

Technical Note

DataLogger Tips and Tricks

1. Overview

This document intends to provide users with tips and tricks for using the DataLogger Server Plug-In. At this point, users should verify that the data source being used is supported. For more information, refer to [DataLogger Option for KEPServerEX OPC Server](#).

2. Tips and Tricks

2.1 Logging Data Consumes Disk Space

Both the amount of data and the frequency at which it is logged will quickly consume disk space. Users must be aware of how their logging choices impact disk space consumption. For example, in testing, Kepware Technologies used a wide data format to log 1000 points of data every 10 milliseconds. It took about three days to consume 5GB of disk space.

Users that require data to be logged at this rate have no way to avoid this; however, most applications require much slower rates.

2.2 Designing the DataLogger Project

The best way to plan the design of a DataLogger project is to start by determining what will be done with the data being logged. Knowing this enables users to create data tables in the most optimized format and to use the Data Source Names (DSN) and DataLogger project to populate tables accordingly. For example, a user that intends to log product data by shift (in order to generate a weekly production report) would create a table with records for Shift #, Up Time, Down Time, Parts Produced, Parts Rejected, and Date/Time Stamp.

The following database design websites are recommended for more information.

- [Five Simple Database Design Tips](#)
- [Tips on Database Design and Use](#)

2.3 Optimizing Data Logging

DataLogger's threading model can be tailored to suit users' specific needs. DataLogger interprets each DSN as a queue to create a new logging thread. For example, if "LogGroup01" uses a DSN named "MySQL_ONE" and "LogGroup02" uses a DSN named "MySQL_TWO," then DataLogger will generate two distinct logging threads even though both "MySQL_ONE" and "MySQL_TWO" may point to the same database. If "LogGroup01"

and "LogGroup02" both use "MySQL_ONE," they will take isochronous (or equal opportunity) turns processing data.

This scalability allows users to log tens of thousands of points simultaneously. This example created two log groups that both logged to distinct DSNs pointing to the same MySQL database. Logging was completed at approximately 50,000 columns per second.

Important: Having more than one DSN connection to the same data source increases the chance of one connection locking the data table in order to insert a record while another connection also attempts to insert a record. This will result in both an error and a loss of data.

2.4 Logging to Microsoft Excel

DataLogger supports logging to existing tables in Excel files via the Microsoft Access Link Tables feature and the ODBC Driver. For information on logging to an existing table in an Excel file, refer to the instructions below.

Note: DataLogger cannot be used to create new tables in Excel.

1. To start, create a new Access DSN that points to the Access database where the Link Tables process will be performed for the specific Excel spreadsheet.

Note: It is recommended that users do not utilize a pre-existing Access DSN for the same database. In one test case, doing so met with poor results.

2. Next, open the associated Access database and then click **File | Get External Data**. Then, select **Link Tables**.
3. Browse to and select the Excel file.

Project Design Tip: Choose descriptive names for the table headers in Excel. This will make it easier to map the OPC server items to the table headers through the Map Item Fields dialog.

2.5 Data Source Handling of Timestamp and Date/Time

Each data source handles the date and time a little differently; however, in some cases, the formatting will be very different.

- **Microsoft Access Timestamp:** Access stores date/time internally as a double precision floating point number. Its syntax is `xxxx.yyyy`. Because the time value is "00:00" at midnight, the `yyyy` part will not be displayed in the Access user interface.
- **Microsoft SQL Timestamp and Date/Time Differences:** "TIMESTAMP" is interpreted by MS-SQL internally as a binary blob used to process image information. It has nothing to do with OPC timestamps, dates, or time information. Users should always utilize "SQL_DATE" or "SQL_DATETIME" to log OPC timestamp information.

Note: DataLogger can either log data to an existing database table or automatically create new database tables. When creating tables automatically, it will suggest a set of column names and SQL data types that are compatible with the specified database. For example,

when DataLogger detects a request to create an "SQL_TIMESTAMP" column in an MS-SQL database, it will post a warning to the Event Log and override the column type (substituting "SQL_DATETIME" instead).

2.6 Error Handling – Database Full

Each data source will have a slightly different error response when the database is full. KEPServerEX will post a message to the Event Log indicating that the RecordSet could not be queried. **It is recommended that users monitor the "_DataLogger.<Logger Group Name>._Error" Tag from their OPC client application to see when an error has occurred. An error state is indicated when the tag's value changes from zero to one.**

2.7 Monitoring the Data Insertion Rate

Each log group has System Tags that specify the amount of time it takes to open the record set for the insertion of a record. If this time is longer than the rate at which data is triggered to log, it is likely that records will be lost. For example, the DataLogger is logging on data change. If the device values change every 250 milliseconds, but the insertion rate is 400 milliseconds, records will be lost if a change occurs while waiting to insert a record.