

SQL Database Optimize

Last Modified on 02/21/2019 10:38 am CST

Issue found:

Due to a missing index and a misconfigured setting, running SELECT statements against the Master Index were holding everything else up. The TraCS vendor suggested 2 fixes. The TraCS vendor will be adding these to future baseline releases. There is a link on the bottom of this document to apply the modifications directly if you want.

Fix:

- Run the AddIndexes.sql which contain the indexes needed for the Data Database, to help speed up queries against the master index tables.
- Run the snapshotIsolation.sql to change the setting -->Snapshot Isolation:
 - By default, SQL will form a queue as it allows activity against tables/databases. This includes reads, which guarantees that the read data is always exactly the latest. In theory, this is a good idea, but can lead to where too many queries can back things up to the point of timing out. The solution is to use snapshot isolation, which makes a quick, temporary copy of a database for queries to read from, only locking tables when it needs to write out to them or change their structure. The copy is updated within seconds, so this doesn't cause any known issues in TraCS, and we recommend larger SQL agencies have this SQL feature enabled. This script will update all 4 TraCS databases.

The scripts can be downloaded with the following link:

<https://trust.dot.state.wi.us/tracs/FAQs10/SQLHelp.zip>

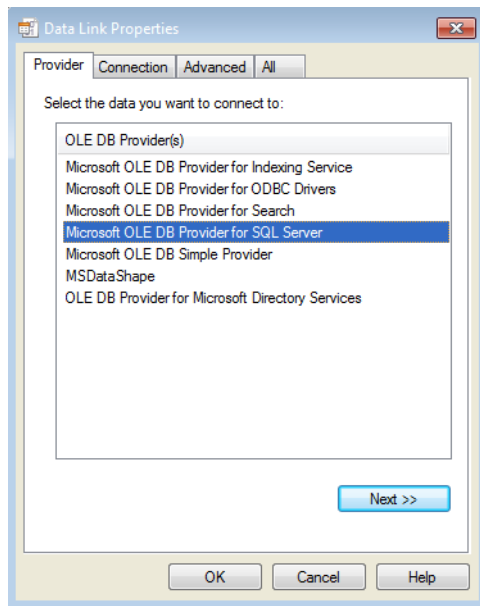
Issue: TraCS cannot connect to the SQL database. You are unsure whether a computer can successfully reach the tracs_data SQL Server database and want to test the connection without using TraCS.

Fix: Field units connect to the tracs_data database to process citation number installation requests and sometimes require a slightly different connection string depending on the network environment in which they are deployed. The TraCS configuration manager does not allow editing and testing of connect strings when running TraCS in mobile mode.

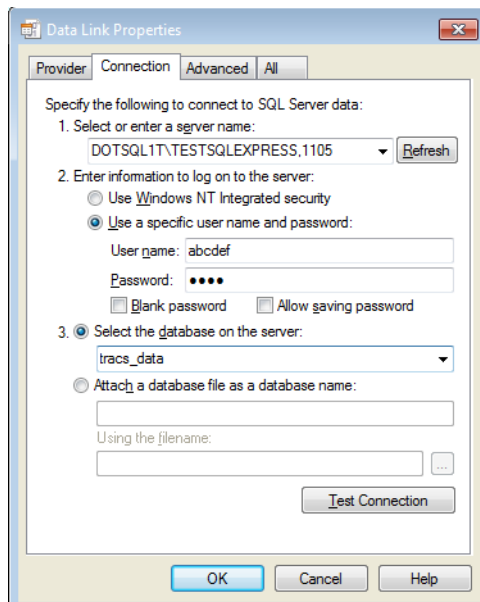
TraCS uses OLE DB to connect to SQL server. Testing the connection using the ODBC administration tool in Windows is not sufficient since ODBC connection strings are different from OLE DB connection strings.

The following instructions will allow you to test OLE DB connection to your tracs_data database.

1. Create an empty txt file on your desktop.
2. Rename the text file to testTraCS_Data.udl
3. Double click the file—it should open in OLE DB Core Services.



4. Click on the Provider tab and select Microsoft OLE DB Provider for SQL Server.
5. Click on the Connection tab and configure the settings to connect to tracs_data, for example:



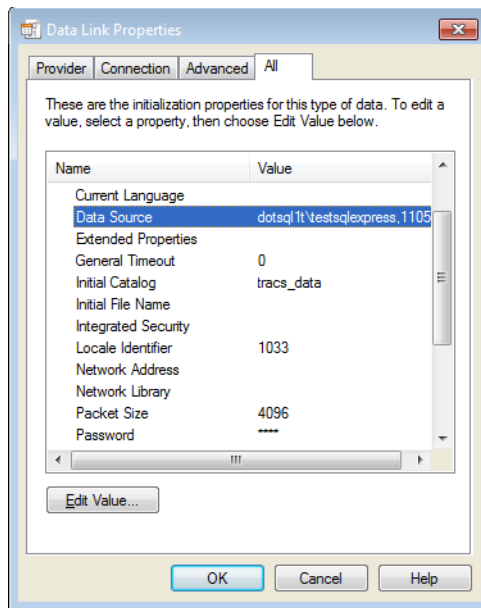
6. Click the Test Connection button. If you are able to connect successfully, skip to step 8.
7. If the connection failed, you may need to make changes on the All tab. Specific settings will be dependent on your network environment. Please work with your network administrator and your SQL Server administrator to determine the required settings. After the Test Connection button returns successful results, proceed to step 8.

Example Scenario:

The field unit is on a different domain from the SQL Server database and SQL Server is configured to only use the Named Pipes protocol.

Option 1: Establish a trust relationship between the two domains. (Using the named pipes protocol with an untrusted domain will not work.)

Option 2: Implement the TCP/IP protocol on SQL server. Then, on the All tab, enter **DBMSSOEN** in the **Network Library** field to force the connection to use the TCP/IP protocol.



8. Use your new knowledge of the correct connection string information and/or changed network environment to configure TraCS.
 - a. Rename testTraCS_Data.udl to testTraCS_Data.txt.
 - b. Open testTraCS_Data.txt in notepad to review the tested connection string.
 - c. At the master computer, update the Data_FieldUnit Db connection string with the updated information by (below are steps contained in [Task 1-6\(c\)](#) of the [Installation Guide](#)):
 - i. Log into Configuration Manager.
 - ii. Click on the Setup tab.
 - iii. Click on the Database Connection button.
 - iv. Expand the data by clicking on the ">" next to Database Connection Strings.
 - v. Click on Data_FieldUnit.
 - vi. Update the connection string as needed based on what was determined in previous steps.
 - d. Perform [Task 4.1\(c\)](#) to create a new distribution.
 - e. Run the new distribution on the field unit.

Note: If you have difficulty determining how to translate the connection string into TraCS settings, please send an email to badgertracs@dot.wi.gov requesting assistance. Be sure to **attach testTraCS_Data.udl** to your message.

Note: If your connectivity issues are due to firewalls, Proxies, or VPNs, please see this helpful article from Microsoft about default port assignments for SQL Server:

General SQL Maintenance: A SQL Server database automatically maintains indexes whenever insert, update, or delete operations are made to the underlying data. As you perform inserts updates and deletes, your indexes will become fragmented both internally and externally. Internal fragmentation is when you have a high percentage of free space on your index pages, meaning that SQL Server needs to read more pages when scanning the index. External fragmentation is when the pages of the index are not in order anymore, so SQL Server has to do more work, especially in IO terms to read the index. If your indexes become too fragmented, at best, your queries will be less efficient but at worst, SQL Server will just stop using the indexes all together, meaning virtually all queries would have to perform a table scan or clustered index scan. This will hurt your performance a lot!

When you reorganize an index, SQL Server uses the existing index pages and just shuffles data around on those ages. This will alleviate internal fragmentation and can also remove a small amount of external fragmentation. It is a lighter weight operation than rebuild and is always online. When you rebuild an index, SQL Server re-sorts the data of the index and uses a new set of index pages. This will alleviate both internal and external fragmentation but is a heavier weight operation and by default causes the index to go offline, although it can be performed as an online operation, depending on your SQL Server version and settings.

Please do not expect to have 0 fragmentation after a Rebuild however. Unless you use a MAXDOP query hint, SQL Server will parallelize the rebuild operation and the more processors involved, the more fragmentation there is

likely to be, because each processor or core, will rebuild their section or fragment of the index individually, without regard for each other. This is a tradeoff between best fragmentation levels and time taken to rebuild the index. For near 0 fragmentation, use MAXDOP 1 and sort the results in TempDB.

Steps to rebuild index. [https://technet.microsoft.com/en-us/library/ms187874\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms187874(v=sql.105).aspx)

To reorganize all indexes in a table:

1. Backup the tables being reorganized.
2. In Object Explorer, Expand the database that contains the table on which you want to reorganize the indexes.
3. Expand the Tables folder.
4. Expand the table on which you want to reorganize the indexes.
5. Right-click the Indexes folder and select Reorganize All.
6. In the Reorganize Indexes dialog box, verify that the correct indexes are in the Indexes to be reorganized. To remove an index from the Indexes to be reorganized grid, select the index and then press the Delete key.
7. Select the Compact large object column data check box to specify that all pages that contain large object (LOB) data are also compacted.
8. Click Ok

Issue Found: *If seeing issues with tempdb page contention*

Fix:

- <https://helpdesk.teginc.com/documentation.aspx?Page=Teg+Public+Safety+Software+System+Architecture+And+Installation+Guide+For+Tracs+And+Mach+Attachment+>
==
- **DO NOT CHANGE RECOVERY MODE IF INTENDING TO DO LOG BACKUPS AND POINT IN TIME RESTORES**

Issue Found: *If seeing issues with small queries getting consistently run in parallel and causing locking*

check that MAXDOP and cost threshold for parallelism are set correctly (i.e. not set to default)

Fix:

- <https://support.microsoft.com/en-us/help/2806535/recommendations-and-guidelines-for-the-max-degree-of-parallelism-confi>
 1. e.g. Our MAXDOP is set to 6 and Cost Threshold to 50, on a dual socket with 12 physical cores total.

Help with Debugging SQL

There are also a bunch of free scripts available which can help with quick debugging

- e.g. <https://www.brentozar.com/first-aid/>
 - sp_Blitz will show general server wide issues
 - sp_BlitzIndex will show index issues

Question: *Should we use Full Recovery or Simple mode for backups?*

- Backup procedures are really at the discretion of the law enforcement agency based on how much risk they

are willing to incur. This article may be helpful in deciding whether to go to Simple:
[https://technet.microsoft.com/en-us/library/ms175987\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms175987(v=sql.105).aspx)

Question: *How do I truncate the SQL Server transaction log that is maintained for backup recovery?*

- This article explains when log truncation should occur: [https://technet.microsoft.com/en-us/library/ms189085\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms189085(v=sql.105).aspx)

Issue Found: *There are known issues with TraCS and SQL Server 2016*

- SQL Server 2016 deprecated some data types. The workaround until fixed by TraCS National is SQL Server 2016 works with as long as compatibility mode is turned on for the TraCS databases.

–GLP 11/27/2017
