Historian Software

Proficy™ Historian

Guide Form Specification

GFT-523

August, 2004

imagination at work

1 System Performance

1.1.1 The data historian shall provide a minimum sustained performance rate of at 20,000 values per second at the largest tag configuration.

1.1.2 The data historian shall support at least 100,000 tags on a single PC

1.1.3 The data historian shall be developed as a high-speed time-based process historian which does not require the use of 3rd party relational databases for operation or installation.

1.1.4 The data historian shall provide 1 millisecond time stamp resolution.

1.1.5 The data historian shall be able to store 32 bit floating point numbers, including time stamps and quality in no more than 4-5 bytes. If storing millisecond then the system may occupy one additional byte for a maximum of approximately 6 bytes per record.

1.1.6 The data historian shall provide automatic time-based indexing between different tags without adversely impacting storage performance. E.g., A user shall not need to add another "column" to each record to store the batch ID since the batch ID column would add several more bytes to each record archived.

2 Administration & Configuration

2.1.1 All administrative functions (e.g., tag configuration, archive maintenance, etc.) must be configurable via a 100% browser based interface (Internet Explorer) without the need for any 3rd party ActiveX controls on the client computer

2.1.2 A non web-based administration tool shall also be available so that a web site is not required for system configuration.

2.1.3 The user shall be able to browse and add tags from any of the data sourced (e.g., OPC)

2.1.4 All configuration changes must be "on-line" without the need to stop & re-start the data historian.

2.1.5 System must provide for the automatic creation of archive files and the ability to automatically overwrite the oldest archive for unattended operation

2.1.6 The system shall provide a method for backing up all on-line/active archives on-line without the need to stop the archive system.

3 Security

3.1.1 System must utilize Windows NT/2000/XP tag and role-based security

3.1.2 Role-based security shall restrict user access to different administration and system functions. At a minimum these shall include:

3.1.3 Security administrators

3.1.4 Tag maintenance

3.1.5 Archive file maintenance

3.1.6 Data collector maintenance

3.1.7 Data readers

3.1.8 Audited writes

3.1.9 Unaudited writes

3.1.10 Tag-based security shall be configurable on a tag-by-tag basis and shall include the following:

3.1.11 Readers

3.1.12 Writers

3.1.13 Administrators

3.1.14 Security system must support both local and domain-based security.

3.1.15 System must provide an "electronic signature" feature to validate the identity of any user for all configuration changes. All client tools provided by the vendor that have the ability to modify the system shall utilize the electronic signature feature.

3.1.16 The electronic signature system shall provide a user configurable dialog message so that the client can customize the message.

4 Audit Trail

4.1.1 The data historian shall come with an automatic audit trail mechanism that stores all configuration changes, user connections, security violations and performance metrics.

4.1.2 The audit trail shall be stored with the data in the archive file covering the same time period so that the user only needs to manage a single file for all data and audit messages for any given period.

4.1.3 The audit trail shall not be modifiable – a user may insert custom messages, but once stored an audit message can not be modified or deleted regardless of the user's security privileges.

5 Tag & Data Collection

5.1.1 The system shall provide a graphical interface to browse and add tags from any supported data source. Added tags will automatically determine the data type, description, tag name from the data source.

5.1.2 All configuration changes shall be performed on-line with no restart required.

5.1.3 The system shall utilize UTC for time stamping data so that it is unaffected by daylight savings time or regional time differences.

5.1.4 Data collectors shall support synchronizing time stamps to a central server time so that all data has the same relative time stamps.

5.1.5 System shall support both polled and exception-based data collection.

5.1.5.1 The minimum polled collection interval shall be 100 milliseconds.

5.1.5.2 The minimum for exception-based collection shall be 1 millisecond.

5.1.5.3 Unsolicited tags shall have a method to throttle/limit the rate of unsolicited data writes. E.g., no faster than every 5 seconds.

5.1.5.4 System shall provide both a collection rate and a collection offset configuration.

5.1.5.5 Collection rates shall be configurable using intuitive seconds, minutes, hours drop down select lists.

5.1.6 System shall support 1 millisecond time stamp resolution.

5.1.7 System shall support the following native data types:

5.1.7.1 Single integer (2 bytes)

5.1.7.2 Double integer (4 bytes)

5.1.7.3 Single Float (4 bytes)

5.1.7.4 Double Float (8 bytes)

5.1.7.5 Scaled floats (scaling a float across 2 bytes)

5.1.7.6 Fixed length string (of any length)

5.1.7.7 Variable length string (of any length)

5.1.7.8 Binary Large Objects (BLOBs of any size)

5.1.8 System shall provide for input scaling (e.g., automatically scale a 0-4096 input to 32-212 degrees F)

5.1.9 System shall provide a dead banding compression algorithm (+/- limits around a process value).

5.1.9.1 The dead banding compression system shall be performed on the remote data collection PCs so that values that do not exceed the dead band are not reported to the server and do not consume network bandwidth.

5.1.9.2 The dead banding shall have a “time out” feature so that a value is stored at a regular interval regardless if it has not exceeded the dead band.

5.1.9.3 The dead banding shall be configurable as a % of the engineering limits.

5.1.10 System shall provide a rate of change compression algorithm (+/- limits around the real-time slope of the process)

5.1.10.1 The rate of change algorithm shall have a "time out" feature so that a value is stored at a regular interval regardless if the slope deviation has not been exceeded.

5.1.10.2 The rate of change compression limits shall be configurable as a % of the engineering limits.

5.1.11 System shall ensure that data spikes are properly represented in trend and client tools. E.g., if a value has been stead-state for several periods and suddenly spikes to a high value trends will properly display this as a spike/step change and not a ramp.

5.1.12 System shall provide for recording time stamps from either a collector PC or from the OPC/PLC tag.

5.1.13 Data collectors shall have an automatic store & forward mechanism to ensure that data is not lost when disconnected to the data archiver.

5.1.13.1 The store & forward mechanism shall not require the user to pre-allocate a buffer file, or set a maximum buffer file size. Instead, the system shall provide for utilizing available disk space up to a user configurable limit.

5.1.13.2 The store & forward mechanism shall automatically detect when the archiver is available and forward data from the buffer files while simultaneously collecting all incoming data.

5.1.13.3 Data collectors shall not require a hardware key or other licensing on the collection PC.

5.1.14 Once installed, data collectors shall automatically register and configure themselves with the data archiver service without the need for any additional configuration on the archiver.

5.1.15 A user shall be able to store comments with any collected data.

6 Calculation Engine

6.1 Architecture and Data Collection

6.1.1 The system shall provide a calculation engine for the automatic calculation and analysis of both incoming and archived data and then storing the results of the calculation in the data historian as a tag value.

6.1.2 The calculation engine shall be configurable so that it can be installed on a PC, or several PCs that are remote from the data historian.

6.1.3 The calculation engine shall have a store & forward mechanism to ensure that calculated results are not lost if the connection to the data historian is down.

6.1.4 The calculation engine shall have a recovery logic system so that any updates to trigger tags (calculation inputs) cause the calculation to re-fire.

6.1.5 The calculation tags shall be configurable to execute both as polled or unsolicited/trigger-based tags.

6.1.6 The minimum poll rate shall be 100 milliseconds

6.1.7 Unsolicited tags execute their calculation whenever an assigned trigger tag receives a new value, time stamp or change in quality.

6.1.8 Calculations shall support an unlimited number of trigger tags.

6.2 Calculation Functions

The system shall support the following calculation functions:

6.2.1 Browse and select any historian tag as an input to the calculation

6.2.2 Current/last stored value of any tag

6.2.3 Previous values of a tag

6.2.4 Next value of a tag

6.2.5 Interpolated values

6.2.6 Current/previous/next time stamp

6.2.7 Current/previous/next quality

6.2.8 Historical time-weighted Average

6.2.9 Historical time-weighted Minimum

6.2.10 Historical Maxim

6.2.11 Historical time-weighted standard deviation

6.2.12 Historical time-weighted total

6.2.13 Historical count of samples

6.2.14 Historical raw average

6.2.15 Historical raw standard deviation

6.2.16 Historical raw total

6.2.17 Time of minimum sample

6.2.18 Time of maximum sample

6.2.19 Total time that a sample was good

6.3 Calculation Configurations

6.3.1 The system shall support filtered calculations so that and of the above calculations are automatically filtered/limited based upon another tag's value. E.g., Calculate the minimum amps (tag 1) over the previous day, but exclude any samples in which for the same time period the line voltage (tag 2) was 0. Or, return the average for tag 1, while the batch id (tag 2) = 'ABC'

6.3.2 Calculations shall support full visual basic scripting within the calculation.

6.3.3 Calculations shall be configurable using the same administration tools (web and non-web) including all visual basic scripting and functions.

6.3.4 The system shall have tools to assess the time that a calculation takes to execute, as well as a means of disabling calculations that exceed a configurable maximum execution time.

6.3.5 The calculation shall be stored in the audit trail along with a full history of modifications and the time, date and user that made the modifications.

6.3.6 The calculation engine shall have a manual recalculation engine so that calculations can be applied to legacy data and values stored alongside the legacy data as if the calculation engine had been executing in real time when the legacy data was archived.

7 Server to Server Engine

7.1.1 The system shall have a server-to-server engine so that tag data can be automatically forwarded from a one historian to a remote historian

7.1.2 The server-to-serve engine shall support all of the functions and features described in the section on the "Calculation Engine" (e.g., calculations, visual basic scripting, re-calculation, etc.)

7.1.3 The server-to-server engine shall provide for the forwarding of audit messages as well as data.

7.1.4 The server-to-server engine shall provide for tag browse and configuration from the destination server's administration interface. The user shall not need to configure server-to-server tags at the remote historian.

8 Client Tools

8.1 Microsoft Excel

8.1.1 The data historian shall come with a Microsoft Excel tool bar so that users can readily extract data and develop reports using Excel.

8.1.2 The excel add-in shall provide for the ability to import or modify tags

8.1.3 The excel add-in shall provide for the ability to import or modify stored data.

8.1.4 The excel add-in shall provide for the ability to import and view comments for any stored data

8.1.5 The excel add-in shall not require the user to know the SQL query language

8.1.6 The excel interface shall provide for automatic recalculation if any cell is changed – e.g., a tag name, time periods, selected calculations and so forth.

8.1.7 The excel interface shall be supplied with sample reports that can be customized by the user.

8.2 Calculation Functions

8.2.1 Browse and select any historian tag as an input to the calculation

8.2.2 Current/last stored value of any tag

8.2.3 Previous values of a tag

8.2.4 Next value of a tag

8.2.5 Interpolated values

8.2.6 Current/previous/next time stamp

8.2.7 Current/previous/next quality

8.2.8 Historical time-weighted Average

8.2.9 Historical time-weighted Minimum

8.2.10 Historical Maxim

8.2.11 Historical time-weighted standard deviation

8.2.12 Historical time-weighted total

8.2.13 Historical count of samples

8.2.14 Historical raw average

8.2.15 Historical raw standard deviation

8.2.16 Historical raw total

8.2.17 Time of minimum sample

8.2.18 Time of maximum sample

8.2.19 Total time that a sample was good

8.2.20 The system shall support filtered calculations so that and of the above calculations are automatically filtered/limited based upon another tag's value. E.g., Calculate the minimum amps (tag 1) over the previous day, but exclude any samples in which for the same time period the line voltage (tag 2) was 0. Or, return the average for tag 1, while the batch id (tag 2) = 'ABC'

8.3 OLE DB

8.3.1 The data historian shall come with an OLE DB interface so that data can be extracted and viewed by client applications such as SQL Server and Crystal Reports.

8.3.2 The OLE DB interface shall provide access to all server time-weighted calculations as described in the Microsoft Excel section above.

8.3.3 The OLE DB interface shall be read only (i.e., Select statements) so that data and configuration can not be modified via SQL.

8.3.4 The OLE DB interface shall provide SET statements and other functions so that time-weighted calculations and reports can be easily extracted. E.g.

E.g., The user shall be able to get the hourly average of a tag or several tags, regardless of how frequently they were collected by entering a statement such as shown below:

SET Intervalmilliseconds= 1Hour, Start Time=Yesterday, EndTime=Today, CalculationMode=Average
Select Timestamp, Value from RawData where Tagname = Tag1 or Tagname= Tag2

8.3.5 SET statements shall include functions as shown below:

8.3.5.1 Today, Yesterday, Now, Beginning of month, etc.

8.3.5.2 Interval = hours, minutes, seconds (for evenly spaced reports)

8.3.5.3 The OLE DB interface shall provide access to all system configuration, audit messages, and data

8.4 HMI

8.4.1 The data historian shall have native connectivity a SCADA/HMI package

8.4.2 The HMI interface shall provide the user with a drop down list to specify different time zones the data shall be represented (since the historian stores data as UTC it can be represented in any relative time zone)

8.4.3 The HMI interface shall provide the ability to select if charts will reflect time changes due to daylight savings time. E.g., when the clocks move forward in the spring the HMI chart will display 1:00 AM and then 3:00 AM.

8.5 Web Interface

8.5.1 The data historian shall have a 100% web-based client interface to browse, chart and display data.

9 Downtime Analysis

9.1.1 The data historian shall have an automatic down time detection and analysis engine

9.1.2 The down time system shall not require the user to modify the PLC or SCADA logic to perform down time analysis or Boolean condition checking.

9.1.3 The down time system shall allow for creating several custom detection models that can automatically pinpoint the location, reasons, fault and equipment of down time.

9.1.4 The down time system shall come with an model that can automatically determine the location of a fault by analyzing the state (blocked, starved, unavailable) of upstream and downstream pieces of equipment.

10 Vendor Requirements

10.1 Development Life Cycle

10.1.1 The vendor must have an established development life cycle that allows for traceability of features and functions throughout that life cycle.

10.1.2 The vendor must have a formal and documented set of quality assurance procedures that are applied to the engineering design, development, and documentation of the software. The presence of a formal quality assurance department shall be required.

10.1.3 The vendor must also demonstrate that its source code for the product is regularly archived both on-site and off-site in facilities suitable to withstand physical harm.

10.1.4 The vendor shall allow for on-site auditing of the development life cycle to ensure good practice.

10.2 ISO 9001 certified

10.2.1 The vendor must be able to demonstrate that it has established procedures.

10.2.2 Vendor needs to be certified under the ISO 9001-2001 guidelines.

10.3 Preferred Vendor / Manufacturer

10.3.1 Pre-evaluation has identified that Proficy™ Historian product from GE Fanuc Automation as the preferred software solution. Any proposed solution must include at a minimum the functionality contained in the current commercially available version of Proficy™ Change Management.

10.3.2 Licensing will be provided to support XX simultaneous users and will include an annualized contract for Proficy™ GlobalCare Support.