

# **History in the Making**

## **An Intimate Look at Data Collection and Reporting**

Joseph C. LaRosa and John C. Clavette  
CDM  
One Glen Lakes  
8140 Walnut Hill Lane, Suite 1000  
Dallas, Texas 75231

### **Introduction**

Tracking historical data has always been an important aspect of the water and wastewater industry. Plants use recorded data for a number of reasons: troubleshoot equipment problems, schedule maintenance procedures, determine efficiencies of specific plant equipment, and, most importantly, validate the quality of the product being produced. Many regulatory agencies have established stringent data collection and storage requirements to which plants must adhere. Data storage can be accomplished online for immediate viewing purposes or archived onto external media (e.g., CD, DVD) for retrieval at a later time. The online storage requirements vary from state to state. Generally, online data storage must be maintained for a minimum of 1 year. However, in some cases, utilities have accommodated storage of data for periods of up to 10 years or even longer.

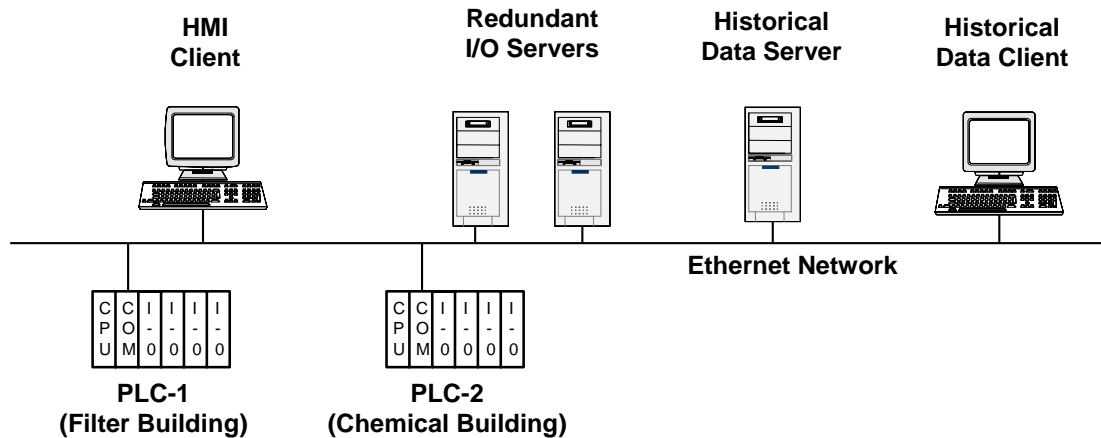
The stored data has a significant importance as it is used to confirm that the quality of the plant effluent meets the mandated specifications set forth by the regulatory agencies. While the overall storage and maintenance of this data may sound like an overwhelming task, the process has been much improved and streamlined since the days of piling stacks of circular charts in the corner of a room. With new hardware and software technologies being researched and developed at astonishing rates, industries are finding more efficient methods of collecting, storing, and presenting their data.

This paper investigates various technologies and methodologies for data collection, storage, and reporting. First, an overview of a typical supervisory control and data acquisition (SCADA) system architecture is presented. Then, database packages that are commonly used for data storage are presented. The remaining sections of the paper discuss many of the important aspects of creating reports, such as common report types, how to establish the required set of data to collect, different methods of data collection, and report development tools.

### **System Architecture**

Prior to delving into the details associated with historical data, it is necessary to first lay down the foundation and to explain the SCADA system components that play a role in making data collection possible. A typical system architecture for a modern-day SCADA system includes various combinations of the following six components: programmable logic controller (PLC), human-machine interface (HMI) I/O server, HMI client, historical data server, historical data client, and a network connecting these components together. One or more of each component can be used, and some components may be integrated together as one machine. For example, one can implement a SCADA system that

combines the HMI client and historical client by implementing the functionality of both components on one personal computer (PC). An explanation of the functionality of each SCADA system component follows.



### ***PLC***

A PLC is an industrial grade programmable device that performs the manual and automatic control functions associated with one or more plant processes and solves important calculations, such as flow totalizations and runtime totalizations. Field equipment is typically hardwired to the PLC via input and output modules, known as I/O modules. In SCADA applications, one or more PLCs are installed at strategic locations at the plant. For example, at a water treatment facility, an engineer may choose to provide one PLC located at the filter building and another at the chemical area. The filter PLC would be programmed for manual and automatic operation of filter valves, backwash pumps, and air scour blowers while the chemical PLC would be programmed for control of the chemical feed pumps and transfer pumps.

### ***HMI Client and HMI I/O server***

The HMI client is a graphical user interface that allows operators to monitor and control plant equipment by viewing graphics, clicking on buttons, and entering process setpoint information. The HMI client allows the user to view and manipulate nearly all information that is available in the PLC.

The HMI I/O server performs several important functions. First, it polls data from the PLCs and updates the HMI client graphics with the newly acquired information. Second, it transfers all updated operator commands and setpoints from the HMI graphics to the PLC immediately after the change is made. Third, it provides updates to the historical data server. Because of the range of essential functions it performs, a failure of the HMI I/O server would be a major inconvenience, if not disastrous, to plant operations. Therefore, redundant I/O servers are typically implemented. If a primary I/O server fails, the standby server is available to provide data to the HMI graphics and historical data server.

The HMI I/O server and client are normally part of an integrated software package that is comprised of the graphics software, the I/O server software, and the hardware drivers. This package may also contain historical database software that will be discussed in the sections to follow.

### ***Historical Data Server***

The Historical data server collects information from the I/O servers and stores it for future use. The architecture of the historical data server depends upon the required level of redundancy as well as the budget available for implementation of the design. For example, if funds are limited, the engineer will likely design the historical data server using a single machine. However, a water treatment facility that is required to report turbidity data to the state once per month will require a system with high reliability. In this case, if funds are available, the engineer may design the server using a redundant server, allowing data collection to continue and data storage to be secure even in the event of a hardware failure. This is discussed further in the historical database section below.

In some cases, the reliability of the historical server is the top priority. A system owner may request that redundant server machines are separated from each other in different buildings. A server of this type would prevent data loss in the event a building is destroyed due to fire or other incident.

For the historical data server to record the data and make it readily available for use, the proper historical database (or historian) software must be installed. This software is focused on in more detail below.

### ***Historical Data Client***

The historical data client is a machine that contains a set of applications allowing operators to access historical data via a user-friendly interface, usually in the form of graphical trends or reports. These applications may reside on multiple machines in the system. One owner may choose to have a stand-alone machine for viewing and printing reports, while trends will be accessed from the HMI client nodes. Other owners may require that all trends and reports be accessible from the HMI client machines. This is driven by owner preference and the availability of funds.

It is important to know the type of trend information and report information the customer needs prior to specifying the historical data client software. For example, if a customer needs to display historical trends and some simple reports with min/max/average calculations, then the historian software may have sufficient built-in tools to accomplish this. More complex queries or customized reports, however, may require more powerful tools.

### ***Network***

In most cases, each of the SCADA system components explained above resides on the same local area computer network (LAN). Most SCADA system LANs are designed to utilize TCP/IP over Ethernet as the network protocols. The network is a pipeline for the

HMI I/O server to poll data from the PLCs, and it provides the HMI clients and historical data server with a highway to access the I/O server's data.

Because the LAN is the only source of communication between the SCADA system components, one can easily see the importance of network reliability. Therefore, integrating redundancy into the network is recommended.

### **Historical Database (Historian) Software**

Without historian software, the historical data server is just another PC. The historian software manages the collection, storage, and archiving of data and can be classified as relational database management systems (RDBMSs), which means that they store data in the form of relational tables as opposed to a "flat file" of information. The flat file database is a database in which all data is stored in one table. Relational databases can use multiple tables to store information, and these tables can each have a different data format. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. Therefore, one database can be viewed in several different ways.

In the case of SCADA systems, the popular RDBMS packages are generally not the best choice for the historian. In most cases, the historian software is chosen based on the HMI SCADA software being used, and is typically of the same manufacturer. One reason for this is SCADA software manufacturers usually develop their historian software for ease of integration with the SCADA application. In other words, configuring the historian software to collect data from specific tags in the HMI I/O server is relatively easy if the same manufacturer's products are used. If a third-party package is used, customized scripting is required to populate the database with SCADA data. Secondly, because of the mass quantities of data being recorded on a daily basis, many of the SCADA software manufacturers develop their historian packages to make use of compression algorithms, which allow faster access to the data and less storage space usage. Lastly, RDBMSs are typically very specialized tools requiring development and maintenance by highly qualified database administrators.

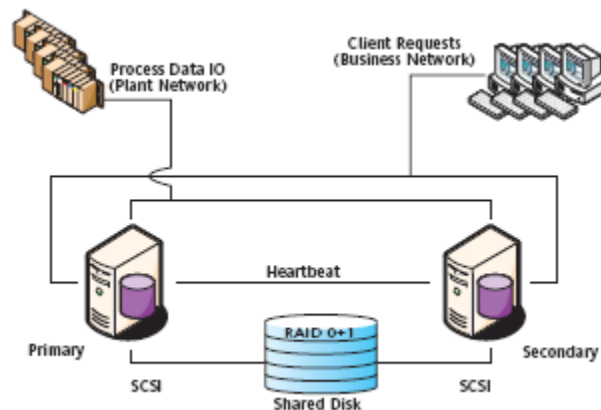
Regardless of what historian the SCADA system is designed around, the system designer should consider placing a high degree of importance on system reliability. Following are some configurations commonly used to increase availability:

#### ***Buffering Data Collectors***

Because the historian computer is typically a stand-alone machine, separate from all of the I/O servers on the network, one needs to account for the possibility of losing a network connection. For this reason, several of the historian packages are developed with data collectors that can be installed directly on the I/O server nodes. These data collectors acquire data from the I/O server and forward it to the historian for storage. The data collectors have buffering capability so that in the case of a network failure, the historical data is stored locally until the network returns to service and the historian can be updated.

### ***Failover Clusters***

The historian node to some owners is one of the most critical components of their SCADA system. These owners will want to ensure the highest level of availability. One means of achieving this is by configuring a cluster, which is comprised of two PCs, a primary and a standby. Both PCs in the cluster have the same historian software installed; however, the primary PC runs the historian software while the other PC in the cluster is idle. If the primary server fails, the secondary server takes over and runs the historian software. In this configuration, the data stored on a separate shared drive, typically a RAID drive offering higher reliability.



### **Defining the Reporting Goals**

It is important that the reporting goals be defined prior to configuring the data collection schemes. By knowing in advance how the historical data is going to be utilized, the developer can make sound decisions on how to collect and store the data. The first step in report design is to create a list of the required reports and sort them in order of decreasing priority. The lower priority reports may or may not be implemented depending on whether funding is available. Following are explanations of typical report types that may assist in creating the report priority list:

### ***Regulatory Reporting***

Some reports may be required or by regulatory agencies such as the Texas Commission on Environmental Quality (TCEQ), formerly known as the TNRCC. The TCEQ has different requirements for different types of water and wastewater treatment as well as other industries. One example of a report required by TCEQ is the Filter Profile Report for Individual Filters. This report is required for public water systems that treat surface water or ground water that is influenced by surface water and are required to conduct additional individual filter monitoring. Another example is the Monthly Optimization Report required by plants that participate in the Texas Monthly Optimization Program offered by the TCEQ. These types of reports should be considered top priority in the list of possible reports to be implemented as part of the reporting package.

### ***Performance and Preventive Maintenance Reporting***

The purpose of these types of reports is to provide plant personnel with information for improving plant performance and when to provide preventive maintenance for equipment. Equipment runtime reports can provide maintenance personnel with valuable information about how often and how long equipment is used, helping them to schedule preventive maintenance and to develop a routine for future usage. Electrical Usage and status reports can provide electricians and management with information on power usage which can help them develop methods to run equipment more efficiently. Other reports, such as sludge retention time (SRT) and chemical dosage reports, can provide operations personnel with information that may allow them to modify certain aspects of process control and increase overall plant performance.

### ***Plant Defined and Non-traditional Reporting***

Many plants have specific reports that are used for educational or process optimization purposes. Any plant-specific report that plant personnel use should not be overlooked when upgrading to an automated reporting system. Reports are usually developed with careful planning, and although they may not be required for regulatory or preventive maintenance, they may provide important information on certain aspects of plant operation.

### ***Trending Requirements***

Historical trends can provide plant personnel with instant information about equipment and analytical data. By comparing data from multiple tags in a trend chart, plant personnel can acquire a better view of how specific processes are performing and can troubleshoot instruments and other equipment. The trend charts should be organized with a structure that most optimally provides useful information to plant staff. Too few tags on each chart can add up to a very large quantity of charts on a system. Too many tags on each chart can make the charts difficult to read and in some cases can affect the system's performance.

Since implementing reports in a report package requires time and money the list of required reports should be minimized to those reports that are most important and fall within the project budget. In some cases, reports that contain overlapping data may be consolidated into one report to help minimize the list. Once an accurate, finalized reports list is defined, the data required for each report and how that report's data is collected shall be defined. The following section provides an explanation of the data collection schemes.

### **Data Collection**

There is a common misconception that all SCADA system points need to be historically collected. The goal is to collect and store data using the least amount of hard disk space possible, while ensuring that all of the necessary data is being collected. Keep in mind that some database packages have limited point counts and have added cost for additional points. One may ask, "How do I define what data is necessary, and how should this data be collected?" The answer to these questions depends on what the data is being used for.

Start by defining the data needed for the finalized list of reports and trends. Additional points are then added as needed to fulfill possible troubleshooting needs or future reporting needs. Points that will likely not be reported or trended are eliminated. Next, identify how the data needs to be collected. There are four data collection algorithms commonly used: time based collection, deadband (change in value) collection, time and deadband collection, and event based collection.

### ***Time Based Collection***

Time based collection is the simplest of the four algorithms. The historian software is configured to collect a sample from a data point based on a time interval such as every minute regardless of how much the data has changed. This is an approach that saves development time if one were to configure the database with all points logged with the same cyclical time period. An additional benefit of cyclical storage is that it allows data points to be recorded based on the reporting requirements. For example, if the desired result is a daily average of 1-minute samples, the data point can be configured to be collected every minute. This ensures that the required 1-minute samples are available for the report, whereas the deadband collection (explained below) may not.

In many cases, cyclical storage may use unnecessary storage space since it records data regardless of whether the value of the data point has changed. For example, if a cyclical storage of 5 seconds is configured on a data point that measures wetwell level, and the wetwell level remains constant for 1 hour, then 720 data points containing the exact same value will be recorded during the 1-hour period. Deadband collection should be considered for this application.

### ***Deadband Collection***

Deadband collection, when configured properly, can significantly reduce the amount of disk space used. The data point is recorded only when the value changes by more than the preset deadband. Thus, in the wetwell level example above, the data point would not have been recorded even once in that hour since the value had not changed during that time period. Deadband collection also allows values to be collected when unexpected changes in process or control cause the data point to spike in a positive or negative direction. For example, if system pressure suddenly drops unexpectedly and then returns to normal, deadband collection would capture this data where time based collection may not.

There are some drawbacks to using this method for data collection. The lines in a historical trend chart are usually plotted from one data value to the next. If a data value is not stored for a long period of time then the line that represents the data point will not be visible until the next value is stored. If the data point is intended to be used on a report that calculates a daily average and no values have been collected for that day then the query would not be able to calculate an average and would result in an error or report a null value.

### ***Time and Deadband Collection***

In some cases, it may be beneficial to configure data points to be collected based on time and deadband. This method is very good for trending and troubleshooting purposes because the deadband component captures the sudden changes in value while the time-based component provides a point of reference for the chart to display a trend line.

### ***Event-Based Collection***

Event-based collection is generally more advanced than the previous three algorithms because customized logic or scripting may be necessary. This method involves recording a value when a particular event or trigger occurs. The event may be a result of logic developed in the PLC or a script that is executed in the HMI application. Or it may even be manually initiated. A prime example of event based collection is the filter profile report for individual filters mentioned above. This report has a number of requirements, but for this example, we will only concentrate on some of them. When a filter is backwashed and returns back into service the turbidity is monitored for spikes, and the maximum spike must be recorded. In addition, values are captured both fifteen minutes and thirty minutes after returning in service, and the filter's runtime is calculated until the filter is backwashed again or placed out of service. Collecting these values using one of the other three algorithms and then trying to retrieve the data based on the criteria mentioned can be difficult and time consuming. However, by allowing the PLC to handle the more complex calculations based on events within the PLC logic, we can minimize the amount of data collected by the historian and the complexity of the customized queries required.

The four algorithms explained above deal with automatic data collection. Most owners also require data to be entered manually as well. For example, some regulatory agencies require samples of water to be tested on a regular basis and the results to be reported. For these test results to be made part of the SCADA system's historical data, they more often than not need to be entered manually. Several methods can be used to accomplish this. In a flat file database, because the architecture does not allow for a new table of data to be added, the simplest approach to adding manual data is to use an HMI graphic display. This display would have fields for entering values that get sent directly to the historian database. In a relational database, the same approach can be taken, or a new table for manually entered values can be created. This table can be written to using any interface that is open database connectivity (ODBC) compliant. For example, a developer may create a customized application in Visual Basic that allows a user to enter the data on a user-friendly form and then sends the data to a table in the historian database.

## **Retrieving Historical Data**

Retrieving data from the historical server seems fairly simplistic but there are many combinations of chronological and data attribute variations that need to be considered when defining how the data will be queried.

### ***Chronological aspects of reporting***

Reports are usually defined by the chronological period of data being displayed. Although there are many types of chronological reports, the most common are hourly, daily, monthly, and yearly. For this paper we define these types as follows:

- Hourly – contains 1-hour worth of data in a variation of minute increments
- Daily – contains 1-day worth of data in a variation of hourly or minute increments
- Monthly – contains 1-month worth of data in daily increments
- Yearly – contains 1-year worth of data in monthly increments

These reports can be generated manually or based on a time schedule. Hourly and daily reports are usually generated based on a schedule and monthly and yearly reports are typically generated manually. There are some cases where monthly reports are generated on a daily schedule with data included up to the previous day.

### ***Report Data Attributes***

Data can be retrieved from the historian and displayed in many ways. The most common are raw samples, interpolated samples, raw calculations, interpolated calculations.

Raw samples are actual values that the historian has collected. For example, if the report is to display all values for a data point that the historian contains for a 1-hour period the number of values may vary if the data was collected based on deadband or time and deadband. If the data point was collected every minute with no deadband then the number of values will be consistent.

Interpolated samples are estimations of actual values that the historian has collected usually based on number of samples or samples per time. For example, if the report was to display all values for a data point that the historian contains based on interpolated samples for a one hour period, the number of values would always be consistent regardless of the number of raw values. The interpolated values would be estimated based on the relative time between the recorded raw values.

Raw calculations are calculated variations of the actual values that the historian has collected. There are many types of calculations that can be applied. Minimum, maximum, and average are among the most used. If the report is intended to display an hourly average of a data point based on the raw values, then each raw value would be added together and then divided by the number of raw values providing an accurate average. If no raw values were collected during the hour then the query would return bad quality, null value or another type of error depending on the report software used or the historian software being queried. Interpolated calculations are similar to raw calculations, except

that if no data was collected during the hour being queried, an interpolated value can be displayed instead of an error.

### **Report Development Tools**

Reports are often the lifeblood of the entire plant operation. When data flows easily from the plant process to the desktops of plant operation and management personnel, faster and smarter decisions can be made. Automation of plant reports can simplify the report generation process, providing plant staff more time for other tasks.

Several tools are available for simplifying the development of automated reports. Historian software packages typically are integrated with a tool set that can be used to access data and assist with simple report development. However, some plant owners may have report requirements that call for a customization beyond what some packages have to offer. For this reason, there are numerous third-party report software applications available on the market that offer advanced tools for report design and generation.

### **Conclusion**

Because rapidly advancing technologies in the automation world are improving the way systems are designed and implemented, many plant owners are upgrading their SCADA systems with the desire that their operations will become more efficient. SCADA system hardware and software are becoming faster and more fault tolerant, and redundancy is becoming a standard practice. In recent years, more and more value is being placed on the acquisition and presentation of historical information. Because of this trend, HMI software manufacturers have significantly improved their products by integrating them with powerful database management systems. These packages not only collect and store data, but also provide the user with wizards and friendly interfaces that ease the configuration and conserve development time.

This paper merely scratches the surface of database configuration and reporting. From simple reporting problems to the most complex reporting problems, a solution can be found. The path to the solution is dependant upon the complexity of the report and the experience of the developer trying to achieve it. In either case, the end product will provide the customer with highly useful information, aiding in the efficient operation of their facility.

### **References:**

- (1) "Industrial SQL Server 9.0 Software Datasheet"  
<http://us.wonderware.com/products/insql>
- (2) <http://www.tceq.state.tx.us/>