

Technical Note

DataLogger Tips and Tricks

The purpose of this document is to provide users with tips and tricks for using the DataLogger advanced plug-In for KEPServerEX®. Please verify that the data source is currently supported before using DataLogger. 🌐 *For more information, refer to the DataLogger webpage.*

1. Logging Data Consumes Disk Space

Both the amount of data and the frequency at which it is logged can consume significant disk space. Users must be aware of how logging choices impact disk space consumption. For example, when KEPServerEX 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.

Most applications do not require such fast rates, but if data is required to be logged at this rate, disk space usage cannot be avoided.

2. Designing the DataLogger Project

The best way to design a DataLogger project is to determine how the logged data will be used. 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 who intends to log product data by shift (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.

🌐 *For more information on database design, see [Five Simple Database Design Tips](#).*

3. Optimizing Data Logging

DataLogger's threading model can be tailored to suit 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," 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 take isochronous (or equal opportunity) turns processing data.

This scalability allows users to log tens of thousands of points simultaneously. In this example, two log groups were created 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 to insert a record while another connection also attempts to insert a record. This results in an error and a loss of data.

4. Logging to Microsoft® Excel

DataLogger supports logging to existing tables in Excel files via the Microsoft Access Link Tables feature and the ODBC Client driver.

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

1. 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:** Do not use a pre-existing Access DSN for the same database; doing so may yield poor results.

2. Open the associated Access database.

3. Choose **File | Get External Data**.

4. Select **Link Tables**.

5. Browse to and select the Excel file.

- **Tip:** When designing a project, choose descriptive names for the table headers in Excel. This makes it easier to map the OPC server items to the table headers through the Map Item Fields dialog.

5. Data Source Handling of Timestamp and Date/Time

Each data source handles date and time a little differently; however, in some cases, the formatting is 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 is not 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 suggests 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 posts a warning to the event log and overrides the column type (using "SQL_DATETIME" instead).

6. Error Handling — Database Full

Each data source has a slightly different error response when a database is full. KEPServerEX post a message to the event log indicating that the RecordSet could not be queried. It is recommended that users monitor the "_DataLogger.._Error" tag from the 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.

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.