Contact Information

AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales, customer training, and technical support, see https://sw.aveva.com/contact.
# Contents

Contact Information .................................................................................................................. 3

Welcome to Wonderware Historian .......................................................................................... 7

   Wonderware Historian Documentation Set .......................................................................... 7

Process data: About tags and values ....................................................................................... 9

   Three-dimensional data: Value, time, and quality (VTQ) .................................................. 10

   Types of tags ......................................................................................................................... 11

Data acquisition: Getting data into Historian ........................................................................ 13

   Sources of data ..................................................................................................................... 13

   About Wonderware Historian data security ....................................................................... 15

   Store-and-forward safeguards ........................................................................................... 15

   Data categories .................................................................................................................... 16

Data storage: Preserving huge amounts of data over time .................................................... 17

   Storage modes ..................................................................................................................... 17

   History blocks and partitions .............................................................................................. 19

   Auto-summarization ............................................................................................................ 20

Data retrieval: Transforming data into information ............................................................... 23

   Retrieval modes .................................................................................................................. 23

   Data query tools: Accessing your data ................................................................................ 27

Data replication: Delivering information to people who need it .......................................... 29

   How tags are used during replication .................................................................................. 30
Welcome to Wonderware Historian

Wonderware Historian is a powerful, high-performance data historian. It can handle the immense volumes of data that modern industrial facilities generate.

Historian can move all of that data fast, acquiring your data for storage and retrieving it when you need it for process analysis and reporting.

Wonderware Historian combines the power and flexibility of Microsoft SQL Server with high speed acquisition and efficient data compression to store time-series data.

With Wonderware Historian, you can:

- **Acquire data from a various sources**
  Historian's Data Acquisition subsystem offers high-speed data capture to acquire plant data from Wonderware I/O Servers, DAServers, InTouch HMI software, Application Server, and other devices.
  Wonderware Historian optimally acquires and stores analog, discrete, and string data. In addition, the SuiteLink protocol used by Wonderware I/O Servers and DAServers provides time and quality stamps as data is added to the system.

- **Store lots of data efficiently**
  The Storage subsystem compresses the data for maximum storage efficiency. In addition to your plant's data, Wonderware Historian stores alarms and events, summary data, configurations, security information, backups, and system monitoring data.

- **Retrieve data when you want it**
  The Retrieval subsystem responds to SQL requests for plant data and allows you to retrieve large amounts of data very quickly. It also supports REST and SDK data retrieval.

- **Replicate your stored data for security**
  The Replication subsystem replicates tags' values to other Wonderware Historian servers with high fidelity or calculates and replicates summaries of those values.

- **Query data using time domain extensions to SQL**
  Wonderware Historian builds on the capabilities of the SQL query language by supporting time-series data and controlling the resolution of returned data in SQL (for example, to give an evenly spaced sampling of data over a period of time).

- **Create, save, and share graphical content**
  Wonderware Historian includes a tool called Insight that lets you quickly search for data and generate clean, clear graphical content that you can save and share.

**Wonderware Historian Documentation Set**

The Wonderware Historian documentation set includes the following guides:
Welcome to Wonderware Historian

- **Wonderware System Platform Installation Guide**
  This guide provides information on installing the Wonderware Historian, including hardware and software requirements and migration instructions.

- **Wonderware Historian Concepts Guide**
  This guide provides an overview of the entire Wonderware Historian system and its key components.

- **Wonderware Historian Scenarios Guide**
  This guide discusses how to use Wonderware Historian to address some common customer scenarios.

- **Wonderware Historian Administration Guide**
  This guide describes how to administer and maintain an installed Wonderware Historian, such as configuring data acquisition and storage, managing security, and monitoring the system.

- **Wonderware Historian Retrieval Guide**
  This guide describes the retrieval modes and options that you can use to retrieve your data.

- **Wonderware Historian Database Reference**
  This guide provides documentation for all of the Wonderware Historian database entities, such as tables, views, and stored procedures.

- **Wonderware Historian Glossary**
  This guide provides definitions for terms used throughout the documentation set.

In addition, the *Schneider Electric License Manager Guide* describes the Schneider Electric License Manager and how to use it to install, maintain, and delete licenses and license servers on local and remote computers.
Process data: About tags and values

Wonderware Historian acquires and stores process data, which is any information related to successfully running a process. That data is stored as *tags*.

The term "tag" originally referred to physical label on a mechanical part or device on the plant floor. Each tag identifies the corresponding device to Wonderware Historian. As each device sends values to the historian, they are recorded by tag.

For example, a boiler might have two tags – one for the temperature gauge and one for the volume meter.

**Using process data to answer your business questions**

Historian stores and retrieves your process data in a way that allows you bring clarity to issues related to your business.

- **Real-time data**
  
  Historian stores data as it comes in from the plant floor.

  Real-time data answers questions like "What's the temperature of that tank right now?"

- **Historical data**
  
  Historian can accept historical information from other systems and retains data captured in real time to use for historical reference.

  Historical data helps to answer questions like "What was the value of this tag every second last Monday?"

- **Summary data**
  
  Historian summarizes certain data to help answer with big-picture questions like "What is the average number of bags produced each week?"

- **Event data**
  
  Historian records process alarms and events as they happen. This data is used to find answers about when things happened; for example, "How often did that boiler trip last month?"

- **Configuration data**
  
  This data describes your system’s configuration and answers questions like "What types of I/O Servers am I using?"

Process data analysis can help improve performance, enhance quality, and reduce costs.

For more information about defining and using tags, see Defining Tags in the Wonderware Historian Administration Guide.
Three-dimensional data: Value, time, and quality (VTQ)

Each time Historian records a data value, it also records a corresponding timestamp and data quality rating. Together, these three things – value, time, and quality – are called a "VTQ".

**Values alone: Only somewhat helpful**

For example, suppose there is a tag named "tank1.temp" that measures the temperature of a tank in the plant.

The tank’s sensor would send periodic temperatures to Historian – 97 degrees, 98 degrees, 102 degrees, 108 degrees, etc.

These mean little unless you know when the temperatures were taken, so each value also has a date/time stamp.

**Value + Time**

Recording each value with a timestamp allows you to see trends, pinpoint process errors, etc.

Historian tracks when a record is sent by the device and when it is received by Historian. This helps to clarify the information if there is a data lag, or if values are added or updated later.

**Value + Time + Quality**

And because errors can occur – from minor mechanical hiccups to major area-wide blackouts – Historian also records a data quality indicator for each record.

If something happens that may affect the data quality, the quality indicator reflects that. That way, you can know if the quality less than optimal for some records, and use that information to report and as accurately as possible.
Where there are gaps, Historian can provide a best guess of what the values were.

Types of tags

Wonderware Historian can handle a wide range of data by supporting these tag types:

- **Analog**
  Measures a continuous physical quantity, such as a tank’s volume or a boiler’s temperature.

- **Discrete**
  Records one of two states for the tag. For example: on/off, open/closed, jam/cleared.

- **String**
  Captures a text expression—with no special format—that is treated as a single data item. A string tag could be used to capture the state of a machine; for example: “started”, “stopped”, “jammed”, or “cleared”.

- **Event**
  Records an instance when a tag meets a preset requirement. For example, a process event tag can let you know when a batch number changes.

- **System**
  Reflects a predefined system variable. System tags are used to collect the system’s performance data. Wonderware Historian system tags have a “Sys” prefix (for example, SysTimeSec).

- **Analog summary**
  Reflects summarized data (minimum, maximum, average, and so on) that is configured to be replicated from one historian to another.
- **State summary**
  Reflects summarized data (minimum time in state, maximum time in state, average time in state, and so on) that is configured to be replicated from one historian to another, or stored locally.

You can configure analog, discrete, string, and legacy history event tags through the SMC. For more information, see Viewing and Configuring Tags in the *Wonderware Historian Administration Guide*. 
Data acquisition: Getting data into Historian

Historian acquires and processes data several times faster than a traditional relational database.

When data is acquired, Historian attaches a timestamp and quality stamp to the value before committing the record to storage.

Historian acquires both original data and revision data. Historian tracks any modifications to the data, and tracks the means by which it was modified.

Sources of data

Historian can accept data from a number of sources. The most typical scenario is data acquisition from an I/O server.

Data acquisition from I/O servers

An I/O server provides data from plant floor devices to Wonderware Historian using the SuiteLink or Dynamic Data Exchange (DDE) protocol.

First, the I/O server collects data values from programmable logic controllers (PLCs), Remote Telemetry Units (RTUs), and similar devices on the factory floor.

Then, the I/O server uses the SuiteLink or DDE protocol to time and quality stamp each data value it collect.

Next, it passes the data values through the Data Acquisition subsystem, or IDAS. IDAS seamlessly handles data values, regardless of their time.

Note: The SuiteLink protocol can be used to collect data from an I/O server on the same or different computer than the IDAS instance. The DDE protocol can be used only when the I/O server is on the same computer as the IDAS instance.

For each data value acquired by IDAS, the timestamp, value, and quality are attached.
Then the values are sent through the Historian Client Access Layer (HCAL) to a Historian Client Access Point (HCAP) on the Historian server, and then to storage. HCAL is a client-side software layer that provides programmatic access to storage, retrieval, and system configuration functionality in the Wonderware Historian.

Historian accepts and historizes each data value according to the storage rules for the tag to which the data value belongs.

For more details, see Configuring Data Acquisition in the *Wonderware Historian Administration Guide*.

**Other data acquisition options**

As this diagram illustrates, Wonderware Historian can accept data from a range of sources.

- **TransactSQL INSERT and UPDATE statements**
  You can insert or update history data in the Wonderware Historian extension tables using Transact-SQL INSERT and UPDATE statements.
  For more information, see Importing, Inserting, or Updating History Data in the *Wonderware Historian Administration Guide*.

- **CSV and LGH files**
  Using the Historian Data Importer utility (aahImport), you can add history data from a file to Wonderware Historian. This utility reads data from InTouch history (LGH) files or comma-separated value (CSV) files, and then sends the data to the Historian server via HCAL.
  Imported data is integrated with data currently stored in history blocks, providing you with seamless access to all your data.
  For more information, see Importing, Inserting, or Updating History Data in the *Wonderware Historian Administration Guide*.

- **App Server, custom SDK client applications, tier-1 replication, and other sources**
  These sources are also able to use HCAL to send data to Historian.

- **Wonderware Historian itself**
  Configuration data comes from the Historian itself.
About Wonderware Historian data security

Wonderware Historian uses an integrated security model to control who can access and update data for the historian. Using this model, each person and computer that accesses or updates historian data is assigned membership in one of three security groups:

- Administrators
- Power Users
- Users

Data can be passed from any computer that's a member of the historian's Power User or Administrator group. (Computers must be on the same domain as the historian.)

When data is ready to be sent from a remote computer, the Wonderware Historian pushes configuration information, including ACLs (access control lists) that define access permissions, to HCAP on the client computer. HCAP launches IDAS on the remote computer and data is sent through HCAL to the historian.

Store-and-forward safeguards

Historian uses a store-and-forward method to protect against data loss if communication is interrupted between the data source and Historian.

Systems using the Data Acquisition Subsystem (IDAS) or Historian Client Access Layer (HCAL) to send data to Historian are able to use the store-and-forward method in case of communication breaks. If the data source loses communication with Historian, the source stores the collected data until communication is reestablished. Then, it forwards the stored data to Historian.
Data categories

Wonderware Historian is able to process and store data in a variety of ways. It categorizes each data record by type to provide a consistent framework for data operations. Each category of data has a separate set of characteristics and is handled differently by the historian.

Original versus revision data

Data acquired by Wonderware Historian can be categorized as:

- **Original data** is the data received from the data source originally—that is, for the first time. For example, a real-time stream of data from an I/O server represents original data. Usually the original data arrives to a Historian in high data volumes for many tags with timestamps close to each other.

- **Revision data**, by contrast, is data that corrects or appends original data. Revision data operations are performed on a per-tag basis and typically have far lower volumes than original data.

Streamed versus non-streamed data

Original data can be streamed or non-streamed.

- **Streamed**: If the data source is able to enforce time order for its output, the data is streamed data. Streamed data has three subtypes:
  - Real-time data is in time order, where the timestamp is in the past relative to the current Wonderware Historian time.
  - Late data is in time order, where the timestamp is far in the past compared to the current Wonderware Historian time.
  - Replication data is data that has been replicated from a tier-1 historian to a tier-2 historian.

- **Non-streamed**: If the time order is not enforced, the data is non-streamed.
Data storage: Preserving huge amounts of data over time

Historian, through its Storage subsystem, saves plant data from various sources to disk.

Historian's efficient storage model includes:

- **A range of storage modes**
  Historian's storage modes allows storage of every meaningful value and extrapolation of all other values at retrieval time.

- **History blocks and partitions**
  Historian efficiently warehouses huge amounts of data using a compact storage format that is optimized for time-series data.

- **Auto-summarization**
  Historian calculates and stores automatic summary records as real-time analog tag values. These auto-summaries are used provide fast, seamless data retrieval, no matter what the granularity.

For information about configuring the Storage subsystem, see Managing Data Storage in the *Wonderware Historian Administration Guide*.

**Storage modes**

Depending on a tag's definition, Historian uses one of these storage modes to retain the values received for that tag:

- **No storage** - No values are stored.
- **Forced storage** - All collected values are stored.

For comparison's sake, this is what forced storage looks like. The red dots represent collected values. All of these values are stored by Historian.
Cyclic storage - Only values that occur at a specified time interval are stored. Using the same collected values as shown above, cyclic storage retains only the values represented by red dots.

Delta storage - Only changed values are stored.

Types of delta storage
Delta storage, as a rule, requires Historian to store any value that is different than the previously received value. Time or deadband rules can be applied for delta storage to further constrain what values are stored.

These are all types of delta storage:
- **Time-enforced delta storage** -- Any changed value is recorded. Additionally, a record must be stored after a given amount of time. This storage mode is used by the Historian SDK.
- **Time deadband delta storage** -- Only changes outside of a particular time deadband are stored.
- **Value deadband delta storage** -- Only changes outside of a particular value deadband are stored.
- **Rate of change (swinging door) deadband storage** -- Only changes outside of a particular rate-of-change (swinging door) deadband are stored.

Measuring change with deadbands
Delta storage retains only those values that have significantly changed from the previously stored value.

For example, if you had a discrete (binary) tag that reflected the state of a power switch, you may not want to record every time the system checks to see that it is switched on. You might really be interested only in when it switches off when it is supposed to be running, and when it gets switched on again.

For analog (numeric) tags, you may only care only about large changes, but not tiny ones. Or, you may want a snapshot of values at certain intervals, and not every one that is reported. You can filter out extraneous value with deadbands.
A **time deadband** is a time filter. It marks the minimum time (milliseconds) between stored values for a single tag. Any value changes that occur within the time deadband are not stored.

For example, these red and blue points are all the values reported for a certain tag. The orange bars represent the time deadband, which starts over with every reported value. Only the red points (P2, P4, P7, P8, P9, P11) are stored. The other points are excluded because they fall within a deadband or outside of the time period.

A **value deadband** is a filter that marks the percentage of the difference between the minimum and maximum engineering units for the tag. Any data values that change less than the specified deadband are not stored.

Here, the orange bars represent the value deadband, which starts over with every reported value. Only the red points (P2, P5, P6, P7, P10, P11) are stored. The other points are not stored because they fall within a deadband or outside of the time period. P9 is not stored because P8 was discarded and it is within the percentage deviation.

A **swinging-door deadband** marks a rate of change deadband, based on changes in the slope of the received values.

For example, specifying a swinging door deadband value of 10 percent means that values are stored if the percentage change in slope of the consecutive data values exceeds 10 percent.

For more information on delta storage modes, see About Delta Storage Mode in the Wonderware Historian Administration Guide.

### History blocks and partitions

Wonderware Historian stores data in history blocks. History blocks use a proprietary file format and are essentially subfolders of the main historian storage folder.

Although the tag values are stored in history blocks on disk, the values appear to be saved to tables in the Runtime database.
Each history block stores all data for a specified duration. The default history block duration is one day, but may be as little as one hour. When there is data to be stored in that time interval, Historian creates a new history block for that data. For example, if a history block is defined for a day's worth of data, when it receives the first data value for the second day, Historian creates a new one-day history block and places the corresponding data into the new block.

As data is acquired, the size of these history blocks grows on a continual basis, being limited only by the size of the hard disk on which the historian resides.

If the historian was not running for some time, or if a history block is deleted, for a certain time period, there may be a gap in the sequence of history blocks -- also known as a block gap.

History block formats are specially optimized for storing time-series data, while general-purpose database management systems typically are not.

Compact storage formats reduce the storage space requirements than would be required in a general-purpose database. Upon retrieval, historical data is presented by the Wonderware Historian OLE DB provider as if it were stored in SQL Server tables.

**Historian partitions**

Historian organizes history blocks within partitions. As real-time data arrives, Historian stores it in history blocks located in the main data partition. At the same time, Historian automatically computes and records a corresponding hourly summary for each analog tag value received. The auto-summary values are stored in auto-summary history blocks within the auto-summary partition.

For more information on history blocks and partitions, see Managing Partitions and History Blocks in the *Wonderware Historian Administration Guide*.

**Auto-summarization**

For every analog tag in the system, Wonderware Historian creates a local replication entity and a one-hour summary tag. As values arrive for an analog tag, Historian automatically computes and records a summary.

Auto-summary values are stored in their own history blocks within the auto-summary partition.

With auto-summarization, Historian can quickly and efficiently retrieve large-volume data for a long duration, even months or years.

*Note:* The auto-summarization feature is enabled by default, but can be disabled.
Data retrieval: Transforming data into information

Wonderware Historian can process SQL-based queries from any number of client applications, including: Wonderware Historian InSight, Wonderware Historian Client, and ad hoc SQL query tools. It can also process queries via the OData interface and from SDK client applications.

When Historian receives a request for data, it performs the following steps:

1. Locates the requested data.
   - Historized process data is stored in history blocks.
   - Configuration data is stored in SQL Server database tables.
   - Replication data is stored in history blocks.

2. Apply a retrieval mode to the data.
   - Because of the enormous amount of data potentially associated with a facility, Historian provides several retrieval modes that help interpret that data, turning the collected data into usable information.

3. Returns the results to the client application.

For more information about retrieving data from Wonderware Historian, see the Wonderware Historian Retrieval Guide.

Retrieval modes

Historian can acquire and store huge amounts of data and allows you to choose from among several retrieval modes to view and interpret the data you need.
**Cyclic**

Retrieves one value per cycle. Whatever the value is when the cycle begins.

**Delta**

Retrieves a value each time the value changes from the previous value. For example, if the value of "4" followed an earlier value of "4", it would not be retrieved. But if "4" followed "3", it would.

**Full**

Every value within a time period is retrieved.

**Interpolated**

Based on values before and after a certain point in time, Historian estimates the value for that time.

In this example, P2 is located exactly at the query start time. Because of this, P2 is returned at that time without need for interpolation. At the following cycle boundary, point PC1 is returned, which is the NULL value represented by P7 shifted forward to time TC1. At the last cycle boundary, point PC2 is returned, which has been interpolated using points P11 and P12.
**Best Fit**

"Best fit" retrieval allows for a compromise between delta retrieval and cyclic retrieval.

Delta retrieval can accurately represent a process over a long period of time, but requires a large number of data values. Cyclic retrieval is much more efficient, but less accurate, because of fewer values.

Best fit provides faster retrieval, like cyclic retrieval, plus the better representation, like delta retrieval.

**Average**

Uses a time-weighted average algorithm to calculate the value for each retrieval cycle.

For the following data values of a tag that uses linear interpolation, the time-weighted average is computed as:

\[
\text{Average} = \frac{((P1 + P2) / 2) \times (T2 - T1) + ((P2 + P3) / 2) \times (T3 - T2) + ((P3 + P4) / 2) \times (T4 - T3))}{(T4 - T1)}
\]

**Minimum**

Returns the minimum value from the actual data values within a retrieval cycle. If there are no actual data points stored on the historian for a given cycle, nothing is returned. If there are NULL values in the cycle, NULL is returned for that cycle.

**Maximum**

Similarly, this mode returns the maximum value of actual data for the retrieval cycle.
**Integral**

Calculates the values at retrieval cycle boundaries by integrating the graph described by the points stored for the tag. In other words, it works much like average retrieval, but it additionally applies a scaling factor. This retrieval mode is useful for calculating volume for a particular tag (for example, gallons of water flowing through a valve over a certain period).

Integral retrieval works with analog tags only. For all other tags, normal cyclic results are returned.

**Slope**

Returns the slope of a line drawn through a given point and the point immediately before it, thus expressing the rate at which values change.

For example, two points P1 and P2 occur at times T1 and T2. The slope is calculated as:

\[(P2 - P1) / (T2 - T1)\]

The difference between T1 and T2 is measured in seconds, so the returned value represents the change in engineering units per second.

**Counter**

The change in a tag's value from the beginning to the end of the period, factoring in any rollover value for the counter. This retrieval mode is useful for determining how much of an item was produced during a particular time period.

**ValueState**

Returns information on how long a tag has been in a particular value state during each retrieval cycle. That is, a time-in-state calculation is applied to the tag value.
**RoundTrip**

Like ValueState retrieval, this mode uses state occurrences within a period for its calculations. RoundTrip retrieval calculates the time between consecutive leading edges of the same state.

---

**Data query tools: Accessing your data**

To report on the data you have stored in Wonderware Historian, you can use:

- Wonderware Historian InSight, a tool included with Wonderware Historian
- Transact-SQL queries
- Any other query tool that can access SQL data sources

Wonderware Historian is part of a client/server architecture that supports desktop client applications, while ensuring the integrity and security of data on the server. This architecture provides common access to time-series data and associated configuration, event, and business data.

The computing power of both the client and the server is exploited by optimizing processor intensive operations on the server and minimizing data to be transmitted on the network to improve system performance.

Microsoft SQL Server acts as the gateway for accessing any type of information in the historian. Historian uses Microsoft linked server technology to plug in its own OLE DB provider. Because of this, any client application that can connect to Microsoft SQL Server can also connect to Wonderware Historian.

For users with client applications, it seems that queries are made to the Runtime database. That database is, in fact, the logical interface to the data.

When it receives a request for data, the Retrieval subsystem retrieves the historized data from history blocks.

**Querying with Wonderware Historian InSight**

Wonderware Historian InSight (included with Wonderware Historian) is a search-based tool that lets you quickly turn your data into easy-to-read charts.
With InSight, you can type the name -- or even a part of a name -- for the tags you want to analyze. Then you can choose the chart type and timeframe to report on. InSight also lets you save and share your data.

For more information on using InSight, see the online help.

**Querying with Transact-SQL**

Historian can handle traditional SQL queries. For example:

**SQL Query**

```sql
select Value as Step,
       Label as Label,
       StateTimeAvg/:Contained as Avg,
       StateTimeMin/:Contained as Fastest,
       StateTimeMax as Longest
from StateSummaryHistory
where TagName = 'Filler1:Step'
  and StartDateTime >= '2016-05-01'
  and EndDateTime <= '2016-05-02'
  and WWCycleCount = 1
  and WWRetrieveMode='cyclic'
```

**Result**

<table>
<thead>
<tr>
<th>Step</th>
<th>Label</th>
<th>Avg</th>
<th>Fastest</th>
<th>Longest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Load</td>
<td>1029</td>
<td>1020</td>
<td>1038</td>
</tr>
<tr>
<td>2</td>
<td>Open</td>
<td>1009</td>
<td>1006</td>
<td>1012</td>
</tr>
<tr>
<td>3</td>
<td>Fill</td>
<td>5710</td>
<td>5612</td>
<td>5842</td>
</tr>
<tr>
<td>4</td>
<td>Settle</td>
<td>2022</td>
<td>2018</td>
<td>2036</td>
</tr>
<tr>
<td>5</td>
<td>Close</td>
<td>4558</td>
<td>1074</td>
<td>5816</td>
</tr>
<tr>
<td>6</td>
<td>Seal</td>
<td>1067</td>
<td>1058</td>
<td>1176</td>
</tr>
<tr>
<td>7</td>
<td>Release</td>
<td>1120</td>
<td>994</td>
<td>1140</td>
</tr>
</tbody>
</table>
Data replication: Delivering information to people who need it

Data from one Wonderware Historian can be replicated to one or more other Wonderware Historians, creating a "tiered" relationship between the historians.

Setting up replication servers is useful for several circumstances, including:

- When you want to replicate data from multiple individual historians (called tier-1 historians) into a single centralized historian (called a tier-2 historian).
- You want to set up a many-to-many relationship between tier-1 and tier-2 historians.
- When your facility has a firewall and you want users on both sides of the firewall to have access to the same data.
- When you want data available from the cloud.

Historian can replicate process data as well as alarms and events.

A historian can act as a tier-1 and a tier-2 historian simultaneously.

Centralized

Cloud
Many-to-many

For more information about setting up and using replication, see Managing and Configuring Replication in the Wonderware Historian Administration Guide.

How tags are used during replication

Data from a tier-1 historian is replicated to a tier-2 historian using tags in the same way that information is collected by an individual Wonderware historian.

The tier for a tag is determined by where it comes from:

- Values for tier-1 tags are gathered directly from an IDAS or HCAL-based data sources. HCAL-based data sources include, for example, Wonderware Application Server or Historian SDK applications, where tags are imported through CSV files, or Transact-SQL inserts and updates.

- Values for tier-2 tags come from another Wonderware Historian.

A historian can act as a tier-1 and a tier-2 historian simultaneously.

A typical scenario for a tiered historian appears in the following example. Tag1 is collected on historian A and all its values are replicated to historian B, where they are stored as values of tag2. At the same time historian B collects data for its tag3 and all its values are replicated to historian C, where they are stored as values of tag4.

In this example, the tags are identified as follows:
- tag1 is a tier-1 tag of Historian A
- tag2 is a tier-2 tag of Historian B
- tag3 is a tier-1 tag of Historian B
- tag4 is a tier-2 tag of Historian C
- Historian A is a tier-1 historian
- Historian B is both tier-1 and tier-2 historian
- Historian C is a tier-2 historian

**Important:** Be careful not to create or modify a replicated tag on a tier-1 historian to have the same tagname that already exists on a tier-2 historian. The system does not prevent you from having a replicated tag on a tier-2 historian receiving data from two or more different tier-1 historians. However, when you retrieve data for that replicated tag on the tier-2 historian using the tagname, an incorrect blend of data from the two (or more) data sources is returned.

There are two types of replication: simple replication and summary replication. Summary replication provides periodic summaries of high resolution data, while simple replication retains the original data resolution.