MilleniumIT Market Data Adapter

MilleniumIT Connectivity Overview

Below is a brief overview of the MilleniumIT market data connectivity.

Market Feeds

There are three market feeds with identical architecture – LSE, JSE and NSX.

Services

A market feed includes four services.

Level 1 service provides instrument definition, instrument status, order book (market by price), statistics, trades and market announcement data via FIX /FAST protocol.

Level 2 service provides instrument definition, instrument status, order book (market by order), opening/closing statistics and trades data via ITCH protocol.

News service provides news data via FIX/FAST protocol.

Indices service provides real time index data via FIX/FAST protocol.

Market Data Groups

A market feed is load balanced by market data groups.

A market data group includes one or more instruments. Each instrument is assigned to just one market data group.

Currently there is one market data group for each market feed (LSE, JSE, NSX).

Sites

A market data group includes two sites disseminating identical data – primary (A) and secondary (B).

Channels

Each site includes up to three channels.

Real Time UDP channel disseminates market data via a multicast group.

Replay TCP channel is used to recover from a small message loss in the Real Time channel. The channel allows a recipient to request retransmission of a range of messages already published in the Real Time channel.

Optional Recovery TCP channel is used to recover from a large message loss in the Real Time channel or a late join to the Real Time channel. The Recovery channel allows a recipient to request a snapshot of the current market data.

Accounts

A recipient must have a valid account to login to the Replay and Recovery channels.

An account password must be changed periodically.

Day Limits

JSE defines a set of per day limits for operations on the Replay and Recovery channels. If any of the limits is exceeded within a day a recipient's account is locked.

	Level 1	Level 2	News	Indices
Replay login limit	1000	1000	5	TBD
Replay request limit	1000	1000	20	TBD
Recovery login limit	500	500		
Recovery request limit	500	500		

Protocol Versions

JSE utilizes the following versions of protocols:

FIX 5.0 SP2, FAST 1.1, ITCH, IPv4

MilleniumIT Market Data Adapter Overview

MilleniumIT Market Data Adapter is a module in FIXAntenna that communicates with the LSE, JSE, NSX market feeds.

Two sets of interfaces are exposed to a client.

One set abstracts a client from MilleniumIT Market Data Adapter implementation.

- Mit::Application
- Mit::ApplicationListener
- Mit::Service
- Mit::ServiceListener
- Mit::Instrument
- Mit::InstrumentListener

The other set abstracts MilleniumIT Market Data Adapter from connectivity implementation. A client can supply its own connectivity implementation. If not supplied by a client the default implementation based on Boost.Asio is created and used by MilleniumIT Market Data Adapter.

- Mit::ConnectionManager
- Mit::UdpConnection
- Mit::UdpConnectionListener
- Mit::TcpConnection

Configuration

MilleniumIT Market Data Adapter can be configured in two ways - via the XML configuration file or the programming interfaces. A mixed approach when some services and/or instruments are configured with the XML configuration file and some services and/or instruments are added programmatically is supported as well.

XML configuration file configures application options, services, service options and instruments.

Application

Application stores services and provides access to them by index and name. Application also stores resources shared across services such as a connection manager.

It is possible to have several applications created at the same time. A client can choose to have either a single application for all the services of all the market feeds or have a separate application for each service. All the IO operations of all the application's services are processed by the same connection manager. The typical solution is to have a dedicated application for each market feed (LSE, JSE, NSX).

Service

Service processes common tasks for FAST and ITCH services such as Real Time channel management, Replay channel management, Recovery channel management, statistics management, status management, password management, etc.

FAST Service

FAST service is a specialized service that processes FAST specific tasks such as FAST message decoding, FAST message replay, FAST message recovery, password change.

FAST service implements the standard sequence processing policy based on field ApplSeqNum (1181).

FAST Instrument Service

FAST instrument service is a specialized FAST service that implements per-instrument sequence processing policy based on the field RptSeq (83).

FAST instrument service accepts only non-instrument messages by default. A client should specify the instruments of interest.

Per-instrument sequence processing is only possible for Level 1 service.

FAST Instrument

FAST instrument processes instrument specific tasks such as message recovery, statistics management, etc.

FAST Parser

FAST parser decodes FAST encoded FIX messages.

ITCH Service

ITCH service is a specialized service that processes ITCH specific tasks such as ITCH unit parsing, ITCH message replay, ITCH message recovery.

ITCH service is capable of converting ITCH messages to equivalent FIX messages to unify message processing on the client site. Whether ITCH messages are converted to FIX or supplied to the client directly is specified by the ItchToFix service option.

ITCH Message Processor

ITCH message processor parses ITCH messages and converts them to the corresponding FIX messages.

ITCH message processor is capable of gathering several ITCH messages into one FIX message (as repeating group entries) to optimize message processing on the client side. Whether several ITCH messages are gathered into one FIX message is specified by the ItchNoDelay service option.

In the unlikely event an ITCH message cannot be converted to the FIX message (e.g. undocumented field value) the entire binary ITCH message is wrapped into a FIX message and passed to a client.

ITCH Parser

ITCH parser defines structures for ITCH units and messages and general purpose functions to construct, parse, validate and dump ITCH units and messages.

Sequence Processor

Sequence processor processes message losses, message reorders and message duplications that may occur in the Real Time channel.

Sequence processor automatically arbitrates between feeds A and B of the Real Time channel to reduce probability of message loss.

Out of order messages are either filtered out or cached in a queue until the gap is filled from the same feed or the second feed or the Replay channel.

In the case a message gap is detected the sequence processor starts a replay procedure. Sequence processor automatically arbitrates between the primary and secondary sites of the Replay channel.

Sequence processor processes late joins and detects the inter-day and intra-day resets.

In the case of a late join or a reset detected the sequence processor executes a recovery procedure. Sequence processor automatically arbitrates between the primary and secondary sites of the Recovery channel.

Service Statistics

Two types of service statistics are maintained - connection statistics and date statistics.

Connection statistics is reset after every service connect operation.

Date statistics is persisted and is reset every new date. The date statistics is checked against the JSE day limits to prevent an account from locking on the JSE side.

Service Status

Status of each service channel is tracked and available via the service connection statistics.

Real Time channel is considered inactive if there are no messages received within the certain time interval.

Replay channel is considered inactive if the last replay procedure failed. If there are no gaps detected within the certain time interval the Replay channel is probed to check the password status and update the Replay channel status.

Recovery channel is considered inactive if the last recovery procedure failed.

Password Management

It is possible to specify separate username / password for each service.

Passwords are managed by MilleniumIT Market Data Adapter automatically. A password is changed either on an explicit client request or when the server reports about a password expiration within a specified time. The new password is either specified by the client explicitly or generated by MilleniumIT Market Data Adapter automatically.

Asio Connection Manager

Asio connection manager serves as a factory for UDP and TCP connections.

Asio connection manager manages a pool of threads and a completion port to process the results of asynchronous operations of an arbitrary (unlimited) number of concurrent UDP and TCP connections.

The size of the thread pool is configurable.

Asio UDP Connection

Asio UDP connection asynchronously receives UDP packets from a multicast group.

Asio TCP Connection

Asio TCP connection synchronously sends and receives data to and from a TCP endpoint.

Logging

All the data sent and received can be optionally logged at three levels for later analysis and troubleshooting.

logClientMessages – log client messages to text log files

logMessages – log UDP/TCP connection messages to text log files

logData – log UDP/TCP connection data to binary log files

Error, warning and trace events are logged to the FIXAntenna log subsystem.

JSE Emulator Overview

JSE emulator is a static library that emulates the JSE Real Time channels by sending previously recorded binary log files to UDP sockets.

JSE emulator emulates the JSE Replay and Recover channels by listening to TCP sockets and responding to a recipient's replay and recovery requests.

Concurrent emulation of feeds A and B of several services is supported.