonstrate the scripting capabilities of PowerShell for performing DBA tasks against SQL Server.

Important:

Scripts are provided for educational purposes only. Idera does not provide technical support for these samples. Scripts should not be used in your production environment without the permission of the SQL Server administrator.

Scenario

The scripts demonstrate various techniques for managing SQL Server using PowerShell and they can be run against a local or remote SQL Server (See Prerequisites for more information).

Each script is designed to be loaded and run using default parameters supplied at runtime. You can load each script into a compatible PowerShell editor. Please review them thoroughly before running them. When running each script, use the PowerShell Integrated Console to supply inputs for the script and review the results.

To run any script that uses SQL Server powershell cmdlets or APIs, you must first run the appropriate Initialize-ISql* script for your version of SQL Server that you wish to manage with these scripts. Some scripts also require you run to PowerShell with elevated privileges using "Run as Administrator". Please refer to the NOTES section in each script for further guidance.

Languages

PowerShell v5.0 or higher is recommended (all scripts have been tested with PowerShell v7.1).

Prerequisites

Before running this sample, make sure the following software is installed all on the same computer:

- SQL Server 2008R2, 2012, 2014, 2016, 2017 and 2019
- PowerShell v5.0 and higher
- SQLPS or SQLSERVER module depending on your version of SQL Server (see Initialize-ISqlPSxxxx scripts)
- AzureRM module (see Initialize-IAzurePowerShell script in the AzureSQLDB folder)
- WinRM Remoting Services (see NOTES section in specific scripts that require this)

What's New in PowerShell SQL Server Scripts v7.0

- New scripts have been added for the latest Azure PowerShell AZ module (*-IAz* scripts)
- A new Database script that uses the SMO .CheckTable() API to check integrity (Start-ISqlCheckTable)
- New Security scripts for performing SQL Server Assessments
- A new General Utility script for getting SQL Inventory data
- A new Initialization script for SQL 2019 to support new SQLSERVER PowerShell module
- Scripts refactored to work with PowerShell 7, .Net Core and SQL 2019
- Get-WmiObject cmdlet usage replaced with Get-CimInstance
- Get-EventLog replaced with Get-WinEvents