

MetaFluent[®] for RMDS

Turning time back into money

The Challenge of RMDS Access

Struggling with the idiosyncrasies of market data access slows down the development of new applications.

Trading firms today develop an increasing number of applications that need to access real time market data and publish their own real-time data to other applications. The hub for this information is the firm's market data platform, which in most institutions is the Reuters Market Data System (RMDS). RMDS provides proprietary APIs that have a significant learning curve and often require an education not only in the API but also in the underlying market data infrastructure. In addition, the symbology and data layouts for information sources such as Reuters can be unfamiliar to developers whose sole focus is not market data. And developers using high-productivity languages like Java and .Net often find limitations in the API functionality available in native Java and C# implementations. These issues slow the time to market for new applications as well as time spent on maintenance over the application lifecycle. And in a trading firm, time is money.



The Problem with Wrappers

Creating "wrapper" APIs and infrastructure imposes large, non-strategic costs on a trading firm.

Faced with these challenges, as well as a desire to mitigate vendor risk, many institutions create "wrapper" APIs that hide the underlying access mechanisms behind an interface of the trading firm's own making. But it is difficult to design an abstraction layer that is simple and efficient yet still exposes all the capabilities of the underlying APIs. Some firms move the abstraction logic to a server, but this challenges them to write efficient infrastructure. Either way, the firm finds itself in the data plumbing business, supporting a "product" with a long lifecycle and a demanding internal customer base. This is an ongoing expense that few trading firms consider strategic.

The Simplicity of MetaFluent

MetaFluent simplifies market data access. Developers use Java and .NET implementations of the standard JMS API. MetaFluent handles market data complexities behind the scenes.

MetaFluent[®] for RMDS provides high-performance pub/sub access to a market data hub through pure Java and .NET implementations of the standard Java Message Service (JMS) interface. JMS is a well-known standard that presents a simple API for messaging and data representation. JMS is familiar to most developers and can be easily learned from the abundant public documentation and examples.

MetaFluent does not require third-party middleware such as a JMS server. Behind the lightweight API is an optimized server that leverages multi-core platforms and scales well across machines through clustering that is transparent to applications.

The point of MetaFluent is simplicity. Applications subscribe to one or more JMS Topics and receive images, updates, and status indications in an intuitive manner. Data are represented as ready-to-use JMS MapMessages. Even Reuters chains are easy to access via single Topic names, with no chain-walking required. MetaFluent for RMDS provides full subscription management, DACS-based content- and topic-based entitlements, and end-to-end data integrity monitoring. Applications can use the same API to publish to the MetaFluent server, which handles distribution to RMDS. All of this functionality comes without baggage such as proprietary APIs, auxiliary topics, add-on libraries, data dictionaries, or application-level protocols. Everything just works. Enabling you to turn more time into money.



Key features

Light-weight, standards-based API

- Pure Java, pure .NET and C client libraries
- Implements the industry standard JMS interface. Many developers know it already.
- Zero client-side config
- Data delivered ready to use
 - MapMessages
 - No parsing
 - No data dictionaries

Both publish and subscribe

- Consumes from ADS/P2PS
- Accepts publications from client apps and converts them to upstream solution, e.g. inserts (RMDS "un-managed publishing").
- Marketfeed now, OMM in development

Dynamic Data-Aware

- Full image/update semantics
- New subscribers receive initial images
- Data-stream exception handling
 - Stale data indications
 - Automatic image refresh on recovery
- Content-based entitlements (see DACS)

Server clustering

- Connection load balancing
- Resilience via redundant load-balancer
- Dynamically add servers to a cluster (no down-time)
- Clustering is transparent to client applications

Feed/service discovery

- An feed/service table is available to client applications
- Lists the services available and shows their states

Standards-based administration

- JMX-enabled for integration with standard monitoring packages
- Secure browser-based interface

Open DACS integration

- Lock- and subject-based entitlements
- Usage tracking
- Dynamic DACS profile updates
- Lock updates

Built-in Reuters data intelligence

- Integrated support for chains using MetaFluent "multi-stream" topics
 - Applications subscribe to single topics
 - No chain walking
 - Rapid chain retrieval

High performance

- Efficient client library
- Optimized binary wire protocol
- Intelligently threaded server
- Optional field filtering
 - Manage data transformation (field names, representations)
 - Further reduce client-side CPU and memory utilization as well as network utilization

Excel Add-In

- Subscribe/publish macros
- Office Ribbon extensions
- Windows XP/Windows 7, Excel 2002, 2007 and 2010
- Small footprint, MetaFluent C API

Vendor-neutral

- NYSE Technologies integration
 - Simultaneous access to RMDS and Wombat data
 - Order book support via the multi-stream topic mechanism
 - DART entitlements

Near-term roadmap

- Transparent symbology mapping
 - Integrate with an external mapping repository (e.g., Bloomberg to RICs)
 - No redundant data caches - Different users access same data instance using different symbology

To arrange demo access or to learn more information, please contact sales@metafluent.com.

