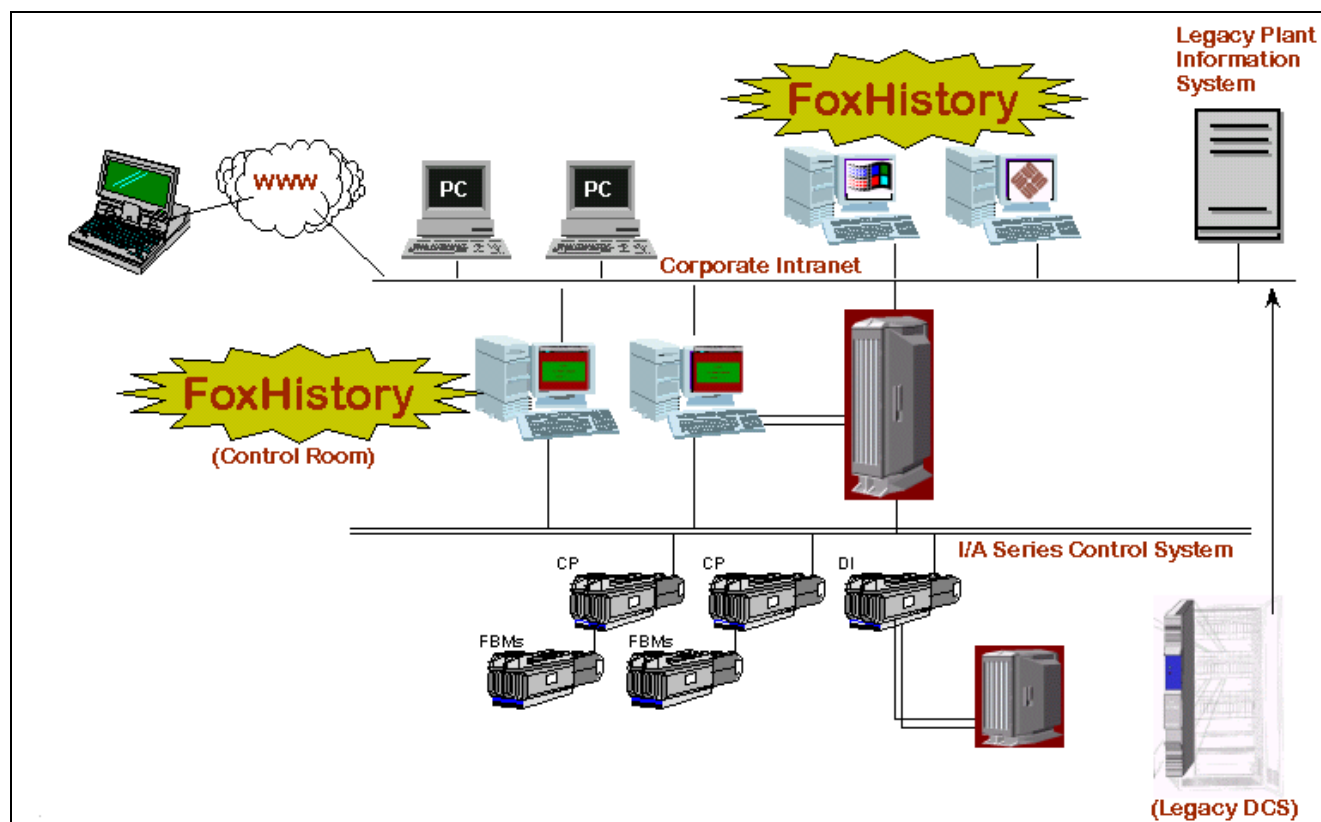


I/A Series[®] Software

FoxHistory



FoxHistory provides enterprise-wide capability for collection, storage, and retrieval of historical process information. FoxHistory can run on both I/A Series systems or on network servers, and can collect information both from I/A Series and proprietary automation systems. FoxHistory repositories are easily integrated with Microsoft personal productivity applications and relational databases.

FoxHistory

Today's manufacturing processes both generate and consume ever larger and more varied amounts of historical information. Manufacturers often need to store this information indefinitely. This information must be both accessible and easy to annotate or augment. Rapid access to this information is important for many reasons including graphical and statistical analysis, regulatory and batch reporting, analyzing product quality, and financial reporting.

Process history is generated by many different devices and systems. Originating at different times and locations, it consists of items such as measurements, alarms, messages, events, and operator actions. It is needed by control system users, network application servers, desktop PCs, and by advanced control and business management applications. Due to this diversity in both source and use, process historical information must be made accessible via the enterprise Intranet using widely accepted, standard client/server technologies. Both Microsoft and Internet standards play increasingly important roles in the distribution and analysis of this information.

In order to meet modern needs for process information, Foxboro has developed FoxHistory. FoxHistory is a software system that enables you to collect, store, and retrieve a wide variety of process historical data using client/server technologies. Information can be stored and accessed over both I/A Series nodes and the enterprise Intranet. You can store and access data using industry standard or proprietary access methods. FoxHistory is capable of supporting today's process systems as well as increasing needs and requirements of tomorrow's manufacturers. FoxHistory can be "retrofitted" to replace I/A Series historians on existing I/A Series systems. Alternatively, it can operate with existing I/A Series historians and applications to give them greater capability and higher performance.

FoxHistory Capabilities

FoxHistory provides many advanced capabilities and benefits. Among these are:

Platform Independence

FoxHistory can, but need not, reside on an I/A Series system. The FoxHistory server package can run on I/A Series stations (AP51, AW51, and AW70), or on network server systems running Solaris or Windows NT operating systems. FoxHistory collectors, where events and real-time data are collected for forwarding to the FoxHistory server, run on these same platforms. Applications using FoxHistory data can reside on any platform supported by FoxAPI (including Windows95, Windows NT, Solaris, VMS, Digital UNIX, HP-UX, AIX, OS/2 and others). Finally, FoxHistory instances running on network servers can utilize both local and network storage resources (such as network storage "appliances"). A wide variety of such computing platforms and bulk storage devices are available. These include computing and storage capacities far greater than required by any control system server platform.

Wide Sizing and Performance Options

FoxHistory serves in a wide range of applications because it sizes from small tasks to plant-wide applications. FoxHistory can be configured to collect as few as 200 to a virtually unlimited number of real-time points. Multiple instances of FoxHistory can exist on a single server. This allows individuals or groups to set up collections for their own specific purposes, without becoming dependent on servers in another organization, while at the same time allowing them to access data from any server plantwide. FoxHistory's ability to run on standard network server platforms provides a full range of hardware performance, mass storage size, and price options.

Collection from Proprietary Systems

FoxHistory is optionally capable of collecting data both from I/A Series and from many different types of Distributed Control Systems (DCSs) and other control products. FoxHistory's Data Collector includes client software for "@aGlance/IT". This allows FoxHistory to connect to any system which is equipped with corresponding @aGlance server software. These servers are available for many common DCSs and plant historian products, allowing FoxHistory to collect data from them and store this data in the same repository with I/A Series process history. Data retrieval mechanisms for FoxHistory work with both types of data seamlessly.

Unlimited Data Types

FoxHistory automatically recognizes and collects all types of I/A Series system data and messages. These data types are integer, float, long, Boolean, character, packed integer, and packed long. It also collects generic data types including arrays of characters, short, unsigned short, integer, unsigned integer, float, and double. Binary large objects (BLOBs) are supported. A BLOB is an arbitrarily large piece of data with unspecified content (such as a video image). A file data-type is any file. The content of the file is stored if data in the file has been updated or when requested by an external trigger.

On-line Configuration

On-line configuration of FoxHistory is supported. Configuration of an FoxHistory is done in a session. The creator of the session can save, abort, or commit the session. A FoxHistory configuration is updated automatically when a session is committed.

Open Database Connectivity (ODBC) Access

FoxHistory includes an Open Database Connectivity (ODBC) driver to make all stored information accessible via the ODBC Application Program Interface (API). ODBC is an important feature because it presents FoxHistory data as a relational database to clients using the ODBC API. By presenting stored information as a relational database, FoxHistory information can be quickly accepted and integrated into your organization. Many applications are already equipped to handle ODBC data sources. This reduces the cost of integrating FoxHistory information with other databases and applications. Many applications, including Microsoft Office applications and all major Relational Database Management System (RDBMS) products, support the ODBC interface. This simplifies integration of FoxHistory with these applications.

RDBMS Integration

FoxHistory provides optional components for the integration of FoxHistory with INFORMIX and Oracle RDBMS products, allowing FoxHistory data to populate these commonly used database systems.

Universal Coordinated Time (UCT)

The history information collected by FoxHistory can be stored in Universal Coordinated Time (UCT) or I/A Series system time. UCT time is appropriate if data is accessed from different time-zones. Automatic time-zone translation is then utilized to translate between local time and UCT.

FoxHistory Compatibility

Improved performance and new capabilities are always welcome, but not if they negate your existing investments in systems and applications! FoxHistory has been designed to work with existing Foxboro and user tools and applications. FoxHistory can co-exist with any existing I/A Series Historian on I/A Series systems. It can also serve existing I/A Series applications including the I/A Series Display Manager, FoxView, FoxAnalyst, and Data for Windows. To these and other I/A Series applications, FoxHistory emulates the I/A Series Historian. FoxHistory includes modified communication servers which emulate I/A Series Historian services. These servers have been modified to access FoxHistory data stores. Existing applications communicate with these servers to retrieve data from FoxHistory data base. Additional servers are supplied to support communication with FoxHistory instances running on network servers. For applications using INFORMIX ESQL/C, the ODBC interface driver is implemented to facilitate the ESQL query of FoxHistory database. The ODBC driver includes all ODBC core functions. A porting tool is provided to assist in porting applications using INFORMIX ESQL/C library functions to ODBC.

FoxHistory Architecture

FoxHistory has a highly modular package architecture. The basic architecture consists of named "instances" of the FoxHistory package. Multiple instances of FoxHistory can reside on a single server (for example, one might be a small instance collecting recent data for support of local CRT displays while another would be a large instance for long-term collection and storage of plantwide data).

Each instance of FoxHistory consists of one server component and one or more collector components (see Figure 1). The server component has several key functions:

- Manage additions to its data repository (which may be local or remote).
- Manage its archives.
- Provide services to applications (such as operator and desktop displays, and remote databases).

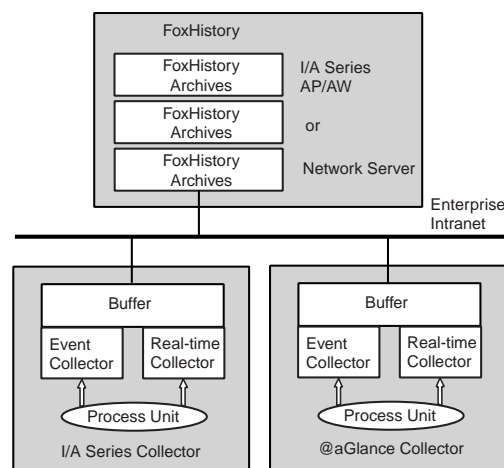


Figure 1. FoxHistory Architecture

Data Collectors

FoxHistory has two types of Data Collectors. The I/A Series Collector is used to capture and buffer data and messages from an I/A Series system. The second type of Data Collector is the optional @aGlance Collector. This collector connects to an @aGlance server in an existing DCS or historian. These servers are available for a variety of DCS, process control systems, and plant historians.

All collectors have both continuous and event interfaces. They also buffer data during periods when the server component is not available.

FoxHistory Data Collection

Real-Time Data Collection

Real-time information is specified by sets of Real-Time Points (RTP) and is collected by real-time collectors or generated by applications (such as manual data entry programs). Data is sent to FoxHistory server to be stored. The RTP can be of any data type. These include I/A Series data types, generic data types, binary large objects (BLOBs), or files.

A RTP value is collected by a real-time collector and stored in FoxHistory database. A RTP has both a fast and a slow collection frequency. A collector always sends the value at the fast frequency but the value is stored at the rate specified in the RTP. A real-time collector gathers and sends a value if one or more of the following criteria are met:

- The value of an RTP changes significantly.
- A RTP is "triggered" by an external trigger (any event or RTP can be designated as the trigger).
- A maximum time between samples (MTBS) is reached.

The fastest collection frequency of a RTP is one millisecond.

New RTPs can be added on-line, and can be removed on-line. A RTP can always be deleted after it has collected data.

Event Collection

Much of the information in a plant cannot be defined as a RTP. Such information may have a variable length or a mixture of data types within the information package. FoxHistory classifies this type of information as events. Examples of events are: alarm messages, annotations, operator actions, batch messages, and manual data entries. An event in FoxHistory consists of a user-defined definition and event instances. The definition consists of assigning keys to every field in the message. The keys are used for sending and receiving the message over a network and for querying the resulting data stores. All FoxHistory users are allowed to specify event definitions. Event types are defined as part of an event group. All message types within a group are automatically "joined" when accessed through the ODBC interface. A message key is defined by a key name, an ODBC key indication, a sequence number, a field type, and the number of elements of the specified type. The key name becomes the ODBC column name. The ODBC specification allows only single-element columns. Compatibility with this restriction is maintained by FoxHistory during the key specification.

Data Retrieval from FoxHistory

Data retrieval from FoxHistory is done via any of four major interfaces (see Figure 2).

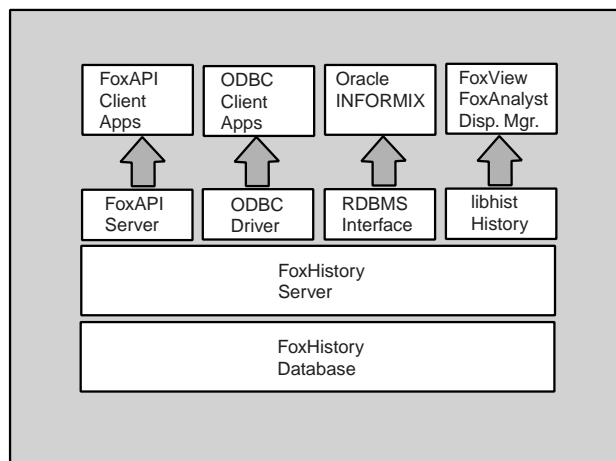


Figure 2. FoxHistory Data Retrieval Interfaces

I/A Series Interfaces

FoxHistory serves existing applications on I/A Series systems such as FoxView, FoxAnalyst, and the I/A Series Display Manager. To these and other applications, FoxHistory emulates an I/A Series Historian. FoxHistory includes modified servers which provide I/A Series historian services. These servers have been modified to access FoxHistory (rather than Historian) data stores. Existing applications (such as FoxView) can communicate transparently with these servers to retrieve data from FoxHistory database. Additional servers are available to support communication with FoxHistory instances which are located on network servers.

FoxAPI

Sometimes very high performance is necessary for access to FoxHistory information. For these cases, an extensive, published, supported, proprietary API (FoxAPI) is available. All FoxAPI functions can be used locally as well as remotely through the Networked FoxAPI product, thus readily integrating process history information into almost any popular computing platform or system.

RDBMS Interfaces

Specialized interfaces are available to integrate FoxHistory databases with major commercial database applications. These are add-in components to FoxHistory. They are component-specific add-ons for support of Oracle, INFORMIX, or Microsoft RDBMS offerings.

FoxHistory Configuration

FoxHistory provides a graphical interface for quickly configuring and monitoring FoxHistory packages on a networked PC (Windows NT or Windows 98). These incorporate a highly intuitive graphical user interface (GUI). Figure 3 shows part of a session where FoxHistory is being set up.

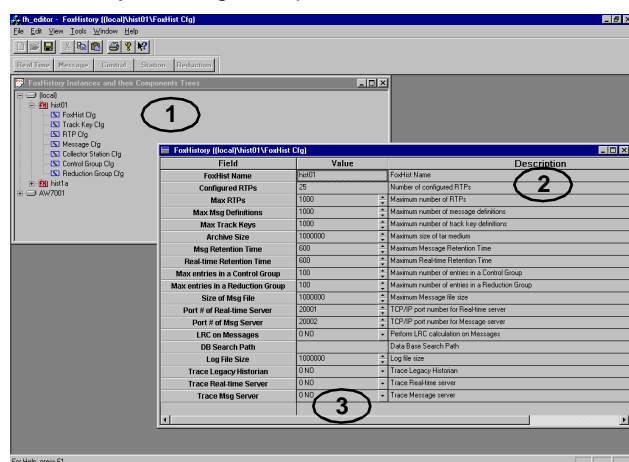


Figure 3. FoxHistory Configuration

1. This window gives the user a hierarchical view of all the FoxHistory packages loaded on his system. There are four on this system. The user then expands the view of any of these four instances and edits its configuration by double-clicking it.
2. This window pops up to show the current configuration of the FoxHistory instance with the logical name "al". The configuration parameters can then be edited or toggled as shown by (3).

Figure 4 shows a further example.

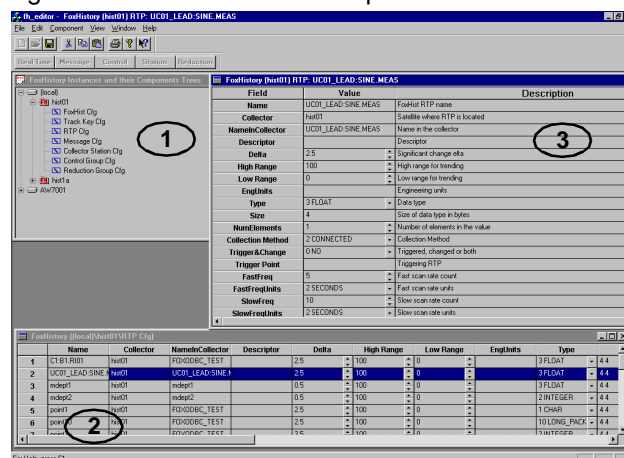


Figure 4. FoxHistory RTP Settings

1. The user picks "Real-time Points" (RTPs) from the same FoxHistory hierarchical window. This switches the user from configuring the general parameters of FoxHistory to configuring the real-time points which are to be collected.
2. The pop-up window shows the points configured for this FoxHistory. When one of these points is picked, detailed information on that point is displayed in a configuration window.
3. The configuration window allows modification of any characteristics of the collection. Note that the configurator allows any selection to be cut and pasted into another. This includes entire configurations. To copy all the RTPs from one FoxHistory to another, just drag and drop.

FoxHistory provides a series of "browsers" to allow any user to view the configuration and collected data, without allowing them to modify either (see Figure 5). A FoxHistory status monitor shows a hierarchical set of status displays which present the user the status of all the installed FoxHistory components.

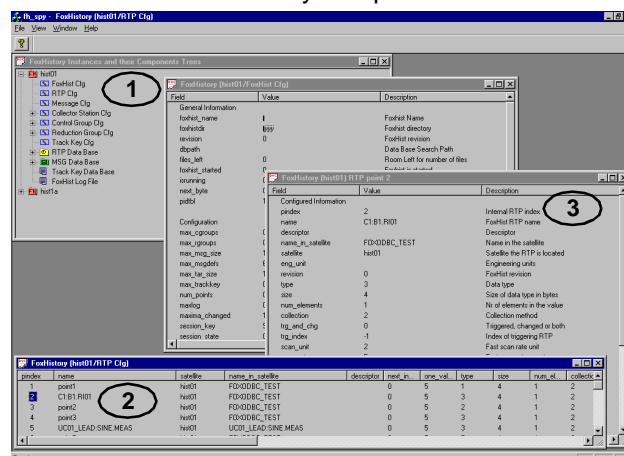


Figure 5. FoxHistory Data Browser

Reduction Files

For the support of I/A Series Historian functions, it is sometimes desirable to view data reduction. A reduction specification defines the reduction type, the reduction period, required percent of valid data, and the RTPs belonging to the reduction files. FoxHistory does not actually perform and store reduced data. Rather, it calculates intermediate values and performs data reductions "on-demand", using the large on-line data repository of FoxHistory.

Control Groups

Sometimes groups of real-time points and events need to be associated. A control group makes this association. All RTPs in a control group are switched between On/Off, Bad/OK, and Fast/Slow by one operation. Event collections are switched between On/Off by the same operation. Multiple RTPs can be assigned as the controlling RTP for the control group.

For a control group any application can control:

- On/Off state of RTPs and Event Messages
- Bad/OK status
- Switching between fast and slow frequency

Archiving

FoxHistory supports a simple and powerful repository archiving system. The FoxHistory data repository can be allocated to any part of the server's file system. The use of this file space is determined by a user-specified sequence. No concatenated storage devices are required to support large repositories. All repository archiving operations can be performed while FoxHistory and its collectors are operating. Historical data can be archived on-line automatically to any system bulk storage device. These can include CD jukeboxes, network storage appliances, and hierarchical storage management devices. Similarly, off-line archiving operations can use any system bulk storage device as an archive medium.

Migration from I/A Series Historian to FoxHistory

Migration tools allow users of the I/A Series Historian to configure and populate a FoxHistory database using their I/A Series Historian archives and real-time collected data. These tools capture the existing I/A Series Historian configuration, and build an equivalent FoxHistory configuration that can be installed on the I/A Series system. An additional tool is used to transfer data from existing Historian archives to the FoxHistory repository.

For applications using Informix ESQL/C, FoxHistory supports ESQL access using ODBC. FoxHistory supplies a toolset to translate Informix ESQL/C to ODBC ESQL. The porting tool converts ESQL/C code for the I/A Series Historian into ODBC for use by FoxHistory. To complete the migration, the application is recompiled and linked using the converted files.

FUNCTIONAL SPECIFICATIONS

Supported Network Platforms

Intel/Windows NT 4.x, SPARC/Solaris 2.4 or later.

Supported I/A Series Platforms

AP51A/B/C/D/E, AW51A/B/C/D/E, and AW70.

Installation

Any number of FoxHistory instances per I/A Series system.

Data Classes

Real-time points, events, manual entries, user-defined classes.

Minimum Number of Collection Points

200

Maximum Number of Collection Points

Unlimited (platform sizing required).

Maximum Number of Samples per Collection Point

Any number subject to the amount of available bulk storage capacity.

Retrieval Interfaces

ODBC and FoxAPI

Data Reduction Operations

Average, maximum, minimum, standard deviation, sum, current value, peaks, and kurtosis.

Archive Media

Magnetic tape, worm drive or any bulk storage device which is part of the server file system.

Supported FoxHistory DCS Data Servers

- Aspen CIM/21 - Requires @aGlance/IT Server Software from Aspen. This software is embedded in some versions.
- Aspen InfoPLUS.21 - Requires @aGlance/IT Server Software from Aspen. This software is embedded in some versions.
- Biles AIM Supervisor - Requires @aGlance/IT Server Software from Biles. This software is embedded in some versions.
- Eltag Bailey - Infi-90 Open - Requires @aGlance/IT Server Software from Eltag-Bailey. This software is embedded in some versions.
- Fisher Rosemount - PROVOX PDS for CHIP and DH - Requires Provox Process Data Server (PDS) from Fisher-Rosemount.
- Honeywell IAC Scan 3000 - Requires @aGlance/IT Server Software from Honeywell.
- Johnson-Yokogawa Centum CS - Requires @aGlance/IT Server Software from Johnson-Yokogawa.
- OSI Software - PI
- Intellution FIX/32 - Requires FIX 6.0 or later.

The Foxboro Company

33 Commercial Street
Foxboro, Massachusetts 02035-2099
United States of America
<http://www.foxboro.com>

Inside U.S. 1-508-543-8750 or 1-888-FOXBORO (1-888-369-2676)
Outside U.S. - Contact your local Foxboro Representative.

Fox, Foxboro, and I/A Series are registered trademarks of The Foxboro Company.

@aGlance is a registered trademark of Intuitive Technology Corporation.

INFORMIX is a trademark of INFORMIX Software, Inc.

MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Excel, SQL-Server, and Access are trademarks of Microsoft Corporation.

Oracle is a trademark of Oracle Corporation.

Sun and Solaris are trademarks of Sun Microsystems, Inc.

Copyright 1997-1999 by The Foxboro Company
All rights reserved