



A New Data Feed Format, Open to the World

Introducing OpenFeed - A New Data Feed Format, Open to the World

by: Eero Pikat

President

Barchart

eero.pikat@barchart.com

One of the reasons that industry organizations like FISD exist, is that most organizations in a given industry doing the same kinds of things, tend to have the same kinds of problems. Sometimes, having an industry group is useful in solving issues in a collective manner.

Delivering data from point A to point Z has been a universal chore for exchanges, data vendors, and consumers alike. Over the years, various standards, formats, and protocols have emerged, with never really a clear winner. FIX and FAST seemed to be the most adopted at one point, but now, yet again, there is a scramble by many parties to move away from this protocol. With various business interests tugging and pulling the exchanges and data vendors, there is a good chance that soon we will, yet again, be faced with a plethora of highly specialized, in some cases half-baked and badly implemented standards that data vendors, trading firms, and many other organizations will need to kludge, wedge, and somehow cram into their systems.

Messaging protocols and the corresponding delivery infrastructure are essential to our industry. Instead of each firm re-inventing the wheel, it would seem that now would be the time to create a new, unified data feed format, open to the world, owned by everyone, without licensing and legal encumbrance.

Timing and maturation are important factors. At nascency, standards bodies tend to fail since they can't account for the marketplace fast enough. For example, consider the standards for web browsers. When the first versions of HTML were proposed, it became clear that the market demanded a lot more than what a few engineers could dream of, and products like Netscape, and then Internet Explorer, added special extensions and features, which then they copied from each other, resulting in a mess lasting from 1995 until very recently. Developers and designers still need to account for the various quirks of different browsers and different versions. But finally, the world has stabilized. It took years of practical, "real world" experience to realize that there should be a separation of data and presentation layers, that forcing proper syntax is a good idea, and that the <BLINK> tag is not a good idea.

We believe that the market data world has reached a sufficient point of maturity. There are now clearly defined users of market data, each with their needs. But we now have, what we think, is a fairly good idea of these users' needs, and can now address them. Furthermore, while still very much a top of mind consideration, the wild swings of balance between processing power versus bandwidth seem to have subsided, and for the most part, we seem to have enough of both, and when we don't, the marketplace seems to deliver quickly enough.



This is the mission of the OpenFeed Project. We have developed, what we believe, is a full featured, standardized data feed format, which takes into consideration the various needs of the different factions of data feed producers and consumers: latency, processing time, extensibility, ease of development, universe of third-party tools, etc.

The beauty of the Internet world is that you can constantly build on the efforts of others. Why re-invent when someone has already solved a specific problem. OpenFeed is built on many other open standards, and we try to yield to an IETF or IEEE standard whenever possible.

OpenFeed is based on Google's Protocol Buffers. Unencumbered by licenses and patent claims, the only rule that OpenFeed must follow is that it continues to the same open licensing agreement as Protocol Buffers, meaning that no one will ever be able to charge royalties for this. Furthermore, there are many providers of developer support for Protocol Buffers, and support can be found for pretty much every development environment--both the industry standards such as C, Java, .net, Python, Perl, etc., and the esoteric (though becoming more mainstream) like Scala and R. The underpinning protocol already handles compression, the ability to have null fields, and various data types. Furthermore, the serialization and deserialization are also already provided. A plethora of free third-party tools automatically generate source code in various languages that implement most of the protocol specification. In short, the rest of the world has already done most of the work for us.

The OpenFeed specification takes both business and technical needs into consideration. Processing time per message is kept to a minimum, and systems can quickly determine whether a packet is needed, and what they want to do with it. The feed works both on multicast as well as P2P network topologies. Open source code is provided to assist and accelerate development and implementation. In short, this is the way that we believe data feeds ought to be.

We always welcome new members and contributors to our groups. We would like to establish an industry panel, where we would have representatives from as many different use-case firms as possible. We don't purport to know every use case, and the more global our input, the better and more robust the protocol will become.