



Movicon NExT

3.3 Historian

Ver.3.4.268

Table of Contents

1. HISTORIAN	1
1.1. GENERAL CONCEPTS	1
1.1.1. <i>Historians</i>	1
1.1.2. <i>Data Loggers</i>	1
1.2. DATABASE CONNECTION	2
1.2.1. <i>Unavailable Database Connections</i>	2
1.3. MAPPING DATA IN DB.....	3
1.4. HISTORICAL PROTOTYPES	4
1.5. HISTORIAN PROTOTYPE SETTINGS	5
1.6. ASSOCIATING TAGS TO THE HISTORIAN	10
 2. DATA LOGGERS	11
2.1. DATA LOGGERS.....	ERRORE. IL SEGNA LIBRO NON È DEFINITO.
2.2. DATA LOGGER COLUMN PROPERTIES	12
2.3. DATA LOGGER SETTINGS.....	13
2.4. ASSOCIATING TAGS TO DATA LOGGERS.....	17

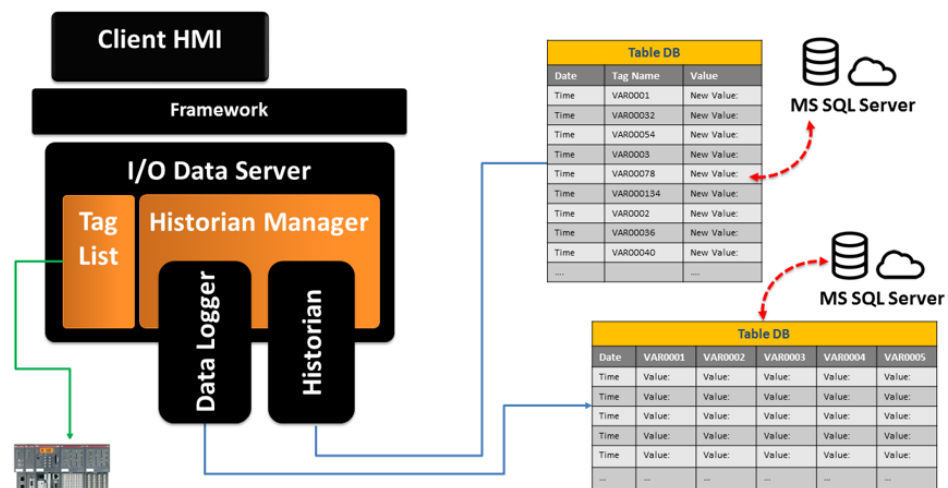
1. Historian

1.1. General Concepts

The I/O Data Server's Historian module records tags managed by the I/O Data Server and defined in the Address Space, using two different recording models:

- **Historians**
- **Data Loggers**

Both recording models record tag values on Database archives, but use different modes and criteria for recording data.



1.1.1. Historians

The Historian records data associated to Historian prototypes according to a model which provides DB tables containing records for each even recorded, in one column. This means that tables will have one column only independently from the amount of tags associated, whereby each record represents the changed value recorded only. The DB is very useful being efficient and ideal for analysing and displaying data in the viewer objects such as the Trends or Data Analysis. The Data Logger model is more suitable for the Report management.

1.1.2. Data Loggers

The Data Logger records data associated to the Data Logger prototypes according to the model providing the DB data tables with records for each recorded event shown in one column for each associated tag.

This means that one table will be created with the same number of columns as there are associated tags, with the recordings of all the tags' values (including those that do not

change) for each record. Even though this may occupy more DB memory, data can be organized in a way that is better for those archives that are used by the Report function.



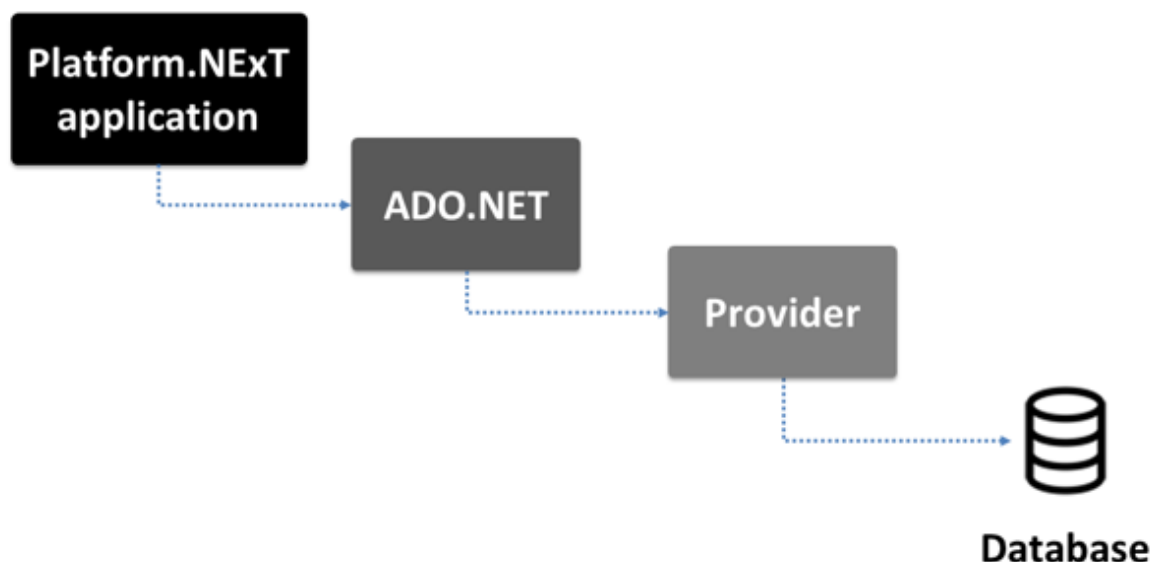
In order to get the best results, programmers should evaluate the best method to use in regard to the type of analysis needed when deciding which data recording model to use.

The baseline to use would be:

- Historian: Process Data
- Data Logger: Production Data

1.2. Database Connection

Connecting to the database is performed independently from the type of database used thanks to the ADO.NET Provider technology. If you do not want the system to use the local default SQL Server database connection, this will allow you to customize another database connection for your project to use for default instead.



1.2.1. Unavailable Database Connections

In the event of an unavailable database connection, data will be written on XML files with the .FlushedData extension.

These files are copied in the project's subfolder: "UFUAServer\Historian".

Using SQL Server Express as database

When you use the Microsoft SQL Server Express the table generated by Movicon.NexT will be set with the 'Simple' recovery model option.

The log file space in this mode is automatically recovered to limit the disk space requisites and, therefore, to avoid having to manage the space in the transactions log.

The recovery model switches from 'Simple' to 'Complete' only when the 'Enable data protection' for the CFR21 management is enabled.

In this way, all modifications made to the table created by Movicon.NExT will be traced and you will need to instigate a log file backup plan in order to manage the transactions log's disk space. In cases where no backup plan is instigated, there might be a risk that the transactions log file will increase in size to occupy all the space available on the disk.

Using SQL Server as database

When you use the SQL Server, the tables generated by Movicon.NExT will be set with the 'Complete' recovery model option.

In this way, all modifications made to the table created by Movicon.NExT will be traced and you will, therefore, need to instigate a log backup plan to manage the transactions log's fixed disk space.



If a backup plan is not implemented, the transactions log file might increase in size to then occupy all the fixed disk space available.

SQL Server CE

As previously mentioned, the standard SQL Server version is installed during the Movicon.NExT setup unlike the Compact Edition of this program which must be downloaded from the Microsoft website using this link:

<https://www.microsoft.com/it-it/download/details.aspx?id=30709>

SQL CE installation is indicated for low performing devices or those which have less memory available. In addition, it provides access protection to data on the database through the use of passwords which can be set using the connection string in accordance with the cf-21 requirements.

Various programs can be used, like the one below, to consult data using external tools:

<http://www.linqpad.net/>

Data Flush

If the connection to the Database in which data is being recorded is interrupted, the Historian will unload data on to xml files with a name using the following syntax:

Historical_YYYYMMddHHMM.FlushedData

These files are copied in the project's "UFUAServer\Historian" subfolder and then removed after the pending queries have been executed.

1.3. Mapping data in DB

According to the type of Tag data defined in the DataLogger, a field will be defined in the SQL Server database to contain the Platform.NExT data. The data correspond to each other as follows:

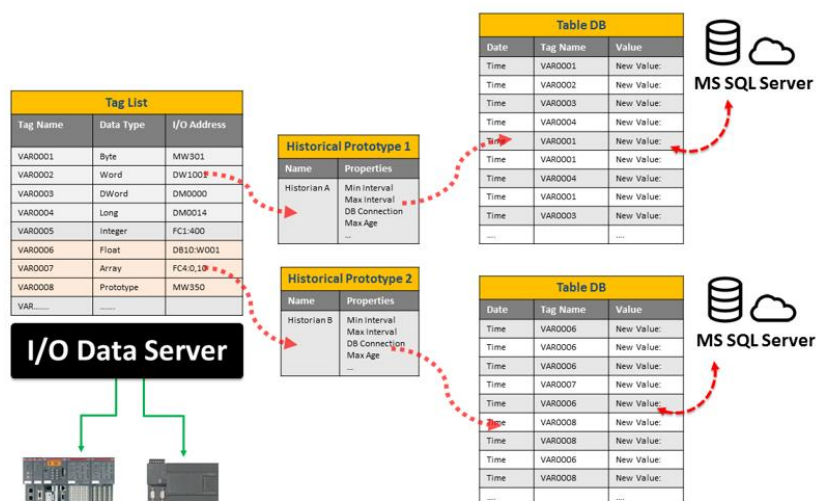
Movicon	SQL Server
Boolean	bit
Byte	tinyint

SByte	smallint
Int16	smallint
UInt16	int
Int32	int
UInt32	BigInt
Int64	BigInt
UInt64	BigInt
Float	Real
Double	float
String	nvarchar

1.4. Historical Prototypes

The Historical Prototypes resource is a I/O Data Server function model that is used for inserting and configuring data recording models that historically log the variations of one or more tags in appropriate Database files according to the configured modalities.

- Each Historical Prototype inserted constitutes a "recording model" whose properties must be configured to establish the recording modalities and the database archive type where the associated tags will be recorded and stored.
- Each time a data recording condition is verified, the tag is recorded and added as a record to the DB table. In this way each record on the table reports information inherent to each individual tag to allow efficient data association management with Trends and Data Analysis.



Data recording modality

Only two parameters need to be set for establishing the data recording criteria based on "Event" or "Frequency".



The fundamental concept of Historian Prototype recording parameters is the criteria with which the record on change or time interval frequency is defined by taking into considering the:

- Min. Interval
- Max Interval

Where the Minimum and Maximum interval define the minimum frequency value when data changes are quite frequent, and the maximum frequency value when data changes are not as frequent, hardly ever or never change.

Example:

1. Setting the Minimum and Maximum value to zero, as for default, the Historian will record the value upon each change.
2. Setting a value different to zero (for example 5 sec.) for both the Min and Max parameters, the Historian will record each time value inserted (eg. 5 sec.) regardless whether data changes or not.
3. Setting a Min value (eg. 1 sec.) and a Max value (eg. 5 sec.), the Historian will record at least every second (Min value) if data changes are very frequent. If the data does not change by the time the Max value is reached (eg. 5 secs) a recording will be executed regardlessly.

1.5. Historian Prototype Settings

Each Historian prototype defined in the project entails the use of a data recording model to record the desired associated tags. The Historian's properties can be configured to record data according to time or on event or both as described below.

General

Name

This field is used for inserting the name of the historical prototype.

Record Only on Quality Good

When enabled, this option only allows recordings to be executed only when the communication quality results 'Good' while the tag in question is connected to the field by using a Communication Driver.

Enabled Tag

This field is used to insert the name of a Tag of the Server to be used in runtime to enable the Historian to record. This Tag will be taken into consideration only when the "Enabled" property is enabled.

If the 'Enabled Tag' is set with the 'zero' value at runtime, the Historian will be disabled and all the Tags connected to it will not be recorded independently from the activated recording mode (on change or time based). Conversely, if the 'Enabled Tag' is set with a value that is different to 'zero' (e.g. 'True', '1', '100', '-10', etc.), the Historian will result enabled and the Tags connected to it will be recorded according to the mode defined.

Note: The use of 'String' or 'Array' variable types are not supported as Historian 'Enabled Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Historian will ignore the 'Enabled Tag' during runtime but will result enabled for recording.

Enabled

This option box allows you to enable the Historian in design mode if disabled (False). In this case, the Server will ignore all disabled Historians (as if they didn't exist) after which it will not be possible to enable them to function during runtime.

This option is enabled for default (True), therefore the Historian results are enabled.

Aggregation Settings

Good Data Percentage

Sets the percentage of good data to be retrieved.

This property indicates the minimum good data percentage within a certain time range, for the StatusCode of the time range specified for the requested processed data to be set to Good. The predefined value is '0'.

Bad Data Percentage

This property indicates the minimum bad data percentage within a given time range requested for the StatusCode in the time range specified for the requested processed data to be set to Bad. The predefined value is '0'.

Stepped

Keeps the last data point constant or the interpolated value based on a horizontal line adaptation.

Use Sloped Extrapolation

This property indicates the mode in which the server executes data interpolation when there are no value limits (e.g. extrapolating the last noted value in future data). A False value indicates that the server will use a SteppedExtrapolation format and will keep the last noted value constant. A True value indicates that the server will transmit the value using the UseSlopedExtrapolation mode. The predefined value is false.

Treat Uncertain As Bad

This property indicates in which mode the server treats data returned with Uncertain StatusCode level in respect to the aggregation calculations. A True value indicates that the server will consider uncertainties as Bad, a False value indicates that the server will consider uncertainties as Good, unless the aggregation definition indicates otherwise. The predefined value is False. Note that the value is still considered as uncertain when the statuscode is calculated for the result.

Advanced Database Settings

Cache Size

The "Cache Size:" parameter is the maximum cache size for errors which are flushed afterwards.

Wait Retry

The "Wait Retry (sec.):" parameter represents the waiting time before retrying the data that raised an error.

Max Retries

The "Max Retries:" parameter indicated the maximum number of errors before flushing data on file.

Execution

Deviation Value

This parameter indicates the deviation value that the variable must obtain in order to record. This value may be an 'absolute' type or 'percentOf' type according to the type selected in the 'Deviation Type' field. When left set at zero, any variable change even if minimal, will activate a recording if the other parameters described below permit it.

Deviation Type

This parameter indicates the recording deviation mode. The deviation value used will be the one set in the 'Deviation Value' property above. The Deviation types that can be used are:

- **AbsoluteValue** = a recording is executed according to the variable value's absolute variation meaning when there is a numeric increase or decrease. The condition that determines this is when the absolute variation is " \geq " in respect to the set threshold. Cases in which the Historical is associated to a string type variable, the recording will be executed when the string type variable simply changes.
- **PercentOfRange** = a recording is executed according to the value variation percentage taken from the difference between the maximum value (Instrument Range High) and the minimum value (Instrument Range Low) set in the Engineering Unit associated to the variable. The condition that determines this is when the variation percentage is " \geq " in respect to the threshold set. This parameter is ignored when the Historical has been associated with a string type variable.
- **PercentOfValue** = a recording is executed according to the current variable's value variation percentage. The condition that determines this is when the variation percentage is " \geq " in respect to the set threshold. In cases where the threshold is set to 100%. This parameter is ignored when the Historical has been associated with a string type variable.
- **PercentOfEURange** = a recording is executed according to the current variable's value variation percentage based on the difference between the maximum value (EU Range High) and the minimum value (EU Range Low) assumed by the variable's associated engineering unit. The determining condition is that the variation percentage should be " \geq " in respect to the set threshold. This parameter is ignored when the Historical has been associated with a string type variable.

Min. Interval

The "**Min. Interval**" represents the time within which each variable variation will not be filtered. Once this time has expired, the variable will be considered and any variation will be recorded. This time count starts after a recording has been executed. Its purpose is to avoid recording very frequent changes that happen one after the other.

Max. Interval

The "**Max. Interval**" represents a time that once expired will prompt a Tag recording independently whether the tag has changed or not. This time determines the fixed tag recording interval time.

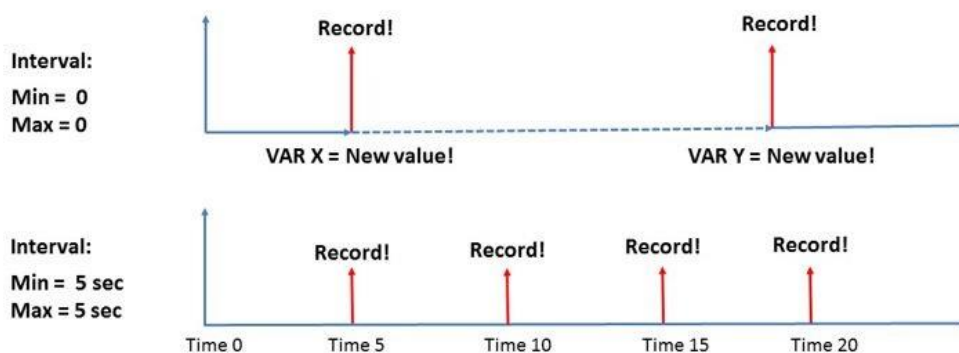
Different modalities can be obtained according to the settings defined for the minimum and maximum recording intervals:

- Minimum and Maximum interval = zero: RECORDING ON EVENT
In this mode all the tag value variations will be recorded whether they happen very

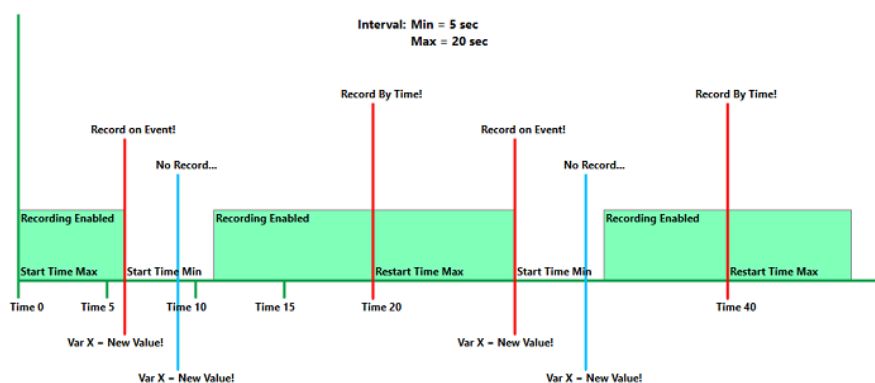
quickly or slowly. Basically, this mode records value variations only when they occur. The ms relating to the RecordDateTime, RecordDateTimeUTC, SourceTimeStamp and ServerTimeStamp columns will also be recorded along with the record on event. Contrary to this, the ms will remain fixed at a constant value.

- Minimum and Maximum interval with same value different to zero: RECORDING BY TIME
This mode records by time according to the defined interval time. For example, when setting both values to 3 seconds, a record will be entered on the table at a frequency of every 3 seconds, independently from whether the tag value has changed or not.
- Minimum interval different than the maximum interval: MIXED EVENT-TIME RECORDING
Setting two different values in the minimum and maximum interval fields, will cause a mixed behaviour of the previous two conditions. In this case the recordings will be executed when changes occur, according to the 'Deviation Type' property settings, by filtering the fastest changes of the set 'Minimum Time'. In addition, fixed timed recording will also be inserted based on the set 'Maximum Time'.

The diagram below can be used to verify recording behaviour with the same Minimum Time and Maximum time values:



Conversely, the following diagram shows the recording behaviour using **different Min. and Max. Interval values**. As indicated by this diagram recordings take place when there is a data change event within the set min. and max. time interval only. If no data change event occurs, the system will nevertheless execute a recording when the max. time elapses. In cases where a data change event occurs, the interval time count will start from zero after the change has been recorded.



Deviation Type

The "**DeviationType**:" parameter indicates the recording change mode whose value is set in the "DviationValue" parameter with the following possible values:

- **AbsoluteValue** = recording is performed based on the variable value's absolute change, intended as numeric increase/decrease. The verified condition is il ">=" in respect to the set threshold
- **PercentOfRange** = recording is performed based on a percentage of the value's variation taken from the difference between the maximum value (Instrument Range High) and minimum value (Instrument Range Low) set in the Engineering unit associated to the variable. The verified condition is ">=" in respect to the set threshold
- **PercentOfValue** = recording is performed based on variation percentage of the variable's actual value. The verified condition is ">=" in respect to the set threshold
- **PercentOfEURange** = recording is performed based on a percentage of the value's variation taken from the difference between the maximum value (EU Range High) and the minimum value (EU Range Low) obtained by the associated engineering unit. In cases where the Historical Log is associated to a string type variable, this parameter will be ignored. The verified condition is il ">=" in respect to the set threshold

Deviation Value

The "**DeviationValue**:" parameter indicated the value to be taken into consideration with reference to the set "DeviationType" value.

Database Settings

Max. Age

The "MaxAge:" parameter defines how long data must be kept in the table. The oldest resulting data of the set value will be deleted. Data will not be deleted from this table when setting this parameter with a zero value. The cancellation of data in the DataBase is performed in blocks using the XPO provider so that delete queries with huge quantity of records do not over burden this operation by requiring more time to cancel. Cancellation procedures go into operation after the first Historical log recording if records exceeding maximum age set are detected.

Connection String

The "Connection String" parameter is used to define a specific connection to the database for each Historical Prototype. When no specific connection has been defined, the historical will use the default connect set in the general "Historian Default Connection" Server connections.

Enable Data File Protection

This field is used when needing to protect audit recordings from unauthorized mishandling of any kind whether unintentional or intentional. This is done by inserting a column with encrypted data in the UFUAAuditDataItem object. The information contained in this column impedes the mishandling of sensible data recordings without compromising its integrity.

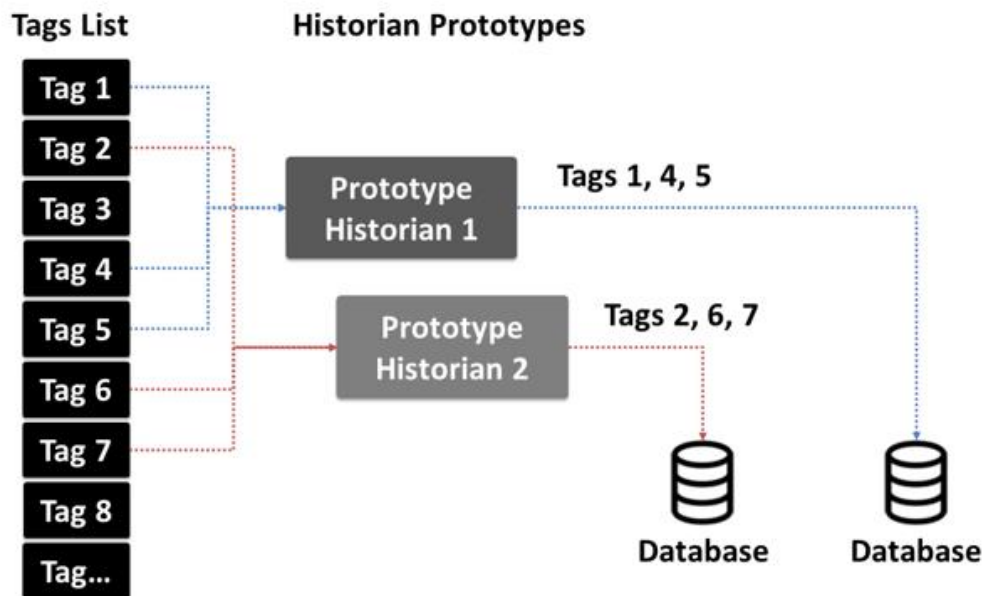
1.6. Associating Tags to the Historian

A Historian prototype is only a recording model or template. To make the recording model functional you will need to associate it with tags.

Associating tags to the Historian prototype is easy.

Simply open the I/O Data Server's Tag list and select the Tag you wish to record (or multi-select all the tags you wish to record), then use the command from the "Assign Historical Settings".

By using this command the tag or tags desired will be associated to the previously inserted Historian model. During runtime all of these tags will be processed by the I/O Data Server and recorded on database in function with the data recording modes set for that specific Historian model.



This scheme shows an example of how tags can be assigned to different Historian prototypes to record on database.

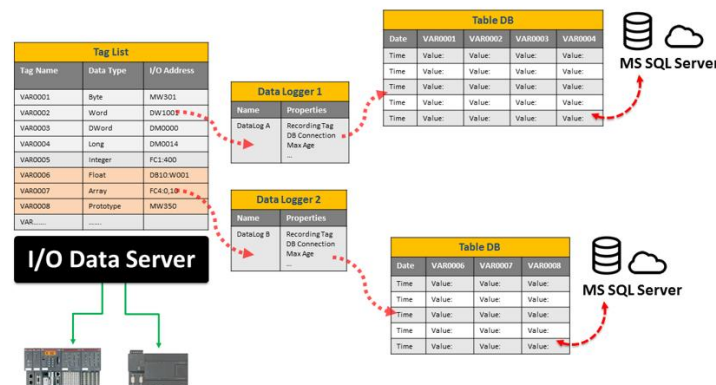


A Historical Prototype can have a vast number of tags associated to it, but only one Historical Prototype can be associated to a tag.

2. Data Loggers

The Data Logger Resource is a functional module belonging to the I/O Data Server that is used for inserting and configuring data recorder engines to record values of one or more Tags in a Database archive in function with predefined recording commands.

- Each Logger inserted is comprised of a data recording engine that has a data column-based structure. Each column corresponds to a value associated to the Tag. Each Data Logger must be configured by means of using its properties to establish the recording modes, the database archive and the archive size with which the tags to be associated will be recorded.
- Each time a data recording event occurs all the Tags associated to the DataLogger columns will be recorded by adding a record in the DB table. In this way each record on this table reports all the information relating to all the associated tags where each tag is represented as a column in the DB table. This type of management is not as efficient as the Historian module functions because it occupied more memory space but it much more practical to use for managing and manipulating data tables such as Analytical Reports.



A column for each tag associated to the DataLogger object is created in the table. Each time a data recording takes place, a record containing all the tags' values will be added and therefore sampled at the same time.

The data recording (entered records) in the Data Logger can take place according to time or command or both. The mode with which the data recording takes place is set in the configuration settings of each individual Data Logger. In addition to the sampling time to establish when to record as defined in the "Recording Time" property, the DataLogger also records on event by means of using the "Recording Tag" property. This property is used for associating a Tag whose value change triggers a data recording. Each time this tag assumes a value other than zero, the DataLogger will record and then automatically reset the recording tag back to zero.

It is also possible to insert a tag to delete data recorded on the table up to that moment ("Resetting Tag").



In cases where Columns are added after a DataLogger recording has already taken place, the new columns will be added to the DB table showing "null" values for previous records.

2.1. Data Logger Column properties

A column is created in the Data Logger table for each tag associated to the Data Logger to report the tag's value. In addition to the column that reports the Tag value, optional columns can also be added for each tag to report further information such as Time Stamp, Quality and other.

Array and Structure Variable Recordings

Array or Structure type variables can be associated to DataLogger columns. In both cases, the DataBase table columns will be created as String type and the data will be recorded as an array of values where each element contains the value of the Movicon array variable's elements and each individual structure variable's member.



Attention! Embedded structures are not supported when using structure variable types. For example, if the structure variable type has a structure type member, it will not be recorded.

General

Column Name

This field is used to insert the name of the column associated to the Data Logger tag.

Column Tag

This field is used to select the tag associated to the Data Logger column.

Time Stamp Settings

Add Source Timestamp Suffix

This is used to add the Source Timestamp column (date and time of last tag modification) to the table structure of the selected tag.

Source Timestamp Column Name

This is used to change the name of the Source Time Stamp column. The name of the column is represented by the name of the tag followed by the indicated column name (e.g. tag002_col1). When this field is left empty, the name of the column will be taken from the name of the tag plus the "_SourceTimeStamp" suffix (e.g. tag002_SourceTimeStamp)

Add Server TimeStamp Suffix

This field is used to add the Server TimeStamp column to the table structure for the selected tag.

Server TimeStamp Column Name

This field is used to change the name of the Server TimeStamp column. The column's title is represented by the tag's name followed by the indicated column name (e.g. tag002_col1). If this field is left empty, the column will be given the name of the tag plus the "_ServerTimeStamp" suffix (e.g. tag002_ServerTimeStamp)

Other Column Settings

Add Quality Suffix

This field is used to add the Quality column to the table structure for the selected tag.

Quality Column Name

This field is used to change the name of the Quality column. The column's title will be represented by the name of the tag followed by the name of the indicated column (e.g. tag002_col1). If this field is left empty, the column will be given the name of the tag plus the "_QualityTimeStamp" suffix (e.g. tag002_QualityTimeStamp).

Add User Name Suffix

This field is used to add the User Column to the table structure for the selected tag.

User Column Name

This field is used to modify the name of the User Column, the column's title will be presented with the name of the tag followed by the indicated column name (e.g. tag002_col1). If this field is left empty, the column will be given the name of the tag plus the "_UserTimeStamp" suffix (e.g. tag002_UserTimeStamp).

2.2. Data Logger Settings

Each Data Logger prototype defined in the project comes with a data recording model for the user to manage by associating the tags desired. Data is recorded according to a time frequency or on event which can be defined when assigning each Tag. The relevant properties to be configured are described below:

General

Name:

Arbitrary name to associate to DataLogger.

Enabled

This option box allows you to enable the Data Logger in design mode if disabled (False). In this case, the Server will ignore all disabled Data Loggers (as if they didn't exist) after which it will not be possible to enable them to function during runtime. This option is enabled for default (True), therefore the Data Logger results as enabled.

Advanced Database Settings

Retry wait time

This field is used for defining the wait time after which the operation will attempt another retry in cases where an error has been detected.

Max. Errors

This field is used for defining the maximum number of errors before flushing data.

Max. Transactions

This field is used for defining the maximum number of transactions to aggregate at one time, in cases with fast triggers in the recording bit, 10 entries are performed with one transaction.

Max. String Size

This field is used for inserting the maximum length value of the string to be recorded in the DataBase.

Max. Cache Size

This field is used to defining the Cache size after which the data will be flushed to file.



If an error should occur while inserting a record, further attempts will be made by using the time interval set in the "Retry wait time" property between each attempt until the number of attempts indicated in the "Max Errors" has been reached. When the number of retries set in the "Max Errors" has been reached, data will be flushed to the xml file indicated in the "Flush path".



In cases where an already existing DataLogger column is renamed, a new column will be created in the table with the new name and the old column will be left untouched.

Database Settings

Connection String

This field is used for inserting the connection string towards a DataBase. In cases where this field is left empty, the connection string for the historical logs set in the project's general I/O Data Server settings (Historian default Connection) will be used.

Max. Age

This field is used to set the max. time that data can be kept on Database before being eliminated. Data that is older than the value set in this field, in respect to the actual time and date, will be eliminated from the database. The Max. Age of data is always checked when new records are stored.

Enable Data File Protection

This field is used to choose whether to protect audit recordings from tampering, unintentional or intentional modification of any kind. This is done by inserting an encrypted data column in the UFUAAuditDataItem object. The information contained within this column will impede any mishandling of sensible data recordings without compromising data integrity.

Table Name:

The name of the table to use for recording data in is specified in this field. When this field is left blank, the name of the DataLogger set in the "Name" will be used.

Use Aggregated Tables

Permits the viewing of heightened time periods for frequently sampled data.

When enabled, the stored data procedure produces these tables:

- <DL table name>_MI for aggregation every minute;
- <DL table name>_HH for hourly aggregation;
- <DL table name>_DD for daily aggregation.

For each numeric column; the columns will be created in the aggregated tables with the _MIN, _MAX, _AVG suffix respectively .

- MIN: minimum value reached within the time range in question
- MAX: maximum value reached within the time range in question
- AVG: average value obtained within the time range in question

Criteria used by tables aggregated by the data analysis:

10 min-> MI

720 min-> HH

14400 min-> DD

This means that data will be taken from the [DataloggerName]_DD table when exceeding 10 days of extractions; when exceeding 12 hours of extractions (but within 10 days) data will be taken from the [NomeDatalogger]_HH; when exceeding 10 minutes of extractions (but within 12hours), data will be taken from the [DataloggerName]_MI; when below 10 minutes of extractions, data will be taken from the [DataloggerName] (the standard DataLogger table).

Skip Check Columns Type

This is set false for default. When set true, the DataLoggers columns and tag types implemented at runtime will be checked for consistency.

Execution

Recording Frequency

This field is used for specifying the time value (gg:h:m:sec) to determine data recording frequency.

Resetting Tag

This field is used to insert a Tag name of the Server to be used at runtime to reset data in the Data Logger table. This Tag is only taken into consideration when the Data Logger's 'Enabled' property has been enabled. Any 'Enabled Tag' setting will not influence the reset command which will be executed regardlessly.

If the 'Resetting Tag' is set to a value that is different from 'zero' (e.g. 'True', '1', '100', '-10' etc.), the Data Logger will delete all recorded data in the table up to that point and then reset the Tag to zero automatically.



'String' or 'Array' variable types are not supported as Data Logger 'Resetting Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Data Logger will ignore the 'Resetting Tag' at runtime.

Recording Tag

This field is used to insert the name of a Server tag to be used at runtime to execute Data Logger recordings. However, this Tag is only taken into consideration when the Data Logger's 'Enabled' property has been enabled and that the 'Enabled Tag' property be set to a value different from zero if inserted.

If the 'Recording Tag' is set to a value that is different from zero during runtime (e.g. "True", "1", "100", "-10", etc.), the Data Logger will automatically reset the Tag to zero after recording it.



'String' or 'Array' variable types are not supported as Data Logger 'Recording Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Data Logger will ignore the 'Recording Tag' at runtime.

Enable Recording Tag

This field is used to insert the name of a Server Tag to be used at runtime to enable the Data Logger to record. This Tag is taken into consideration only when the Data Logger's 'Enabled' property is enabled.

If the 'Enable Recording Tag' property is set to the 'zero' value at runtime the Data Logger will not be enabled and all the Tags connected to it will not be recorded independently from the activated recording mode (on command or time based). However if the 'Enabled Tag' is set at a different value that is not zero (e.g. "True", "1", "100", "-10", etc.), the Data Logger will result as enabled and the Tags connected to it will be recorded according to the set mode.



'String' or 'Array' variable types are not supported as Data Logger 'Enable Recording Tags'. If one of these unsupported variables is inserted, a warning message will show in the Server log at startup and the Data Logger will ignore the 'Enable Recording Tag' at runtime and result enabled for recording.

Allow Duplicated Rows

Allows the recording of Data Records with the same Time Data.

Table Columns

Reason Column Name

This field is used for specifying the name of the column relating to the information on what caused the value to change. The default name will be used when this field is left empty.

User Column Name

This field is used for specifying the name of the column relating to the user (user who activated recording command). When this field is left blank, the default name will be used instead.

MSec Column Name

This field is used for entering the name of the column relating to the recording time's milliseconds detail. The default name will be used when this field is left empty.

Local Time Column Name

This field is used for specifying the name of the column relating to the data in Local format. When this field is left blank, the default name will be used instead.

UTC Time column Name

The name of the column relating to the data in UTC format. The default name will be used when left empty.



If an error should occur while inserting a record, other attempts will be made by using the time interval set in the "Retry wait time" property between each attempt until the number of attempts indicated in the "Max Errors" has been reached. When the number of retries set in the "Max Errors" has been reached, data will be flushed to the xml file indicated in the "Flush path".



When an already existing DataLogger column is renamed, a new column will be created in the table with the new name and the old column will remain untouched.

2.3. Associating Tags to Data Loggers

A Data Logger prototype is only a "Recording model" commonly known as a Template. To make this recording model operative you will need to associate Tags to it. The operations used to associate tags to a DataLogger prototype are very simple.

After having created your Data Logger you can then use the "Add Data Logger Column" command from the task ribbon displayed in the same resource. This command will ask you to specify Tag and a few properties for the column you wish to add as described below.

DataLogger Columns

After having defined a DataLogger it is necessary to associate Tags to be recorded. To associate a Tag to a DataLogger a new Column must be created in the DataLogger and configured based on the following properties:

Column Name

name of the DataLogger column.

Source Time Stamp Suffix

when enabling this property, a column will be created and added to the table with the "Source Time Stamp" value. The name of the column will be composed with the default name plus the suffix which will be inserted in the edit field on the right.

Server Time Stamp Suffix

when enabling this property, a column will be created and added to the table with the "Server Time Stamp" value. The name of this column will be composed with the default name plus the suffix which will be inserted in the edit field on the right.

Quality Column Suffix

when enabling this property, a column will be created and added to the table with the Tag quality value. The name of the column will be composed with the default name plus the suffix to be entered in the edit field on the left.

User Name Suffix

when enabling this property a column will be created and added to the table to contain the name of the user you activated the recording.

The name of the column will be composed with the default name plus the suffix to be entered in the edit field on the left. The text contained in this field does not always report a user name. For example, if the recording is commanded by the Communication Driver, the user is indicated as Anonymous.

Column Tag

the name of the variable to be associated to the DataLogger column is selected in this field.

