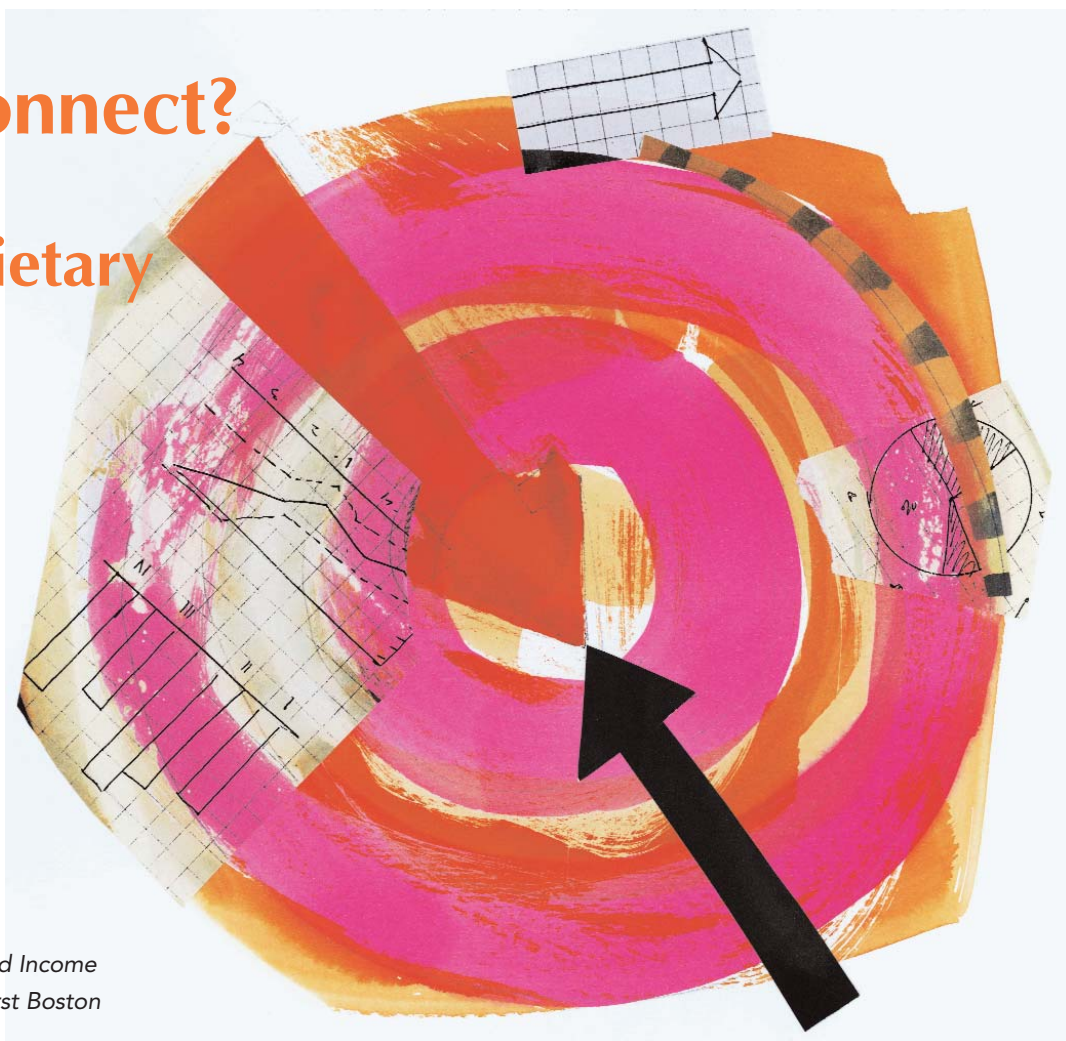


# How to connect?

## Using proprietary or standards based APIs



By Dan Tsou, Vice President, Fixed Income  
FIX Connectivity, Credit Suisse First Boston



**Electronic trading has transformed financial markets. Equity, futures, options, fixed income and FX markets have all developed platforms and methods of automation for electronic trading. The global equity markets, for example, have seen enormous automation and a concurrent increase in trading volumes and a decrease in trade costs. Underlying much of the transformation of financial markets is the increasing adoption of industry standards such as FIX.**

Manual trading has been replaced with automated electronic processes that integrate peripheral support systems and allow for sophisticated applications utilizing real time data and advanced analytics. By connecting the systems of counterparties and markets, electronic trading has opened new pools of liquidity and ushered in an era of quantification that has leapfrogged traditional relationship trading.

At the core of electronic trading technology is the Application Programming Interface (API) that enables systems to automate this electronic interaction. The role of the API is to provide a system engineer with well defined

methods needed to support the trade process such as how to connect, how to receive prices, how to send an order, and how to receive a trade. As electronic trading matures to meet the changing business demands of different markets, the number of options and complexity of API solutions have also increased.

The focus of this article is a discussion about the significance of using proprietary APIs versus using APIs based on the industry standard Financial Information eXchange (FIX) as the bridge between counterparties and within markets. Information was gathered for the article by surveying market participants responsible for connectivity at

a number of large firms. Comments from survey participants are included in the survey response section.

### Hypothesis

- In markets where an industry standard protocol exists, proprietary APIs decline in significance.
- The costs of maintaining proprietary APIs on multiple platforms (C++, Java, COM, etc) can be much more costly to support than FIX messaging.
- Industry standards tend to replace proprietary APIs if the processes being automated are well defined and mature. Otherwise, the protocol will not be able to keep up with the evolving demands.
- Proprietary APIs are sometimes viewed as a means to gather and retain clients by tying their systems with the proprietary technology. However, most firms with the sophistication to integrate one proprietary API are just as capable of integrating other APIs so the retention effects are not as meaningful as believed.
- Proprietary APIs will always be viable to firms that require connectivity with a small list of service providers.

### What is a proprietary API

The category of proprietary APIs is so broad because there are virtually no limits to what they can do and how they do it. Anytime there is a need to pass information from one system to another an API can be developed to facilitate the interaction. Some examples of proprietary APIs popular in the financial markets are Reuters' API for price distribution, TradeWeb's API for all trade related communication, OMGEO's API for trade processing, and Bloomberg's API (although it is important to note that Tradeweb and Bloomberg also have industry standard based APIs for FIX connectivity).

The distinguishing qualities of proprietary APIs are their unparalleled combination of performance, features and ease of use. Proprietary APIs are built to meet the needs of a specific system so the design can be highly optimized. Generally speaking, as an API's features increase, its performance and ease of use will decrease in order to support the needs of the different functions.

Proprietary APIs are created by service providers to allow their customers to integrate the provider's processes into their business. In order to meet all their customer needs, major service providers may have to offer several different

versions of the API to support the major system development environments such as Java, C++, COM, UNIX, Win32, and Linux.

### What is an industry standard API

Industry standard APIs serve the same functional purpose as proprietary APIs but they are designed to meet the needs of a much broader audience. Often industry standard APIs serve an entire industry (or a portion of the industry) rather than a specific provider and facilitate interacting with multiple service providers.

Industry standards are often the catalyst for market transformation because they establish a common language for providers and customers to interact. For example, HTTP (Hyper Text Transfer Protocol) is an industry standard that defines how information is exchanged.

Although industry standards can drive exponential growth for industries, standards are challenging to create due to the consensus required across an entire industry. Because of the compromises required to reach consensus and the degree of flexibility required, industry standards are often comparatively slower in performance, narrower in features, and harder to use.

For service providers, industry standards significantly reduce the complexity of their API, since they no longer have to provide support for variants such as Java, C++, COM, etc. More importantly, standards based APIs often lead to an exponential growth in client adoption which can mean more revenue and reduced costs from automation efficiencies.

For customers, standards based APIs reduce the complexity of integrating different APIs from multiple providers and broadens the number of service providers they can use.

### Connecting to many or few

In markets, like fixed income, where an industry standard's adoption has not reached critical mass the decision on which type of API to use can be traced to the fundamental business and technology profiles of the customers.

For a larger or more sophisticated firm that needs automated connections to multiple partners, the

predominant choice is the industry standard given its high degree of reuse. It's not practical for firms to develop different software for each of their business partners as would be required with proprietary APIs.

### Firms utilizing advanced automation:

Integration with multiple counterparties  
 Multiple asset classes  
 Multiple trading process automations  
 Dedicated development teams and environments

For customers with plans to connect to only a few business partners, it can be advantageous to use a proprietary APIs. Proprietary APIs often provide more features and tools to simplify the integration effort.

### Firms utilizing early stage automation:

Small number of counterparties  
 Small numbers of processes to be automated  
 Small development staff  
 Limited technology infrastructure

### Excerpts from survey responses

- *Will FIX reduce the importance of proprietary API's? If so, what are the conditions under which that could occur?*

I do not necessarily think that FIX will reduce the importance of proprietary API's, but hopefully will make integrations between the vast majorities of sell and buy-side firms more straight forward and uniform. There will likely still be a need for proprietary API's for certain types of specialized customers (i.e. hedge funds) where performance of market data and execution may necessitate non-FIX implementations.

- *What are the advantages of using proprietary APIs?*

There are a number of advantages.

- 1) Specialization - The proprietary APIs are tailored to the system environment in which they live. These have grown as the organization has grown and are highly specialized to the organization's flavor of the business process being supported.
- 2) Speed - Standard protocols add layers of packaging and unpackaging that aren't required in specialized API's. This can be especially important in volume sensitive applications.
- 3) Support - It takes time for managers and developers to

understand a standard and what it can do and more importantly what it can't do. These are self evident to the team who built an API."

- *Have proprietary API's been useful in retaining clients?*

Yes. Anytime we can step up and meet a clients needs we are improving that relationship. If that need requires us to build an API then we will attempt to meet that requirement. However, we would prefer that these interfaces be handled using standardized interfaces.

Not to a large extent

- *Which types of clients are inclined to choose a proprietary API?*

Hedge funds.

Mid- to small-sized investment managers prefer the simplicity and network-security aspects of our proprietary API over FIX."

2 types in my space. 1. Smaller asset managers with outsourced or very small IT departments. In this case we need to meet them because they don't have the technical wherewithal to implement the standard. (This could change as vendors adopt the standard in their systems) 2. Fixed income firms that have established equity trading operations. These firms are usually already FIX enabled and have internal expertise.

- *What are the major limitations of FIX?*

FIX has a format that doesn't fall within current accepted programming practices. 2) A network protocol that either introduces a 3rd party engine vendor into the relationship or requires expensive programming expertise. 3) A messaging protocol that needs a layer of queuing and routing between the engine and the application. 4) A mystique has evolved that to be secure and fast FIX must run under a private network provider.

### Performance

From a technology and support point of view FIX as a protocol has very few limits it supports pre-trade, trade and post-trade messaging formats. With the release of FIX 4.4 support was added for standing

settlement instruction. The limits of FIX are those from outside sources such as; critical mass, reference data and initial capital cost investment.

In fixed income it's missing critical mass. It's getting there but obviously slowly. As more firms get the time and money to implement a FIX based solution the more useful it becomes, like a fax machine is more useful if more people have fax machines."

- *How do costs compare between supporting FIX messaging and proprietary APIs?*

That's hard to say as in fixed income we have only just begun. The intuitive answer is that supporting one protocol connected to multiple clients is cheaper than managing multiple APIs to the same clients. As the standards allow ever more clients to be connected in a shorter timeframe, ways of managing that growth must be addressed. Therefore you should expect to spend more money in absolute terms as more clients connect and this infrastructure needs to be fault tolerant, global and monitored continuously. Proportionally, however I would still expect the support costs to be smaller."

Assuming constant network costs and similar end-user application development costs between implementing FIX and our proprietary API, the one substantial difference is the cost of buying or implementing the FIX engine - a cost that is not a factor in implementing the proprietary API.

FIX requires larger 3rd party costs while proprietary APIs require more in-house development resources. Where FIX saves money is in reducing the number of client-specific installations where the client has their own API.

If a firm implements FIX as a strategy all future developments will need to have FIX functionality embedded into its code, architecture, and application. Instead of having multiple protocols, message formats to map to developers will need to support, monitor and maintain a single source, which is certainly more cost effective than having multiple source codes.

### Points that matter

Customers want an easy to use and cost effective solution. The solution providers want to meet their

customer needs and make it difficult for customers to leave. The industry standards committees want to increase utilization of the standard. For electronic trading, FIX and proprietary APIs are two key solutions that will shape the evolution. Below is a summary of important points when evaluating FIX and proprietary APIs:

- *Proprietary APIs can offer the best combination of performance, features, and ease of use*

For example, since there is no limitation on how simple the API is, the proprietary API can be designed to carry very small messages with predefined structure, thus maximizing its performance. The proprietary API is also free to create its own communication protocol designed for high speed performance.

In contrast, industry standards such as FIX have to carry the necessary overhead data to provide enough flexibility to allow many systems to use it.

For customers only requiring a small number of connections in the foreseeable future, proprietary APIs may be the best option.

- *Proprietary APIs are impractical for customers that requires automation with many partners*

If a customer needs to connect with many trading partners, it is significantly more efficient and easier to use an industry standard such as FIX.

If proprietary APIs are used instead, it would require having the resources to develop and maintain different software for each of the APIs.

For customers with near term plans to connect to more than a few partners, using an industry standard based API is the most efficient approach. **FIX**

*This article has been edited by Cate Long of Multiple-Markets, Co-chair, Americas Education and Marketing Subcommittee, Global Fixed Income Committee.*

### Any thoughts on this or other articles?

Please send any comments, referring to this article as Vol 1 Issue 4 GL1, direct to Edward at [edward@fixglobal.com](mailto:edward@fixglobal.com)