A GATEWAY TO INTELLUTION'S FIX SYSTEM

J.M. Nogiec†, FNAL, Batavia, IL 60510, USA

Abstract

A gateway system to Intellution's FIX control system has been developed at the Fermilab's Magnet Test Facility (MTF). The gateway enables exchange of data between a FIX-based automation system and other systems deployed in the same network environment. The gateway has been deployed as part of a sophisticated system used at MTF to test superconducting accelerator magnets. The same architecture can be directly applicable to other projects where gateway solutions are looked-for.

1 INTRODUCTION

The Fermilab's Magnet Test Facility (MTF) performs tests of superconducting accelerator magnets and components with help of two systems: the Distributed Monitoring and Control System (DMCS) [1][2][3] and the Intellution FIX control system [4]. The former system monitors test subjects and is responsible for quench detection and protection. It also controls test conditions, provides data visualization, and archives monitoring and quench data for on-line and off-line analysis. The latter system controls the MTF's refrigerator and is responsible for providing cryogenic conditions required for testing. In order to conduct tests and correctly interpret test results, data from both systems have to be simultaneously available for analysis and presentation. To satisfy this requirement a gateway system between both the DMCS and the FIX system has been developed. The gateway allows for bi-directional access to cryogenic data stored by FIX from the DMCS scanning system.

2 ARCHITECTURE

Intellution's FIX industrial automation software is a popular Windows-based HMI/SCADA solution which provides process monitoring and control, data collection, graphic display, historical trending, alarming, data archiving, and security for an unlimited number of I/O points.

The gateway to FIX software allows client computers to access FIX data over the network. With its help, a client application can read and write to a FIX database that resides on the remote computer running the FIX automation software (Figure 1). Client applications interact with the FIX system via the gateway API. The API functions convert client database access operations into request messages that are sent to the gateway server over the network. The server, which is located in one of the FIX control nodes, receives the request, performs the actual database access operation, and sends a response back to the client. The FIX gateway system operates on top of the socket-based inter-process communication layer using the TCP protocol.

Since the gateway API is available on multiple platforms it allows for integration of systems in heterogeneous environments. Client applications running under VxWorks and UNIX (Solaris, SunOS, or Linux) can interact on the data level with the FIX system running under the Windows operating system.

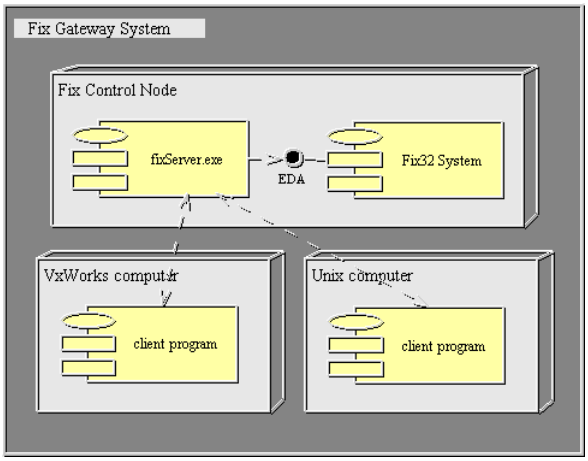

Figure 1: Components of the gateway.

3 GATEWAY SERVICE

At the heart of the gateway system lays a multi-threaded server with one worker thread per connection. The server awaits connections on a specified port, where the port can be specified by either an environment variable or a registry key. Once the connection is established, the server receives the request, performs the actual database access operation (using the FIX Easy Database Access package), and sends a response back to the client. The gateway server continues to serve requests until the connection is terminated. The server automatically terminates connections that are inactive for one hour. To facilitate portability and avoid problems with platform dependent encoding, exchanged messages have text formats.

A maximum number of concurrently active client connections can be specified to avoid degradation in service speed and provide an auxiliary mechanism for protection against misbehaving clients. The server can execute either in the stealth or verbose modes. In the verbose mode, diagnostic messages are printed on the screen.

†nogiec@fnal.gov

4 GATEWAY ACCESS

The gateway is accessible programmatically via its API implemented as a library and a couple of non-GUI programs intended for scripting.

4.1 Protocol

The gateway access protocol is a variation of a simple request-reply protocol (see Fig. 2). It has a form of a synchronous RPC, where user's code is blocked until a reply is received or time-out occurs. The relatively small size of messages and moderate speed requirements allows for exclusive use of ASCI message formats, which makes the protocol highly portable.

Following current software development trends, the next revision of the gateway will replace the proprietary protocol by an XML-based communication protocol such as increasingly more popular SOAP. As a result, the system will retain its platform independent formats of messages but will migrate toward standardized communication protocols.

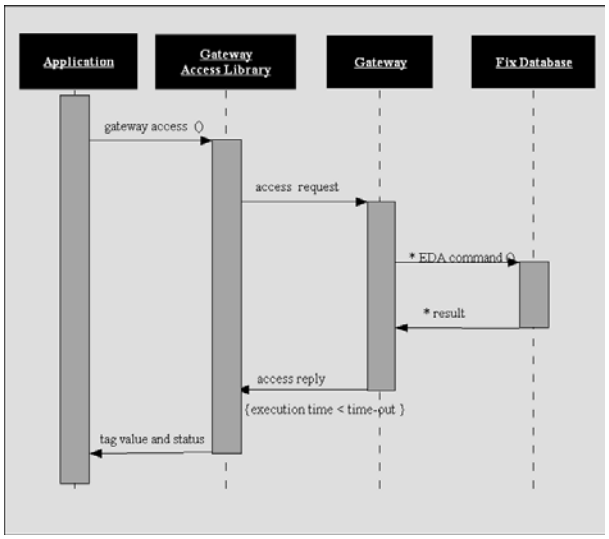

Figure 2: Gateway Access Sequence.

4.2 Gateway API

The system offers simple, yet powerful API. Values are accessed in FIX databases using node, tag, and field names. The FIX gateway library allows applications to open a connection to a given node, access given tags, and close the connection. Tag names can be up to 30 alphanumeric characters long and must start with a letter.

The function *fixa_open()* is used to establish a connection to a given node, and port. It returns a non-negative descriptor on success, which is used later when calling other library functions to reference the connection. The *fixa_get()* function retrieves a value of a given tag, whereas *fixa_put()* updates a given tag. The *fixa_close()* function is invoked to terminate the connection with the gateway server. All the functions return completion codes that indicate if the operation has been successful and specify the type of exception.

The FIX API is available for the Solaris and VxWorks operating systems and could be easily ported to other Unix-like platforms, such as Linux.

4.3 Scripting

Two client programs *fixGetData* and *fixPutData* have been developed to demonstrate the proper usage of the gateway API, to facilitate writing scripts, and to assist in testing of the FIX gateway system. Both programs are available for the Unix and VxWorks operating systems.

The *fixGetData* program can be used to get the value of a given tag. It accepts a node, port, and a tag as its parameters. The program has also an option to periodically obtain and print the tag value. The *fixPutData* program can be used to set a new value in a given tag. It accepts the same node, port, and tag parameters as *fixGetData* plus the new value of the tag.

5 INTEGRATION WITH DMCS

Distributed Monitoring and Control System (DMCS) is a multi-platform system consisting of a set of distributed computers connected via a local area network. It combines functionality of both a general-purpose monitoring and control system and a measurement tool.

The I/O subsystem of DMCS has a layered architecture (see Fig. 3). At the highest layer is an elaborate scanning system that examines process variables. Process variables correspond to the physical quantities measured, monitored, controlled, or altered by the system and are evaluated periodically, or in response to some events. Some process variables (*input process variables*) reflect directly measured or monitored values, whereas other contain values used to regulate or control the system *(output process variables*). Still other process variables called *software process variables* may contain system control parameters, internal system variables, statistics, long-term trends, etc. All process variables have calculations assigned to them. Each input and output variable is additionally associated with a data acquisition device. Devices are accessed uniformly by device-independent I/O operations implemented in the device independent I/O layer. This layer translates I/O operations into driver or handler calls. The system supports various devices accessible via the VME and GPIB instrumentation buses and transparently integrates PLC-based I/O systems.

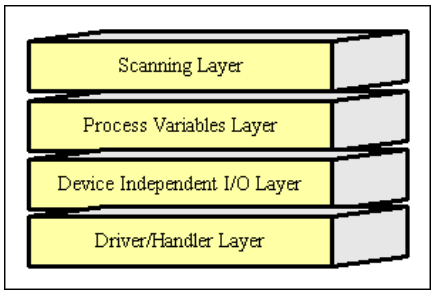

Figure 3: Layered I/O Architecture of DMCS.

In order to transparently access data via the gateway, the device independent I/O layer of DMCS had to be extended to include support for the new device called FIX. In that layer, device independent I/O operations (open, close, setup, input, output, control) are translated into calls to the gateway's API. After adding the support for the FIX device, process variables can specify that device to be used to acquire data in the same way as standard analog input or output devices. To simplify the mapping between DMCS's process variables and FIX's tags, identical names were used for both when referring to the same measured or controlled physical quantity.

6 DIAGNOSTICS

The gateway has been developed with consideration for its long-term support and maintenance, where troubleshooting and performance monitoring play important role.

6.1 Troubleshooting

Various problems can be encountered when accessing remote FIX databases. Typical problems include wrong port or node name of the gateway, FIX system or gateway not running, or attempts to perform an illegal operation (e.g., attempt to update non-settable tags or set the value outside the allowed range, unknown or misspelled tag)

In order to aid in troubleshooting, the gateway server uses the standard Windows application event log to record errors and the program start and termination events. The server also stores information about its most recent execution, including all major gateway parameters, in the Windows registry.

6.2 Performance monitoring

In a distributed environment with multiple independent clients, monitoring load and inner workings of the server could come in handy when debugging faulty client programs or improving performance. The *PERFMON.EXE* application that ships with Windows can be used to monitor the number of threads that are running in the gateway server process. Each new connection should increase this number by one.

7 SUMMARY

A flexible gateway system to the FIX control system has been developed at the Fermilab's MTF. Although the system has been developed as the gateway for Intellution's FIX system, it could be easily adapted to provide remote access to other systems. The gateway system allows for bi-directional exchange of data in heterogeneous computing environments and scales well with load. The multi-threaded server guarantees the suitable performance in a concurrent distributed environment. Due to its socket-based protocol with ASCII messages, the gateway allows for easy access from different computing platforms. A standard API and basic support for scripting allow for developing both production data acquisition applications and ad-hoc test solutions. Information logging allows for troubleshooting and access monitoring.

8 REFERENCES

[1] J.M.Nogiec et al., "An Open Distributed Monitoring and Control System", Proceedings of the International Conference on Computing in High Energy Physics CHEP'97, Berlin, April 1997

[2] J.M.Nogiec, E.Desavouret, D.Orris, J.Pachnik, S.Sharonov, J.Sim, J.C.Tompkins, and K.Trombly-Freytag, "A Distributed Monitoring and Control System", PAC'97, Vancouver, 1997

[3] http://sdsg.fnal.gov/dmcsweb

[4] http://www.intellution.com/