



Professional Association for SQL Server



Using SQLDiag to Troubleshoot SQL Server Problems

Brad M McGehee, Director of DBA Education

Red Gate Software

www.bradmcgehee.com

Co-Sponsored by:



Microsoft

Media Sponsors:



What We Are Going to Learn Today

- What is SQLDiag
- What Can SQLDiag Do
- How to Use SQLDiag
- A Quick Look at SQLDiag Output



What is SQLDiag

- SQLDiag is a command-line tool included with SQL Server 2005/2008 that collects general diagnostics data that can be used to help troubleshoot many different kinds of problems.
- It's main purpose is to collect a large amount of diagnostic information about a SQL Server instance as quickly as possible.
- While it is designed to collect data, it is not designed to analyze the data.
- It can be run as a console application, or as a service. Today, we will focus on the console use of SQLDiag.



What Can SQLDiag Do

- In general, SQLDiag collects this type of data:
 - Information on your SQL Server's hardware
 - SQL Server instance configuration information
 - The current “running state” of a SQL Server (DMV snapshots, etc.)
 - Windows event logs
 - SQL Server logs
 - Performance counters (including Windows and SQL Server counters)
 - Capture your Profiler black box traces
 - Capture custom SQL Server Profiler traces (a server-side trace)
- It can also be customized to collect additional information than what is provided by default with SQL Server.



How Might You Use SQLDiag

- Capture data about specific problems that you can duplicate.
- Capture data on a longer-term basis, for intermittent problems.
- Capture snapshots before and after any significant changes are made to the server.
- Capture baselines of your SQL Server instances at regular instances for comparisons.
- Send the information collected to Microsoft PSS in order for them to help resolve your most difficult problems.
- By exploring the code used by SQLDiag, and by viewing the results, you can learn a lot about how Microsoft goes about collecting data they feel is more important when troubleshooting SQL Server-related problems.



Pros & Cons of Using SQLDiag

- Pros:
 - Provides you with a single tool to collect diagnostic information.
 - Provides in-depth diagnostic information.
 - It is customizable, allowing it to better fit your unique needs.
 - Multiple instances of SQLDiag can run at the same time.
 - Works with clustered SQL Servers.
 - It can capture data on multiple SQL Server instances at the same time.
 - The overhead of SQLDiag itself is minimal, but there is a performance hit when collecting the data, similar to collecting it manually using other methods.
- Cons:
 - Hard to use, unless you like working with command lines, T-SQL, and XML.
 - Collected data is not easy to analyze, even with available tools available to help you out (Profiler, Performance Monitor, SQL Nexus, RML Utilities, PAL).



SQLDiag Security

- To run SQLDiag, you need to be a local windows administrator of the physical server, plus you must be a member of the SQL Server sysadmin group.



What Happens When SQLDiag Runs

- When you run SQLDiag, it:
 - It checks the parameters it has been given.
 - Uses an XML configuration file to find out what data it needs to collect. The default XML file is called “SQLDiag.xml”.
 - It runs a T-SQL script, called MSDiagProcs.sql, inside of tempdb, which creates some temporary SPs, and then runs them to collect much of its diagnostic data.
 - It also runs any custom “steps” specified in the XML file.
 - Some data is collected when it first runs, some is collected on a periodic schedule, and some is collected before SQLDiag is closed.
 - By default, collected data is stored in the /sqldiag folder.
- Demo files, their locations, and their contents.



Using SQLDiag at the Command Prompt

- SQLDiag can be run with defaults, or with various options:

```
{ [/?] }  
|  
{ [/I configuration_file]  
  [/O output_folder_path]  
  [/P support_folder_path]  
  [/N output_folder_management_option]  
  [/M machine1 [ machine2 machineN] | @machinelistfile]  
  [/C file_compression_type]  
  [/B [+]start_time]  
  [/E [+]stop_time]  
  [/A SQLdiag_application_name]  
  [/T { tcp [ ,port ] | np | lpc | via } ]  
  [/Q] [/G] [/R] [/U] [/L] [/X] }  
| { [START | STOP | STOP_ABORT] }  
| { [START | STOP | STOP_ABORT] /A SQLdiag_application_name }
```



Demo

- How to use SQLDiag using its default settings.
- See how SQLDiag runs.
- Briefly review the output produced (but save detailed explanation for later).



/I Configuration File

- In order for SQLDiag to know what data to collect, it uses an XML configuration file.
- If you don't specify one, then it uses the default SQLDiag.xml file.
- Besides the default option, there are two others available with SQL Server 2008:
 - SD_General.xml
 - SD_Detailed.xml
 - Or, you can find others on the Internet, or create your own.
- The command is:
 - `sqldiag /I "SD_General.xml"`
- Demo



/O Output Folder Path

- By default, the output of SQLDiag is placed in a subfolder of the folder where SQLDiag.exe is located, with a name of SQLDiag.
- If you want to choose a different location and folder name, you must specify it's path, like this:
 - `sqldiag /O "C:\sqldiag_results"`
- Demo (if time)



/N Folder Management Option

- By default, SQLDiag will ask you if you want to overwrite the any previous files.
- The /N options provides two other options:
 - 1 = Automatically overwrite the older data
 - 2 = SQLDiag automatically renames the folder to SQLDIAG_00000, SQLDIAG_00001, and so on. This way, all of your history is preserved, and is not overwritten.
- The command is:
 - `sqldiag /N1` or `sqldiag /N2`
- Demo



/X Snapshot Mode

- By default, SQLDiag starts, and continues to run until you stop it. If you add the /X option, SQLDiag will start, execute once, then automatically shut down.
- The command is:
 - `sqldiag /X`
- Demo (if time)



Switches Can be Used Together

- Of course, switches can be combined as needed, like:
 - `sqldiag /I "SD_Detailed.xml" /O "C:\sqldiag_results" /N2`
- Demo (if time)



How to Use the Data Collected by SQLDiag for Troubleshooting

- SQLDiag (depending on the XML file) outputs a series of files:
 - Black Box trace files
 - Newly captured Profiler trace files
 - Newly captured Performance Monitor trace files
 - Instance_name_MSINFO32.TXT
 - SQL dump files
 - Instance_name_sp_SQLDiag_Shutdown.OUT
- Demo (as time permits)



SQLDiag Best Practices

- Collect data on a local disk, but one that does not include the MDF or LDF files.
- Don't collect too much information for too long. Depending on the configuration, SQLDiag can collect a huge amount of data very quickly.
- Feel free to modify the SQLDiag XML files so that only the data you want to collect is collected. No point wasting resource collecting data you will never use.
- If you are collecting baseline data, just take snapshots.



Take Aways From This Session

- While SQLDiag is not for everyone, it does offer a powerful way to collect SQL Server instance information in one easy to execute command.
- SQLDiag is very flexible and can be customized, but you will have to figure out how to customize it, as it is not documented.



Find Out More

- ◆ Free E-Books:
 - ◆ www.sqlservercentral.com/Books
- ◆ Check these out:
 - ◆ www.SQLServerCentral.com
 - ◆ www.Simple-Talk.com
 - ◆ www.Red-Gate.Com
- ◆ Contact me at:
bradmcgehee@hotmail.com
- ◆ Blogs:
www.bradmcgehee.com
www.twitter.com/bradmcgehee

[For a free 14-day trial of the Red Gate SQL Toolbelt, click here.](#)

