TrendWorX64 Logger Performance Optimization

APPLICATION NOTE

September 2013

Description: Guide to TrendWorX64 Data Logging Performance Optimization
OS Requirement: Vista x64/ Windows 7 x64/ Windows 8 x64/ Windows Server 2008 x64/ Windows Server 2008 R2 x64/ Windows Server 2012
General Requirement: Familiarity with TrendWorX64 Logger Configurator and Microsoft® SQL Server®

Introduction

This document is meant to provide a database organization and management scheme that allows you to control the database storage and table creation for better performance of TrendWorX64 Logger. The Table(s) Management tab in the TrendWorX64 Logger Logging Group allows you to set up an "archive" scheme, which can optimize historical replay and data logging.

Managing the Database Table Organization

Scheduling Historical Table Management

For example, you could configure data-logging as follows:

- The data-logging load is 250 tags per second.
- TrendWorX64 Viewer historical viewers are configured to start up with an initial display of 1 hours' worth of data.
- The historical data will be maintained for 7 days.

In the case above, you should configure TrendWorX64 Logger to create four tables per day by creating a new table every 6 hours and maintaining a total of 28 tables. This approach will:

- Speed up initial historical replay, because only the most recent table will be opened and the size of each table will remain reasonable.
- Speed up data logging, because each table will be a reasonable size.
- Improve system maintenance and backup tasks by avoiding the creation of enormous database tables.

In general, for frequent data logging it is recommend that you configure the data logging activity such that the size of the historical tables will not exceed 80 MB.

Other Considerations

For non-networked applications in which all GENESIS64 software is running on a single computer, making sure that the database is not on the same hard drive as the HMI displays can improve performance. Frequent display switching requires disk access, which can slow down data logging.

Data logging requires additional memory resources. In addition, databases allow you to control the amount of memory used. In the case of Microsoft SQL Server, memory is handled in a centralized fashion using the Microsoft SQL Server Management Studio.

Configuring Memory Management for Microsoft SQL Server

Microsoft SQL Server introduces a dynamic memory-management scheme that uses system memory based on need and availability. In any case, it is recommended that for Microsoft SQL Server data-logging applications you do not allocate more than two thirds of the total physical RAM available for database server memory usage. In addition, it is critical that the PC on which Microsoft SQL is installed has virtual memory at least two times the total amount of physical RAM available.

Data-Storage Requirement

Data storage requirements refer to the size of the target database once data collection and data storage begin. Microsoft SQL Server and Server Express can be configured to increase the database size automatically, if needed, while data logging is activated.

Microsoft SQL Server Express is a freely downloadable and distributable version of Microsoft's SQL Server relational database management system, comprised of a database specifically targeted for embedded and smaller-scale applications. SQL Server Express lacks a concurrent workload-governor to "limit performance if the database engine receives more work than is typical of a small number of users." It has a number of technical restrictions (which make it unsuitable for large-scale deployments) including:

- Maximum database size is 10 GB per database in SQL Server 2012 Express and 2008 R2 Express, 4 GB for SQL Server Express 2008 Express and earlier; compared to 2 GB in the former MSDE). The limit applies per Express database (MDF file, log files excluded).
- No SQL Server Agent service.
- Hardware-utilization limits:
    - Single physical CPU, but multiple cores allowable.
    - 1 GB of RAM (runs on a system with any RAM amount, but uses only 1 GB at most).

Note: For full comparison see the Microsoft web pages such as http://msdn.microsoft.com/en-us/library/cc645993(v=SQL.110).aspx

Microsoft SQL Server Editions (non-Express Editions) are not bound by the restrictions above. Databases can span hard drives and computers for a single database that is terabytes in size.

To compute an estimate of required database size, on a per-logging group basis, use the following equation:

Total Number of Records = (Number of Tags x Logging Interval) / (36 x Data-Collection Rate)

Required Size = (Total Number of Records) x (1024 Bytes/Record) x (1.25)

Example

For example, assume you want to data log 100 tags with a data-collection rate of 1 second and a logging interval of 24 hours (converted to seconds). This will require the following:

- Total Number of Records = (100) x (24 hours x 60 min/hr x 60 sec/min) / (36 x 1 sec) = 240,000 Records
- Required size = (240,000) x 1,024 x 1.25 = 307 MB per day

Note: When you are logging data to an Oracle i9 database, you can expect a 20 percent increase in database size because of the data types supported by Oracle. When logging to Microsoft SQL 2005 or 2008 databases, you can expect a 20 percent increase in database size.

However, the above example is a worst-case scenario that assumes all tags are updated every second from the OPC server. In reality, OPC servers are event-based, so they will update only when the value of the tag changes. The TrendWorX64 Logger also supports various data-logging filters, as well as Absolute/Percent Deadband. These can be used to reduce the database-storage requirements effectively.

To help you in maintaining the target database size and avoiding situations in which the database size grows uncontrolled, the TrendWorX64 Logger supports a cyclic database table structure, which if configured will always maintain the configured number of tables into the database by overwriting existing data as it rolls over each table. Refer to the TrendWorX64 Logger Configurator documentation for more information.

In addition, keep in mind that ordinary database maintenance tasks, such as defragmentation, compaction, and table index updates, may require additional space on the target computers and drives, which can vary depending on the database tasks under operation. It is recommended that you allow for additional space in these cases.

Installing Microsoft SQL Server Express from the GENESIS64 DVD

The TrendWorX64 Logger supports data logging to Microsoft SQL Server. The GENESIS64 product installation DVD will automatically install Microsoft SQL Server 2012 Express, a powerful and reliable data management system that delivers a rich set of features, data protection, and performance for embedded applications, lightweight Web Sites and applications, and local data stores; as well as SQL Server Management Studio. SQL Server Management Studio is an administrative tool to be used with Microsoft SQL Server databases.

Deploying Microsoft SQL Server requires several simple steps at startup that are very important to overall data logging and system performance.

During Installation

When installing Microsoft SQL Server, (if possible) specify a different drive for the "master" and "tempdb" databases of Microsoft SQL Server from the location of the actual historical database. If installing Microsoft SQL Server 2005 or 2008, please install the latest service pack, available from Microsoft.

Note: When installing Microsoft SQL Server 2008, remember that TrendWorX64 Version works with Microsoft SQL Server 2008 in single-instance mode. Starting with Microsoft SQL Server 2008, you can install multiple instances of Microsoft SQL Server 2008 on the same computer. Each one is a separate install, having its own name (instance name). Thus, clients connect to it via instance name or server name.

After Installation

After installing Microsoft SQL Server, go to the Microsoft SQL Server Management Studio and:

- Set the "master" database size to 80 MB and the transaction log to 20 MB.
- Set the "tempdb" database size to 120 MB and the transaction log to 20 MB.

The initial sizes of "masterdb" and "tempdb" are small. Microsoft SQL Server will increase them on the fly, but it will consume resources. It is always better to start from a decent initial size. To perform this task, select each database above, right-click to display the context menu, and select Properties. This opens the Database Properties dialog box, shown below. Select the Files page which will enable you to adjust the database sizes.

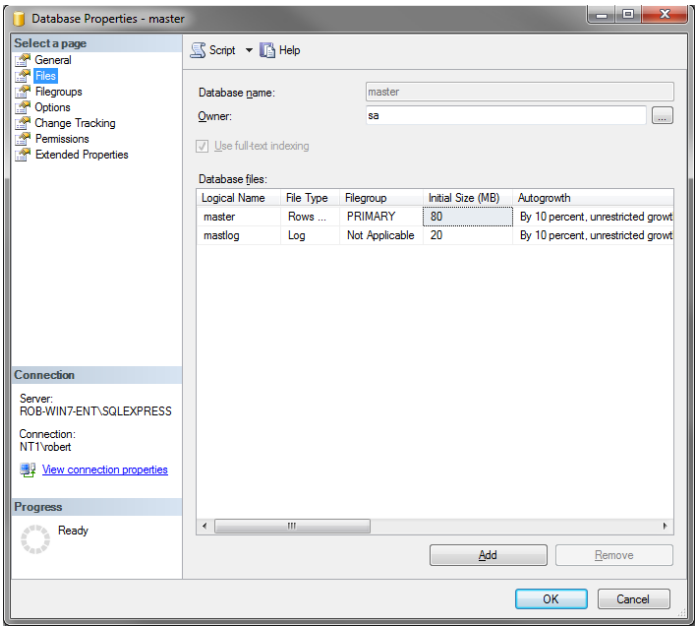

Figure 1 - Database Properties Files Page Dialog Box

Once the sizes have been modified, click OK to apply the changes.

Setting Server Memory Options

By default, Microsoft SQL Server will be configured to use all available PC memory, which can cause problems with a heavy-duty load or an inadequately equipped PC.

You can use the Microsoft SQL Management Studio to adjust the server memory settings. For Microsoft SQL Server installations, start the Microsoft SQL Management Studio, select the Microsoft SQL Server, right-click, and then select Properties. This opens the Server Properties dialog box. Select the Memory page where you can modify the server memory settings using the Memory tab of the SQL Server Properties dialog box.

SQL Server Properties

It is recommended that you use as a maximum value no more than two thirds of the total physical RAM available on the PC. For example, assume that the PC has 6 GB of RAM. Set the Microsoft SQL Server RAM to a maximum value of for 4 GB of RAM.

Note: You may have to stop and restart the Microsoft SQL server for these changes to be applied.

Controlling Transaction Log size

For Microsoft SQL 2008 databases, go to the historical database Properties dialog box and select the Options page, as shown in the figure below. Under Recovery Model, select Simple from the drop-down list. This enables Transaction Log truncation so that the log will not increase in size.

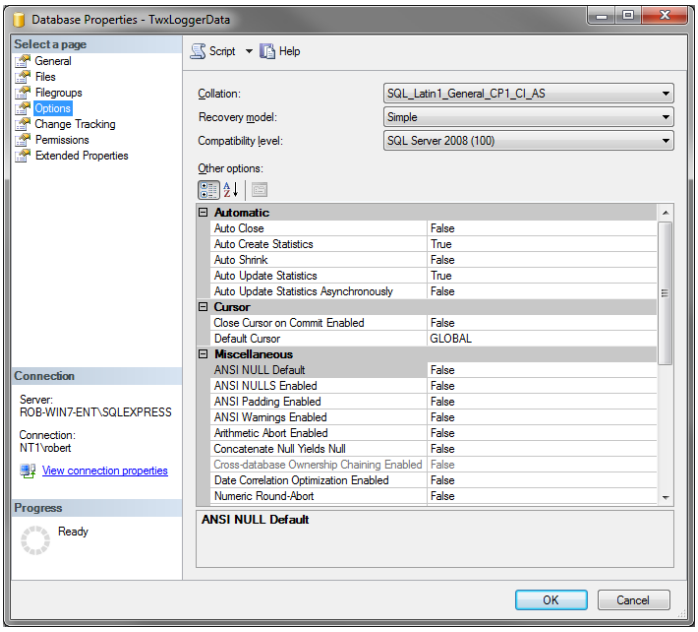

Figure 2 - Database Properties Files Page Dialog Box

Limiting the Transaction Log Size

For Microsoft SQL 2008 databases, make sure that the Truncate Log on checkpoint option is checked in the Options page of the database Properties dialog box. In addition, the Auto Close should be set to False for fast logging; Auto Create Statistics and Auto Update Statistics options should be set to True.

Adjusting PC Virtual Memory Settings

Make sure that the PC has adequate virtual memory (paging file) for Microsoft SQL Server and all other Windows applications. For example, assume that the PC has a total of 6 GB of RAM. Set Microsoft SQL Server to use a maximum of 3 GB RAM. Make sure the paging file is at least 6 GB.

To adjust the virtual memory settings on your PC, select the My Computer icon on the desktop, right-click and select Properties from the pop-up menu. Once the System Properties dialog box is displayed open the Advanced tab of Advanced System Properties, as shown below.

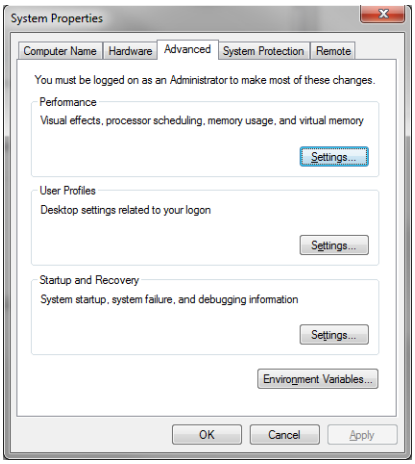

Figure 3 - Windows System Properties Dialog Box

Click the Settings button in Performance to open the Performance Options dialog box.

Click the Change button to display the Virtual Memory settings dialog box. Here you can modify your virtual memory settings.

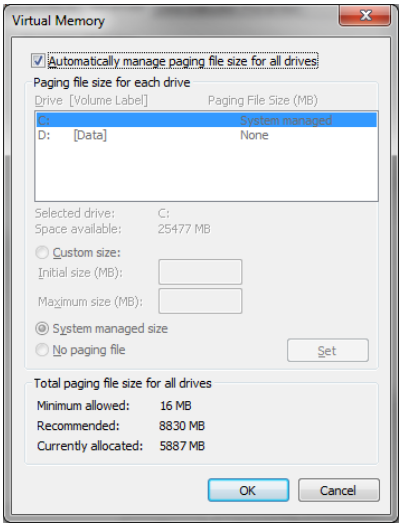

Figure 4 - Virtual Memory Settings Dialog Box

Microsoft SQL Server will use as much memory as it needs to boost its performance. If you have chosen to install SQL Express instead, remember that SQL Express will only use up to 1 GB of RAM (see section on Data-Storage Requirement above). However, if it has to give back physical RAM, it will substitute with virtual memory. In some cases, it will use more virtual memory than physical RAM. Also note that Microsoft SQL Server 2008 uses less memory than Microsoft SQL Server 2000 and performs better, at the expense of using more disk space.

Optimizing Data-Logging Configuration

To optimize the data-logging configuration:

- Use deadband or a data-compression filter. This can drastically improve performance while saving disk space usage.
- Log to disk as slowly as possible. That is, do not write to disk every sample, unless your data collection rate is 15 minutes or longer.
- Create the historical database with a good initial size (30-50 percent of the estimated final size). Microsoft SQL Server will increase the size on the fly, but it will consume resources. It is always better to start from a decent initial size.
- Avoid using many database groups to log to the same database. Each database group uses an independent connection to the database, which consumes more resources.
- Use fewer logging groups (with more tags per group) rather than many logging groups (with few tags per group). Microsoft SQL Server will have fewer tables to maintain, so performance will be better.
- Do not let the size of data-logging tables exceed 80 MB. Create a new table instead. The bigger the tables get, the slower data logging will be.
- Run Microsoft SQL Server in Windows Server rather than Windows Workstation to increase performance.

Managing Database Storage Requirements

Within Microsoft SQL Server Management Studio, you can set up maintenance plans that can defragment the database and optimize and recompute table index information. Ideally, these plans should be activated during inactive data-logging periods. When applied to a database on a periodic basis, they can significantly improve database performance and reduce database disk space usage.

Example Configurations

- **Example 1.** If you wanted to data log 100 signals per second, you would write to disk every minute or 60 samples, creating a new table every 4-6 hours.
- **Example 2.** If you wanted to data log 500 signals per minute, you would write to disk no faster than every 5 minutes or five samples, creating a new table once a day or once a week.

Data Logging Load Distribution

The TrendWorX64 Logger tag database uses a hierarchical organization, which involves database groups and logging

groups within each database group. This enables you to organize data-logging activity logically, in smaller loads, and in your desired areas of interest. It is recommended that you do not add more than 500 tags per database and/or logging group (depending on available hardware and the overall data-logging load, as well as the database type). This way the database tables will become more manageable, and the data-logging activity will not require extensive system resources. In general, for frequent data logging it is recommended that you configure your data logging activity such that the historical tables created will not exceed a size of 80 MB.

Note: Microsoft SQL Server has database-maintenance capabilities even when active connections are open. In certain cases, for an above-average data-logging load it may be preferable to create databases with a large initial size in order to avoid frequent database expansion operations.

Note: When data logging to Oracle, it is recommended that the Oracle 9i client tools are installed and utilized on the TrendWorX64 Logger and that no data-logging group contains more than 200 tags. It is not recommended to use Oracle for fast data-logging rates. It is also recommended that writing to the Oracle database is kept as slow as possible. For example, if the data-collection rate is 1 minute, writing to disk should be set to be no faster than 5 minutes.

Database Access Scheduling

The TrendWorX64 Logger has flexible options for controlling the frequency of data logging to the database. Using these tools, you can control the frequency of database writes and spread them in such a way that not all groups write to the database at the same time. Internally, TrendWorX64 Logger tries to achieve this to a certain degree, but the configuration gives you even more flexibility in doing so. In terms of configuration, TrendWorX64 Logger allows you to control the interval at which data will be logged to the database, regardless of the data-collection interval.

Scheduling Database Access for Load Balancing

Using the Logging Group page in the TrendWorX64 Logger you can configure the system:

- On a per-group basis.
- To write to the database when a certain number of records have been collected.
- At a certain time.
- On interval.
- On a condition.
- Using a combination of options.

This way, you can "spread out" the write activity to the database and balance the data-logging load more successfully.

Note: TrendWorX64 Logger should not access the database for writes any faster than every 5 seconds.