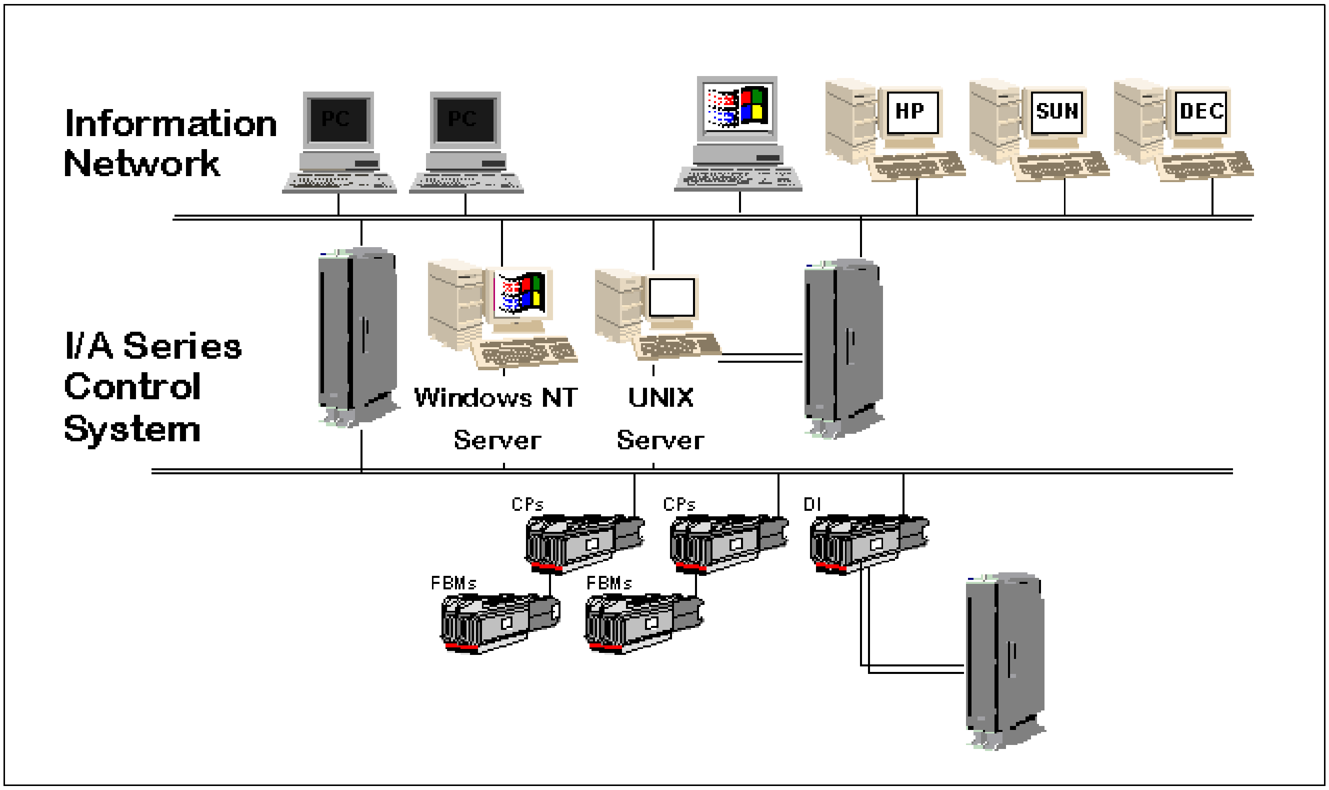
I/A Series Information Suite
AIM*API and Networked AIM*API Integration Software



*Foxboro AIM*API and Networked AIM*API integration software provides a consistent interface for application development and integration with the wealth of real-time and historical information supplied by I/A Series systems.*

OVERVIEW

The Foxboro I/A Series has always been more than just a capable control system. It is the foundation of a real-time plant information system that supports timely and effective business decisions. Thanks to unique Invensys enabling technology, you can link business systems with the control environment for better production planning, inventory control, and human resource allocation.

AIM*API and Networked AIM*API, integral parts of the I/A Series Information Suite, are powerful, reliable tools for integrating information across the enterprise. Whether the integration is within a single shop or across continents, AIM*API dramatically reduces implementation time and total project cost. The field-proven AIM*API provides a common functionality on and off platform, and gives you the freedom to choose between migrating and integrating applications between various generic computing platforms and the I/A Series system. AIM*API enables you to match computing platforms and applications to meet a variety of performance requirements while working within the constraints of plant standards, stringent requirements for maintainability, and limited personnel resources.

BENEFITS

There are several benefits provided by AIM*API and Networked AIM*API:

- Platform independently eliminates re-engineering applications.
- Common interface for applications allows use of multiple host computers including Sun, Windows NT 4.0, IBM.
- AIM*API based applications are easily moved between computing platforms utilizing AIM*API and Networked AIM*API.
- Networked AIM*API software based applications behave like local I/A Series applications.
- Protocol-specific details are transparent to Networked AIM*API based applications.

The key benefits of using AIM*API and Networked AIM*API are ease of program integration and protection against underlying platform, system, and low-level API modifications/changes. As the I/A Series and other networked computing platforms continue to evolve, improve, and upgrade, you do not need to change your applications. This applies not only to Foxboro and user-developed programs, but third-party applications using AIM*API as well.

AIM*API is the “on-platform” API for 50 Series AP/AWs and 70 Series AWs. Networked AIM*API includes 50 Series or 70 Series server support, networking support (using standard networking protocols such as TCP/IP and DECnet) and client software which is native to a particular plant computing platform. Clients currently supported by Networked AIM*API are:

- Sun (Solaris) TCP/IP
- Digital VAX (VMS) TCP/IP or DECnet
- Digital Alpha (OpenVMS) TCP/IP or DECnet
- HP RISC (HP-UX) TCP/IP
- IBM (RS 6000) TCP/IP.

AIM*API and Networked AIM*API Integration Software

AIM*API provides access to:

- Connection Services
- General Information
- Grouping I/A Series Data Objects
- I/A Series Data Object Access
- Change-Driven Extensions (CDX)
- Compound Summary Access (CSA)
- I/A Series Historian Access
- AIM*Historian Integration
- I/A Series Files (Read/Write)
- AIM*API Configuration Information
- Initialization File Access
- Error Logging Control.

Connection Services

Typically, a client application opens a connection to a server, sends and receives data and closes the connection. A client may simultaneously have open connections to multiple servers, governed by user configuration at the client. Additionally, a server may maintain simultaneous connections with multiple clients. These connections are controlled by authorization at the server. The following functions provide this type of service:

an_close_server	Close the connection to server
an_exit	Gracefully close the application
an_get_server	Get the current server
an_open_server	Open the connection to a server
an_set_server	Set the current server

General Information

This set of routines enables the client application to obtain general information related to the state of the current client and server:

an_get_client_info	Get client information
an_get_server_info	Get server information

Grouping I/A Series Data Objects

I/A Series data objects may be grouped together forming sets of data objects. The following functions are utilized for the maintenance of object data sets:

clsset	Close a set
getidx	Translate a data object name to an index
getmidx	Translate multiple data object names to indexes
getnam	Get the number of data objects in set, read change deltas, and data object names
getscn	Get the number of data objects in set, read scan rate, write scan rate, and write change deltas
get_set_name	Get a data set name
get_set_num	Get a data set number
gsinfo	Get the call type, queue#, number of data objects in set, value types, and data object indexes
gsnent	Get the number of data objects in set
put_set_name	Name a set
readnam	Translate the data object index to name
sbopen	Open a set without continuous update
scopen	Open a set with continuous update
sqopen	Open a set with continuous update and queuing of changes

I/A Series Data Object Access (Fast Access)

This access method is appropriate when I/A Series data objects are being accessed frequently. It is only useful for connectable data objects and consists of the following functions:

bread	Buffered read of a set of data objects
bwrite	Buffered write of a set of data objects
mreaidx	Read a value, status, and change count for multiple indexes
mreawidx	Read a write value, status, and write change count for multiple indexes
qread	Dequeue changes from a set's change queue
readccnt	Read the number of change-driven updates received for this data object
readmnam	Translate multiple data object indexes to names
readnch	Return total number of change-driven updates received by AIM*API since last restart
readnwr	Return the total number of change-driven updates received by AIM*API since last restart
readsta	Read a status
readval	Read a value
readwcnt	Read a number of writes to a data object
readwval	Read a write value
writccnt	Set read change count for a data object
writnch	Set read change count
writnwr	Set write change count
writwcnt	Set write change count for a data object
wrtval	Write data object value to shared memory

I/A Series Data Object Access

This method is appropriate when I/A Series data objects are accessed infrequently. Both connectable and nonconnectable data objects of all value types may be accessed. The following functions support this method:

sread	Read of a string object
swrite	Write of a string object
uread	Read of non-string objects
uwrite	Write of non-string objects.

Change Driven Extensions (CDX)

The CDX mechanism is extended into the client application to enable change-driven reads and writes with I/A Series data objects. In a networked situation, this mechanism significantly reduces traffic levels. The I/A Series data objects in the CDX are a subset of the I/A Series data objects in the AIM*API data object sets in the AP/AW. You specify an access type, read delta, read scan rate, and write delta for each I/A Series data object. The following functions support these services:

an_add_objects	Add data objects to CDX
an_dq_changes	Read the unsolicited updated values sent by the server. The client application can either wait indefinitely or time out.
an_get_current	Get current values for data objects in CDX
an_get_object_info	Get data object information for CDX
an_mod_objects	Modify data objects in CDX
an_poll_changes	Poll for changes
an_readccnt	Read a read change count
an_readnch	Read the read change count total
an_readnwr	Read the write change count total
an_readwcnt	Read a write change count
an_rem_all	Remove all data objects from CDX
an_rem_objects	Remove data objects from CDX
an_start_changes	Enable automatic change updates
an_stop_changes	Disable automatic change updates
an_writccnt	Set a read change count
an_write_objects	Set data objects in CDX
an_writnch	Set read change count total
an_writnwr	Set write change count total
an_writwcnt	Set a write change count
mwrtval	Write multiple data object values to the server.

Compound Summary Access (CSA)

CSA functions provide the user with read access to I/A Series object information. The following function calls support this capability:

an_all_sta	Get the name of all letterbugs, their host letterbugs, and their stations by types
an_all_cmp	Get the names of all compounds in specified station
an_all_blk	Get the names of all blocks in specified compound
an_blk_info	Get parameter information for a specified block type
an_obj_def	Get parameter information for a specified compound or block parameter
an_obj_descr	Get the block description for specified object

I/A Series Historian Access

The Historian access function enables the user to read and write Manual Data Entry (MDE) values in the Historian without knowing how the information is actually stored. The following function calls support this capability:

an_col_grp_names	Get name and description of all collection groups of specified type
an_col_grp_member s	Get name and associated parameters of all members of specified group
an_red_grp_info	Get reduction operations and associated parameters for specified reduction group
an_mde_members	Get name and associated parameters of all members of specified MDE group
an_put_mde_data	Add, modify, or delete an MDE entry
an_hist_values	Get Historical data values (except messages)
an_get_msgs	Get Historical messages
an_num_hists	Get the number of Historians in the I/A Series system
an_num_hist_grps	Get the number of collection groups in the specified Historian
an_num_hist_pts	Get the number of collection points in the specified Historian
an_hist_names	Get the domain name of all Historians in the I/A Series system
an_hist_lbug	Get the letterbugs of 50 Series stations containing the specified Historians
an_hist_pts	Get the name of all points being sampled by the specified Historian
an_pt_allhists	Get the domain name of all Historians associated with the specified point
an_pt_onhists	Get the domain name of the first Historian and the first “on” Historian associated with the specified points
an_pt_grps	Get the groups that a point is in for a specified historian
an_set_client_id	Set the client ID for client and server use in log file
an_rread	Read object’s range

I/A Series File Access

In addition to I/A Series data object access, functions are provided to support direct I/A Series file access.There is no need to run ftp or mount network files. The following functions provide read/write access to all or part of an I/A Series file:

iarfil	Read a file from an I/A Series AP/AW
iawfil	Write a file to an I/A Series AP/AW
pfread	Read part of a file from an I/A Series AP/AW
pfwrit	Write part of a file in an I/A Series AP/AW

*AIM*Historian Integration*

AIM*API contains a suite of functions that provide fast, reliable access to process data and event information collected and maintained by the AIM*Historian (and legacy I/A Series FoxHistory instances). AIM*API enables you to retrieve sample data, reduced historical data, and event information, remotely configure AIM*Historian instances, and write data for real-time objects (RTPs) that are configured for manual data entry (MDE). The functions are:

fh_Cfg	Create and configure AIM*Historian instances
fh_FdbGetValue	Get sample data for a specified RTP
fh_FdbMsgQuery	Retrieve an event message from the AIM*Historian message database
fh_FdbPutMsg	Insert an event message into the AIM*Historian message database
fh_FdbPutValue	Insert a value into the AIM*Historian database for an RTP configured for MDE
fh_FdbRedArray	Retrieve values from an array and perform specified reductions on the sample data
fh_FdbReduction	Retrieve values from an RTP database and perform specified reductions on the sample data for specified time span
fh_GetAimHistVer	Get the version of the AIM*Historian software currently on a station, the names of instances on the station, and the software version in use when each instance was configured
fh_GroupControl	Set the states of the RTPs in a control group, turning collection on or off, specifying slow or fast collection frequency, and marking status as good or bad
fh_MsgControl	Turn collection on or off for event messages in a control group
fh_MsgName	Get message name and group name for a specified message number
fh_MsgNr	Retrieve message number for a specified message name and message group
fh_RTPControl	Set the states of a specific RTP, turning collection on or off, specifying slow or fast collection frequency, and marking status as good or bad
fh_RTPDef	Retrieve the parameters of an RTP configuration
fh_RTPIndex	Get the RTP indices for specific RTP tags
fh_RTPName	Get the RTP names for specified RTP indices

*AIM*API Configuration Information*

Configuration information related to AIM*API sizing parameters is available via the following functions:

get_maxch	Get the maximum number of changes in the sqopen() change queue
get_maxds	Get the maximum number of data sets
get_maxgw	Get the maximum number of gateways
get_maxien	Get the maximum number of entries in the list
get_maxlst	Get the maximum number of AIM*API lists
get_maxobj	Get the maximum number of data objects
get_maxqo	Get the maximum number of sqopen() sets.

Initialization File Access

By default, an AIM*API based client application automatically initiates an an_read_init function with the initialization file name of an_init.cfg. A function is provided to enable the client application to specify a different file name:

an_read_init	Read initialization file

Error Logging Control

AIM*API supports error and trace messages. Keywords for message logging are defined in the initialization file to support PrintErr and TraceLevel. Using this set of functions, a AIM*API client application can enable or disable logging at will:

an_noprint_err	Disable printing of AIM*API error messages
an_print_err	Enable printing of AIM*API error messages

Utilities and Services

AIM*API includes the following menu-driven utilities and services for testing and troubleshooting AIM*API implementations:

an_ping	Check access to an AIM*AT server
an_what	Print version information for the AIM*API software on a specific server
foxdis	Monitor the read and write change load on the gateway, the current value, and status of objects for tuning the read and write change deltas
apispy	View data and check operating status of AIM*API instance
apitst	Test AIM*API functions
ia2rem	Copy a file from an I/A Series AP
rem2ia	Copy a file to an I/A Series AP
setclose	Close an AIM*API set
sxopen	Open an AIM*API set

Supported I/A Series Server Platforms for AIM*API

- 50 Series AP/AW
- 70 Series AW

Supported I/A Series Server Platforms for Networked AIM*API

- 50 Series AP/AW - TCP/IP or DECnet (with optional DECnet connectivity software)
- 70 Series AW - TCP/IP

Supported Client Platforms and Networks

- Sun - TCP/IP
  – Solaris (Version 2.2 or greater)
- Digital Vax - TCP/IP and DECnet
  – VMS (Version 4.7 or greater)
- Digital Alpha - TCP/IP and DECnet
  – OpenVMS
- HP-UX RISC based systems - TCP/IP
- IBM RS 6000 - TCP/IP

50 Series Server Requirements

- V3.x, V4.x or V6.x I/A Series software
- Optional information network port installed such as Ethernet, token-ring, ATM or fast Ethernet
- DECnet Connectivity Software Option (only required if client platform is using the DECnet protocol)

I/A Series AP/AW50 Sizing Information

- Up to 30 simultaneous client computing platforms connected to an I/A Series server.
- The maximum numbers of servers that a client computer can connect with simultaneously is determined by the client’s computing and networking capability.
- The maximum size of an I/A Series server data object database is 32,000 data objects. The default is 5,000 data objects.
- CPU loading on a 50 Series platform when the server is receiving 500 data object changes per second from a client computing platform is as follows:
  – AP/AW51A: 10% or less
  – AP/AW51B: 5% or less
  – AP/AW51C: 2.5% or less
  – AP/AW51D: 1% or less
- Up to 150 Ethernet packets per second at 512 bytes per packet has no significant impact on the general performance of the I/A Series server.

70 Series Server Requirements

- V6.x I/A Series software
- Optional information network port installed

I/A Series AW70 Sizing Information

- Up to 30 simultaneous client computing platforms connected to an I/A Series server.
- The maximum numbers of servers that a client computer can connect with simultaneously is determined by the client’s computing and networking capability.
- The maximum size of an I/A Series server data object database is 32,000 data objects. The default is 5,000 data objects.

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.

MB 021 Printed in U.S.A. 0999

An Invensys company