Powershell based Database Assessment

Last Modified 3/18/21

Nick Patti

Contents

[About 1](#_Toc39844469)

[Limitations 1](#_Toc39844470)

[Customer server 1](#_Toc39844471)

[DBA computer 1](#_Toc39844472)

[Potential Features 2](#_Toc39844473)

[Recent Features 2](#_Toc39844474)

[Instructions 2](#_Toc39844475)

# About

This process will automate the data collection portion of a database assessment. At a high level, you would copy the files down to a server in the customer environment, run the powershell script, and then copy the output to your laptop. On your laptop, you run another powershell script to compile the output into a nicely formatted excel file.

The purpose of using the powershell script is to streamline the data collection. The entire process generally takes about 15 minutes to run per server to generate the output and another 5 minutes per server to compile it. This means no more copy/paste for 100+ scripts and then adjusting formatting, column widths, etc.

# Limitations

## Customer server

1. Local Admin permissions
2. Windows Server 2008 R2 or newer
3. Powershell version 3.0 or newer
4. SQL SMO (typically included with SSMS installations)

## DBA computer

1. Must have office/Excel installed
2. Powershell version 3.0 or newer (5+ is recommended). You can check this via $PSversionTable

# Potential Features

Below are a list of potential features. If any of these would provide value, please contact me and I can work on implementing it.

* Summary/Recommendations – have standardized recommendations generated automatically for DBA review

# Recent Features

1. Added invoke-sqlcmd2 which offers more flexibility with connections (we can script in SQL authentication if anybody uses it). Also lets us set an unlimited timeout
2. Set the default timeout to 60/360 seconds for normal/slow queries to prevent any runaway queries.
3. Created a “Rerun” script that will loop through the failed scripts and attempt to run the failed scripts again with an unlimited timeout. This should be monitored carefully as the scripts will not timeout, but can help for slow running queries to make sure data is captured.
4. Updated a couple scripts to work with case-sensitive collations
5. Updated the Recent Backup script to include backup location
6. Updated the Error Log script to now include only the errors (ignoring several informational messages) to make it easier to digest/review
7. Added ability to run Mini / Performance / Config / Custom assessments alongside the full assessment
8. Added progress bars to better display progress
9. Made variables/input more user friendly by prompting for settings with defaults already defined
10. Changed the logic on when to call invoke-sqlcmd vs invoke-sqlcmd2
11. General bug fixes/code cleanup/ SQL Script tweaks

# Instructions

## SETUP

1. Copy the entire folder down to your laptop in a location you want to save all your assessments and compilations. You are free to move the compiled spreadsheets out to customer specific folders after it is compiled, but we will need a standard location for excel to process these files.
2. Copy the .zip file down to a local server in the customer environment. Extract the file; I normally place it so the structure is C:\Navisite\Powershell\Assessment, but it can be placed elsewhere or on other drives. The script uses relative positioning so it will work regardless of what drive it is placed on.
3. Navigate to the servers.txt file, and populate it with a list of servers you want it to run against. As long as you can register the instance using Windows Authentication from the server that hosts the scripts, then it should be able to collect the data. You can do a single instance, or multiple instances (the most I have tested is 10 instances at once, but it could likely do more). The list should look like below:

serverA

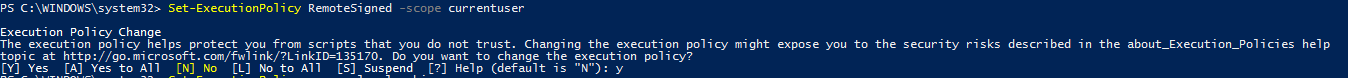
ServerB

ServerC\instanceName1

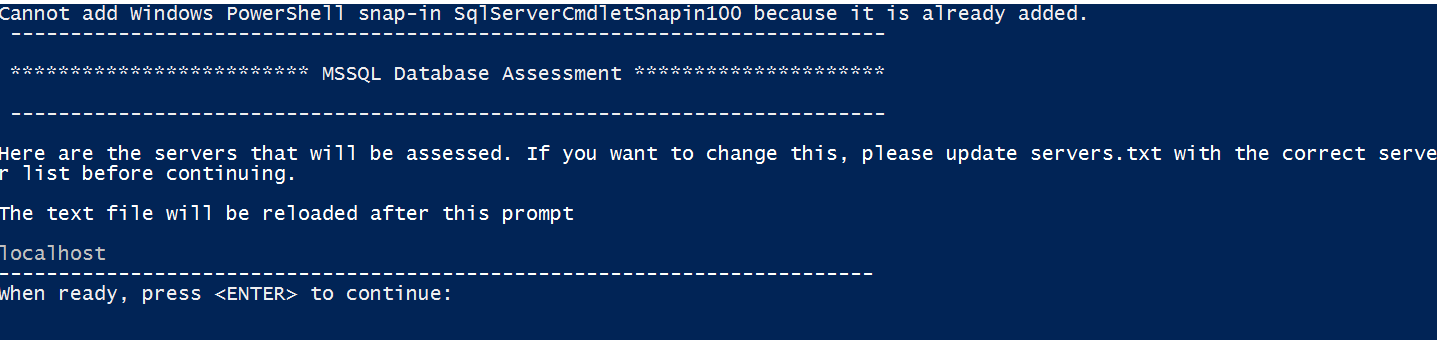
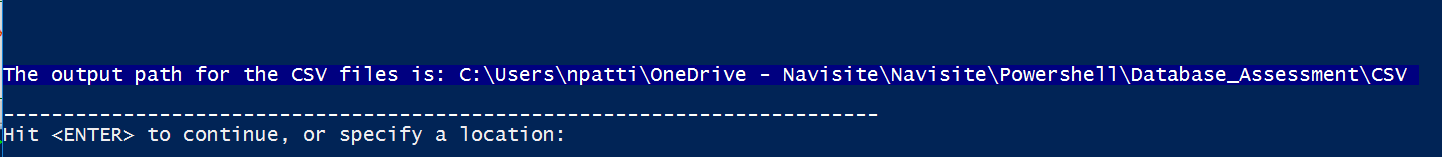
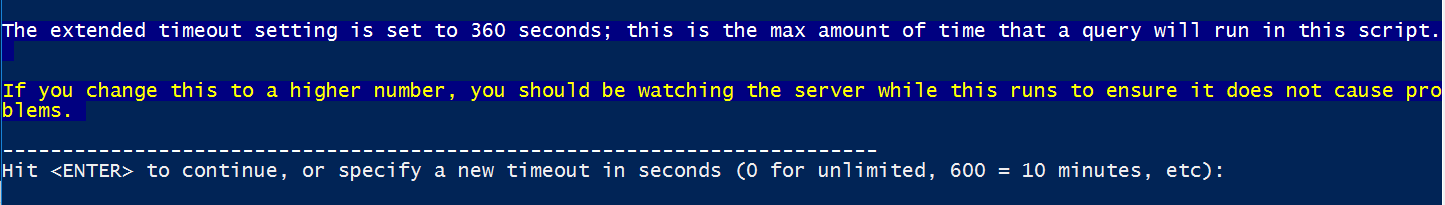
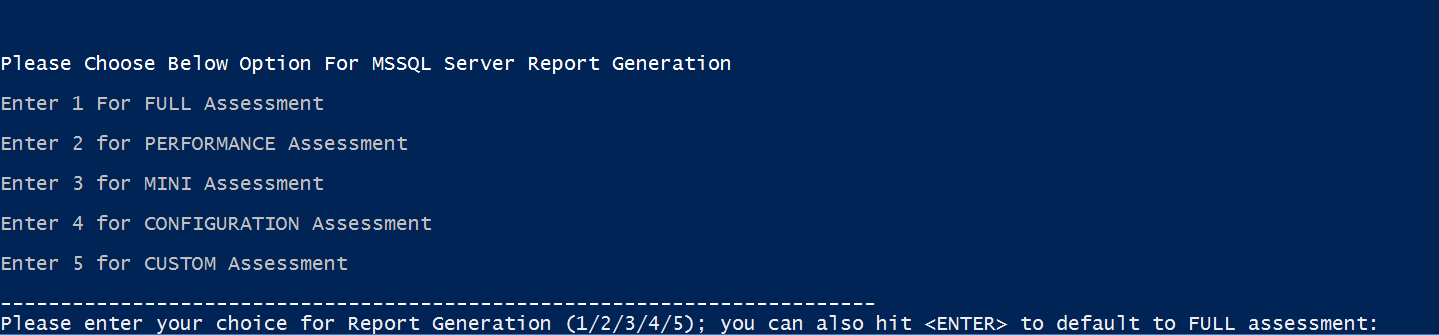
ServerC\instanceName2

serverD

1. Launch powershell as administrator and set the execution policy:
   1. Set-ExecutionPolicy RemoteSigned –scope currentuser

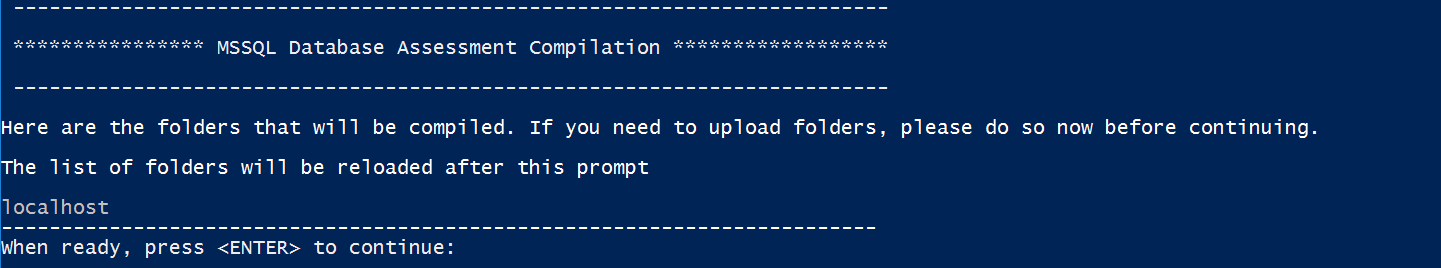
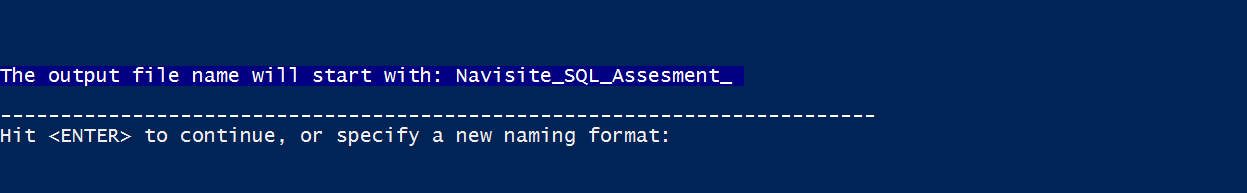
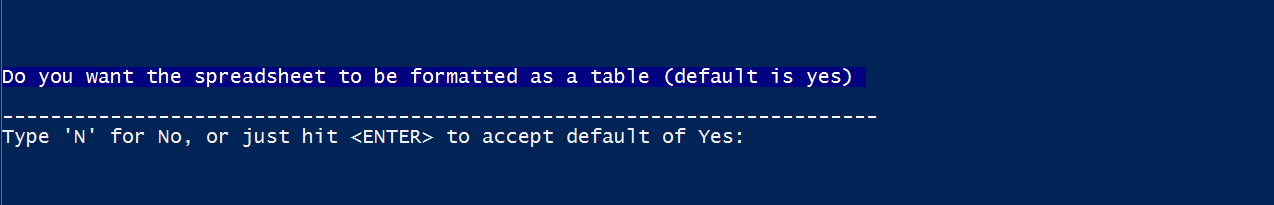
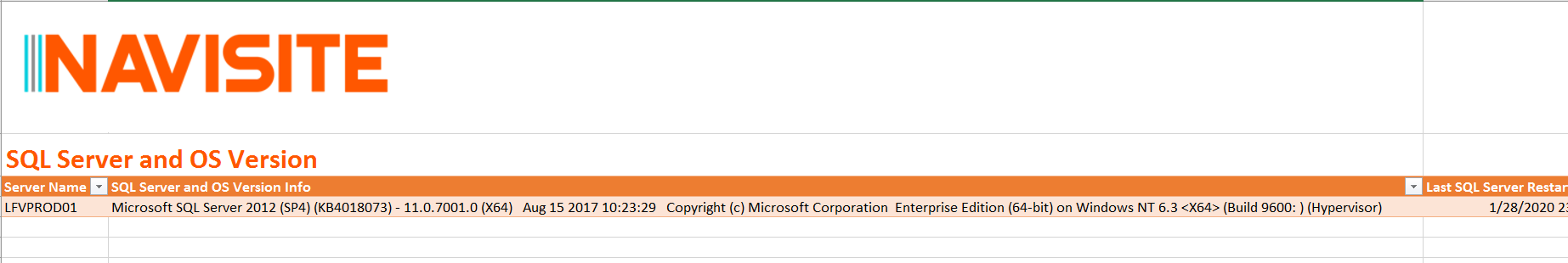


## Running the Assessment Scripts

1. Open the Run script (Powershell\_Assessment\_Run\_v5.ps1) – you can open it in Powershell ISE or by running the file.
2. Setting input options:
   1. The first prompt will show the servers it plans to run against; at this time, reference the list and if you need to change the server list, update servers.txt. After you hit <ENTER> it will reload the list of servers.
      1. 
   2. The second prompt sets the variable for where you want the CSV files located; the default location (based on where you placed the scripts) is provided as the default. Hit <ENTER> to accept, or enter a new location. It will validate the path you provided; if the path cannot be found you will need to stop the script and restart the process.
      1. 
   3. The next prompt is the extended query timeout setting. By default, most queries (90%) will run in under a minute each. There are around 10 queries that are known to take longer, and will rely on the setting defined here. The default is 360 seconds (6 minutes). You can set this to any integer, or set to 0 for an unlimited timeout. Hit <ENTER> to accept or enter a new value.
      1. 
   4. The final option prompts for which type of assessment you would like. The default option is a FULL assessment but you can also choose Performance, Mini, or Configuration.
      1. 
      2. If you wish to choose a CUSTOM assessment, you will need to update the config file with the list of scripts you want included. This can be done by updating the file CUSTOM.txt in the Configs folder; enter the three digit code for each script you want ran. You can use the MINI.txt file as an example
   5. After setting all options, the assessment will start.
3. Watch for errors, note any issues with scripts running, and review the output logs.
   1. If you had any failures, you can execute the “RerunFailedAssessmentScripts.ps1” script, which will loop through all CSV Assessment folders and rerun any failed scripts with an unlimited timeout. This can be helpful for any queries that timed out or were deadlocked; if a script fails due to a different reason (e.g. SQL Agent script on Express edition), it won’t fix the script, it will just try it again.
   2. If needed, you can rerun scripts manually and paste into the spreadsheet after you compile it.
4. Copy the CSV folder back to your laptop, placing in your directory for the assessments.

## Compiling the Data

**\*\*\*\*EFFECTIVE 1/2021\*\*\*\* We need to take an extra step in order to get the script to run. This is because of limited permissions on our laptops. You should continue to use your laptop for the compilation (not VDI).**

1. Determine to the location of your powershell scripts on your laptop; this is normally C:\Navisite\Powershell\ but depends on where you save these scripts to.
2. Launch powershell (normally, not as administrator); run the below command, using the location from step 1. This will allow you to run the rest of the compile scripts.
   1. Get-ChildItem -Recurse C:\Navisite\Powershell\Database\_Assessment | Unblock-File
3. If step 2 fails, open the text file, and change line 2 to match your assessment directory. Copy the entire command and run it in a new Powershell window. This will create a digital certificate that will sign the assessment scripts.
   1. 
4. If step 2 or 3 was successful, you can proceed as normal; otherwise you will need to troubleshoot the methods to sign or unblock the scripts.
5. Open the Compile script on your laptop (Powershell\_Assessment\_Compile\_v5.ps1) – you can open it in Powershell ISE or by running the file.
6. Setting Compilation options
   1. The first prompt when you run the scripts will show you the servers it intends to compile; review this list and copy up the CSV folders from the customer environment if you haven’t already. Pressing <ENTER> will reload the list of files to process
      1. 
   2. The second prompt will ask you for a naming convention. The default is Navisite\_SQL\_Assessment followed by the assessment type (FULL/MINI/PERF), followed by the servername (serverA, serverB, etc), and then followed by a date time stamp. Press <ENTER> to accept or provide a new naming convention
      1. 
   3. The last prompt will ask whether you want the sheets formatted as tables. The first example below shows WITH table formatting, and the second example is WITHOUT table formatting. The default is to use table formatting, so hit <ENTER> to accept or type “N” without quotes to refuse table formatting.
      1. 
      2. 
      3. 
   4. All other variables are determined by the script itself, including the assessment type, the template, the logo location, etc.
7. Once you have started the compilation, wait for it to complete. **Be sure you don’t try to copy/paste anything while it runs, as it relies on your clipboard to compile the spreadsheet**. This process usually takes about 5 minutes per server.
8. Review the spreadsheet and verify the file for accuracy and completion.