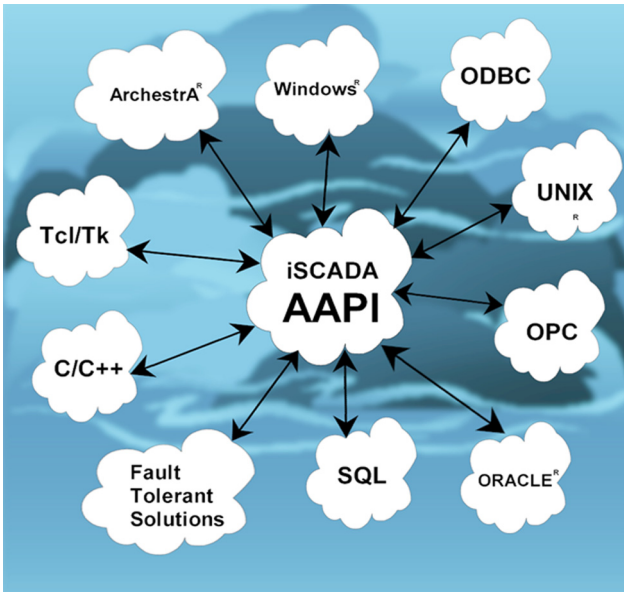
I/A Series® Intelligent SCADA Master Station Advanced Application Programming Interface (AAPI)



INTRODUCTION

The Advanced Application Programming Interface (AAPI) provides a user development environment to integrate distributed or local applications with the I/A Series Intelligent SCADA (iSCADA) Master Station (Refer PSS 21S-2M1 B3).

The AAPI provides access to iSCADA data and functionality and provides a straightforward, powerful platform for building redundant and fault-tolerant applications. Using the AAPI, users and developers are free to focus on solving application domain problems. The interface allows low-cost application reuse, system migration, and the extension of system life.

When combined with Invensys ArchestrA™, the defacto industry standard for systems integration, the AAPI OPC DA Server allows end users and system integrators to provide Windows®.NET™ based applications with real-time data access from iSCADA systems.

KEY FEATURES

- Intuitive SQL based data access and functionality
- Interface bindings for Tcl/Tk, C/C++, and ODBC
- Synchronous and asynchronous query execution
- Change-driven notification of data updates
- Field point control with execution feedback
- Support for both UNIX® and Windows® platforms
- Fault-tolerant application development infrastructure
- Query batching for efficient reading/writing
- Real-time access of iSCADA Master Station data by ArchestrA™ enabled applications

INTUITIVE SQL BASED INTERFACE

Three AAPI interface bindings provide simple programming language wrappers for enhanced Structured Query Language (SQL). Except where asynchronous data publication is required, standard SQL grammar is used for iSCADA Master Station data access and functionality.

Scripting Application Development

The AAPI Tool Command Language (Tcl/Tk) interface binding provides a scripting interface for testing, prototyping, and small to medium application development. The AAPI Tcl/Tk package is fully integrated with standard Tcl/Tk interpreters to provide application developers with a complete application development environment. Tcl/Tk script, distributed with the AAPI, provides interactive SQL session access and display of iSCADA data and functionality.

C/C++ Application Development

For advanced application development the AAPI includes a C/C++ interface binding. The AAPI C/C++ interface supports full integration with existing event dispatchers provided by standard programming environments such as the Windows® event loop used for Windows® GUI applications.

invensys®
FOXBORO®

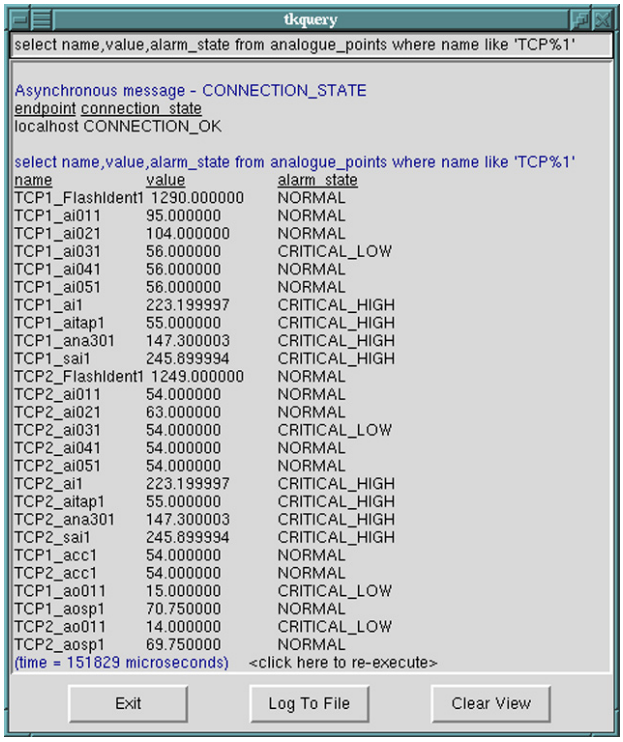


Figure 1. Tcl/Tk interactive SQL Session

ODBC Interface

For data exchange with existing Microsoft Office® applications, the AAPI provides an Open Database Connectivity (ODBC) driver. Thus providing Microsoft Excel® and Microsoft Access® applications, with direct read and write access to the iSCADA system.

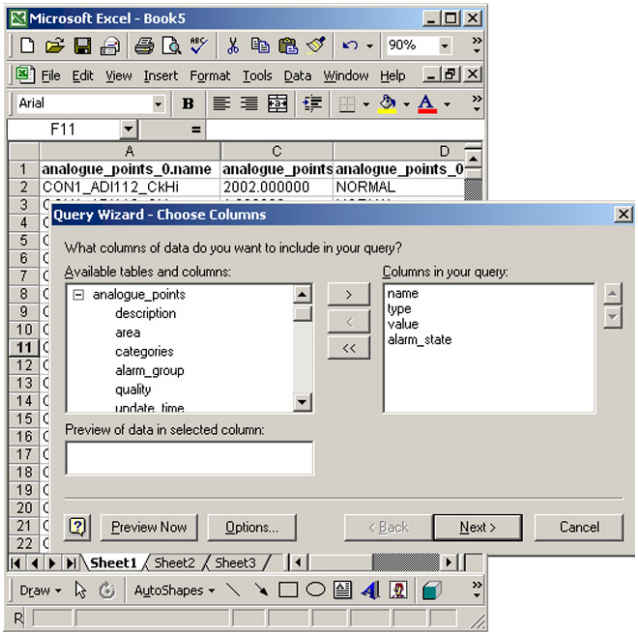


Figure 2. ODBC Connectivity with Microsoft Excel®

Synchronous and Asynchronous Queries

Queries to the AAPI can be executed synchronously or asynchronously.

Synchronous queries block the calling application and return a result set or error condition after the query is completed. Synchronous queries are easily extended to asynchronous queries by including a callback function or procedure when executing the query.

Asynchronous queries can be used to improve programming efficiency by allowing flexibility in waiting for and processing large result sets and improve data throughput rates using batched queries.

Asynchronous Message Notification

The AAPI extends the standard SQL syntax to provide a publish/subscribe mechanism for asynchronous notification of iSCADA point changes. Applications can subscribe to specific columns in the iSCADA points table. If the point attribute is updated in the iSCADA system, the AAPI publishes the new value to the application using a supplied callback function or procedure.

Asynchronous message notification is also available for application tables. Refer to “Support for Fault-Tolerant Applications” for application table support.

Asynchronous Point Change Notifications

The AAPI publish/subscribe logic has been designed to minimize CPU usage and iSCADA system resources. Figure 3 is an example of CPU usage of the AAPI components for various publications data transfer rates. The AAPI provides a level of decoupling between applications and the iSCADA real-time system whereby additional applications, monitoring the same points, do not increase the load on real-time iSCADA system process.

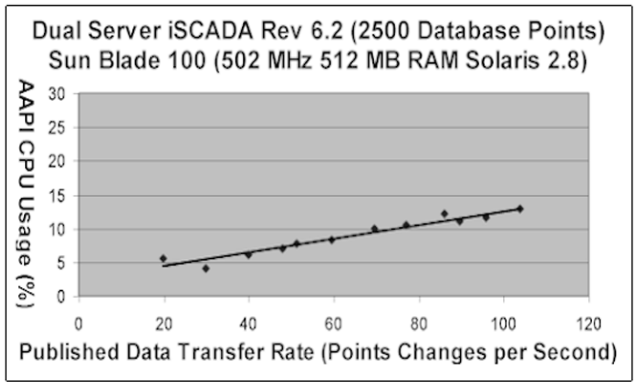


Figure 3. Published Data CPU Loading

Access Control

Access to the AAPI is controlled with username and password authentication.

REAL-TIME OBJECT ACCESS

General

The AAPI maps a subset of the iSCADA object schema into an object-relational table structure.

The base points table contains the current value, quality information, point description and classification by category, area, and alarm group, and scanning device name.

The analogue points table, in addition to the base points table, contains the alarm limits, engineering units and limits, control type and controllable limits, and periodic calculation point type and source point.

While the digital points table, in addition to the base points table, contains the current and normal state, valid point range, alarm and event processing mode, periodic calculations information, and control type.

Historical data

The AAPI provides read access to the iSCADA history database. Historical value and quality data can be retrieved based in local or UTC time ranges. Historical data is returned in an efficient change based format.

Field Point Control

The AAPI provides programmatic access to iSCADA controllable points. Using SQL functions, an application can reserve and execute controls on field points. Typically, an application firstly reserves the control point. Once the control reserve is established, the application then sets the control point using the control execute function. The AAPI supports trip close, momentary raise/lower, scaled raise/lower, and set point controls.

Systems List Access

Applications can generate system events within the iSCADA event sub-system.

MULTI-PLATFORM SUPPORT

The AAPI client libraries are available for UNIX® and Windows® operating systems (Refer AAPI release notes for currently supported platforms).

The AAPI provides platform independence for Tcl/Tk applications. Applications written in Tcl/Tk run on the supported platforms without changes to the AAPI interface commands in the source scripts.

Applications written using the C/C++ interface require only minor modifications for the supported platforms.

FAULT-TOLERANT INFRASTRUCTURE

The AAPI extends the iSCADA fault-tolerant platform to the application development environment. AAPI clients can reside, in redundant configuration, on the SCADA hosts or on remote application hosts. The AAPI middleware manages fail-over due to host, network, or client failure.

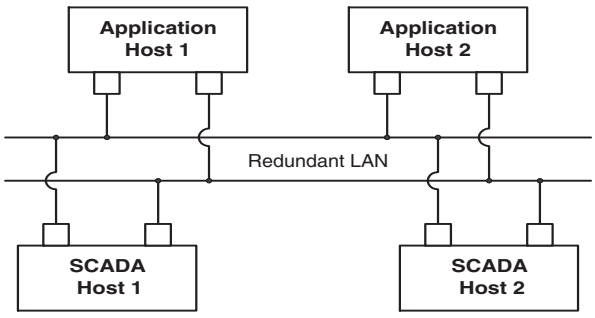


Figure 4. Example Redundant Configuration

Support for Fault-tolerant Applications

Two key features, application groups and application state tables, support the development of fault tolerant applications.

When an application starts, it can elect to join an application group where applications exist as redundant instances of the same application. The first application to join the group is elected the group master. If the master application fails then the AAPI middleware selects a new master. Each remaining members of the group can subscribe to be notified if it is elected as master.

Application state tables provide a mechanism for applications within the same group to share data. Each application within a group can create or drop tables and select, delete, insert, or update rows within the table.

BATCH PROCESSING

Accessing real-time objects one-at-a-time incurs significant performance costs due to the overhead of multiple round trips between processes. The AAPI reduces the inter-process and network overhead by batch processing transfers. The batch processing facility is activated by wrapping a series of queries with the AAPI batch start and batch run statements.

For example, on a 502Mhz Sun Blade 100, a client application can expect batched data retrieval rates of 250 real-time point objects per seconds or update rates of 90 real-time point object per second. Using the subscription mechanism (real-time point data is cached) client applications can retrieve up to 3000 real-time values per second (actual performance is affected by system configuration).

ORACLE® EXTENSION

For direct integration with Oracle® databases an optional feature available with the AAPI is the Oracle® data provider. This data provider supports direct query access through the AAPI to tables in an Oracle® applications or enterprise database. The Oracle® data providers allows AAPI applications to configure a mapping between Oracle® tables and virtual AAPI tables, select, update, insert and delete rows in Oracle® tables, and execute table joins across the iSCADA and Oracle® databases.

In conjunction with the high-availability features of Oracle®, the AAPI can be used to develop fault-tolerant solutions spanning across iSCADA and Oracle® data sources/stores.

ORDERING INFORMATION

Part Number	Description
SY-1102092	**IA SCADA AAPI ODBC SQL 5 NT Clients** - I/A Series Intelligent SCADA AAPI Windows ODBC driver. (Licensed per connection)
SY-1102093	**IA SCADA AAPI C/C+** - I/A Series Intelligent SCADA AAPI C/C++ Windows or UNIX client libraries. (Licensed per station)
SY-1102094	**IA SCADA AAPI TCL - 5 Windows or UNIX Clients** - I/A Series Intelligent SCADA AAPI TCL Windows or UNIX client libraries. (Licensed per connection)
SY-1102095	**IA SCADA AAPI Unix Server** - I/A Series Intelligent AAPI server, read/write access to real-time data, read/write access to historical data, and read/write access to events. (Licensed per station)
SY-1102096	**IA SCADA AAPI ORACLE Data Provider** - I/A Series Intelligent SCADA AAPI read/write access to ORACLE. (Licensed per ORACLE database)
SY-1102097	**IA SCADA AAPI OPC DA Server** - I/A Series Intelligent SCADA AAPI Windows OPC DA Server. (Licensed per Windows server host station)

This product uses software developed by Spread Concepts LLC for use in the Spread toolkit.
For more information about Spread, see http://www.spread.org.

33 Commercial Street
Foxboro, MA 02035-2099
United States of America
www.foxboro.com
Inside U.S.: 1-866-746-6477
Outside U.S.: 1-508-549-2424
or contact your local Foxboro representative.
Facsimile: 1-508-549-4999

Foxboro Australia
10 Brandl Street
Eight Mile Plains, Brisbane
Queensland 4113, Australia
Telephone: +61 7-3340-2111
Fax: +61 7-3340-2100

MB 21A Printed in U.S.A. 1203