

# Algorithmica History Server

## PRODUCT BRIEF

## Algorithmica History Server — enterprise-wide management of historical data

The Algorithmica History Server (AHS) allows the user to create a local database of historical information, originally collected from a real-time financial data service. The time-series are kept in a database of the user's choice, which may then be accessed via different client applications for display and manipulation.

Quantitative analysis, trading, and risk management typically require large amounts of time-series data. Whether the data is used for automatic pricing in legacy systems or for generating trading propositions in desk-

top applications, the need for quality-assured and coherent data is the same.

Installing AHS to centrally manage data collection ensures maximum cost benefits from large scale filtering and monitoring. Adapting a centralized data management strategy has been the choice of most large financial institutions.

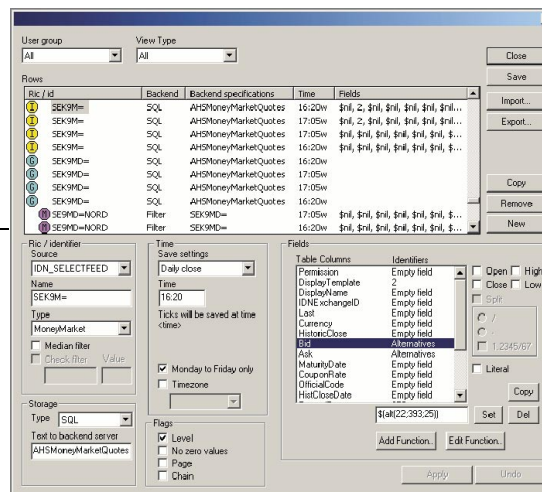
By using the Data Manager Client, the organization will be able to define, build and maintain a local database, storing thousands of financial time-series from different data sources. The configuration can be changed at any time without interrupting the running services.

The AHS is designed to be independent of the chosen database solution. No expensive new software or hardware is necessary. Any applications based on in-house databases can still be used. Substantial cost savings can be achieved by reduced maintenance and educational efforts when using familiar technology.

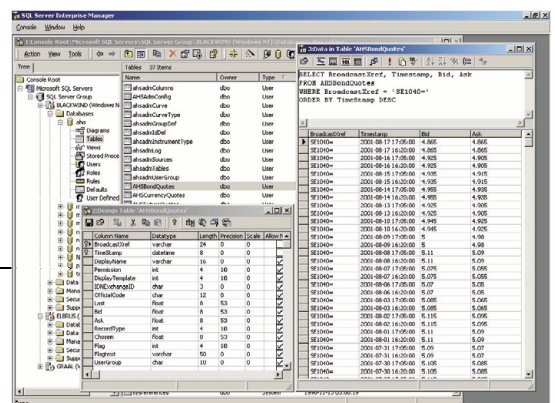
Time series data can be stored on all ODBC-compliant relational databases. AHS works well together with Microsoft SQL Server, Oracle and IBM DB2.

Data can also be formatted to fit a Fame database or other specialized time-series formats.

Control and maintain the collection of thousands of real-time series with the user-friendly Data Manager Client



With the Algorithmica History Server in place, extract time series using your favourite database tools



## Sampling of data

### General features

Storage frequencies can be configured for single instruments and for complete markets.

Typical storage frequencies used when working with a real-time feed include:

- tick by tick
- multiple of minutes
- multiple of hours
- single or multiple time of day

When entering instruments into the system, desired filter rules can be applied.

Typical filters include:

- data spikes filter
- zero value filter
- relative difference filter
- stale data filter
- median filter

More advanced filters can easily be added, such as:

- yield-to-price conversion
- implicit volatility conversion
- calculated proxy replacement

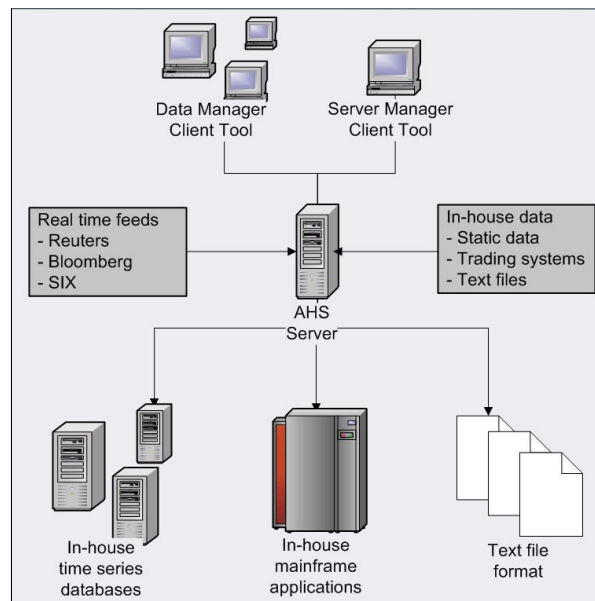
## System overview

Thanks to the fact that AHS is built around continuous monitoring of live feeds, it is not as sensitive to disturbances as common snap-shot applications.

Instruments can be entered either by their specific id or as a part of a complete chain of instruments. Entering data using templates is another

convenient feature of the Algorithmica History Server.

By running the process “backwards” and contribute data from internal pricing systems back into the real-time feed network, the Algorithmica History Server may very well be used as a real-time pricing engine.



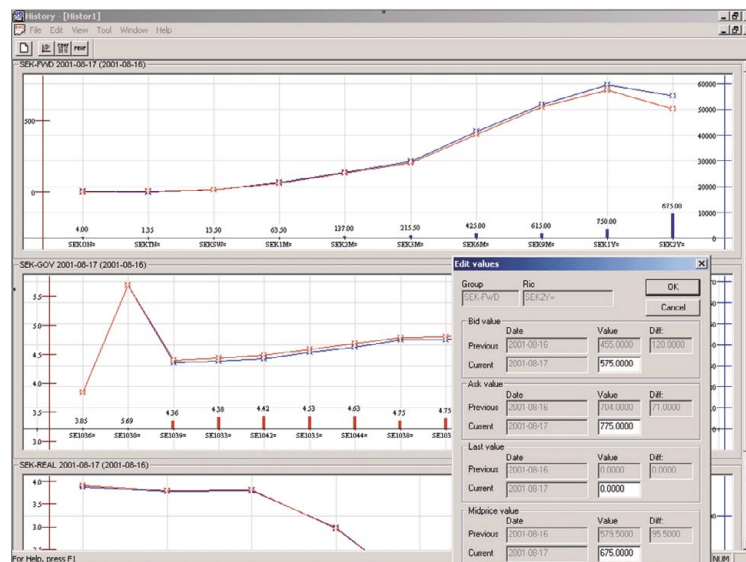
The Algorithmica History Server is available for Microsoft Windows Server and Sun Solaris

## Supported feeds

The AHS server can capture real-time data from most data vendors and handle several sources simultaneously.

Currently supported feeds include:

- Bloomberg
- Markit
- Reuters SSL
- SIX p-server
- Thomson
- Text file formats



## Detection of corrupt data

Using standardized or custom filters, AHS can significantly speed up the time-consuming task of cleaning and verifying incoming real-time data.

Using the graphical visualization component, the operator can scan large amounts of time-series data in minutes.

Data that can be mapped onto curves is also presented in curve format.

Any changes to the historical data are recorded, thus enabling a complete audit trail. This is an important feature for data used in pricing systems or other systems with high demands for data integrity.