THOMSON REUTERS MATCHING API

THOMSON REUTERS MATCHING DATA FEED DIRECT USER GUIDE

Document Version 1.5.5
Date of issue: 3 July 2015

Contents

About this document

Intended readership

This document is aimed at technical staff support or developing with Matching Data Feed Direct (MDFD) in MAPI.

In this document

This document defines technical requirements for the RFA based Matching Data Feed Direct (MDFD)

What's new?

In version 1.5.5

- Added new FIDs providing Iceberg related configuration & limits to OMM 147 and 148
- Updated section 2.1 to include minimum version numbers for UPA
- Added section 2.1.1 clarifying that UPA & RFA both provide the same market data

In version 1.5

- Added the new configuration related OMMs (147, 148, 149 & 175) to chapters 4 and appendices
- Updated section 3.2.2 to reflect that conflation period length now varies by instrument rather than service
- Improved and expanded descriptions of existing concepts and functionality throughout the document based on commonly raised questions during development

In version 1.0.23

- Removed references to Market Price model as that is no longer available in MAPI (PCN 6645)
- Updated section 3.3.2.2 clarifying how an empty book is represented and that map entries in general are not provided in any particular order
- Removed historical section 3.3.6 concerning upgrades from AutoQuote (AQ)

In version 1.0.21

- Updated section 2.1 to remove references to RFA version 6.4.2.1.L1 and 6.4.0.E4 as these versions are no longer supported. Added reference to the RFA compatibility matrix located on the Thomson Reuters customer zone.
- Updated section 3.3.2.3 added reference to the new FX Spot Instrument attributes document located on the Thomson Reuters customer zone.
- Updates Appendix D to reference the new FX Spot Instrument attributes document located on the Thomson Reuters customer zone.
- Removed Appendix G HTG Depth of Book instruments are defined in the referenced FX Spot Instrument attributes document located on the Thomson Reuters customer zone.

In version 1.0.20

- Updated References section with links to documents on the Thomson Reuters customer zone.
- Updated section 3.3.2.3 to reflect additional Spot Instruments that have been enabled for Depth of Book in production.
- Updated Appendix G for instruments that are enabled for DOB on the Host to Go test environment.

In version 1.0.18/ 1.0.19

- Updated section 3.3.2.3 to reflect additional Spot Instruments that have been enabled for Depth of Book in production.
- Updated Appendix G for instruments that are enabled for DOB on the Host to Go test environment.

In version 1.0.17

- Updated section 2.1 regarding qualified RFA versions
- Updated sections 3.3.2.2 and 3.3.2.3 to reflect additional Spot Instruments that have been enabled for Depth of Book.
- Updated Appendix G for instruments that are enabled for DOB on the Host to Go test environment.

Feedback

If you have any comments on this document please contact the Thomson Reuters TPG Documentation team.

.

Chapter 1 Introduction

The Matching API is comprised of two main parts, one dedicated for price discovery and another dedicated for order management. The Reuters Foundation API (RFA) is used for price discovery and an industry standard FIX interface has been introduced for order management and post trade processing.

The Matching RFA implementation provides Spot and Forward prices from Matching platform. It provides a low latency market data feed utilizing the RFA Open Message Model (OMM) format to provide fast access to Matching data.

1.1 Scope

This document defines technical requirements for the RFA based Matching Data Feed Direct (MDFD) product. Although the MDFD product is implemented using the RFA the requirements and message definitions specified herein only apply to the Thomson Reuters Matching service.

1.2 References

Reference #	Document Name	Link
1	Matching FIX Interface User Guide	https://customers.reuters.com/a/support/paz/pazDocs.aspx?dId=481996
2	Thomson Reuters Matching Market Data	https://customers.reuters.com/a/support/paz/pazDocs.aspx?dId=489406
3	FX Spot Matching Instrument Attributes	https://customers.reuters.com/a/support/paz/pazDocs.aspx?dId=548547
4	RFA Developers Guide for Java (released with the RFA package)	https://customers.reuters.com/developer/
5	RFA Developers Guide for C++ (released with the RFA package)	https://customers.reuters.com/developer/
6	RFA Configuration Guide (released with the RFA package)	https://customers.reuters.com/developer/
7	RDM Usage Guide (released with the RFA package)	https://customers.reuters.com/developer/
8	RFA Compatibility Matrix	https://customers.reuters.com/developer/api_compatibility.aspx

1.3 Terms and Acronyms

Term/Acronym	Definition
FID	Field identifier
Instrument	Term used to denote an FX Currency pair
MDFD Server	Market Data Feed Direct server
OMM	Open Message Model
RDM	Reuters Domain Model
RFA	Robust Foundation API (formerly Reuters Foundation API)
RFA Token	Authentication token retrieved from FIX Gateway
RSSL	Reuters Source Sink Library
Symbol	RFA identifier for a given instrument

Chapter 2 RFA Package supported versions, environments and sample tool

The MDFD provides price discovery support via the Reuters Foundation API (RFA). The RFA library (available in both Java and C++ variants) is used to connect to the MDFD server, handling the encoding and decoding the market data messages. Thomson Reuters does not release the network protocol used by the library to communicate with the MDFD server and clients must use the library in order to access the market data.

2.1 Qualified RFA/UPA Versions for Usage with Matching

At the time of writing the *minimum versions* supported by MAPI are:

- RFA 7.2.1.L1 (C++ & Java)
- UPA 7.0.1.L1 (Java)
- UPA 7.4.0.L1 (C++)

Clients may wish to use later versions in order to provide compatibility with newer operating systems, architectures, compilers or virtual machines etc. The market data provided by the different versions though does not vary and therefore it providing at least the minimum is used clients will be able to access all of the market data published in MAPI.

For additional details on operating system and programming language compatibility please refer to the compatibility matrix on the Thomson Reuters Customer Zone website and the README file inside a specific library version.

However clients are not permitted to use versions below the minimum under any circumstances.

2.1.1 RFA vs. UPA

The choice of which library to use will not affect the market data that is delivered.

The market data concepts, messages and fields described in this guide apply equally to RFA & UPA. Both use the same underlying protocol and connect to the same market data servers. The difference is the interface the library provides to the application developer.

UPA provides a lower level interface without the extra features RFA has by default (e.g. ability to load configuration from external sources, automatic management of connection heart beating and failover). A UPA developer therefore has more flexibility on how to implement those features but will not have the benefit of it already being included.

Developers should review the Developer Guides provided with each library to evaluate which one suits their design & requirements best.

2.1.2 Library Version Numbering

Each library version has a suffix of *.L<num>* or *.E<num>*.

The *E* signifies it is an emergency release which fixes a specific issue rather than providing new functionality or improvements. Emergency releases have not undergone a full regression test and therefore should only be used by a client if they are specifically affected by the problem the release addresses.

RFA & UPA are used by many products supporting many different connection types and message formats. It should not be automatically assumed that an emergency release is fixing a problem that would have an impact for MAPI clients.

The main (non-emergency) releases are those with a suffix of *.L<num>*.

2.1.3 Library Packages

The library is downloaded via the Thomson Reuters Customer Zone website (https://customers.reuters.com/developer/).

Versions are available for Java or C++ programming languages and Windows, Linux or Solaris operating systems.

The downloaded archive file contains numerous documents and supporting files as well as the runtime library itself:

File or Directory	Applicable to	Purpose
README	All languages, all platforms	Contains details of: • Version history/change log, • Hardware, Software and compiler requirements • Installation instructions
/Docs	All languages, all platforms	RDM Usage Guide RFA Developers Guide RFA Configuration Guide RFA Migration Guide
/Docs/refman/rfa	C++, all platforms	C++ API reference
/Docs/refman/rfajava	Java	Java API reference in Javadoc format
/etc/RDM	All languages, all platforms	The standard RDM dictionaries containing the field and enumerated values definitions. Note this does not include the custom definitions (non-RDM) discussed in section 4.
/Examples	All languages, all platforms	A series of examples covering RFA functionality. Note these are not MAPI specific. The MAPI sample client discussed in section 2.2.
/Include	C++, all platforms	The C++ include files
/Legacy	All languages, all platforms	Documentation and examples related to older connection types and messages formats. This is not relevant to MAPI
/Libs	All languages, all platforms	Contains the Java JAR file or C++ libraries needed to compile or run an application.
/MessageFiles	C++ on Linux or Solaris	Files containing the strings needed to write log messages
/Tools	Java	A set of simple tools for importing/exporting RFA configuration settings via the Java Preferences API.

File or Directory	Applicable to	Purpose
/ValueAdd	All languages, all platforms	Source code and documentation for a set of useful classes and utilities built on top of the public API

2.2 RFA Sample Tool and Source Code

The samples provided with the library (in the */Examples* directory described above) are not specific to MAPI. Some may use connection types or message formats that are not available in MAPI. Even examples based on RSSL connections and Open Message Models (OMM) format may still refer to service names, models or symbols that are not known in MAPI. As such the configuration and/or source code in these examples may need to be modified before they can successfully be used against a MAPI MDFD server.

Instead it is recommended that clients using RFA in MAPI for the first time start with the MAPI RFA sample client. This generates subscription requests for items which exist on a MAPI MDFD server and it also handles the requirement of obtaining token from the FIX gateway needed to successfully authenticate.

This sample is released to clients and included with all the other MAPI documentation and samples on the BizBuzz site (http://transactions.thomsonreuters.com). It is available in both Java and C++ variants.

The MAPI sample applications and accompanying documentation are built against version 7.2.1.L1. Clients that use newer versions of the RFA library should consult the *README* and *RFA Migration Guide* provided with the specific library version to ensure they are aware of any changes that might affect how it is built, configured or used.

Chapter 3 Functional Requirements for Matching specific RFA Client Applications

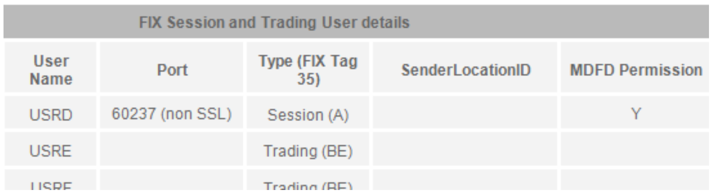
3.1 Authorization / Permission Management

The MDFD servers use a token for authentication rather than managing a separate set of usernames and passwords.

The token, hereafter referred to as the *RFA Token*, is provided when the client successfully logs into the FIX Gateway with an appropriate user. It is provided in the FIX tag *Text(58)* on the Logon (35=A) response from the gateway.

When viewing the credentials document for an environment, the RFA token should only be taken from those marked with MDFD Permission = Y. Using a token from any other user will result in the subscriptions requests being rejected.

Further details on how to login to the FIX Gateway can be found in the *FIX Interface Users Guide*.



FIX Session and Trading User details

User Name	Port	Type (FIX Tag 35)	SenderLocationID	MDFD Permission
USRD	60237 (non SSL)	Session (A)		Y
USRE		Trading (BE)		
USRE		Trading (BE)		

3.1.1 Login Request and Permissions

To logon to the MDFD server the client sends a subscription request using the standard *Login* OMM and provides the RFA token in the Name attribute (see section 4.7.1).

The permissions granted will be directly linked to the credit code (aka TCID) of the FIX session from which it came. The token will only allow a client to view instruments that the TCID is allowed to trade and subscription attempts to other instruments will be denied. Similarly it will prevent access to screened prices for all other TCIDs except the one to which the token is linked.

Clients who have multiple TCIDs will therefore need to make multiple RFA connections. Each one would use a different token obtained from the appropriate FIX session. This will ensure that they can access the relevant instruments and screened prices for each TCID.

3.1.2 Logout

Logging out is achieved by closing the *Login* OMM subscription.

3.1.3 RFA Token Expiry

RFA tokens are valid for the whole trading week from Sunday through to Friday.

Once a token has been obtained it remains valid for the remainder of that week irrespective of what happens to the FIX session from which it was retrieved. It is the client's choice whether they logout of the FIX session after receiving the token, or keep it connected and use it for submitting orders.

However clients that choose to no longer keep the FIX session open are still expected to use the FIX protocol properly and logout from the gateway gracefully. Abrupt disconnections and problems with the sequence numbers should only occur in abnormal circumstances (e.g. network problems, application crashes etc.) - they must not be part of the normal operation of the application.

A token can never be used in a different trading week to which it was obtained. Tokens always expire at the weekend regardless of how late in the week it was generated. A client that retrieves a token towards the end of the week, perhaps only hours or minutes before trading finishes would still be required to obtain a new token in order to access market data from Sunday onwards.

3.2 Market Data Concepts

The Market Data within MAPI provides the following areas information:

- Available prices (screened or unscreened)
- Last Traded prices
- Worst Prices

All of this is delivered using the same Market By Price OMM in the same event. There are *not* separate subscriptions for each of the 3 types of information. For details of the fields with the Market By Price payload see Appendix A.

Please be aware though that the data provided is only a summary - it is not a complete list of every order and every trade that occurs. For instance there are limits on how many different prices are shown (see *Depth of Book*). There is also a limit on the maximum quantity reported (see *Regular Amount*). Finally data is only calculated at set intervals rather than every time an order is placed/cancelled/matched (see *Conflation Period*).

3.2.1 Depth of Book

The available prices in MAPI are summarized so that for the top few prices on each side it will give the total quantity and the number of orders that the quantity is comprised of. The detail of the individual orders from which this is summarized is not reported.

For example:

BUY			SELL		
ORDER_PRC (FID 3427)	**ORDER_SIZE (FID 3429)**	**NO_ORD (FID 3430)**	**ORDER_PRC (FID 3427)**	**ORDER_SIZE (FID 3429)**	**NO_ORD (FID 3430)**
1.3027	7 mio	4	1.3029	3 mio	1
1.3026	4 mio	1	1.3030	15 mio	3
1.3025	25 mio	10	1.3032	1 mio	1

The top level (highest bid/ & lowest offer price) is referred to as *top of book* or depth 1. The next level is depth 2 and so on. Each instrument has a limit on the maximum number of depths that can be shown and once that has been reached any liquidity at lower levels will not be visible.

Each instrument also has a threshold specifying the maximum difference from top of book allowed. Prices that are further away than this will not be visible regardless of how few depths that would leave remaining.

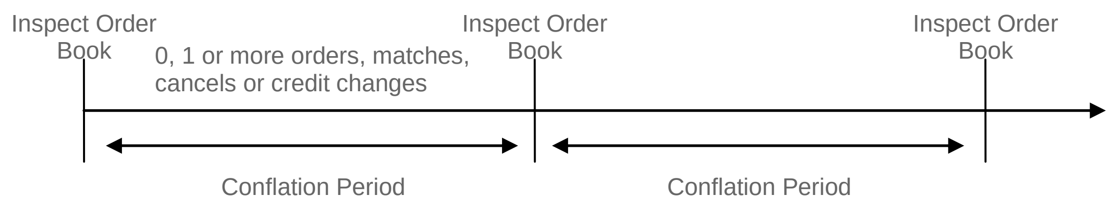
Therefore as the market moves it is entirely possible for the number of different prices reported to vary on one or both sides. At the extreme an empty book will be shown by having no prices on either side. This will occur at the beginning of the week before trading starts as well as at the end of the week when trading has finished. It is also possible that an empty book occurs any other time during the week dependant on market conditions.

The maximum number of depths is given *in BOOK_DEPTH (FID 6454)* and the threshold for price difference is given in *PRICETHOLD (FID 6455)*. These limits are set globally Thomson Reuters; the same limits apply to all participants and they are not user configurable.

The number of depths and threshold are also included in the Spot Instrument Attributes document on Customer Zone.

3.2.2 Conflation Period

Regardless of how much activity occurs on the instrument the MDFD servers only inspect the Matching order book at fixed intervals. This means the data that is published to clients is conflated over a period of time rather than being full-tick.



This means it is not possible to determine exactly what individual changes occurred between two consecutive updates for an instrument. Any number of orders could have been placed, matched or cancelled in between time. Additionally credit for any of the counterparties involved could have been consumed or replenished.

The conflation period is defined per instrument. This is set by Thomson Reuters and applies to all MAPI participants; it is not user configurable.

3.2.2.1 Conflation Rates

Previously the conflation rate was determined by the service. Only *TRFX_MDFD_Standard* was actually made available but other services were mentioned in documentation as a future possibility.

However these additional services will not be introduced. Instead all MAPI participants will use the same service and the conflation rate will depend on the specific instrument involved.

Service	Conflation Period (pre MAPI 1.5)	Conflation Period (MAPI 1.5)
TRFX_MDFD_Standard	250ms	Varies by instrument
TRFX_MDFD_Enhanced	To be determined & never released	Not available
TRFX_MDFD_Premium	To be determined & never released	Not available

Initially when MAPI 1.5 is released all instruments will still have a conflation period of 250ms.

Future changes to the rates used by specific instruments will be announced via a Product Change Notification (PCN) which is the same process used for any other instrument related changes.

3.2.3 Regular Amount

The *Regular Amount* puts an upper limit on the quantity (and therefore the number of orders) that will be published for a particular price.

If the quantity actually available exceeds the Regular Amount then market data will not show the full total but instead show only the Regular Amount instead.

A flag will be set though to differentiate between there being exactly the Regular Amount available and there being more than the Regular Amount. However it will not be possible to identify by how much it has been exceeded – whether it is only 1 mio more or 100 mio more the output will be the same.

The number of orders will also be limited in this case and it reports how many orders the Regular Amount is comprised of rather than the full number.

The Regular Amount is defined per instrument and is reported in *REG_AMOUNT (FID 6306)*. This is set by Thomson Reuters and applies to all participants; it is not user configurable.

The flag indicating whether the regular amount has been exceeded is *ORDSIZEIND (FID 6305)*.

The Regular Amount is also included in the Spot Instrument Attributes document on Customer Zone.

Scenario	Meaning
ORDER_SIZE (FID 3429) < REG_AMOUNT (FID 6306)	ORDER_SIZE is reporting the actual qty available.
ORDER_SIZE (FID 3429) = REG_AMOUNT (FID 6306) ORDSIZEIND (FID 6305) = 0	ORDER_SIZE is reporting the actual qty available. Co-incidentally it is also exactly the same number as the Regular Amount.
ORDER_SIZE (FID 3429) = REG_AMOUNT (FID 6306) ORDSIZEIND (FID 6305) = 1	The Regular Amount has been exceeded. ORDER_SIZE is reporting the Regular Amount, but the full quantity is at least 1 mio higher, maybe more.
ORDER_SIZE (FID 3429) > REG_AMOUNT (FID 6306)	This is not possible. ORDER_SIZE will never report a numerical qty higher than the Regular Amount.

3.2.4 Worst Price/Aggregate Price

The *Worst Price* (aka Aggregate Price) takes into account the various prices and quantities available and reports the price that would need to be submitted in order to match a pre-defined quantity called the *Standard Quantity*.

For example consider the scenario where there are 3 different bid prices available with various quantities available at each price:

- 10 mio @ 1.5289
- 7 mio @ 1.5288
- 19 mio @ 1.5287

If the Standard Quantity for this instrument was 20 mio then purchasing it would require matching multiple prices. The 20 mio would be comprised of:

- 10 mio @ 1.5289 (all 10 mio at this price)
- 7 mio @ 1.5288 (all 7 mio at this price)
- **3 mio** @ 1.5287 (only 3 of the 19 mio at the worst price)

The worst price in this case is 1.5287 as that would be needed to fill the final 3 mio of the original 20.

Likewise if the Standard Quantity was actually 15 mio instead then it would be:

- 10 mio @ 1.5289 (all 10 mio at this price)
- **5 mio** @ 1.5288 (5 of the 19 mio at the worst price)

The worst price in that case is 1.5288.

The worst price is only reported a *price* for each side. It is not a list providing a breakdown of the qty needed at each price point, nor is it an average value taking account of the different underlying values. It is simply the worst numerical value (that is the lowest bid or highest offer) that would need to be placed in the FIX *Price(44)* tag of a limit order for it to match the full *Standard Quantity*.

A price that was closer to the top of the book than the Worst Price would not match the full Standard Quantity (unless the market moved sufficiently in the meantime of course). Likewise a price that was further away from the top of the book would be unnecessary to match the Standard Quantity.

The worst price will only be calculated if all of the following are true:

- It is a *Screened* subscription
- The instrument has a *Standard Quantity* defined and it is more than the minimum order size
- There is at least the *Standard Quantity* available …
- … and that quantity is within the *Worst Price Threshold* from the top of the book

The *Standard Quantity* is defined per instrument and is reported in *STD_AMOUNT (FID 6302)*.

The *Worst Price Threshold* is defined per instrument as is reported in *WPT (FID -10056)* of the Reference Data Model.

These are set by Thomson Reuters and apply to all participants; they are not user configurable.

The *Worst Price* is calculated for each side separately and is reported in BID (FID 22) and ASK (FID 25).

The *Standard Quantity* and *Worst Price Threshold* are also included in the Spot Instrument Attributes document on Customer Zone.

3.2.5 Credit Screening

Available prices can be provided as either *Screened* or *Unscreened*.

Unscreened prices are not checked for credit and as such they could involve orders which cannot be matched by a particular counterparty. These may also be referred to as the market best prices.

In contrast screened prices ignore all orders where there is not enough remaining credit left to allow even a partial match. As a result these prices will either be equal to or worse than the unscreened prices. Whilst screened prices are not always going to as good as the unscreened prices, they are the best prices that the participant could actually match. These may also be referred to as the tradable prices.

A subscription request is made for either the screened or unscreened version as explained in section 4.1, but the choice of subscription type only affects the available prices. The last trade price is never screened and the worst price is only available in a screened subscription.

The table below summarises which items are screened and which are unscreened for a particular subscription type:

Item	**Unscreened Subscription**	**Screened Subscription**
Available Prices	Unscreened	Screened
Last Traded Price	Always unscreened	Always unscreened
Worst Price	Not available	Screened

Clients that wish to view both screened and unscreened prices will need to make two subscriptions – one for each type.

If a screened subscription shows no available prices then there are two possibilities:

- No liquidity at all (the market is empty)
- Liquidity exists but lack of credit means none of it is available to this participant

The difference between the two possibilities can be determined by comparing the screened with the unscreened data:

Unscreened	**Screened**	**Meaning**
No Liquidity	No Liquidity	Normal market conditions when there is no liquidity.
Liquidity	Liquidity	Normal market conditions when there is liquidity. Note unscreened and screened might not be identical, but there is some liquidity in both.
Liquidity	No Liquidity	Lack of sufficient credit. All the orders are being filtered out due to credit screening. Last trade data may still update though indicating other participants still have credit and are continuing to trade.
No Liquidity	Liquidity	This is an impossible scenario. Credit screening only removes orders from the calculations; it never adds or changes them. Therefore if there is no unscreened liquidity there cannot be any screened liquidity either.

3.2.6 Global Market Data Sequence Number

A global market data sequence number is provided within the Matching Data Feed Direct service. The market data sequence number can be used by receiving RFA applications to maintain temporal ordering of market data updates received across multiple RFA endpoints within the same or different geographic region. The market data sequence number is unique for the given currency pair and weekly trading session. The SEQ_NO (FID 3859) field is available for unscreened and credit screened market data updates on the Market By Price OMM. Since all market data updates within Matching are conflated at predefined intervals the sequence number does not monotonically increase across market data update intervals and will be reset at the start of each trading week.

3.2.6.1 Credit Screened Market Data Updates

Credit screened market data updates may have different values for price and quantity than their corresponding unscreened market data updates with the same sequence number. This is due to differences in bilateral credit relationships across Matching participants. Additionally it is possible for credit screened market data updates received across two RFA endpoints in different geographic regions to have different values for price and quantity for the same sequence number. This is due to the asynchronous nature of credit update processing across regional distribution centres. To account for this scenario we recommend that a timestamp be used to uniquely identify each update.

3.2.6.2 Regional Market Data Updates

The Matching v1.2 release enhances the trigger mechanism used to start each market data conflation cycle. The new mechanism utilizes a centrally generated trigger from our Matching service instead of the local Network Time Protocol (NTP). This new functionality was introduced to provide better alignment of market data content across regions. All market data updates within Matching are sourced from our primary data centre (London UK) and distributed globally via regional data centres.

Chapter 4 Open Message Models (OMM)

The various sets of data distributed by the MDFD servers are provided in the Open Message Model (OMM) format.

Some of these OMMs are standard models provided by the RFA API. They are documented in the RDM Usage Guide (provided with the library in Docs/RDMUsage.pdf) and so may be familiar to developers who have used the RFA API in other products.

Other models are what the RFA library refers to as *User Defined*. These are for purposes that are very specific and therefore unlikely to be re-usable by any other product. As such these models:

- do not appear in the RDM Usage Guide
- do not have their names defined as constants in the C++ or Java libraries
- have model numbers greater than 100
- have negative field identifiers and do not appear in the standard dictionaries provided with the RFA library

In order to parse the user defined fields MAPI specific dictionaries need to be used. These include both the standard and user defined definitions and so they can be used as direct replacements for the standard files (etc/RDM/RDMFieldDictionary and etc/RDM/enumtype.def) that were provided with the library itself.

The MAPI dictionaries are available on BizBuzz (http://transactions.thomsonreuters.com) alongside all the other documents and samples used for MAPI development.

The Dictionary OMM on the MDFD server includes these user defined fields and therefore clients who subscribe to the Dictionary OMM will automatically get these additional definitions.

The models available on a MAPI MDFD server are:

Purpose	OMM Model	Type	Provides
Matching Pricing	Market By Price	Standard	Available, Worst and Last Trade prices
Matching Configuration	Symbol List	Standard	List of RFA symbol names
	Reference Data	User Defined	List of instruments and their properties
	Currency Reference Data	User Defined	List of individual currencies
	Configuration Data	User Defined	List of asset class and their properties List of tenors
	Presentation Data	User Defined	Arranges related instruments into higher level groupings to facilitate navigation/display in GUIs.
Generic RFA functions	Login	Standard	Used to connect and authenticate but does not provide data itself.
	Source Directory *	Standard	A list of services along with: • current status • dictionary requirements • supported OMMs
	Dictionary *	Standard	Provides the field and enumerated value dictionaries, as an alternative to loading them from local flat files.

** Please see the RDM Usage Guide for details on how to subscribe and use data provided in these models. They do not require additional MAPI specific instructions and as such they are not discussed any further in this section.*

4.1 Market by Price

This model provides the market data and it is the main purpose RFA is used in MAPI. (The other models only support RFA or provide configuration information).

It provides 3 different aspects of pricing information:

- Available prices (screened or unscreened)
- Last Traded Price
- Worst Price

All three are reported in the same message at the same time. A user does not subscribe individually to each part and the server does not send them out as separate messages either.

A subscription is made for each RFA symbol (section 4.2) and an application must subscribe to each symbol it is interested in. This in turn means an application only receives market data for the instruments it specifically wants rather than receiving data for every instrument regardless of interest.

Whether the market data is screened or unscreened depends upon the RFA symbol used. If *both* screened and unscreened data is required then the client simply makes two subscription requests – once for each symbol.

4.1.1 Last Trade Price

There are several fields in the Market by Price message providing data related to the last trade. It provides the last trade price, side, date & time. It *does not* provide the last traded quantity.

As explained in 3.2.2 all the market data is conflated and therefore it is only reporting the very *last* trade within the conflation period rather than every trade. Also RFA updates can omit fields when they haven't changed for efficiency reasons.

This means some or all of the last trade related fields might not be present in any given message.

The table below lists which fields would be present in each scenario to help understand the above principles:

Scenario	Price PRIMACT_1(393)	Side ACT_TP_1(270)	Date VALUE_DT1(875)	Time VALUE_TS1(1010)
No trade	No	No	No	No
Trade at the same price and same side	No	No	Yes	Yes
Trade at the same price but different side	No	Yes	Yes	Yes
Trade at the same side but different price	Yes	No	Yes	Yes
Trade at different side and different price	Yes	Yes	Yes	Yes

4.1.2 Request Message

Component	Values	Comment
MessageModelType	RDMMsgTypes.MARKET_BY_PRICE rfa::rdm::MMT_MARKET_BY_PRICE 8	Constants are defined in both Java and C++ APIs or alternatively the numerical value can be used.
Attribute Info		
ServiceName	TRFX_MDFD_Standard	
Name	*RFA Symbol Name – e.g.* GBP= GBP=<*TCID*>	Unscreened subscriptions use the plain RFA symbol name Screened subscriptions have the client's TCID (aka branch or credit-code) appended after the equals symbol See the Symbol List model for details of the RFA Symbol Syntax.

4.1.3 Response Message

The response message is detailed in Appendix A.

4.2 Symbol List

This model will return the complete list of RFA symbols that exist in Matching. The symbol names are used in the Name attribute of the Market By Price message described above in 4.1.2.

RFA symbols are not identical to the values used in the Symbol(55) tag of a FIX message. Generally RFA symbol names in MAPI:

- Are in upper case
- Do not include the separating forward slash ("/") or any spaces
- Have an equals symbol at the end ("=")
- For screened prices the client's TCID is given after the equals symbol

The correct Symbol(55) tag value for a specific RFA symbol can be found in DSPLY_NAME (FID 3) of the Market By Price message.

The general syntax of the MAPI RFA symbol names is:

CCY1CCY2[RESTRICTION][TENOR]=[TCID for screened prices]

Where either the base or the terms is the USD though only the other currency is listed in the name:

CCY[RESTRICTION][TENOR]=[TCID for screened prices]

Please note though that Matching only trades certain currency pairs and the tenors & restrictions for each pair can vary. The tables below explain what the various codes mean but it does not imply that any combination of these codes can be used to produce a valid instrument. A string might have the correct syntax for a RFA symbol name but it does not guarantee that it exists.

The symbol list will return every unscreened instrument that currently exists in Matching and attempting to subscribe to any other symbol name will *always fail*.

4.2.1 Restrictions

If a currency pair has different restrictions for domestic vs. offshore and both instruments are available in Matching then they can be differentiated as follows:

Restriction	Description
DOM	Domestic
OFF	Offshore

If no restriction is given in the symbol name then is does *not* necessarily imply that the instrument can be traded anywhere. There may still be a restriction but as it is the *only* instrument traded in Matching different values are not needed to avoid symbol name clashes.

4.2.2 Tenors

The tenor identifies when settlement will take place. If no tenor is given then it implies normal spot settlement (e.g. t+2 for most instruments, t+1 for usd/cad etc.)

Tenor	Description
Spot	
<none>	Normal Spot settlement
T	Today (t)
TOM	Tomorrow (t +1)
Forwards	
ON	Overnight
TN	Tomorrow Next
CS	Cash Spot
<*n*>D	Spot - <n> Days
SN	Spot Next
SW	Spot Week
<*n*>W	<n> Weeks
<*n*>M	<n> Months
<*n*>Y	<n> Years
IM1	International Monetary Market date 1
IM2	International Monetary Market date 2

The list will provide all legal symbol names in the system regardless of whether the permission exists to actually subscribe to that symbol. It is not customized for each client. Currency restrictions and an organisation's own business rules mean that it is normal for there to be some entries in the list which are not accessible.

If the unscreened version of a symbol name does not exist in this list then a Market By Price request for it will most certainly fail; either the symbol represents a currency pair/tenor that Matching does not trade, or it is syntactically invalid and could never represent an instrument in the first place.

But applications which take this list and simply iterate through trying to subscribe to each one in turn will generate a large number of rejections. This could make it harder for support teams to tell the difference between a genuine problem and an expected failure that occurs all the time. Therefore we strongly recommend that client applications maintain configuration of the particular symbols needed and only attempts to subscribe to those ones (after verifying they exist in the Symbol List).

The response message is detailed in Appendix B.

4.2.3 Screened vs. Unscreened

The list only provides the unscreened symbol names.

Every unscreened symbol has a screened version too and the screened symbol name is created by simply appending the TCID after the equals sign.

For example a Market By Price subscription for EURGBP= would return the unscreened prices, whereas EURGBP=ABCD would return the screened prices for TCID ABCD (assuming your RFA token was taken from a FIX session belonging to the TCID ABCD).

Attempts to request screened prices for a different TCID will always be rejected.

4.2.4 Request Message

A separate request must be made for each asset class – it is not possible to receive a list of both Spot and Forward symbols in a single list.

If the Name attribute in the request contains only the asset class then response will just be a list of symbol names.

However if the service name that will be used for the Market By Price requests is appended to the asset class then the response provide extra fields.

Component	**Values**	**Comment**
MessageModelType	RDMMsgTypes.SYMBOL_LIST rfa::rdm::MMT_SYMBOL_LIST 10	Constants are defined in both Java and C++ APIs or alternatively the numerical value can be used.
Attribute Info		
ServiceName	TRFX_MDFD_Symbols	
Name	FX_SPOT *or* FX_FORWARD_SWAP FX_SPOT:<*Market By Price Service Name*> FX_FORWARD_SWAP:<*Market By Price Service Name*>	The asset class to which the symbols belong and optionally the service name that will be used in the Market By Price subscription later. e.g. FX_SPOT:TRFX_MDFD_Standard

4.3 Reference Data

This model will return the list of instruments for a particular asset class.

Unlike the Symbol List model though, this provides more than just the RFA symbol names. Along with the names it will also include various properties for each instrument including the pip size, decimal places, Minimum Quote Life (MQL) limits and much more.

The majority of these properties are needed in order to submit and manage orders via the FIX protocol. For instance orders where the prices has too many decimals or is not a multiple of the pip-size will be rejected, and cancellations will not be accepted until an order has been open for at least the MQL limit.

Applications that do not obtain these properties from the *Reference Data* model will therefore need to rely on manual configuration in order to operate correctly.

The response message is detailed in Appendix C.

4.3.1 Request Message

Component	Values	Comment
MessageModelType	147	This is *User Defined* model and consequently the Java & C++ APIs do not contain named constants for this model.
Attribute Info		
ServiceName	TRFX_MDFD_Symbols	
Name	FX_SPOT FX_FOWARD_SWAP	The asset class

4.4 Currency Reference Data

This model will return a list of currencies (note this is an individual currency, not a currency pair).

The message consists of a Map where the key of each item is the *Currency ID*. This ID is the foreign key used in other models when they need to refer to a specific currency.

The response message is detailed in Appendix D.

4.4.1 Request Message

Component	Values	Comments
MessageModelType	175	This is *User Defined* model and consequently the Java & C++ APIs do not contain named constants for this model.
Attribute Info		
ServiceName	TRFX_MDFD_Symbols	
Name	*	An asterisk.

4.5 Configuration Data

This model will returns configuration information that is not specific to a particular instrument.

It contains 3 different areas with each one represented in its own Map:

- Asset Classes
- Market Data Groups
- Tenors

The information is needed in order to use the FIX API correctly. However unlike the information in the Reference Data model which focuses on instruments, this information is mostly at the asset class level. As such this data reports the more fundamental configuration related to Spot vs. Forward Swaps behaviour and many applications will have hardcoded that behaviour in without any real difficulties.

The response message is detailed in Appendix E.

4.5.1 Request Message

Component	Values	Comments
MessageModelType	148	This is *User Defined* model and consequently the Java & C++ APIs do not contain named constants for this model.
Attribute Info		
ServiceName	TRFX_MDFD_Symbols	
Name	PV6	

4.6 Presentation Data

This model will return a list of instruments and display groupings along with attributes that are used to control its display on a keystation or New Matching GUI.

The data allows the instruments to be arranged into a tree like structure, with asset classes at the top level, and then the currency pairs, followed by (for forward swaps) the tenors.

e.g.

- FX Spot
    - aud/cad
    - aud/chf
    - aud/cnh
    - ...
- FX SWAP
    - aud/usd
        - aud/usd ON
        - aud/usd TN
        - aud/usd SN
        - aud/usd SW
        - ...
    - ...

None of the information in this model is required to be able to successfully submit or manage an order. However applications that wish to display the list of instruments in logical groupings may wish to make use of this data.

The response message is detailed in Appendix F.

4.6.1 Request Message

Component	Values	Comments
MessageModelType	149	This is *User Defined* model and consequently the Java & C++ APIs do not contain named constants for this model.
Attribute Info		
ServiceName	TRFX_MDFD_Symbols	
Name	*	

4.7 Login

This standard RDM is used to connect and authenticate with the MDFD server.

4.7.1 Request Message

Component	Values	Comments
MessageModelType	RDMMsgTypes.LOGIN rfa::rdm::MMT_LOGIN 1	
Attribute Info		
Name	<RFA token> e.g. 4cd7666a-b053-44f7-91d0-7db22a0bbd62	This is the value retrieved from the Text(58) tag of the FIX Login (35=A) response. Text(58) starts with the prefix “RFA:” however that must be stripped off when populating the *Name* attribute.
ApplicationId	174	
Position	<IP address or hostname>/net e.g. 192.0.2.1/net	

Chapter 5 RFA for Matching Data Feed Direct

The MAPI MDFD servers use Open Message Model (OMM) messages over a Reuters Source Sink Library (RSSL) connection type.

Other message or connection types that a RFA library version might support will not be accepted by the MAPI MDFD server. Therefore unless a client is intending to also use RFA elsewhere outside of MAPI there it is not necessary to learn those other areas of the API.

Specifically the client acts as an OMM *Consumer* which sends subscription requests to the MDFD server in order to receive the relevant market data responses.

OMM *Providers* on the other hand are used to publish data via RFA. All market data from a MAPI MDFD server though is sourced from orders & trades within Matching and clients cannot contribute data via RFA directly.

5.1 OMM Concepts

OMM messages are an efficient but flexible way of transporting complex data structures made up of primitive data types and even other nested complex data structures.

5.1.1 OMM Message Types

There are 5 types of message.

Message Type	Direction	Purpose
REQUEST	Client -> MDFD server	Requests to open a subscription. It will typically contain the service, message model, name and possibly some other attributes to define what the subscription is for. When the REQUEST is submitted via the *registerClient()* API method a handle is returned which can be used to refer back to the particular subscription at a later point.
REFRESH	MDFD server -> client	Contains the data that was subscribed to. A REFRESH message is a complete snapshot and does not rely on the client already having previous data. It is used to provide clients with the initial data when a subscription is first opened, as well as in some connection recovery scenarios.
UPDATE	MDFD server -> client	Contains incremental changes to the subscribed data. As its only an incremental change, the data provided needs to be combined with the data previously received in order to see the full picture. It is used to provide clients with updates to the subscribed data throughout the lifetime of the subscription.

Message Type	Direction	Purpose
STATUS	MDFD server -> client RFA library -> client	A message that *only* contains changes to the OMM statuses. STATUS messages are used at times when there is a need to inform of status changes but there is no other data to send. In cases where there is *both* status and data to provide an UPDATE or REFRESH message will be used instead (as those too can contain the OMM statuses). The RFA library can also generate status messages locally and this is how the client can be informed of status changes when there are problems communicating with the MDFD server.
CLOSE	Client -> MDFD server	Closes a subscription. Unlike a REQUEST this message is not directly created by the client. Instead the client simply calls the unregisterClient() API method and provides the *handle* for the particular subscription. The API library then generates the CLOSE message and sends it to the MDFD server.

Given Matching runs for the whole trading week (Sunday -> Friday) and the FX markets can change frequently UPDATE messages are by far and large the most commonly observed message type in MAPI. REFRESH messages are typically only seen when starting a subscription or after the connection recovers in scenarios where it is not practical for the server to determine what incremental UPDATE is necessary.

In RFA many tasks are actually implemented as OMM subscriptions – it is not just limited to the market data itself. For instance:

- To login or logout of the MDFD a client subscribes or unsubscribes using the Login OMM
- Field and enumerated value dictionaries can be downloaded using the Dictionary OMM
- The current state of services on the MDFD server are reported on a Source Directory OMM
- The list of RFA known symbols is provided on the Symbol List OMM

5.1.2 OMM Statuses

As well as the payload data, OMM subscriptions have state information to help clients determine the current state of a subscription.

The status is not a single field but rather two separate statuses. One status for the stream (subscription) and another for the data contained within it.

5.1.2.1 Stream State

The Stream State specifies whether the subscription is currently open or not. When a stream is OPEN it means the RFA library is managing the subscription and data will be received when the server is able to provide it.

Streams in a CLOSED or CLOSED-RECOVER state indicate the subscription is finished and no further data will be received.

CLOSED is used if the client requests closure (e.g. when no longer interested in it, or as part of an application shutdown), or if the server refuses to provide the data (e.g. client does not have permission, requesting non-existent data etc.).

CLOSED-RECOVER is used in scenarios where the RFA library would normally manage recovery, but the client has expressly disabled that functionality (by setting attribute SingleOpen = 0 in the

Login OMM). Has the recovery not been disabled (SingleOpen = 1, the default) then a stream state of CLOSED-RECOVER would not be used and instead the event would have stream state = OPEN, data state = SUSPECT.

5.1.2.2 Data State

The Data State specifies whether the data provided on the subscription is currently valid.

When data is SUSPECT it means there is problem that would prevent new updates from being received. Therefore any existing data the client has cached should be treated with caution as it might be out-of-date.

A data state of OK means that it is correct.

Stream State	Data Sate	Meaning
OPEN	OK	Subscription is open and data is up-to-date
OPEN	SUSPECT	Subscription is open, but cache data might be out-of-date as any new updates cannot be received at present.
CLOSED	OK	Subscription is closed but not due to an error of any kind. E.g. client has unsubscribed, or sent a request for a non-streaming subscription (a one-off response without any further updates later on).
CLOSED	SUSPECT	Subscription is closed due to an error. Requests that immediately result in this state are permanent failures. The MDFD server received the request and understood it but has refused to allow it. Clients need to correct their request for trying again. Retrying the same request again will fail and clients need to correct their request.
CLOSED-RECOVER	SUSPECT	Subscription is closed due to a temporary error, but the RFA library will not keep trying to recover from it. This is used instead of OPEN SUSPECT if the client logs in with attribute SingleOpen = 0.

5.1.2.3 Status Code & Status Text

These two pieces of information provide extra details as to the reason behind the current Stream & Data states. This information can be useful when supporting and debugging applications using RFA and it is strongly recommended that the values are logged.

5.1.3 Handles

Each time a subscription request is made via the registerClient() API call a handle is returned. The handle is a unique identifier that is needed to refer back to the subscription later on.

The handle can be used to:

- Unsubscribe, via unregisterClient()
- Pause or Resume a subscription, via reissueClient()
- Request a full REFRESH, via reissueClient()
- Match responses back to the right subscription, via the event's getHandle()

The handle object can only be compared for equality though – there is no way to directly access its value.

5.1.4 Connection Management and Data Recovery

RFA offers full connection management and market data recovery management.

There is a heartbeat mechanism in place between the RFA library and MDFD server to make sure the network connection remains open and the application at both ends is still alive. Client applications do not need to monitor the heart beating (in fact the API provides not means by which to do so).

Should the library detect a problem communicating with the MDFD server it will raise an event reporting that there is an issue. The event will have Stream State = OPEN & Data State = SUSPECT to indicate that data cannot currently be received and anything the application previously cached should be treated with caution.

The same mechanism is used if the problem is further upstream (between the MDFD server and the components it connects to). This means that regardless of where the problem is located the client is notified in a consistent way.

The status text will contain more information as to what the problem is. This information is not intended to be parsed by applications as its formatting and contents can vary. However the information is useful for support purposes and it is recommended applications log this information.

As long as the Stream State = OPEN though client applications do not need to take any action to recover the connection. The RFA library will automatically retry and once the situation recovers it will receive any UPDATE or REFRESH messages that are necessary to bring its cached data back up to date.

The recovery does not replay all the history though. The client will receive whatever is necessary so the current data is correct but any intermediate changes will have been lost.

The connection recovery can be disabled by setting the attribute SingleOpen = 0 in the Login OMM. In that case when there is a problem communicating the RFA library will simply close the subscription (stream state = CLOSED-RECOVER). It will be up to the client to determine when the problem has been resolved and then reconnect & resubscribe as necessary. (See section 3.4.8 of the *RDM Usage Guide* for further details).

Client application is expected to download RDM dictionaries during initial connection attempt while the Matching service is in non-trading mode. Client should avoid dictionary download during trading mode as it can cause significant bandwidth utilization and can cause interruption in normal trading operation.

5.2 Consumer Application Responsibilities

The following are the responsibilities of a consumer application:

1. Setting up the application, which involves the following:
    - Initialize OMM Consumer.
    - Login to market data system with correct RFA token as acquired during FIX user logon.
2. Send requests:
    - Request for OMM Data which may result in an open event stream using registerClient() method
    - Pause/Resume or obtain a REFRESH on an existing open stream using reissueClient() method
    - Request to close an existing open stream using unregisterClient()
3. Handling inbound events which encapsulate the response messages.
4. Cleaning up and shutting down.

RFA C++ Client application should use OMMPerfMode parameter (as explained in RFA Config Guide) to run the RFA in latency mode. Client application can also use API Call out feature to run their data feed application in more performant manner. Please refer to section 13.1 – 13.4 of the RFA Developers Guide.

The equivalent configuration parameter to OMMPerfMode in RFA Java is the shareConnections session parameter. By default, the value of the parameter is false, which disables the connection sharing between sessions and also reduces the latency. This means RFA Java applications run in minimized-latency mode by default. In the setting, RFA Java uses the Merged Thread Model in which the Session and the Connection uses the same thread, and the event queue between the Session and the Connection is removed. Please refer to section 10.1 – 10.4 of the RFAJ Developers Guide for more information.

5.2.1 Processing RFA Events

Clients receive the responses (REFRESH, UPDATE, STATUS messages) via an Event object. The Event is passed to the client as the argument to the clients processEvent() API callback.

As well containing the OMM response, the Event also contains other useful items such as the Handle to which the response relates.

When processing the event the client needs to take account of both the OMM message type and the OMM status information (if present). The message type (section 5.1.1) determines whether there is any data present and if so whether it is incremental or a full snapshot. The status provides changes to the stream and data state.

5.2.2 Processing Map Entries

Some message models make use of a data type called a Map. The depth of book data in the Market By Price model is one such use where each price & side is represented by its own Map Entry.

Map Entries (of which the Map is made up) allow specific entries to be added, updated or deleted in later updates without having to resupply the whole Map again.

A Map Entry contains:

- An action (ADD, UPDATE or DELETE)
- A key, giving the entry a label so it can be referred to in later messages
- A field list, or any other data type, containing the data related to this entry

Actions:

- ADD: add a new entry
- UPDATE: change a previously notified entry
- DELETE: remove a previously notified entry

The order in which map entries are provided is not specified and can vary at any time.

Any sorting if necessary will need to be done by the client and then maintained as entries are added, updated and deleted later on. Care should be taken to ensure that when processing the Map Entries the correct changes are made no matter what order the ADD, UPDATE and DELETE actions occur within the Map. For example if an ADD was listed before a DELETE it would (briefly) take up more space than a DELETE followed by an ADD.

Once all the actions have been applied the Map will be in a consistent state again. However the transitional state whilst only part–way through processing will not necessarily be valid and therefore caution should be taken before displaying or acting on data at any such stage.

For the Market By Price model specifically it should be noted that the limits defined by BOOK_DEPTH (FID 6454) & PRICETHOLD (FID 6455) are for information only. The MDFD servers will have already taken these into account when producing the Map Entries and so a client application just needs to process those and does not need to make further edits of its own.

However it can be useful to assert that after processing all the Map Entries in the event that the client’s record of the prices is still within the limits. This can then allow a client application to fail fast should its record of the current state conflict with the known limits.

5.2.3 Market Data Un-subscription

Subscriptions to items are cancelled by closing the handle returned by the successful subscription. Whilst this operation is processed the API may receive price update messages.

RFA user will receive an indication of the success or failure of the operation.

If the operation is successful, the RFA will no longer receive updates for which he has unsubscribed (closed).

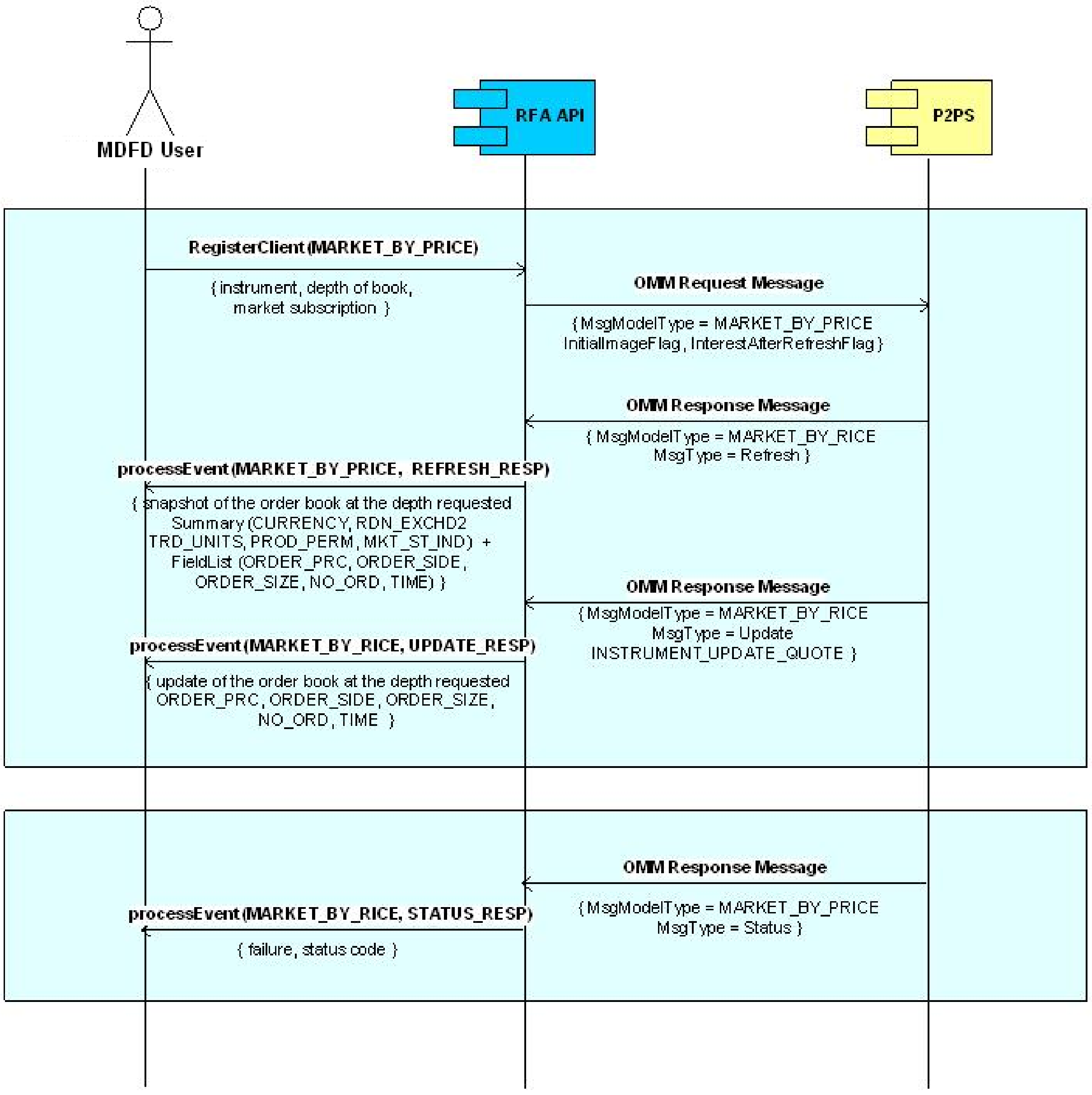
The user should still be able to place orders for that particular instrument in the system (via the FIX Gateway).

(Whilst the RFA library also has historical support for other formats and connections types none of these are available in MAPI).

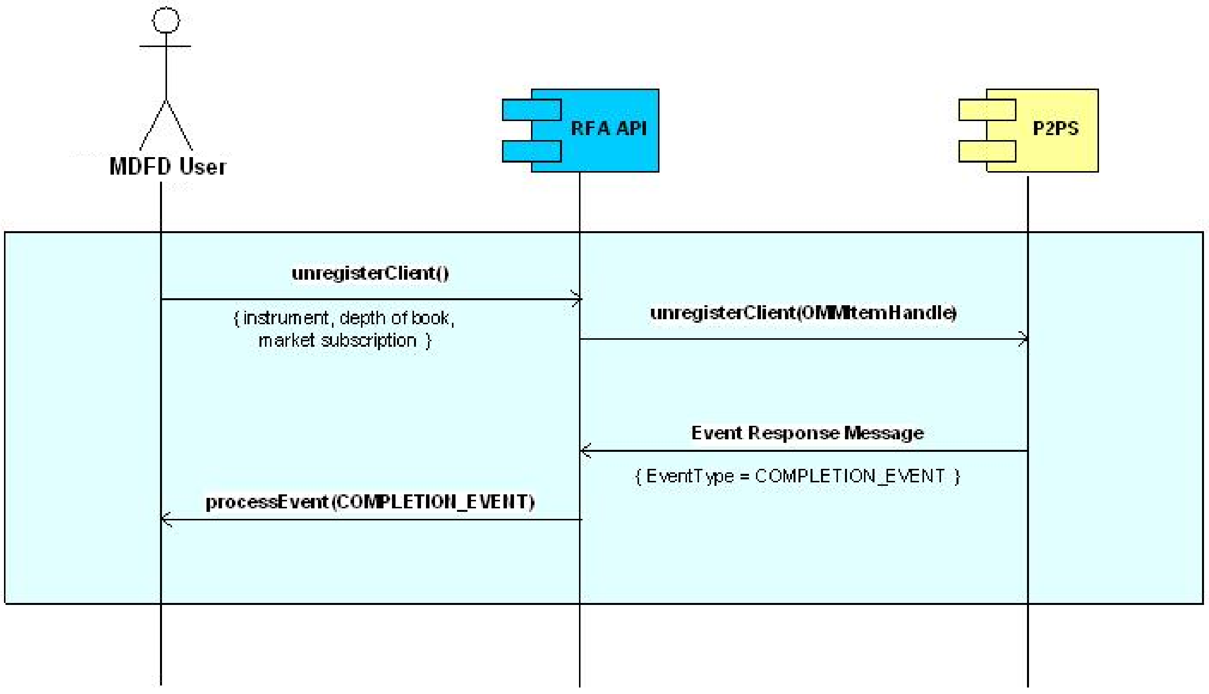
```
Below is an example of a refresh message received at first event update

Message Type = REFRESH_RESP,
Msg Response Type =SOLICITED,
Msg State Code = NONE,
Msg Data State = OK,
Msg Stream State = OPEN,
Msg State Text = ""

PROD_PERM (1) - 7583,
RDNDISPLAY (2) - 153,
DSPLY_NAME (3) – aud/usd,
CCY1 (517) - AUD,
CCY2 (518) - USD,
REG_AMOUNT (6306) - 35,
STD_AMOUNT (6302) - 20,
TRD_UNITS (53) - 4,
PRICE_CONV (573) - 1,
TENOR (4402) - SPOT,
PRIMACT_1 (393) - 1.0359,
ACT_TP_1 (270) - 16,
VALUE_DT1 (875) - 18 JUL 2012,
VALUE_TS1 (1010) - 15:58:41:306 GMT,
LOTSZUNITS (54) - USD,
LOT_SIZE (55) - 1,
QUOTE_DATE (3386) - 18 JUL 2012,
BID (22) - 0.0000,
ASK (25) - 0.0000
PRICETHOLD(6455) – 0.002
BOOK_DEPTH(6454) – 3
QUOTIM_MS(3855) – 63656467
SEQ_NO(3859) - 555618

Below is an example of an UPDATE Action when the volume changes on an existing price

2012-07-18 13:13:58,713 DEBUG RFA Client - Message State:
Type = OMM_ITEM_EVENT,
Message Type = OMM_SUB_RESP_MSG,
Msg Message Type = UPDATE_RESP,
Msg Response Type =SOLICITED
Key = B.1.0356,
Action = UPDATE,
ORDER_SIZE (3429) - 5,
ORDSIZEIND (6305) - 0,
NO_ORD (3430) – 1

Below is an example of an ADD and DELETE action when a better price is added and an old
   price is removed

Key = S.1.0362
Action = ADD
ORDER_PRC (3427) – 1.0362
ORDER_SIZE (3429) – 5
ORDSIZEIND (6305) – 0
ORDER_SIDE (3428) – 2
NO_ORD (3430) -1

Key = S.1.0359,
Action = DELETE,
Field list = [empty]
```

```
MARKET_BY_PRICE with Depth of Book disabled will receive price updates in individual
   UPDATE_RESP message events
Event 1: Message_Type=UPDATE_RESP QUOTIM_MS(3855)=68300469
Action=ADD 5 @ 1.2262 Key S.1.2262
Event 2: Message_Type=UPDATE_RESP QUOTIM_MS(3855)=68300470
Action=UPDATE 3 @ 1.2262 Key S.1.2262
Action=ADD 5 @ 1.2163 Key B.12163
Action=DELETE Key S.1.2262
Event 3: Message_Type=UPDATE_RESP QUOTIM_MS(3855)=68320472
Action=DELETE B.1.2163


MARKET_BY_PRICE with Depth of Book enabled will receive information about each level in
   iterating MAP entry of single UPDATE_RESP events in one event update
Event 1: Message_Type=UPDATE_RESP QUOTIM_MS(3855)=63656467
Action=ADD 5 @ 1.0357 Key B.1.0357
Action=ADD 5 @ 1.0360 Key S.1.0360
Action=Add 5 @ 1.0361 Key S.1.0361
Action=ADD 5 @ 1.0359 Key S.1.0359
Action=ADD 5 @ 1.0355 Key B.1.0355
Action=Add 5 @ 1.0356 Key B.1.0356

Event 2: Message_Type=UPDATE_RESP QUOTIM_MS(3855)=63656468
Action=ADD 5 @ 1.0354 Key B.1.0354
Action=ADD 5 @ 1.0362 Key S.1.0362
Action=UPDATE 3 @ 1.0356 Key B.1.0356
Action=UPDATE 3 @ 1.0355 Key B.1.0355
Action=UPDATE 5 @ 1.0361 Key S.1.0361
Action=UPDATE 15 @ 1.0360 Key B.1.0360
Action=DELETE Key B.1.0357
Action=DELETE Key S.1.0359
```

5.2.4 Screened/Unscreened Market Rate subscription & updates


MDFD User
RFA API
P2PS
RegisterClient (MARKET_BY_PRICE)
{ instrument, depth of book, market subscription }
OMM Request Message
{ MsgModelType = MARKET_BY_PRICE InitialImageFlag, InterestAfterRefreshFlag }
OMM Response Message
{ MsgModelType = MARKET_BY_RICE MsgType = Refresh }
processEvent (MARKET_BY_PRICE, REFRESH_RESP)
{ snapshot of the order book at the depth requested Summary (CURRENCY, RDN_EXCHD2 TRD_UNITS, PROD_PERM, MKT_ST_IND) + FieldList (ORDER_PRC, ORDER_SIDE, ORDER_SIZE, NO_ORD, TIME) }
OMM Response Message
{ MsgModelType = MARKET_BY_RICE MsgType = Update INSTRUMENT_UPDATE_QUOTE }
processEvent (MARKET_BY_RICE, UPDATE_RESP)
{ update of the order book at the depth requested ORDER_PRC, ORDER_SIDE, ORDER_SIZE, NO_ORD, TIME }
OMM Response Message
{ MsgModelType = MARKET_BY_PRICE MsgType = Status }
processEvent (MARKET_BY_RICE, STATUS_RESP)
{ failure, status code }

5.2.5 Market Rates Un-subscription


MDFD User
RFA API
P2PS
unregisterClient()
{ instrument, depth of book, market subscription }
unregisterClient(OMMItemHandle)
Event Response Message
{ EventType = COMPLETION_EVENT }
processEvent(COMPLETION_EVENT)

Appendix A Market By Price Model

The summary data section of the Map provides the fields that are not specific to any particular depth.

Map Summary Data		
FID ID	**ACRONYM**	**Description**
1	PROD_PERM	Permission code
2	RDNDISPLAY	Display template number
3	DSPLY_NAME	Display name for the instrument. This is the value that must be used in the Symbol(55) FIX tag when placing orders etc.
517	CCY1	Base ISO currency *for ex. EUR*
518	CCY2	Terms ISO currency *for ex. USD*
15	CURRENCY	Currency the price is specified in *for ex. USD*
525	VDATE_P1C1	Value Date Period 1 (Spot or near leg of an FX Forward Swap) Note this field is not a DATE data type. It is a string representing a date in the format DD MMM YYYY.
527	VDATE_P2C1	Value Date Period 2 (Forward or far leg of a FX Forward Swap) Not used for FX Spot currency pairs Note this field is not a DATE data type. It is a string representing a date in the format DD MMM YYYY.
53	TRD_UNITS	Maximum number of decimal places
573	PRICE_CONV	Price Convention 1 = Normal 2 = Inverse
4402	TENOR	Period to maturity for the instrument *(e.g. 1M, 2M, 1D, etc. For SPOT, it will include just SPOT*
393	PRIMACT_1	Last Traded Price
270	ACT_TP_1	Last Traded Side 16=B (Bid/Paid) 18=A (Ask/Given) Value is reported from the aggressor’s point of view.
875	VALUE_DT1	Last Trade Date
1010	VALUE_TS1	Last Trade Time
54	LOTSZUNITS	Lot size units
55	LOT_SIZE	Lot size
3386	QUOTE_DATE	Date this order book information was calculated from.

Map Summary Data		
3855	QUOTIM_MS	Time this order book information was calculated from. Note this field is not a TIME data type. It is the number of milliseconds since midnight GMT.
22	BID	Worst Bid Price (see section 3.2.4)
25	ASK	Worst Ask Price (see section 3.2.4)
6302	STD_AMOUNT	Standard Amount for this instrument. This is the quantity on which the Worst Price calculation is based (see section 3.2.4)
6306	REG_AMOUNT	Regular Amount for this instrument. This is the maximum quantity that will be reported at each price (see section 3.2.1).
6454	BOOK_DEPTH	The maximum number of depths that will be provided for this instrument (see section 3.2.1).
6455	PRICETHOLD	The maximum difference from top of book price that will be provided for this instrument (see section 3.2.1).
3859	SEQ_NO	Provides a global market data sequence number for the given currency pair. This sequence number will be reset at the start of each trading week. (see section 3.2.6)

Each Map Entry represents one price on one side.

Map Entry		
<Map Entry Key>		Unique label so that future UPDATE or DELETE actions can refer back to this entry
FID ID	**ACRONYM**	**Description**
3427	ORDER_PRC	Order Price
3428	ORDER_SIDE	1 = BID 2 = ASK
3429	ORDER_SIZE	Total Volume available at this price. This field is limited by the Regular Amount. If ORDSIZEIND = 1 then there is more qty available at this price than this field reports.
6305	ORDSIZEIND	Indicates whether the Regular Amount has been exceeded. 0 = <blank> (Not exceeded. There is exactly ORDER_SIZE available) 1 = + (Exceeded. There is more than ORDER_SIZE available)

Map Entry		
3430	NO_ORD	Number of orders that make up the ORDER_SIZE (FID 3429) value. This field is limited by the Regular Amount. If ORDSIZEIND = 1 then there can be more orders available at this price than this field reports.

Appendix B Symbol List Model

When the Name attribute is just the asset class (e.g. *FX_SPOT* or *FX_FORWARD_SWAP*):

Map Entry		
<Map Entry Key>		RFA Symbol Name
FID ID	**ACRONYM**	**Description**
	<Empty Field List>	

When the Name attribute also includes the Market By Price service name (e.g. *FX_SPOT:TRFX_MDFD_Standard*)

Map Entry		
<Map Entry Key>		RFA Symbol Name
FID ID	**ACRONYM**	**Description**
6454	BOOK_DEPTH	Maximum number of depths that will be reported for this instrument.

Whilst the Symbol List responses contain only zero or 1 fields, a large amount of data can be returned nonetheless. As a result it is possible that the response will have to be sent as a multi-part REFRESH, especially for the Forward Swap symbols.

Appendix C 147 - Reference Data Model

Map Entry		
<Map Entry Key>		ID number of instrument.
FID ID	**ACRONYM**	**Description**
-10020	MRKTNAME	Instrument name
-10392	SYMBOL	Symbol
-10239	RIC	Reuters instrument code.
-10021	LOTSIZE3	Quantity increment. Quantities that are not a multiple of the lot size will not be accepted.
-10022	MN_ORD_QT	Minimum order quantity
-10061	STD_QTY	Standard Quantity
-10462	RGL_QTY	Regular Amount
-10024	PIPSIZE	The price increment. Prices that are not a multiple of the pip size will not be accepted.
-10025	LEFT_DP	The maximum number of permitted left decimal places.
-10026	RGT_DP	The maximum number of permitted right decimal places.
-10053	BASECURID	ID number of the base currency. Currency IDs are defined in the *Currency Reference Data* model.
-10054	QTDCURRID	ID number of the quoted/terms currency. Currency IDs are defined in the *Currency Reference Data* model.
-10055	TENORID	ID number of the Tenor. Tenor IDs are defined in TNRMAP *(FID -10402)* of the *Configuration Data* model.
-10056	WPT	The maximum difference from top-of-book allowed when calculating the *Worst Price*. Given in pips.
-10099	PT	The maximum difference from top-of-book allowed when showing multiple depths. Given as a decimal.
-10057	RL_DOL_ID	The instrument used to convert into the currency credit limits are maintained in.

Map Entry		
-10058	PERIODTXT	The name of the period (name of the tenor referenced by TENORID).
-10059	INS_GP_ID	An ID to uniquely identify a market group. A market group defines a collection of markets.
-10417	VOL_CUR	Defines whether the market is directly quoted (quoted in dollars) or indirectly quoted (not quoted in dollars). 1 = BASE 2 = QUOTED
-10060	CLS_ID	Indicates if the market is eligible for CLS. 0 = FALSE 1 = TRUE
-10068	T_ORD_MNS	The minimum number of milliseconds allowed when using FIX tags ExpireTime(126) or ExposureDuration(1629) to specify an end time for the order.
-10069	T_ORD_MXS	The maximum number of milliseconds allowed when using FIX tags ExpireTime(126) or ExposureDuration(1629) to specify an end time for the order.
-10435	KS_TT	Legacy value used by the keystation to identify the tenor.
-10438	RRQDR	The quote format for FWD swaps.
-10535	MINQUL	The minimum number of milliseconds an order must remain positioned before a request to cancel it is allowed.
-10556	FARLEG_DP	The number of decimal places permitted on the far leg quantity.
-10624	RGTSMDPS	Indicates how many of the rightmost digits of the price to treat as the fractional part of a pip
-10780	MNOIBO	Maximum number of open Iceberg orders per instrument , per trading user
-10782	MN_IBO_TP	Minimum permitted tip size for an Iceberg order
-10783	MX_IBO_TP	Maximum permitted tip size for an Iceberg order

Appendix D 175 - Currency Reference Data Model

Field List		
<Map Entry Key>		ID number assigned to this currency.
FID ID	**ACRONYM**	**Description**
-10046	CURRISO	ISO 4712 code of the currency
-10047	CUR_DS_CD	Display name of the currency
-10048	MD_CUR_TH	Threshold for medium quantity
-10049	LG_CUR_TH	Threshold for large quantity
-10050	SM_CR_IND	Whether to represent *small* quantities as a letter, or the actual numeric value. 0 = ALPHABETIC 1 = NUMERIC
-10051	MD_CR_IND	Whether to represent *medium* quantities as a letter, or the actual numeric value. 0 = ALPHABETIC 1 = NUMERIC
-10052	LG_CR_IND	Whether to represent *large* quantities as a letter, or the actual numeric value. 0 = ALPHABETIC 1 = NUMERIC
-10407	RDMCUR	Not used.
-10406	LTSZUN	Not used.

Appendix E 148 - Configuration Data Model

Field List		
FID ID	**ACRONYM**	**Description**
-10482	SYSTYPE	The host-wide system type (PRODUCTION or HOST TO GO) 1 = PRODUCTION 2 = HOST_TO_GO
-10663	ASSCLS	A Map where each entry represents a different Asset Class
<Map Entry Key>		ID number of Asset Class.
-10036	->ASCLSNAME	The name of a particular asset class within the Host.
-10037	->CRDMETH	The method of credit used within this asset class. BINARY indicates that the credit relationship can be in one of two states, whereby they either have or do not have credit with counterparty. LIMIT indicates that a credit relationship be unlimited, zero and limited. Limited credit is whereby the credit is greater than zero but not exceeding the MXCREDIT for the asset class/credit universe. 0 = LIMIT 1 = BINARY
-10038	->MATCHBEH	The matching behaviour, either FIRM or SOFT. 0 = FIRM 1 = SOFT
-10083	->MXCREDIT	The maximum credit amount that can be given when the CRDMETH (FID -10037) = 0 (LIMIT).
-10521	->SUPCLS	Indicates if asset class supports CLS. 0 = FALSE 1 = TRUE
-10125	->CRUNIID	ID number of the credit universe. This is not needed to in order to use MAPI and the definitions these IDs link to are not available.
-10522	->SUPMQ	Indicates if asset class supports more quantity. 0 = FALSE 1 = TRUE
-10523	->SUPUQ	Indicates if asset class supports upping the quantity. 0 = FALSE 1 = TRUE

Field List		
-10527	->PRC_RUL	The price rule for an asset class 0 = POSITIVE 1 = POSITIVE_OR_NEGATIVE 2 = ANY_VALUE
-10045	->CURRID	ID of the currency that credit limits are held in. Only used when *CRDMETH (FID -10037) = 0 (LIMIT)*
-10528	->CURMLT	The currency multiplier for a credit universe.
-10775	ITDM	A map of tip delays for Iceberg orders
<Map Entry Key>		ID number assigned to this tip delay range Used in FIX tag `StrategyParameterValue(960)` when `StrategyParameterType(958)=D`
-10776	MINTIPD	Minimum delay in the range in milliseconds
-10777	MAXTIPD	Maximum delay in the range in milliseconds
-10182	MKTGRPMAP	A map of market group definitions keyed by market group identifier.
<Map Entry Key>		ID number assigned to this Market Group.
-10183	->MKTGPNAME	The display name of the market group.
-10019	->ASSCLSID	ID number of the Asset Class. Asset Class IDs are defined in *ASSCLS (FID -10663)*.
-10402	TNRMAP	A map of tenor data keyed on tenor identifier.
<Map Entry Key>		ID number assigned to this Tenor.
-10403	->TNRNM	The name of the tenor (e.g. '3M').
-10404	->TNRVAL	The value of the tenor in units of TNRUNI
-10405	->TNRUNI	The unit of the tenor. 1 = D 2 = W 3 = M 4 = Y 5 = SN 6 = ON 7 = TN 8 = IM 9 = CS 10 = ME 11 = YE 12 = SW

Field List		
-10661	->TN_DOR	An integer that can be sorted to arrange the list in order of tenor length. (Sorting by TNRVAL or TNRUNI won't result in the proper order)

Appendix F 149 - Presentation Data Model

Field List		
FID ID	**ACRONYM**	**Description**
-10043	MRKMAP	A Map containing the individual instruments
<Map Entry Key>		ID number of this instrument
-10086	-> HL_PIP_CT	The number of pips in the quote to highlight on the display.
-10087	-> HL_PIP_ST	The starting quote pip position counting from the right of the decimal point to where highlighting will begin.
-10050	-> SM_CR_IND	The format used to display small quantities. 0 = ALPHABETIC 1 = NUMERIC
-10051	-> MD_CR_IND	The format used to display medium quantities. 0 = ALPHABETIC 1 = NUMERIC
-10052	-> LG_CR_IND	The format used to display large quantities. 0 = ALPHABETIC 1 = NUMERIC
-10062	-> AL_DISPID	Determines what characters to display if the quantity is within the small/medium/large threshold, and that threshold is set to use ALPHABETIC indicators. 0 = ML – display as S, M or L (dependant on the threshold) 1 = R – display as R
-10063	-> LP_RND	Flag to indicate if a 1/2 pip quote is rounded; if true quote must be rounded such that the last pip is either a 0 or a 5. 0 = FALSE 1 = TRUE
-10550	-> MARKETDG	The market display group that this market belongs to. This ID number refers to an entry in MKTDGMAP (FID -10552).
-10551	-> DISPORD	The position of this market/market display group within its market display group
-10552	MKTDGMAP	A Map containing the Market Display Groups which are the non-leaf nodes in the tree-like structure. Each entry in MRKMAP (FID -10043) above is assigned to one of these groups.
<Map Entry Key>		ID number of this Market Display Group.
-10554	-> MKTDGNAME	The name of the group.

Field List		
-10555	-> PRNTGRPID	The ID number of its parent group, or null if it is a top-level node.
-10551	-> DISPORD	The sorted position of this entry compared to other entries under the same PRNTGRPID

© 2014 Thomson Reuters. All rights reserved. Republication or redistribution of Thomson Reuters content, including by framing or similar means, is prohibited without the prior written consent of Thomson Reuters. 'Thomson Reuters' and the Thomson Reuters logo are registered trademarks and trademarks of Thomson Reuters and its affiliated companies.		**For more information** **Send us a sales enquiry at** **financial.thomsonreuters.com/sales** **Read more about our products at** **financial.thomsonreuters.com** **Find out how to contact your local office** **financial.thomsonreuters.com/locations**

Document Version 1.5.5
Date of issue: 3 July 2015

THOMSON REUTERS