Universal Trading Platform
Market Data

Cash Markets

Version 2.6 **March 2010**

Disclaimer

The registered office of NYSE Technologies SAS is

6-8 Boulevard Haussmann 75441 Paris Cedex 09 France

Telephone: +33 (0)1 73 03 14 14 Fax: +33 (0)1 73 03 03 04

Siège social : 6-8 Boulevard Haussmann 75009 Paris

SAS au capital de 2 869 962 €

RCS Paris 425 100 294 - N°TVA : FR 87 425 100 294

NYSE Technologies is part of the NYSE Euronext Group.

www.nyseeuronext.com

1. Cash Markets Processing Information

1.1. *Introduction*

1.1.1. Contact Information

For any questions about this specification please contact your UTP Helpdesk (see contact information in appendix A):

1.1.2. Universal Trading Platform Market Data Overview

The Universal Trading Platform Market Data Feed provides high speed real-time market data for NYSE Euronext European Cash markets and Qatar Exchange Cash Market.

The data feed has the following high-level features:

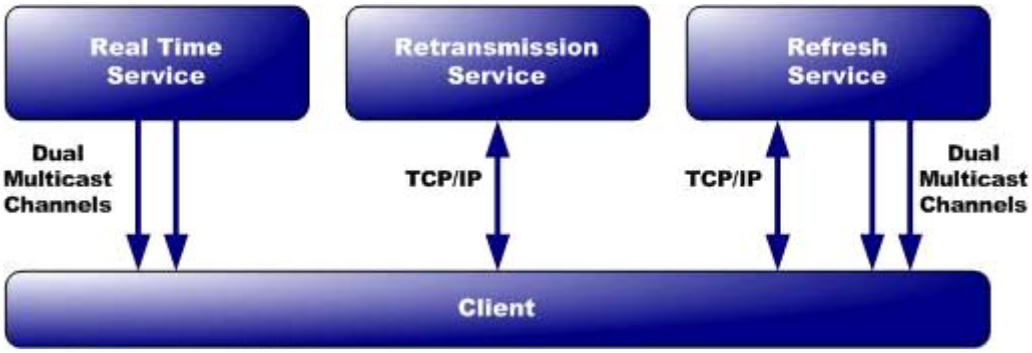
- Multicast technology
- High Availability
- Ultra-low latency
- Reliable network solution
- High level of scalability
- Access to wide range of market data sets

This chapter provides detailed information about the features of the feed, to support the development of client applications by Members, Independent Software Vendors and Quote Vendors.

The following chapters of this document provide details that are specific to each of the market data sets, including formats for each message type.

1.2. *Access to Market Data*

Clients connect to multicast addresses for the real time market data messages and refresh data, and can also connect to a TCP/IP server for packet retransmissions. Requests for retransmission and refresh are performed via TCP/IP



1.2.1. Real Time Market Data

Real-time market data is message-based over the UDP IP protocol with fixed length binary and ASCII fields.

It uses the push-based publishing model. This means that data will be published based on its availability. Once an update is available, it will be published to the appropriate multicast group.

For business & capacity reasons, market data will be split across a number of multicast groups organized into predefined data sets.

Each multicast group will deliver a set of data for a certain market segment.

The client application will be responsible for issuing Multicast subscriptions to one or more of the multicast groups assigned to each product.

Please refer to chapter 4 and 5 for detailed content of Production and External User Acceptance multicast groups (in the case of Euronext implementation).

The process of subscribing to a multicast group ID is also known as 'joining' a multicast group. Upon session termination, the client's host system should issue an 'unjoin' message. This will terminate delivery of data to that host's local network. If a client application terminates without issuing an 'unjoin' message, the network will eventually issue a 'timeout' for the multicast group subscription that will automatically terminate delivery of the Multicast packets to the host's local network.

The 'join' and 'unjoin' processes are standard functions. No specific instructions are provided here, as they are specific to the user's operating system and programming language.

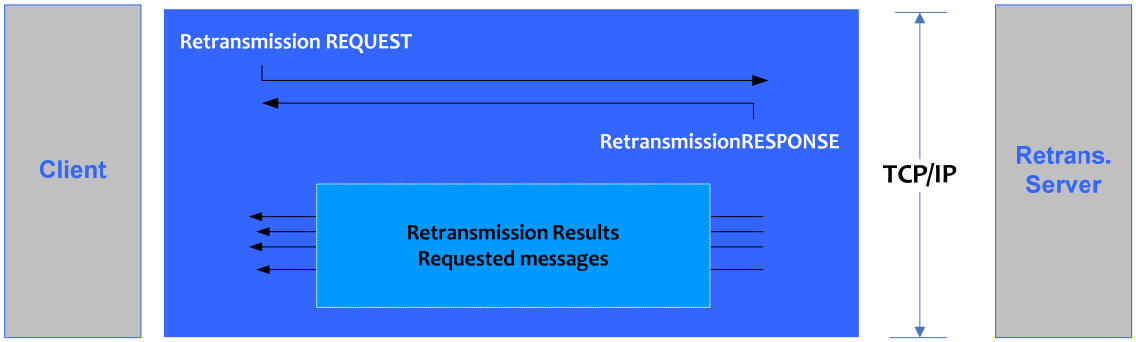
1.2.2. Retransmission Functionality

The retransmission functionality is designed to allow the user to recapture a small number of missed packets.

It is not intended that clients use the retransmission functionality to recover data after long outages or on late start up. Accordingly, the number of packets that the user can request is limited. The number of retransmission requests permitted per user is also limited per day, in order to avoid over-usage of this function that would provoke an overload of the network and the XDP servers.

The client makes a TCP/IP connection with the retransmission server, and receives the requested messages also via the same TCP/IP channel.

The following diagram shows the sequence of messages and the transport protocols employed when making a retransmission request.



The retransmission request will include a Source ID (username) which will be validated by the exchange system. It is important to note that only one Source ID can be used per application session.

The retransmission request may be rejected for any of the following reasons:

- Invalid Source ID;
- Invalid packet sequence number;
- Source ID currently used;
- PSN no longer in cache;
- Total number of packets requested in the current day exceeds the predefined system limit; [1]
- Number of retransmission requests in the current day exceeds the predefined system limit. [2]

[1] For Euronext Cash Markets, this limit is set today at 1,000

[2] For Euronext Cash Markets, this limit is set today at 1,000

In the case of such a failure, the user will receive an error message to advise of the reason for failure.

1.2.3. Refresh Functionality

The Refresh Server supplies (on demand) a snapshot including reference data, last trade price, high, low and the order book.

The refresh messages are compressed using the ZliB compression format, which delivers a significant reduction in network latency.
ZliB is part of the Opensource program, compression and decompression libraries are available on Internet.

The refresh server is designed to allow the user to update the market state within their applications before restarting in real-time, following a data outage or late start.

The client makes a TCP/IP connection to the Refresh Server for requesting the refresh, whilst also joining the refresh multicast channels for receiving the refresh messages.

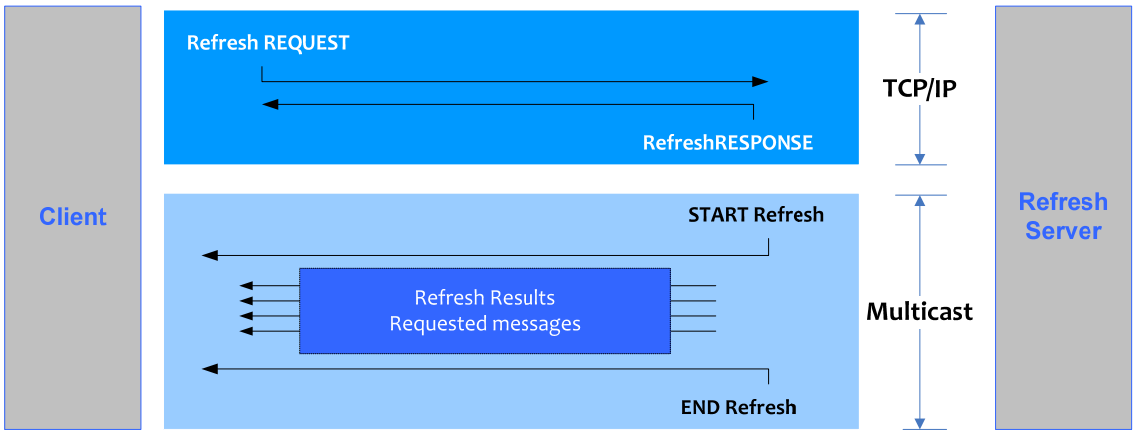
The refresh server will respond to a request with a Refresh Response message to indicate whether the request was accepted or rejected.

The refresh messages will then be sent on the multicast channels. This will be preceded by a Start of Refresh message and followed by an End of Refresh message. No dedicated retransmission service is available for the refresh, if packet loss is detected, clients should submit another refresh request.

The refresh request should include a Source ID (username) which will be validated by the exchange system. It is important to note that only one Source ID can be used per application session.

A pair of refresh multicast channels will be provided for each corresponding real-time service. The contents of the refresh and message formats will correspond to the contents and message formats contained in the appropriate real-time service.

The following diagram shows the sequence of messages and the transport protocols employed when making a refresh request.



The refresh request may be rejected for any of the following reasons:

- Invalid Source ID;
- Invalid Service ID;
- Incorrectly formatted message sent;
- Incorrect packet type sent;
- Total number of refreshes requested in the current day exceeds the predefined system limit; [3]

[3] For Euronext Cash Markets, this limit is set today at 1,000

- Rejected due to unavailability of refresh data [4]

In the case of such a failure, the user will receive an error message to advise of the reason for failure.

1.3. *Processing Guidelines*

1.3.1. General Processing Notes

The following processing notes apply to all messages:

- All fields will be sent for every message;
- Only field values will appear in the published messages (e.g., no names or 'tags' will appear in the message);
- The field names that appear in the message format documents are for reference purposes only;
- All the fields are contiguous, with reserved fields for alignment issues;
- All field sizes are fixed and constant;
- Binary fields are provided in network byte order (Big Endian format);
- ASCII string fields are left aligned and null padded;
- Segmentation of messages across packets will not be supported. This means a message will never straddle a packet boundary.

1.3.2. Packet Structure

All packets of data sent on the Universal Trading Platform Market Data Feed will have a common packet header followed by one or more messages (with the exception of some technical messages that do not contain any messages).

The packet header format is the same for all packets, and contains packet length, number of messages within the packet, packet sequence number, etc.

The format of each message in the packet depends on message type, but each message will start with message size and message type.

The maximum length of a packet is 1400 bytes.

A packet will only ever contain complete messages. A single message will never straddle multiple packets.

The message size will never exceed the maximum packet length (less the packet header size).

Packet Header	Message 1	Message 2	…	Message n

The packet header provides information including the total packet length, a packet sequence number, the number of messages within the packet and a send timestamp. The format is as follows:

Packet header				
Field	**Offset**	**Size (Bytes)**	**Format**	**Description**
PacketLength	0	2	Binary Integer	Length of the packet (in number of bytes) including the 16-byte packet header.
PacketType	2	2	Binary Integer	Identifier for the type of data contained in the packet.
PacketSeqNum	4	4	Binary Integer	Packet sequence number, unique per broadcast stream (=multicast group) (except if reset during the day)
SendTime	8	4	Binary Integer	SourceTime in milliseconds indicating the packet broadcast time. The number represents the number of milliseconds since the previous Sunday 00:00 UTC.

[4] Such event is very unlikely to happen, and would take place only if both Primary and Secondary XDP servers would have lost part of their contectual information. In this event, a reinitialization from the Disaster Recovery XDP server would be triggered by the Exchange.

Packet header				
Field	**Offset**	**Size (Bytes)**	**Format**	**Description**
				Indicates the time the message was sent by the market data application.
ServiceID	12	2	Binary Integer	Numeric value identifying the multicast group.
DeliveryFlag	14	1	Binary Integer	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Integer	The number of messages that are contained within the packet.

The format of each message within a packet will vary according to message type.

However, regardless of the message type, each message will start with a 2-bytes message length followed by a 2-byte message type.

Message				
Field	**Offset**	**Size (Bytes)**	**Format**	**Description**
MsgSize	-	2	Binary Integer	Size of the message in bytes, excluding these two bytes
MsgType	-	2	Binary Integer	Type of message. Refer to message specification section of this document for details.

1.3.3. Sequence Numbers

All messages conform to the line level sequencing. Each channel (a.k.a. multicast group) has its own packet sequence number. Clients can use packet sequence numbers to determine the following:

- Missing (gapped) packets
- Unordered packets
- Duplicate packets

1.3.4. Detecting and Recovering Missed Data

UDP is an 'unreliable' protocol and therefore may drop packets from line A and line B.
The Universal Trading Platform Market Data Feed provides 3 different mechanisms for recovering missed data:

- Line arbitration – using dual multicast channels
- Retransmission server – recovery of a limited number of packets
- Refresh server – snapshot of current market state

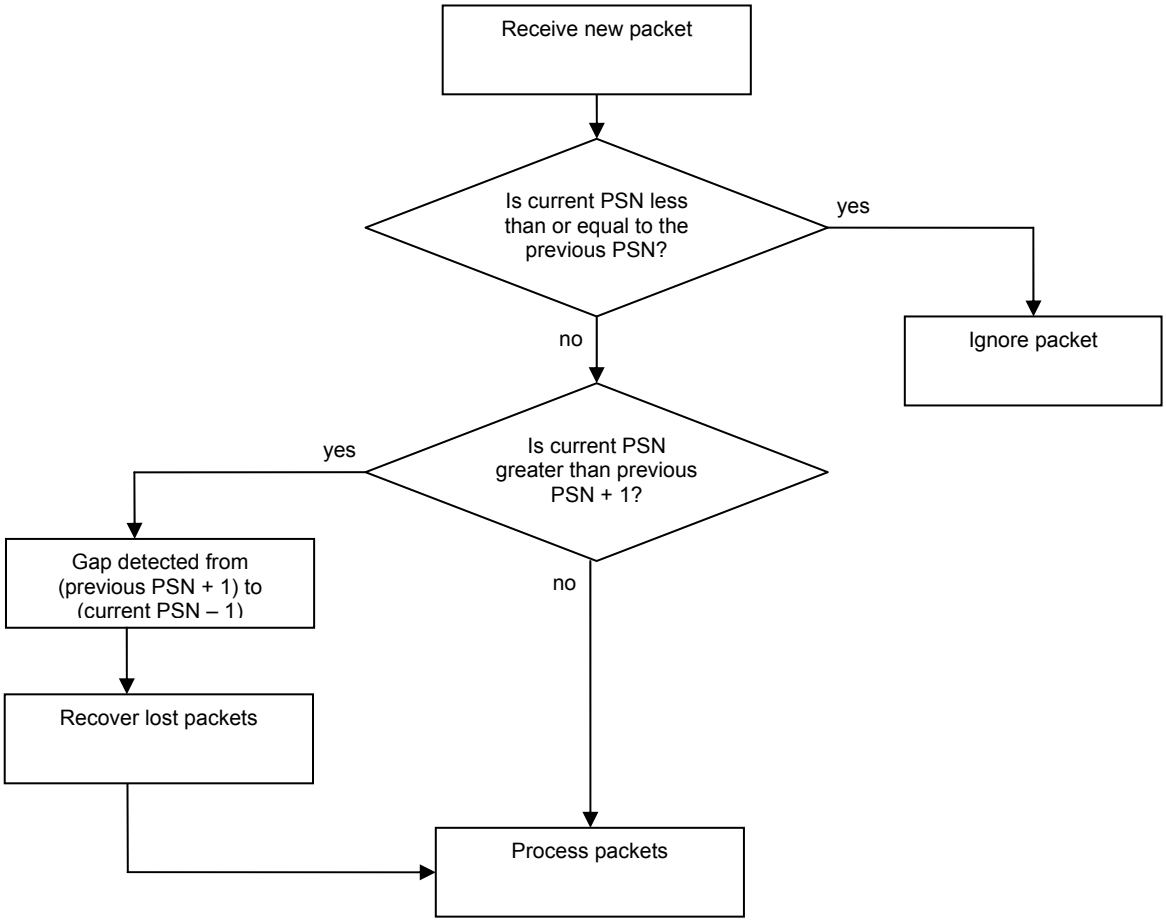
These mechanisms should be used as follows:

Event	Action
Packet lost on one of the two lines	Try to recover data from the other line with a configurable timeout
Dropped packet(s) on both line A and line B	Recover dropped packet(s) from retransmission server
Late start up or extended intraday outage	Request a refresh of the current market state and then continue with real time messages

1.3.4.1. Gap Detection

Each packet has a Packet Sequence Number (PSN). PSNs start at one (1) and increase one by one and without gaps with each subsequent packet. Users should use the PSN to detect gaps in the transmission of messages.

The following diagram illustrates how the PSN should be used to detect gaps in the feed.

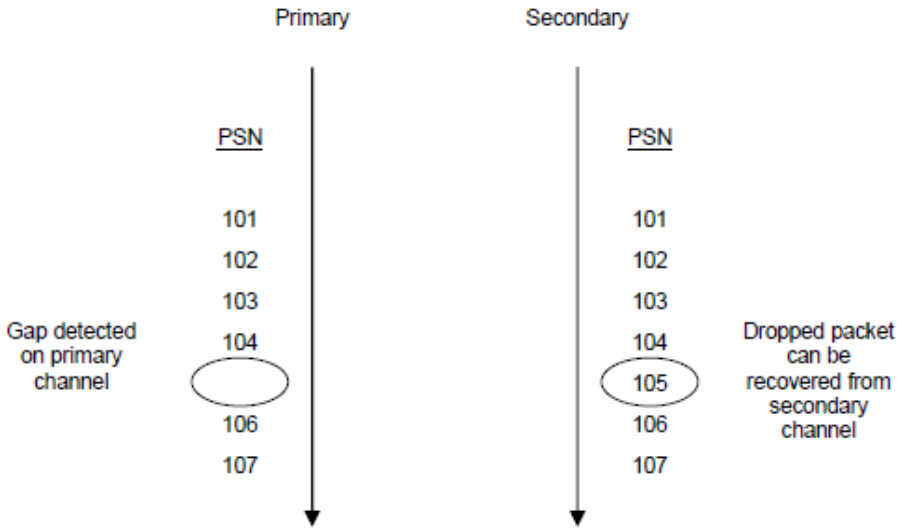


1.3.4.2. Line Arbitration

Client applications should check the Packet Sequence Number (PSN) for every packet received.
PSNs are unique and increase monotonically (i.e. one by one) for each service.
Line A and line B are identical in terms of:

- Packet contents
- PSNs
- Sequence in which packets are sent

Client applications should listen to both channels in real-time. Clients should look at packets coming from both lines and process the ones that arrive first, regardless of whether they came from line A or line B. It is advisable to apply the 'first come – first served' rule.



1.3.4.3. Retransmission Server

If a packet is lost from both line A and line B, clients then make a TCP/IP request to have the packets resent. Packets are resent from the Retransmission Server.

After a client establishes a TCP/IP connection, the Retransmission Server will periodically send heartbeat request messages to the client. Clients must respond to this request with a heartbeat response within a specific timeframe – otherwise, the Retransmission Server will close the connection.

Note that it is upon the Client to decide if he prefers to maintain a TCP/IP connection for Retransmission at all times, or if he prefers to establish it only when he needs it.

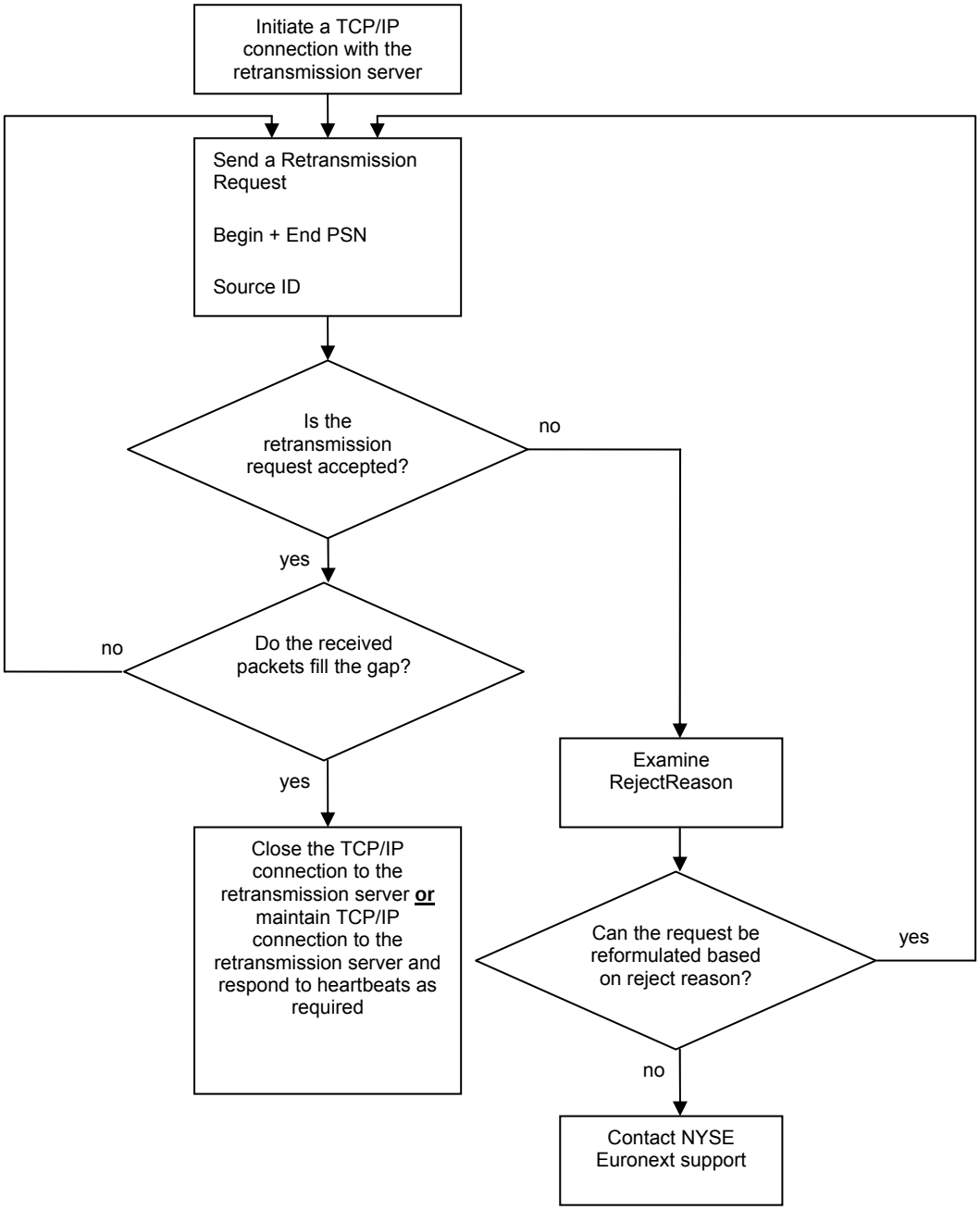
The client makes a TCP/IP connection to the Retransmission Server for both requesting and receiving retransmitted packets.

Retransmission requests should contain a Start PSN, an End PSN and a Source ID. The Source ID identifies the client application, and will be supplied by the exchange. The request can be rejected for a number of reasons as defined in section 2.7.

The number of retransmissions allowed per client per day is limited (See chapter 4 and 5 for detailed content of Production and External User Acceptance retransmission server limitations, regarding the Euronext implementation).

The length of each retransmission is limited to a pre-defined number of packets. In case the information to retransmit is too large, it is suggested to use the Refresh service.

The following diagram illustrates the process of requesting dropped packets from the retransmission server:



1.3.4.4. Refresh Server

If a client starts their application late, or experiences an outage, the refresh server should be used to provide the means to get back in synchronization with the real time market. Clients send and receive the refresh request and response messages over a TCP/IP connection, and join the refresh multicast groups to receive the content of the refresh.

After a client establishes a TCP/IP connection, the Refresh Server will periodically send heartbeat request messages to the client. Clients must respond to this request with a heartbeat response within a specific timeframe – otherwise, the refresh server will close the connection.

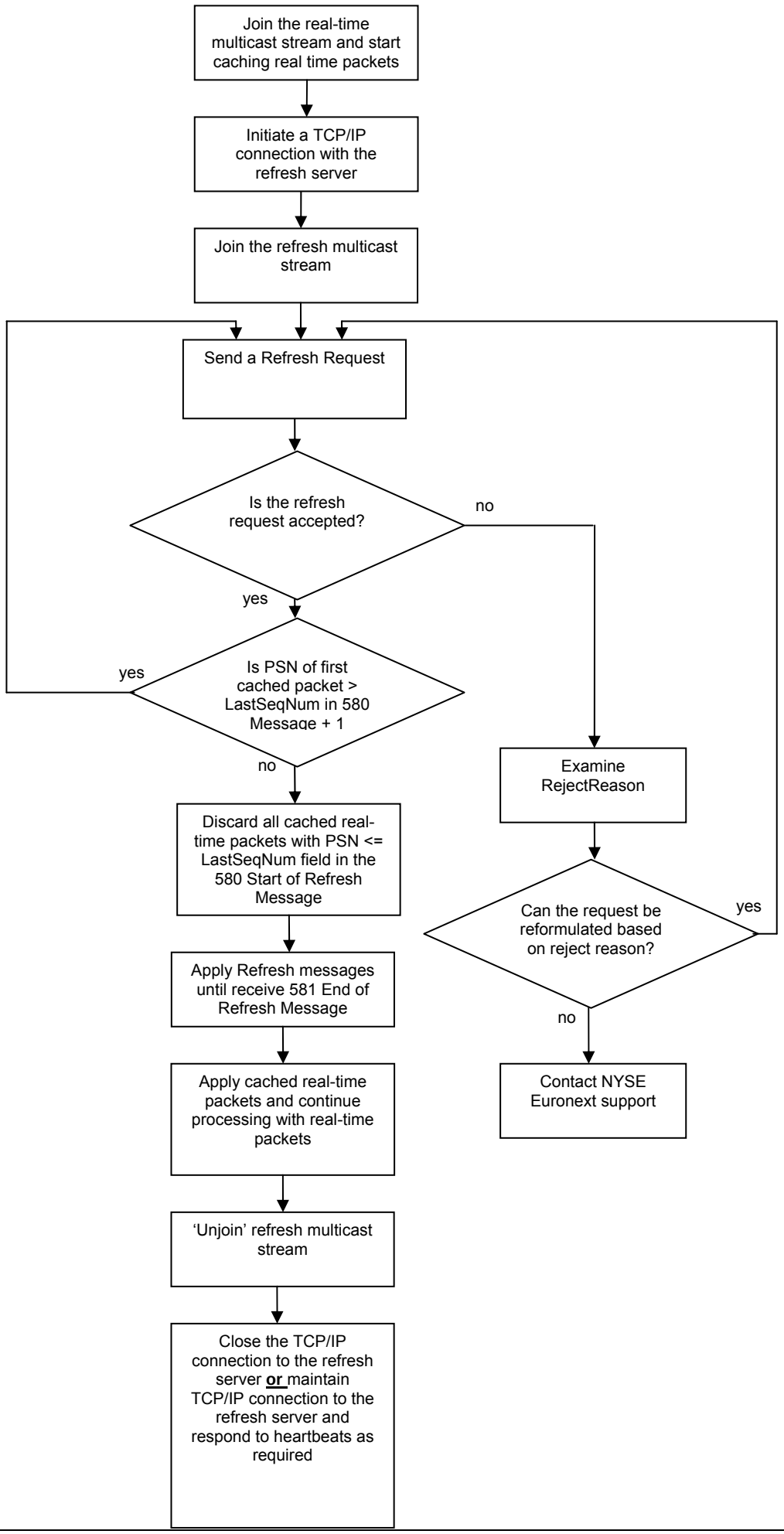
The client makes a TCP/IP connection to the Refresh Server to request a refresh of packets, and at the same time join the multicast refresh groups. Refresh requests should contain a Service ID and a Source ID. The Service ID will define for which multicast channel a refresh is required. The Source ID identifies the client application, and will be supplied by the exchange. This will be identical to the Source IDs allocated for the

retransmission server. The request can be rejected for a number of reasons as defined in the refresh response message (section 2.9).

Once a successful request has been received by the server, a refresh response will be sent to the client. Clients should then process the refresh content from the refresh multicast groups and synchronize this with the real time data.

The number of refresh's allowed per client per day is limited (See chapter 4.9 for detailed content of the refresh server limitations regarding the Euronext implementation).

The following diagram illustrates the process of requests from the refresh server:


Join the real-time multicast stream and start caching real time packets
Initiate a TCP/IP connection with the refresh server
Join the refresh multicast stream
Send a Refresh Request
Is the refresh request accepted?
no
yes
Is PSN of first cached packet > LastSeqNum in 580 Message + 1
yes
no
Examine RejectReason
Can the request be reformulated based on reject reason?
yes
no
Contact NYSE Euronext support
Discard all cached real-time packets with PSN <= LastSeqNum field in the 580 Start of Refresh Message
Apply Refresh messages until receive 581 End of Refresh Message
Apply cached real-time packets and continue processing with real-time packets
'Unjoin' refresh multicast stream
Close the TCP/IP connection to the refresh server **or** maintain TCP/IP connection to the refresh server and respond to heartbeats as required

1.4. *Operational Information*

Measures are in place to safeguard against unexpected system failures.

1.4.1. Exchange System Failure

1.4.1.1. Dual Multicast Lines

Under normal operating conditions, the exchange system will send real-time messages to two unique multicast addresses. This provides clients with two redundant data feeds. The client application should be designed to handle the loss of one of the two multicast channels without any interruption to service.

1.4.1.2. High Availability

The High Availability (HA) functionality of the market data publisher is set up to ensure there is no loss of service for clients if there is any kind of outage in the exchange on the primary publisher, for example a hardware failure. The failover to the secondary publisher will occur without any gap in market data packet sequence numbers. The HA failover has been designed to be as transparent as possible for clients, as the connectivity in terms of multicast groups and ports will not change. However, clients should note that there are specific technical details that should be considered:

- For details of retransmissions and refresh behavior that should be included as part of application logic, please refer to sections 4.6.3, 4.6.4, 5.6.3 and 5.6.4.

1.4.1.3. Disaster Recovery Site

In order to mitigate any serious outage in the primary data centre, a secondary data centre is online in standby mode in the secondary data centre, in case of a serious incident.

Clients should ensure all configuration surrounding the secondary data centre is included as detailed in chapter 4 (in the case of Euronext implementation).

1.4.2. Client System Failure

Real-time market data is made available on two different multicast groups. This offers clients the possibility to set up more than one receiving system processing the same data. In the event of a client system failure, the backup client system should continue to process the real-time data sent on the second multicast group.

1.4.3. Gap Detection

The Universal Trading Platform Market Data Feed provides a unique, sequential packet sequence number for each multicast channel. This will allow recipients to identify ‘gaps’ in the message sequence and, if appropriate, reconcile them ‘locally’ with an alternate channel or request retransmission of the missing/corrupted data packet.

Refer to Gap Detection for more details on gap detection (Chapter 1.3.4.1).

1.5. *General Processing Notes*

1.5.1. Environments

This specification provides information on how to develop applications to subscribe to Euronext and/or Qatar Exchange Cash market data. A number of environments are available for clients, all of which have various data products separated into different multicast channels. The environments available, and the different multicast groups available in each (known as Service IDs) are defined in appendix F:

Channel definitions for each of the above can be found in chapter 4 (Production Feed Configuration) and chapter 5 (External User Acceptance Feed Configuration), in the case of Euronext implementation.

1.5.2. Multicast Streams

Dual multicast streams are made available for the distribution of real-time and refresh data.

Data is provided across multiple multicast streams. Users should refer to chapters 4 and 5 for information on what data is carried in each multicast group (in the case of Euronext implementation).

Clients should connect to multicast stream(s) for which they require data.

1.5.3. TCP/IP Channels

TCP/IP channels are made available for retransmission and refresh requests and responses.

The user can choose to disconnect/reconnect in between requests. However if choosing to remain connected, the user will need to respond to heartbeat requests from the exchange.

1.5.4. Date and Time Conventions

Dates and Times use UTC (Universal Time, Coordinated).

The base for timestamps in Packet headers is the number of milliseconds since the previous Sunday 00:00:00.000 UTC (so in the night from Saturday to Sunday).

The base for timestamps in Message bodies is the number of milliseconds since previous midnight 00:00:00.000 UTC.

For example Wednesday 15:30:00.000 UTC is indicated as 315000000 in a Packet Header or 55800000 in a message body.

1.5.5. Sequence Numbers

The feed contains two sequence numbers:

- The packet sequence number is part of the packet header, and should be used for retransmission requests. It is unique per service and common across a pair of dual multicast streams. Note that the packet sequence number is only unique for market data packets, heartbeats use the PSN of the last packet.

- The source sequence number is assigned by the source system to this message. Whilst this sequence number increases serially, it does not increase one by one.

1.5.6. Price Formats

Prices in the feed are represented by two fields, an integer value and a scale code. All prices in the feed share a common scale code, which is represented in the PriceScaleCode.

The value should be calculated using the following formula:

$$Value = \frac{Integer}{10^{ScaleCode}}$$

For example, a price of 27.56 is represented by an Integer of 2756 and a PriceScaleCode of 2.

1.5.7. Data Types

All “Binary Int.” formatted fields are numeric unsigned binary. All “Binary Int. (signed)” formatted fields are signed binary integer. Binary data is in network byte order (Big Endian).

All “ASCII Str.” and “ASCII Ch.” fields are alphanumeric, left justified and null padded.

1.5.8. Instrument Identifiers

An instrument is identified by its SymbolIndex, across all feeds that relate to that instrument. The SymbolIndex is arbitrarily assigned by the feed, and will not change for the lifetime of the instrument.

The SymbolIndex can take a different value for the same instrument depending on the environment (Production or Test).

Standard security identifiers (for example ISIN, Euronext and QE Trading Code) can be found in the 555 Reference Data message (as detailed in the Market Information Appendix).

2. Technical Message Specifications

2.1. *Introduction*

There are two types of messages transmitted as part of this protocol: control and data. Control messages do not contain data, they allow conversing parties to exchange session-specific information (e.g., 'reset sequence number'). Data messages are product specific.

2.2. *Packet Header Format*

All messages will contain a common packet header. The table below describes the header field. The design is intended to minimize the development burden on behalf for clients. Meaning that, all clients may implement line-level protocol processing once, and then only need develop parsing algorithms for their choice of message.

Field	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet (in bytes) including the 16-byte packet header
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. 1 - (PSN) Sequence Number Reset 2 - Heartbeat Message 10 - Retransmission Response message 20 - Retransmission Request Message 22 - Refresh Request message 23 - Refresh response message 24 - Heartbeat Response Message
PacketSeqNum	4	4	Binary Int.	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in milliseconds indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC.
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream (multicast group). Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

2.3. *Packet Sequence Number Reset*

This message is sent to 'reset' the Packet Sequence Number at start of day, in response to failures, etc. Note that this message will contain a valid sequence number. The message format is shown below.

2.3.1. Packet Sequence Number Reset Processing Notes

Packet Sequence numbers normally begin at one (1) and increase one by one with each subsequent packet. There are two scenarios where the packet sequence number is reset (besides the start of day). Firstly, if the value should exceed the maximum value that the SeqNum field may contain, it will be reset to one (1). Secondly, if the system fails and it recovers, it sends a Packet Sequence Number reset message. The PacketSeqNum field of that packet will be set to one (1) and the NextSeqNumber field will be set to two (2).

Note that the packet sequence number reset is always sent as the first message of the day.

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Sequence Number Reset*				
NextSeqNumber	16	4	Binary Int.	Contains the packet sequence number value that the client should expect in the immediately succeeding data packet. Note that this packet will contain its own valid packet sequence number in the header portion of the message.

2.4. *Heartbeat*

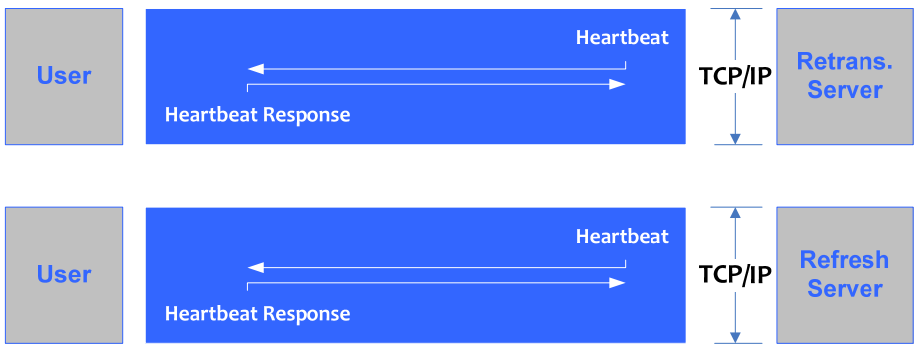
Heartbeat messages are sent in the multicast streams as well as in the active TCP/IP retransmission and refresh sessions.

2.4.1. General Heartbeat Processing Notes (TCP and Multicast)

- This applies to the TCP channels for retransmissions and refresh, and also the multicast channels for real time and refresh data.
- Heartbeat messages will only contain the packet header (with PacketType = '2'). The packet will not contain a message body.
- Heartbeats will only be sent on the multicast channels when there is no market data. Heartbeat frequency since the last packet, is:
    - 2 seconds in the multicast streams.
    - 30 seconds in the active TCP/IP retransmission sessions.

2.4.2. Retransmission and Refresh Heartbeat Processing Notes (TCP)

- Clients may receive a heartbeat message if they have an active TCP/IP session with the retransmission or refresh server.
- To determine the health of the user connection on the TCP/IP channel, the retransmission or refresh server will send regular heartbeat messages to the user. The heartbeat frequency is 30 seconds The time out for this heartbeat response message is set at 5 seconds. If no response is received by the server within this timeframe, the TCP/IP session will be disconnected.
- Clients that choose to establish and remain connected to the retransmission or refresh server intraday must respond to a heartbeat message with a heartbeat response message. Users can choose to either disconnect following each retransmission or refresh request, or remain connected to the retransmission or refresh server.



2.5. *Heartbeat Response*

Clients that choose to establish and remain connected to the retransmission server intraday, must respond to a heartbeat message with a heartbeat response message.

Note that the fields in the packet header should be filled as follows:

PacketLength = 36

PacketType = 24

PacketSeqNum = optional

SendTime = optional

ServiceID = optional

DeliveryFlag = 0

NumberMsgEntries = 1 (only 1 heartbeat response message should be sent per packet)

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Heartbeat response*				
SourceID	16	20	ASCII Str.	This field represents the Identifier of the source (client) requesting retransmission. Field is null padded, left aligned.

2.6. *Retransmission Request*

This message is sent by clients requesting missing messages identified by a sequence number gap. Upon receipt of a valid retransmission request message, the requested message(s) will be sent. The requested message(s) have the same message format and content as the original sent by the system.

Note that the fields in the packet header should be filled as follows:

PacketLength = 44

PacketType = 20

PacketSeqNum = optional

SendTime = optional

ServiceID = Service ID of the broadcast stream corresponding to the request, in other words the broadcast stream for which messages need to be recovered by the client.

DeliveryFlag = 0

NumberMsgEntries = 1 (only 1 retransmission request should be sent per packet)

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Retransmission Request*				
BeginSeqNum	16	4	Binary Int.	Begin Sequence Number of the requested range of messages to be retransmitted. Note the Sequence Number refers to the PacketSeqNum in the header. **Remark**: The broadcast stream from which a retransmission is requested has to be stated in the field ServiceID in the Packet header of the RetransmissionRequest message.
EndSeqNum	20	4	Binary Int.	End Sequence Number of the requested range of messages to be retransmitted. Note the Sequence Number refers to the PacketSeqNum in the header. **Remark**: The broadcast stream from which a retransmission is requested has to be stated in the field ServiceID in the Packet header of the

Field	Offset	Size (Bytes)	Format	Description
				RetransmissionRequest message.
SourceID	24	20	ASCII Str.	This field represents the Identifier of the source (client) requesting retransmission. Source-ID is pre-set by the Exchange and subject to validation. Field is null padded, left aligned.

2.7. *Retransmission Response*

This message will be sent immediately via TCP/IP in response to the client's request for retransmission messages.

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Retransmission Response*				
SourceSeqNum	16	4	Binary Int.	This field contains the request message sequence number assigned by the client. It is used by the client to couple the request with the response message.
SourceID	20	20	ASCII Str.	This field represents the Identifier of the source (client) requesting retransmission. Field is null padded, left aligned.
Status	40	1	ASCII Str.	Indicates whether the retransmission request was accepted or rejected. Valid values: 'A' - Accepted. 'R' - Rejected.
RejectReason	41	1	Binary Int.	Indicates the reason for the rejection. Valid values: 0 Message was accepted 1 Rejected due to permissions (the ServiceID is not granted for the SourceID or a connection is already open for this SourceID) 2 Rejected due to invalid sequence range 3 Rejected due to max sequence range reached (> thresholds) 4 Rejected due to max request reached in a day (> thresholds) 5 Rejected - Requested packets are no longer available 6 Rejected - Retransmission request incorrectly formatted
Filler	*42*	*2*	*ASCII Str.*	*For future use.*

2.8. *Refresh Request*

This message is sent by clients requesting a refresh. The system will provide the appropriate message(s) in response.

Note that the fields in the standard packet header should be filled as follows:

PacketLength = 36

PacketType = 22

PacketSeqNum = optional

SendTime = optional

ServiceID = Service ID of the broadcast stream corresponding to the request (the stream for which the refresh will be applied).

DeliveryFlag = ignored in the request.

NumberMsgEntries = 1

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Refresh Request Messaget*				
SourceID	16	20	Ascii Str..	Identifier of the Source ID requesting the refresh.

2.9. *Refresh Response*

This message will be sent immediately via TCP/IP in response to the client's request for a refresh.

PacketLength = 44

PacketType = 23

PacketSeqNum = Contains the packet sequence number if sent in the refresh request

SendTime = Ignored in the response.

ServiceID = Service ID of the broadcast stream corresponding to the request (the stream for which the refresh will be applied). This will be the same value as in the initial request.

DeliveryFlag = Ignored in the response.

NumberMsgEntries = 1

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Refresh Response Message*				
SourceSeqNum	16	4	Binary Int	This field contains the request *packet sequence number* assigned by the client. It is used by the client to couple the request with the response message.
SourceID	20	20	Ascii Str..	Identifier of the source requesting the refresh.
Status	40	1	Ascii Str..	Indicates if the Refresh request has been accepted. Valid values: 'A' for accepted, 'R' for rejected.
RejectReason	41	1	Binary Int.	Indicates the reason for rejection. Valid Values: 0 – Message was accepted, 1 – Request rejected due to permissions (the ServiceID is not granted for the SourceID or a connection is already open for this SourceID), 2 – Request rejected due to incorrect ServiceID, 3 – Refresh request incorrectly formatted, 4 – Request rejected due to incorrect packet type sent, 5 – Rejected due to max request reached in a day (> thresholds) 6 – Request rejected due to unavailability of refresh data, eg: - Requesting reference data refresh before the real time reference data has been fully broadcast. - Requesting orderbook refresh before the morning market sheet retransmission. 7 – Refresh request rejected as sent to incorrect server (secondary instead of primary).
Filler	*42*	*2*	*Ascii Str.*	*For future use*

2.10. *Refresh Messages*

2.10.1. Refresh Compression

The refresh messages disseminated over the refresh multicast channels are compressed using the ZliB compression format. As mentioned before this method of compression considerably reduces network latency.

- The complete specifications for the ZliB compression format can be found here:

- http://www.zlib.net/zlib_docs.html

- Clients should use the DeliveryFlag field in the packet header to determine if a packet is ZliB compressed or not.

- All messages delivered over the refresh multicast channels will be compressed. The messages delivered over the real time multicast channels will not be compressed.

2.10.2. Refresh Packet Type

In order to allow refresh data to consist of different message types, defined by different generic packet types, a new packet type has been created (999). For example:

- The 555 reference data message will be sent within a generic market information packet type 995. A trade full information 240 message will be sent within a generic trade message packet 998. In order to combine these two message types within a single multicast stream for refresh, the new packet type below has been defined.

Field Name	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet including the 16-byte packet header.
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. Only generic packet types are used, and if a packet contains different generic packet types, then the packet type 999 is used. Possible values: 999 – Refresh combined packet type 994 – Generic Quotes *all current quotes (5/10BBO's)* 995 – Generic Market Information *Current Data dictionary – stocks & indices* 996 – Generic Indices *Current indices values* 998 – Generic Trades *Last trades for each instrument* 230 – Generic Orderbook *Current snapshot of the order book*
PacketSeqNum	4	4	Binary Int.	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and monotonically and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in millisecond indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream. Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

2.10.3. Start Refresh – 580 Message

A refresh cycle begins with a *Start Refresh* message and ends with an *End Refresh* message on the multicast channels.

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *Start Refresh Message*				
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	580 – Start Refresh Message
LastSeqNum	20	4	Binary Int.	Contains the last cached PacketSeqNum that the Refresh is valid to.

2.10.4. End Refresh – 581 Message

A refresh cycle begins with a *Start Refresh* message and ends with an *End Refresh* message on the multicast channels.

Field	Offset	Size (Bytes)	Format	Description
Defined below are the 'body' fields of the *End Refresh Message*				
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	581 – End Refresh Message
LastSeqNum	20	4	Binary Int.	Contains the last cached PacketSeqNum that the Refresh is valid to.

3. Real Time Message Specifications

3.1. *Market Information*

3.1.1. Overview

The Equities Market Information service uses the push-based publishing model. This means that data will be published based on its availability. Once information is available, it will be published to Equities Market Information clients. The Equities Market Information message reflects the last information for each NYSE Euronext traded security and information on the market itself.

List of the message types in the Equities Market Information Feed:

- 505 - Stock state change message
- 513 - Euro & Interbank rates
  Not used for Qatar Exchange
- 516 - Class state change
- 523 - Mail
- 530 - IMP (Indicative Matching Price)
- 531 - Market imbalance
  Not disseminated – For future use
- 534 - Authorized out of session limits
  Only used with TCS (Euronext). Not used for Qatar Exchange
- 535 - TCS State change
  Only used with TCS (Euronext). Not used for Qatar Exchange
- 537 - Collars
- 539 - Session Timetable
- 540 - Display bid and ask
  Only used with NSC (Euronext). Not used for Qatar Exchange
- 541 - Daily Summary
  Not disseminated – For future use
- 545 – Global Market Activity Summary
  Only used with Qatar Exchange
- 550 - Start referential
- 551 - End referential
- 555 - Referential

3.1.2. Packet Header Format

All messages are preceded by a common packet header format. The following table describes the header fields of a Equities Market Information message.

Field Name	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet including the 16-byte packet header.
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. If all messages within a packet are of the same message type, the packet type will be equal to that message type. If not, the packet type can be set at 995. Possible values: 995 - Generic Market Information Message. 505 - Stock State Change message - 68 Bytes 513 - Euro & Interbank rates - 56 Bytes 516 - Class State Change - 44 Bytes 523 - Mail - 984 Bytes 530 - IMP - 52 Bytes 531 - Market Imbalance - 64 Bytes 534 - Authorized Out of Session Limits - 56 Bytes 535 - TCS State Change - 40 Bytes 537 - Collars - 48 Bytes 539 - Session Timetable - 100 Bytes 540 - Display Bid an ask - 44 Bytes 541 - Daily Summary - 84 Bytes 545 – Global Market Activity Summary – 42 Bytes 550 - Start Referential - 24 Bytes 551 - End Referential - 24 Bytes 555 - Referential - 509 Bytes
PacketSeqNum	4	4	Binary Int.	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and monotonically and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in millisecond indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream. Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

3.1.3. Stock State Change - 505 Message

3.1.3.1. Message Overview

A 'Stock State Change' message is generated by the trading engines to announce a change in the state of an instrument.

3.1.3.2. Message Sending Rules

This message is sent for tradable instruments:

- When an instrument has been halted.
- When an instrument has been unhalted.
- When an opening or an opening auction is programmed
- When a programmed opening or opening auction is cancelled
- Prior to the trading session, in order to carry over states of instruments halted the day before.

This is sent for indices:

- When the broadcasting of an index is temporary suspended. In such a case, the index level code is set to 3 (real-time forerunner), instead of 2 (real-time index level).
  Not used in this case for Qatar Exchange

3.1.3.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '505' Stock State Change.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	505 – Stock State Change
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SystemID	32	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	36	2	Binary Int.	Number of micro seconds to be combined with SourceTime
Filler	*38*	*2*	*Binary Int.*	*For future use.*
StartDateHalting	40	8	ASCII Str.	Date when suspension of trading started for an instrument. Valid values: YYYYMMDD 00000000 if not provided
StartTimeHalting	48	6	ASCII Str.	Time when suspension of trading started for an instrument. Valid values: HHMMSS 000000 if not provided
ProgOpeningTime	54	6	ASCII Str.	Instrument opening time that has been programmed by the Cash Market Operations. Valid values: HHMMSS 000000 if not provided.

OrderEntryRejection	60	1	ASCII CH.	Indicates whether order entry is allowed or forbidden. Valid values: '0' Not applicable • For instruments traded on NSC: 'I' Inhibited 'A' Authorized • For instruments traded on UTP: 'N' Order entry allowed 'Y' Order entry forbidden Null Not provided
InstrumentState	61	1	ASCII Ch.	Indicates the state of the instrument. Valid values: '0' Not applicable • For instruments traded on NSC: 'S' Suspended 'R' Reserved 'G' Frozen Null Open or not provided • For instruments traded on UTP: 'A' Auction 'H' Halted 'C' Closed Null (or space) Inherited (following the state of the class the instrument belongs to) Remark: There is a relation between InstrumentGroupState/Class (Msg 516) and InstrumentState: in case the state of the Group/Class to which the instrument belongs differs from the state of that individual instrument, the most restrictive value supersedes: If the Instrument State shows "tradable" but its class/group is declared "not tradable" (e.g. Interrupted, Forbidden), effectively trading is not allowed in this instrument.
InstrumentTradingStatus	62	1	ASCII Ch.	Indicates whether trading on an instrument is suspended, halted or resumed: Valid values: Null Not provided • For instruments traded on NSC: 'S' Suspended 'R' Trading resumed after security was reserved/suspended 'H' Reserved, price beyond upper limit 'B' Reserved, beyond lower limit 'P' Simple reservation • For instruments traded on UTP: 'S' Suspended • For Indices: 'S' Calculation Suspended 'R' Calculation Resumed

HaltReason	63	1	ASCII Ch.	Indicates the origin of halting for an instrument Valid values: '0' Not applicable • For instruments traded on NSC: 'A' Automatic (The instrument was reserved automatically because the price was beyond the upper or lower limit) 'M' Manual (The instrument was reserved manually by the Market Operations) Null Not provided • For instruments traded on UTP: 'R' Halted. No liquidity provider 'C' Opening price outside collars 'M' Manuel halting by Market Operations Null or space Instrument not halted or information not available
ActionAffectingState	64	1	ASCII Ch.	Code indicating the event that caused a change in the instrument state. Valid values: '0' Not applicable • For instruments traded on NSC: 'P' Deferred opening or auction has been programmed for the instrument 'M' Instrument reserved or halted manually by the Cash Market Operations 'C' Instrument is currently being traded 'O' Instrument is now open (after R or S) 'R' Instrument has been automatically reserved after opening of its group 'D' Deferred opening or auction has been cancelled 'A' Order entry has been authorized for the instrument 'I' Order entry has been forbidden for the instrument 'E' Orders in the book have been eliminated 'N' Instrument is being initialized (beginning of the trading day) 'F' Instruments in Fast-Market state. 'S' Return to normal activity (Slow-Market state). 'G' Instrument has been frozen or unfrozen. 'U' Modification of the state of a warrant due to a change of state of its underlying instruments 'W' Programming of a deferred opening for a warrant due to an action of the warrant robot 'X' Cancellation of a deferred opening of a warrant due to an action of the warrant robot 'Y' Beginning of a one side only period Z' End of a one side only period Null Not provided • For instruments traded on UTP: 'C' Trading on the instrument at the opening (sent before O) 'D' Cancelled programmed opening 'M' Instrument manually halted by Market Operations 'N' Instrument is being initialized (beginning of the trading day) 'O' Instrument opened 'P' Deferred programmed opening 'R' Automatic halting at the class auction
InstrumentStateTCS	65	1	ASCII Ch.	Indicates if trade declaration/reporting is authorized in TCS. Used only with TCS Valid values: '0' Not applicable 'S' Instrument is forbidden 'A' Instrument is authorized Null Not significant

PeriodSide	66	1	ASCII Ch.	This indicator indicates the side phase for an instrument being in a Bid only situation or Offer only situation. This indicator is only available on the NSC VW (NSC for Warrants) system. Valid values: '0' Not applicable 'A' Bid only period 'V' Offer only period Null Not significant
Filler	*67*	*1*	*Binary Int.*	*For future use.*

3.1.4. Euro & Interbank Rates - 513 Message

3.1.4.1. Message Overview

A Euro & Interbank Rates message informs the market participants about the interbank rate of the Euro currency in a foreign currency (valid for a day). The Euro Interbank Rate is the official rate published by the French National bank "Banque de France".

3.1.4.2. Message Sending Rules

This is sent once for each trading day and for each currency.

3.1.4.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '513' Euro & Interbank Rates.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	513 – Euro & interbank Rates
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
EuroInterbRate	32	4	Binary Int.	Official rate for the Euro, expressed in another currency(To be calculated with the RateScaleCode)
ChangeRate	36	4	Binary Int. (signed)	Percentage variation for Euro interbank rate compared to the previous known rate (previous day)(To be calculated with the ChangeScaleCode).
SystemID	40	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	44	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
TypeOfRate	46	2	Binary Int.	Nature of the rate. Valid values: 24, Interbank currency rate, always filled.
Filler	*48*	*2*	*Binary Int.*	*For future use.*
TickDirection	50	1	ASCII Ch.	Direction of the change in the Euro Interbank rate in relation with the previous known interbank rate (previous day). Valid values: '+' - Euro Interbank rate is Higher than the previous value '-' - Euro Interbank rate is Higher than the previous value '0' - Euro Interbank rate is the same as the previous value
Currency	51	3	ASCII Str.	Currency in which the relevant Euro Interbank currency rate data is given. (ISO3A- ISO 4217).
ChangeScaleCode	54	1	Binary Int.	To be combined with ChangeRate
RateScaleCode	55	1	Binary Int.	To be combined with EuroInterbRate

3.1.5. Class State Change - 516 Message

3.1.5.1. Message Overview

The Class State Change message indicates a change in the state of a class / instrument group.

There is a relationship between InstrumentGroupState / ClassState (516 message) and InstrumentState (505 message). In case the state of the Group/Class to which the instrument belongs differs from the state of that individual instrument, the most restrictive value supersedes the other. If the Instrument State shows “tradable” but its class/group is declared “not tradable” (e.g. Interrupted, Forbidden), effectively trading is not allowed in this instrument.

3.1.5.2. Message Sending Rules

This message is sent each time a class / instrument group changes state during the trading day.

3.1.5.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = ‘516’ Class State Change.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	516 – Class State Change
SourceSeqNum	20	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SystemID	28	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	32	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
InstrumentGroupCode	34	3	ASCII Str.	Instrument Group / Class Identification
InstrumentGroupState	37	1	ASCII Ch.	For NSC traded instrument-groups only. (for UTP traded classes check field Class State in this message) Instrument Group status. Valid Values: ‘P’ - Pre-Opening, ‘C’ - Beginning of day consultation, ‘O’ - Opening, ‘S’ - Market Session, ‘R’ - Trading at last, ‘N’ - Surveillance Intervention, ‘F’ - End of Day Inquiry, ‘Z’ - Interrupted, ‘I’ - Forbidden.
SessionType	38	1	ASCII Ch.	Market Session. Valid values: ‘E’ - Early ‘C’ - Core Session ‘L’ - Late
Filler	*39*	*1*	*Binary Int.*	*For future use.*

ClassState	40	4	ASCII CH.	For UTP traded instruments only. (For NSC traded instr.groups check the field InstrumentGroupState in this message) Class status representing the current trading market phase for instruments belonging to that class and whose status is inherited. 'Valid values: EAMO' Early Monitoring 'COCA' Core Call 'COAU' Core Auction 'COCO' Core Continuous 'CLCA' Closing Call 'CLAU' Closing Auction 'TAL' Trading At Last 'COMO' Core Monitoring 'LAMO' Late Monitoring 'HALT' Halted

3.1.6. Mail - 523 Message

3.1.6.1. Message Overview

In Euronext, this message will become obsolete once the regulated order books for all instruments have migrated from the NSC trading platform to the UTP trading platform (scheduled for Q4-2009).

This message supplies a part or a complete electronic message (E-mail), in order to inform members and rebroadcasting members about quotation conditions on a given listed instrument or a set of listed instruments or otherwise about a technical problem.

3.1.6.2. Message Sending Rules

This message is sent manually by Cash Market Operations to inform member firms about events of general interest that occurred in the market (suspension of securities, deletions of order books, new listings of securities, various technical messages...)

Note 1: A long mail message can be split into several pages. Information in the mail header enables users to rebuild the entire mail message

Note 2: For Qatar Exchange, this message may change – and "long mail" messages won't be available (only single page messages).

3.1.6.3. Message Structure

The table below describes the body fields of a Equities Market Information message, MsgType = '523' Mail.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	523 - Mail
SourceSeqNum	20	4	Binary Int.	1. This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SystemID	28	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	32	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
EMailMsgNumber	34	2	Binary Int.	Helps in rebuilding a mail message that was sent in several messages. Mail number
NumberOfMessage	36	2	Binary Int.	Helps in rebuilding a mail message that was sent in several transmission (between 1 to 20). Number of messages making up the mail message.
SequenceNumber	38	2	Binary Int.	Current sequence number message.
InstrumentGroupCode	40	3	ASCII Str.	Instrument Group of Identification
Filler	43	1	ASCII Str	*For future use.*
PriorityIndicator	44	1	ASCII Ch.	Urgency of the mail, valid values: 'O' - Urgent message. 'N' - Non urgent message.
TypeOfMessage	45	1	ASCII Ch.	General contents of the message. Valid Values: 'B' - Market Information contents, 'T' - Technical Information, 'R' - Market and Technical information
AddressType	46	2	ASCII Str.	Indicates the users to whom the message is addressed.
Title	48	80	ASCII Str.	Title of the mail.
Text	128	854	ASCII Str.	Text, line break (@), page break (#)
Filler	*982*	*2*	*ASCII Str.*	*For future use*

3.1.7. Indicative Matching Price IMP - 530 Message

3.1.7.1. Message Overview

The Indicative matching price message indicates an instrument's theoretical opening price (also known as an indicative opening price). Processing rules differ, depending on the different trading engines:

- NSC traded instruments (Warrants, Certificates, Lening&Borrowing)
  - An IMP-message is sent if the theoretical price or - volume varies.
  - If the theoretical price remains undetermined, but the reason for this indetermination changes, then a Theoretical opening price message is sent with zero values.
- UTP traded instruments
  - An IMP-message is sent if the indicative matching price or – volume varies.
  - If the indicative matching price remains undetermined, but the reason for this indetermination changes, then a Theoretical opening price message is sent with zero values.
- SmartPool
  - Indicates either the VWAP price (before the first opening), or the mid-point price (before the next opening phases), or the closing price (during Pre-closing phase) of an instrument listed on SmartPool.

3.1.7.2. Message Sending Rules

- NSC traded instruments
  - This message is sent each time the IMP (Indicative Matching Price) / TOP (Theoretical Opening Price) changes in Call / Pre-Opening phase or when a security is halted or reserved.
    (If an IMP/TOP can be calculated is depending on the contents of the order book).
- UTP traded instruments
  - On the UTP trading engine, the IMP is calculated in a preset frequency (set at 15 msec for Euronext)

3.1.7.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '530' Indicative matching price.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	530 - Indicative Matching Price
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
IMPrice	32	4	Binary Int.	Indicative Matching Price: Price at which transaction on a security would occur if Opening occurred at this moment (To be calculated with the PriceScaleCode).
Variation	36	4	Binary Int. (signed)	Percentage variation of the last IMP against the previous day's reference price for the stock concerned, ie the last known price can be the last adjusted closing price or the last price indication (if an indication was entered after the last traded price).(To be calculated with the VariationScaleCode).
SystemID	40	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	44	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
PriceScaleCode	46	1	Binary Int.	To be combined with IMPrice
VariationScaleCode	47	1	Binary Int.	To be combined with Variation

IMVolume	48	4	Binary Int.	Indicative Matching Volume: Volume that would be exchanged if Auction occurred at this moment. Provided only for instruments traded on UTP.

3.1.8. Market Imbalance - 531 Message

3.1.8.1. Message Overview

The market imbalance message will be defined in a future release of the specifications.

PLEASE NOTE: MsgType 531 is defined for future use only.

3.1.8.2. Message Structure

The table below describes the body fields of an Equities Order Book message, MsgType = '531' Market Imbalance.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	531 - Market Information Message – Market Imbalance
SymbolIndex	20	4	Binary Int.	Index of the Symbol- Stock representation
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
SourceTime	28	4	Binary Int.	This field specifies the Order generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
Volume	32	4	Binary Int.	The indicative match volume
TotalImbalance	36	4	Binary Int.	The total imbalance volume
MarketImbalance	40	4	Binary Int.	The market imbalance volume. The value can be negative.
PriceNumerator	44	4	Binary Int.	Indicative price match (according to PriceScaleCode).
SystemID	48	4	Binary Int.	The ID of the originating Exchange/System of the message.
SourceTimeMicroSecs	52	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
AuctionTime	54	2	Binary Int.	Projected auction time.
PriceScaleCode	56	1	Binary Int.	To be combined with PriceNumerator.
AuctionType	57	1	*ASCII Chr.*	'O' - Open 'M' - Market 'H' - Halt 'C' - Closing
Filler	*58*	*6*	*ASCII Str.*	*For future use,*

3.1.9. TCS Authorized Out of Session Limits - 534 Message

3.1.9.1. Message Overview

The message Authorized Out of Session Limits (outside the Trading Session) carries the authorized limits for trades executed outside the trading session for a given instrument.

These limits are calculated by the TCS reporting engine according to the last price and possibly the Weighted Average Spread for the instrument.

This message is not sent for the following types of instruments:

- Instruments listed on the Block trading market,
- Instruments that have been delisted from the regulated markets,
- Instruments listed on the Alternext market (for trades outside the central order book on these instruments, neither the price nor the traded amount is checked by TCS),
- Instruments belonging to the Dutch Fund category,
- Exchange-traded funds (ETFs) - called "trackers'" in Euronext terminology.

3.1.9.2. Message Sending Rules

This message is sent:

- Every morning for all securities submitted to a price control using the TCS operation type "off-market"
- If the security is traded on a continuous basis, the message is sent when the security's stock group enters the Surveillance Intervention phase
- If the security is traded by fixing, the message is sent at each fixing for that security
- By manual intervention by Cash Market Operations.

3.1.9.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '534' Authorized Out Of Session Limits.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	534- Authorized Out Of Session Limits
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
MinPriceTrades	32	4	Binary Int.	Minimum price for normal out of session trades(To be calculated with the TradeScaleCode).
MaxPriceTrades	36	4	Binary Int.	Maximum price for normal out of session trades(To be calculated with the TradeScaleCode).
MinPriceBlockTrades	40	4	Binary Int.	Minimum price for out of session block trades(To be calculated with the BlockTradeScaleCode).
MaxPriceBlockTrades	44	4	Binary Int.	Maximum price for out of session block trades(To be calculated with the BlockTradeScaleCode).
SystemID	48	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	52	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
TradeScaleCode	54	1	Binary Int.	To be combined with MinPriceTrades and MaxPriceTrades
BlockTradeScaleCode	55	1	Binary Int.	To be combined with MinPriceBlockTrades and MaxPriceBlockTrades

3.1.10. TCS State Change - 535 Message

3.1.10.1. Message Overview

A TCS state change is sent by the TCS reporting engine to announce the next trading day. It makes it possible to determine the trading date and the related indicative Net Asset Value (iNAV) for valuing the Dutch Funds trades.

3.1.10.2. Message Sending Rules

This message is sent:

- For a daily auction, once a day for an Investment fund instrument group only in order to indicate the change of a trading day.
- For a weekly, monthly or quarterly execution cycle, only once during the concerned cycle for an Investment funds instrument group only in order to indicate the change of a trading cycle.

3.1.10.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '535' TCS State Change.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	535- TCS State Change
SourceSeqNum	20	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SystemID	28	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	32	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
InstrumentGroupCode	34	3	ASCII Str.	Instrument Group concerned by the change of trading cycle
ChangeTradingCycle	37	1	ASCII Ch.	Indicates the cut-off for an investment fund group in order to take into account the trading cycle change. Valid values: 'C' - Change trading cycle.
Filler	*38*	*2*	*ASCII Str*	*For future use*

3.1.11. Collars - 537 Message

3.1.11.1. Message Overview

A Collars message informs Clients of modifications in authorized price fluctuations for an instrument. This message is subjected to selective transmission according to the class/group to which the instrument belongs.

3.1.11.2. Message Sending Rules

This message is sent:

- At the start-up of the UTP trading system in the morning
- Each time Cash Market Operations changes a dynamic collar / static threshold for an instrument
- After the first auction of the given instrument.

3.1.11.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '537' Collars.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	537- Collars
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
HighCollar	32	4	Binary Int.	Higher bound of the collar price range (To be calculated with the HighScaleCode).
LowCollar	36	4	Binary Int.	Lower bound of the collar price range (To be calculated with the LowScaleCode).
SystemID	40	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	44	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
HighScaleCode	46	1	Binary Int.	To be combined with HighCollar
LowScaleCode	47	1	Binary Int.	To be combined with LowCollar
CollarType	48	1	ASCII Ch.	Defines the type of collars: D Dynamic collars S Static thresholds
Filler	*49*	*3*	*ASCII Str*	*For future use*

3.1.12. Session Timetable - 539 Message

3.1.12.1. Message Overview

The session timetable message indicates the timetable listing all the states changes of a Class / Instrument group for the current trading day.

3.1.12.2. Message Sending Rules

This message is sent:

- Automatically for each Class / Instrument group at the beginning of the trading day, to indicate the times at which the market session will change from one phase to another
- On an exceptional basis, it may be sent during the trading day in cases where normal hours have changed or multiple openings are re-programmed during the day

3.1.12.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '539' Session timetable.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	539- Session timetable
SourceSeqNum	20	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SystemID	28	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	32	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
InstrumentGroupCode	34	3	ASCII Str	Indicates for which Class / Instrument Group code the different times are provided.
SessionType	37	1	ASCII Ch	Market Session. Valid values: 'E' - Early session 'C' - Core session 'L' - Late session.
TimePreOpening1	38	6	ASCII Str.	Indicates the pre-opening time of the Class / Instrument Group for the session
TimeOpening1	44	6	ASCII Str.	Indicates the opening time of the Class / Instrument Group for the session
TimeClosing1	50	6	ASCII Str.	Indicates the closing time of the Class / Instrument Group for the session
TimePreOpening2	56	6	ASCII Str.	Indicates, if applicable, the pre-opening time of 2nd auction
TimeOpening2	62	6	ASCII Str.	Indicates, if applicable, the opening time of 2nd auction
TimeClosing2	68	6	ASCII Str.	Indicates, if applicable, the closing time of 2nd auction
TimePreOpening3	74	6	ASCII Str.	Indicates, if applicable, the pre-opening time of 3rd auction
TimeOpening3	80	6	ASCII Str.	Indicates, if applicable, the opening time of 3rd auction
TimeClosing3	86	6	ASCII Str.	Indicates, if applicable, the closing time of 3rd auction
EodTime	92	6	ASCII Str.	Time of End of day inquiry / Late Monitoring
Filler	*98*	*2*	*ASCII Str.*	*For future use*

3.1.13. Display Bid and Ask - 540 Message

3.1.13.1. Message Overview

A Display Bid&Ask message describes the bid or offer for an instrument for which there is an unfilled bid or offer.

For instrument traded on UTP, this message will no longer exist.

3.1.13.2. Message Sending Rules

This message is sent for instruments traded on NSC:

- In the Pre-Opening phase if the theoretical opening price (TOP) is outside the upper (SIH) or lower (SIB) intermediate threshold, the message displays the bid or ask price instead of the equilibrium price, in case of price extremes. In this case, the bid and ask prices are made equal to the SIH and SIB.
- In the Pre-Opening phase if some of the 'at best' orders could not find a counterpart and thus the theoretical opening price (TOP) could not be determined. In this case:
    - If at best buy orders remain:
      The Type of last price has a value of 02 and the TOP is made equal to the upper intermediate threshold (SIH)
    - If at best sell orders remain:
      The Type of last price has a value of 03 and the TOP is made equal to the lower intermediate threshold (SIB).

3.1.13.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '540' Display Bid or Ask.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	540 - Display Bid & Ask
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
Price	32	4	Binary Int.	Bid or Ask price(To be calculated with the PriceScaleCode)
SystemID	36	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	40	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
PriceScaleCode	42	1	Binary Int.	To be combined with Price
Side	43	1	Binary Int.	Type of price. Valid values: 2 - Bid (Buy) Order 3 - Ask (Sell) Order.

3.1.14. Daily Summary - 541 Message

3.1.14.1. Message Overview

A Daily Summary message summarizes an instrument's opening trades.

The Opening Summary message is sent by the trading engines after an instrument opening (fixing) that has been traded to summarize the opening (fixing), or if the first trade(s) occurred during continuous trading.

PLEASE NOTE: MsgType 541 is defined for future use only.

3.1.14.2. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '541' Daily Summary.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	541 - Daily Summary
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
FirstPrice	32	4	Binary Int.	First price(To be calculated with the ScaleCode). of the trading session. Can differ from the Open price if takes place during Early session or Late session (if any).
OpenPrice	36	4	Binary Int.	Price resulting from the opening auction.
LowestPrice	40	4	Binary Int.	The lowest price(To be calculated with the ScaleCode) of all the trades that have occurred for a given listed instrument on a given day.
HighestPrice	44	4	Binary Int.	The highest price(To be calculated with the ScaleCode) of all trades that have occurred for a given listed instrument on a given day.
ClosePrice	48	4	Binary Int.	Official closing price(To be calculated with the ScaleCode).
LastPrice	52	4	Binary Int.	Last price of the trade(To be calculated with the ScaleCode). Can differ from the Close price if takes place during Early session or Late session (if any)
Volume	56	4	Binary Int.	Total number of instruments traded during the core session
SystemID	60	4	Binary Int.	The ID of the originating Exchange/System of the message
CapitalAmount	64	8	Binary Int.	Sum of the capital traded on a given day for a given listed instrument. This sum is expressed in the trading currency of the instrument(To be calculated with the CapitalScaleCode)
SourceTimeMicroSecs	72	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
Date	74	8	ASCII Str.	Date of the trading day on which trades occurred that are included in the Daily Summary for a given listed instrument.
ScaleCode	82	1	Binary Int.	To be combined with LowestPrice, OpenPrice, LowestPrice, HighestPrice, LastPrice
CapitalScaleCode	83	1	Binary Int.	To be combined with CapitalAmount

3.1.15. Global Market Activity Summary – Message 545

3.1.15.1. Message Overview

This message is used only at Qatar Exchange

This message provides an overall summary of the global market activity.

3.1.15.2. Message Sending Rules

The global Market Activity summary is issued in real time, every NN seconds (where NN is a parameter typically set to 15 to 60 seconds).

3.1.15.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '545' Global Market Activity Summary.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	545 – Global Market Summary Activity
SourceSeqNum	20	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SystemID	28	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	32	2	Binary Int.	Number of micro seconds. To be combined with SourceTime field.
NbTrades	34	4	Binary Int.	Number of Trades executed on the market since the beginning of the Trading Day
QtyShares	38	4	Binary Int.	Cumulated Quantity traded (number of shares) on the market since the beginning of the Trading Day
AmountTraded	42	4	Binary Int.	Total cumulated amount traded on the market since the beginning of the Trading day (To be calculated with the PriceScaleCode)
PriceScaleCode	46	1	Binary Int.	Applicable to all prices in the message
Filler	*47*	*3*	*ASCII str.*	*reserved for future use*

3.1.16. Start Referential – Message 550

3.1.16.1. Message Overview

The Start referential message is sent to signal the start of transmission of the Instrument Reference Data Flow. It is sent in the morning and in the evening.

3.1.16.2. Message Sending Rules

This message is sent:

- Each time referential messages (555) are sent, prior to the first one.

3.1.16.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '550' Start referential.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	550 - Start referential
Indicator	20	1	ASCII Ch.	This field indicates the start of the Instrument characteristic flow. Always takes the value 'S'
Filler	*21*	*3*	*ASCII Str.*	*This is a filler, reserved for a future use*

3.1.17. End Referential - 551 Message

3.1.17.1. Message Overview

The End referential message is sent to signal the end of transmission of the Instrument Reference Data Flow. It is sent in the morning and in the evening.

3.1.17.2. Message Sending Rules

This message is sent:

- Each time referential messages (555) are sent, after the last one.

3.1.17.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '551' End referential.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	551- End referential
Indicator	20	1	ASCII Ch.	This field indicates the end of the Instrument characteristic flow. Always takes the value 'E'
Filler	*21*	*3*	*ASCII Str.*	*This is a filler, reserved for a future use*

3.1.18. Referential - 555 Message

3.1.18.1. Message Overview

The referential message indicates the main characteristics of a listed instrument:

- Characteristics of the instrument itself
- Trading characteristics of the instrument
- Previous trading day price and amount of capital traded.

For morning messages, the characteristics are valid for the trading day at the start of which it was transmitted.

For evening messages, the characteristics are valid for the next business date (the trading day D+1 following the Day D of its transmission).

3.1.18.2. Message Sending Rules

This message is sent:

- Every morning. There is a 555 message per instrument, tradable or broadcasted for the current day.
- Every evening. There is a 555 message per instrument, tradable or broadcasted for the next trading day.
- 555 messages will be sent in between a start and an end reference data messages (550 and 551).

3.1.18.3. Message Structure

The table below describes the body fields of an Equities Market Information message, MsgType = '555' Referential.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	555 - Instrument Characteristic
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
LastAdjPrice	32	4	Binary Int.	Last traded price of the previous trading day after application of the adjustment coefficient (To be calculated with the LastAdjPriceScaleCode). Provided only in the message sent in the morning. Not provided for SmartPool and European instruments.
SystemID	36	4	Binary Int.	The ID of the originating Exchange/System of the message
PrevVolumeTraded	40	4	Binary Int.	Number of shares traded on this security during trading day identified by 'DateOfLastTrade'. Provided only in the message sent in the morning. Not provided for SmartPool and European instruments.
FixPriceTick	44	4	Binary Int.	Indicates the amount of the fixed tick size, if it has been defined at the instrument level. (according to FixPriceTickScaleCode). Provided only for tradable instruments.
SourceTimeMicroSecs	48	2	Binary Int.	Number of micro seconds. To be combined with SourceTime
Stock Exchange Code	50	2	Binary Int.	Indicates the Market Place. Possible values: Refer to appendix C
TypeOfInstrument	52	2	Binary Int.	Type of instrument. Possible values: Refer to appendix D (Stock Type).

Field Name	Offset	Size (Bytes)	Format	Description
EventDate	54	8	ASCII Str.	Date of the last modification of the characteristics of the instrument except the following ones (which are modified every day): - Previous day's adjusted closing price (LastAdjPrice) - Previous day capital traded (PrevDayCapitalTraded) - Number of shares for this security traded on previous day (PrevVolumeTraded) - Date when the security last traded (DateOfLastTrade)
InstrumentName	62	18	ASCII Str.	Instrument Name
PeriodIndicator	80	1	ASCII Ch.	Indicates for which day the characteristics of the instrument will be effective. Valid values: 'M' - Morning, effective for the current day. 'E' - Evening, effective for the next trading day.
TypeOfMarketAdmission	81	1	ASCII Ch.	Indicates the type of market to which a security has been listed. Valid values: 'A' - Instruments traded on the primary market 'B' - Instruments traded on the secondary market 'C' - Instruments traded on the New Market 'D' - Non regulated market / instruments traded on the free market ('Marché libre') 'E' - Non regulated market / Alternext 'F' - Non listed 'G' - Regulated Market / Non equities 'H' - Regulated Market / Equities / Segment A 'I' - Regulated Market / Equities / Segment B 'J' - Regulated Market / Equities / Segment C 'K' - Regulated Market / All securities / Special Segment 'L' - Regulated Market / Equities / Other instruments 'S' - OPCVM, SICOMI non listed (French Investment Funds) '6' - Off Market '7' - Gold, Currencies, and Indices of Euronext '9' - Foreign
IssuingCountryCode	82	3	ASCII Str.	Country code of location for the corporate headquarters of the company that issued the instrument. (ISO 3166-3A)
TradingCurrency	85	3	ASCII Str.	Code of the currency in which the instrument is traded. (ISO 4217-3A)
InstrumentGroupCode	88	3	ASCII Str.	Designates the class/group of instruments to which the security belongs.
InstrumentCategory	91	1	ASCII Ch.	Indicates to which category, the security belongs. Valid values: 'A' - Stock, 'O' - Bond, 'W' - Warrant and certificate, 'T' - Tracker, 'D' - Miscellaneous
InstrumentCode	92	12	ASCII Str.	Isin Code, International Securities Identification Number, according to the ISO-6166.
DateOfLastTrade	104	8	ASCII Str.	Date of the last trading day during which a trade was executed for the instrument. (YYYYMMDD). Provided only in the message sent in the morning. Not provided for SmartPool and European instruments.
UnderlyingRepoISINCode	112	12	ASCII Str.	Underlying security (security used in the loan quotation system) for loan contracts at given value on centralized lending market. Used on L&B segment only
RepoExpiryDate	124	8	ASCII Str.	Represents the inclusive date until which a lending/borrowing contract can be traded. Usedon L&B segment only
FirstSettlementDate	132	8	ASCII Str.	Represents the first possible settlement date for a given instrument with the instrument's depository. When this date is not provided, it means

Field Name	Offset	Size (Bytes)	Format	Description
				that the first possible settlement date is the same as the instrument's flotation date. This item is provided solely for Amsterdam-listed instruments of the type As If and When Issues. In other words, it is provided solely for new issues for which the first settlement date is a considerable length of time in the future, or is still not known even though it is already possible to trade the instrument. As long as the date remains unknown, this is a fictitious date that must be modified as soon as the real date is known. In terms of instrument types, the instrument can be either a bond or a warrant. This item is determined as followed: • if the marketplace = 038 (Amsterdam), then • if the instrument is a bond, then the first possible settlement date is the settlement date for the issue price if this item is not set to zero. • if the instrument is a warrant, then the first possible settlement date is the settlement date for the issue price (taken from the Warrant Characteristics message) if this item is not set to zero. In all other cases, this item is not provided. Used by the clearing house in the rule for determining the theoretical settlement date for a trade. Permitted Values: Nulls - If not provided - YYYYMMDD format. - 20111111 is the date used for Dutch warrants for which the settlement date is unknown at the time the instrument is floated.
TypeOfDerivatives	140	1	ASCII Ch.	Type of derivative instrument associated to the security. Valid values: '1' - INAV (Indicative Net Asset value for ETF-Exchange Traded Funds) '4' - Supports short terms options, '6' - Supports long term options '9' - No.
BICDepositary	141	11	ASCII Str.	Identifies the depository organization for the shares of an instrument, or when the same organization manages several systems, this item identifies the relevant settlement system for settling trades on a given listed security. The ID is allocated by SWIFT, according to the BIC standard (ISO 9362). Used by the Clearing 21® system to determine the relevant system for settling trades in a Brussels-listed instrument (the Bank of Belgium's X/N system or VIF system), and thus, the time at which the transactions must be sent during the operating day, and the format in which they must be sent.
ICB	152	4	ASCII Str.	Identifies for a listed instrument, the economic subsector of the issuing company in the ICB (Industry Classification Benchmark) classification.
MIC	156	4	ASCII Str.	Identifies the market to which an instrument y belongs by its MIC (Market Identification code), according to ISO 10383. NYSE Euronext owns the following MICs: ALXA ALTERNEXT AMSTERDAM ALXB ALTERNEXT BRUSSELS ALXP ALTERNEXT PARIS ENXB EASYNEXT BRUSSELS ENXL EASYNEXT LISBON MLXB MARCHE LIBRE BRUSSELS TNLA TRADED BUT NOT LISTED AMSTERDAM TNLB NYSE EURONEXT - TRADING FACILITY BRUSSELS WQXL MARKET WITHOUT QUOTATIONS LISBON XAMS EURONEXT AMSTERDAM XBRU EURONEXT BRUSSELS XLIS EURONEXT LISBON XMLI MARCHE LIBRE PARIS XPAR EURONEXT PARIS

Field Name	Offset	Size (Bytes)	Format	Description
				XSMP SMARTPOOL Qatar Exchange owns the following MICs: DSMD Doha Securities Market/Qatar Exchange
UnderlyingWISINCode	160	12	ASCII Str.	Gives the trading code of the underlying listed security of a warrant.
Depositary List	172	25	ASCII Str.	Identifies the possible main depository organizations (max 5) for the shares or bonds for an instrument. Used by the clearing house to determine the relevant system for settling trades. Valid values: '00001' - Euroclear France '00002' - CIK (Belgium) '00003' - NECIGEF (the Netherlands) '00004' - X/N (BoB's system) '00005' - VIF (non-fungible Belgian instruments) '00006' - Euroclear Bank '00007' - NIEC '00008' - Physical '00009' - Euronext Paris non Euroclear France '00010' - Interbolsa '00000' - No depository organization Nulls - Not significant
MainDepositary	197	5	ASCII Str.	Identifies the default (or main) depository organisation of the instrument (for Euronext, between the possible 5 depositaries registered for Euronext markets) used by priority for the settlement (e.g.: multi-listed instruments which have several depositories). This data has to be treated in consideration of the data DepositoryList. Used by the clearing house to determine the relevant system for settling trades. Valid values: Same as data "DepositoryList"
TypeOfCorporateEvent	202	2	ASCII Str.	Indicates the last type of corporate event that has occurred on an instrument, such as detachment of rights, or of coupons. The data item is automatically calculated by the adjustment application but in case of problem or error, the data item value could be modified manually, particularly for purging the order book in case of absence of corporate event. This data has to be treated in consideration of the date of the event included into the header of the message. Valid values: '00' No specific event '01' Dividend payment in cash or in stocks '02' Interest payment (bonds for which the price is not expressed in % of the nominal, only) '04' Split '05' Bonus (i.e. attribution) '06' Subscription '07' Share allocation '08' Share swap '09' Reverse split '10' Merger '11' Final bond redemption '12' Capital amortization '13' Draw announcement (Belgian bonds only) '14' Block trade of controlling interest '15' Optional corporate events(dividend option) '16' Complex corporate event '17' Purge of the order book (purge is initiated manually in the absence of a corporate event, e.g. following the modification of the variable tick of the listed security)
TimeLagEuronextUTC	204	5	ASCII Str.	Used for Euronext/MiFID: Effective difference time between CET (Euronext time) and UTC. To be

Field Name	Offset	Size (Bytes)	Format	Description
				interpreted in conjunction with the time difference between MiFID regulators and UTC. Valid values: - SHHMN format (with S = + / - , HH = Hour, MN=Minutes) - Always provided (MiFID instrument) - " 0000" means no difference time.
TimeLagMiFIDRegUTC	209	5	ASCII Str.	Used for Euronext/MiFID: Effective time difference between MiFID regulators and UTC. To be interpreted in conjunction with the time difference between CET (Euronext time) and UTC. Valid values: - SHHMM format (with S = + / - , HH = Hour, MM=Minutes) - Always provided (MiFID instrument) - " 0000" means no difference time
CFI	214	6	ASCII Str.	Classification code of a financial instrument defined by the ISO-10962 standard. The structure of the CFI code: - The CFI reflects characteristics that are defined when a financial instrument is issued, and remain unchanged during its entire lifetime. - The CFI consists of six alphabetical characters: 1st character indicates the highest level of classification (Categories) 'E' - Equities 'D' - Debt instruments 'R' - Entitlements (Rights) 'O' - Options 'F' - Futures 'M' - Others/Miscellaneous 2nd character indicates specific groups within each category: Groups e.g. for equities: - Shares - Preferred shares - Convertible preferred shares - Units, i.e. unit trusts/mutual funds etc. - Others 3rd to 6th character indicate the most important attributes to each group: Attributes e.g. for equities: - Voting right - Ownership/transfer restrictions - Payment status - Form
QuantityNotation	220	4	ASCII Str.	Specifies the nature of the amount expression used for negotiating the instrument on the market. Valid values: 'UNT' - In unit (i.e. number of shares), left padded 'FMT' - In facial amount (i.e. bonds expressed in %), left padded Null - Not applicable
IndexSetOfVarPriceTick	224	2	ASCII Str.	When the instrument's tick is defined via a Variable Tick Table: The key for the variable price tick table consists of two data items [the index for a set of variable price ticks, the lowest value in a range of prices]. The index data item refers to a set of lines in that table which make it possible to determine the price tick for an instrument, based on the price range in which a given price for the instrument falls (i.e. a price to be rounded off or a limit to be checked). When a listed security is created, or when the characteristics of an existing listed security are modified, this data item is (re)initialized to the "typical" index value for this listed security (if a value is not provided for the Fixed Price Tick data item). In Euronext, this "typical index" is derived from the following three characteristics: -, - Trading currency (type of unit in which the listed security's price is

Field Name	Offset	Size (Bytes)	Format	Description
				expressed), - Broad instrument category associated with the listed security, - Trading group.
MarketFeedCode	226	2	ASCII Str.	"Market data flow" to which the instrument belongs. Possible values are listed in appendix A
MICList	228	24	ASCII Str.	Identifies the markets on which a security is listed by its MIC (Market Identification Code). - For a security listed on a single market, the listing MIC code is the same than “Market identification code (MIC) of the listed security” - For a security listed on several Markets: . The first MIC is the same than the “Market identification C ode (MIC) of the listed security . The others MIC indicate the other listing places
IndustryCode	252	4	ASCII Str.	Not used
IndustryText	256	100	ASCII Str.	Not used
FinancialMarketCode	356	3	ASCII Str.	Refer to appendix B
USIndicators	359	7	ASCII Str.	Not used
Filler	*366*	*2*	*ASCII Str.*	*For future use*
PrevDayCapitalTraded	368	8	Binary Int.	Cumulative capital traded for all previous day trades on a security (To be calculated with the PrevDayCapitalTradedScaleCode). Provided only in the message sent in the morning. Not provided for SmartPool, NAe and European instruments.
NomMktPrice	376	8	Binary Int.	Amount of the nominal value of the security (To be calculated with the NomMktPriceScaleCode).
LotSize	384	8	Binary Int.	Expressed in number of shares or in an amount or a volume of the capital, of the lot size. The lot size is a minimum tradable quantity that is set for each instrument by Euronext or Qatar Exchange. The quantity of an order entered by a trading member on the market must be a multiple of the lot size. This number is also called the "Quotité de Marché" (Minimum market tradable quantity). For bonds, this data has to be considered with the data “Amount of par value for instrument for calculating trade amount”. This item is calculated in the following way: For Brussels-listed bonds that are quoted in %: • if 1 ≤ market par value ≤99 999 999, then the Instrument Lot Size is the integer part of the market nominal (and, moreover, the lot size and the par value for trade amount are set to 1). For Amsterdam-listed bonds quoted in %: • if 1≤ initial par value ≤ 99 999 999, then the Instrument Lot Size is the integer part of the initial nominal (and, moreover, the lot size and the par value for trade amount are set to 1). For Lisbon-listed bonds quoted in %: • if 1≤ market par value ≤ 99 999 999, then the Instrument Lot Size is the integer part of the market nominal (and, moreover, the lot size and the par value for trade amount are set to 1). • if the market par value is not an integer, until 4 decimals (over, the number is rounded to 4 decimals) the Instrument Lot Size is set to an integer multiple of the market par value (and, moreover, the par value for trade amount is set to 1). In all other cases, the Instrument Lot Size is the item Instrument Lot Size that is provided by PGD-OST (this is a whole number of shares or bonds). Note 1: Only positive values are possible. If not provided, will be set to 0

Field Name	Offset	Size (Bytes)	Format	Description
				Note 2 Only integer values that are equal to or greater than one are accepted until Euronext systems have been adapted for using quantities expressed as a par value amount (Decimalization project). According to LotSizeScaleCode.
NumberInstrumentCirc	392	8	Binary Int.	Number of shares issued/bonds outstanding after payments
SharesOut	400	8	Binary Int.	Not used
AuthShares	408	8	Binary Int.	Not used
OfficialQuotationList HeadingNumber SectionNumber	416	 2 1	 Binary Int. Binary Int.	Gives the section, heading and sub-headings in which the security is in the “Bulletin de la Cote Officielle”. Used in Euronext only Valid values (Number for section): 0 - French national funds and treasury bonds. 1 - Bonds, warrants, public and semi-public sector. 2 - Bonds, warrants, French common bond funds. 3 - French domestic equities and odd lots (cash). 4 - Other securities in French francs (cash), and international issues. 5 - Foreign securities (cash). 6 - Monthly settlement securities. 7 - Second Market. 8 - Hors Cote and Eurocac. 9 - Miscellaneous (gold, foreign exchange, indices...).
RepoIndicator	419	1	ASCII Ch.	Indicates whether the security listed underlies any loan contracts, meaning it has been admitted to the Deferred Settlement system and/or to the lending market. Valid values: ‘0’ Instrument neither eligible for the SRD, nor eligible for the Loan and Lending Market. ‘1’ Instrument eligible for the SRD and for the Loan and Lending Market ‘2’ Instrument eligible for the SRD but not for the Loan and Lending Market ‘3’ Instrument eligible for the Loan and Lending Market but not for the SRD ‘8’ Non significant
LastAdjPriceScaleCode	420	1	Binary Int.	To be combined with LastAdjPrice.
TypeOfUnitExp	421	1	Binary Int.	Unit in which the security is quoted. Valid values: 1 In Units, 2 As a % of nominal, 3 As a % of nominal (including accrued interest). 8 In Kilograms, 9 In Ounces.
MarketIndicator	422	1	Binary Int.	Indicates the market regulations governing the market on which the security is traded. Valid values: 0 Not used 1 Cash 9 Not defined.
PrevDayCapitalTradedScaleCode	423	1	Binary Int.	To be combined with PrevDayCapitalTraded.
TaxCode	424	1	Binary Int.	Tax deduction code to which the security belongs. Valid values for Euronext Paris: 0 Not eligible to PEA 3 Eligible to PEA, 9 Not provided
NomMkdtPriceScaleCode	425	1	Binary Int.	To be combined with NomMkdtPrice.
LotSizeScaleCode	426	1	Binary Int.	To be combined with LotSize.
FixPriceTickScaleCode	427	1	Binary Int.	To be combined with FixPriceTick.
Mnemo	428	5	ASCII Str.	Mnemonic code of the instrument.

Field Name	Offset	Size (Bytes)	Format	Description
TradingCode	433	12	ASCII Str.	Trading code of the instrument.
Filler	*445*	*3*	*ASCII Str*	*For future use*
StrikePrice	448	4	Binary Int.	The specified price of an option contract at which the contract may be exercised, whereby a call option buyer can buy the underlying or a put option buyer can sell the underlying. The buyer's profit from exercising the option is the amount by which the strike price exceeds the cash instrument price (in the case of a call), or the amount by which the cash instrument price exceeds the strike price (in the case of a put). In general, the smaller the difference between spot (cash instrument price) and strike price, the higher the option premium. Also called exercise price. According to StrikeScaleCode. Permitted values Only provided for warrants or other derivative instruments.
StrikeCurrency	452	3	ASCII Str	Code for the currency of the strike price applied for an instrument whose maturity is reached (ISO 4217-3A). Permitted values Only provided for warrants or other derivative instruments.
StrikeScaleCode	455	1	Binary Int.	To be combined with StrikePrice
CurrencyCoef	456	4	Binary Int.	Change ratio coefficient of currency applied to instrument: Used in conjunction with one of the change rate indicators in order to apply this coefficient to a currency among two available currencies defined for the instrument: - Trading currency - Currency code of strike price for Derivative Instrument The currency to which this coefficient will be applied depends on just one of the two values set to the related indicators (defined below in the integrity constraints). This coefficient is used when the currency is not compliant with the ISO 4217 (3A) standard as pence (GBP) or cent (USD) expression of an official currency. In this case the formula to apply in order to retrieve the price expressed in an official currency is: Real price in Trading currency = Traded price (i.e. 1565 pence) x Change ratio coefficient value (0.01)
CurrencyCoefScaleCode	460	1	Binary Int.	To be combined with CurrencyCoef
TradingCurrencyIndicator	461	1	Binary Int.	Change rate indicator for currency of Instrument Traded Price: Indicates if the change rate will be applied to the currency defined for traded prices of the instrument. Permitted values - 0: Change rate not applied to the traded price. - 1: Change rate applied to the traded price. Null: Not applicable
StrikeCurrencyIndicator	462	1	Binary Int.	Change rate indicator for currency of the strike price Indicates if the change rate will be applied to the currency defined for strike prices of the instrument. Permitted values - 0: Change rate not applied to the strike price. - 1: Change rate applied to the strike price. Null: Not applicable
MarketSegment	463	2	ASCII Str.	Qatar Exchange only: Market segment. Valid values: NM Normal market OT Block trading ID Index
Algo	465	1	ASCII Str.	Qatar Exchange only: Indicates which trading algorithm is used. Valid values: F FIFO O FIFO Origin

Field Name	Offset	Size (Bytes)	Format	Description
LocalName	466	30	ASCII Str.	Qatar Exchange only: Instrument name in local language (Arabic character set ISO-8859-6 for QE)
SmallTrade	496	4	Binary Int.	Qatar Exchange only: Instrument trade amount under which a trade is considered as small
Filler	*500*	*12*	*Binary Int.*	*For future use*

3.2. *Trades*

3.2.1. Overview

The Equities Trade service uses the push-based publishing model. This means that data will be published based on its availability. Once a Last Sale (a.k.a. Last Trade) is available, it will be published to Equities Trades Clients.

The Equities Trade message reflects the last sale in each traded security.

Below is a list of the functional message types in the Equities Trade Feed:

- 220 - Trade Creation
- 221 - Trade Cancellation
- 240 - Trade Full Information
- 241 - Price Update
- 242 - TCS Trade
  Not used for Qatar Exchange
- 243 – (TCS) Trade Publication
  Not used for Qatar Exchange
- 244 - Settlement Price
  Not used for Qatar Exchange
- 245 - Auction Summary.
- 246 – Notice of Interest
  Not used for Qatar Exchange
- 247 – VWAP-Closing Price
  Only used at Qatar Exchange

3.2.2. Packet Header Format

All messages are preceded by a common packet header format. The table below describes the header fields of a Trade messages.

Field	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet including the 16-byte packet header.
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. If all messages within a packet are of the same message type, the packet type will be equal to that message type. If not, the packet type will be set at 998. Possible values: 998 - Generic Trade Message. 220 - Trade Message – Creation 60 bytes 221 - Trade Message – Cancellation 44 bytes 240 - Trade Message – Full Information 80 bytes 241 - Trade Message – Price Update 60 bytes 242 - Trade Message – TCS Trade 124 bytes 243 - Trade Message – Trade Publication 160 bytes 244 - Trade Message – Settlement Price 44 bytes 245 - Trade Message – Auction Summary 68 bytes 246 - Trade Message – Notice of Interest 36 bytes 247 - Trade Message – VWAP-Closing Price 60 bytes
PacketSeqNum	4	4	Binary Int.	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and monotonically and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in millisecond indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC.
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream.

Field	Offset	Size (Bytes)	Format	Description
				Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

3.2.3. Trade Creation - 220 Message

3.2.3.1. Message Overview

This message is sent by the trading engines (NSC and UTP). It indicates one of the following:

- A trade generated during or after the first opening of the instrument, or
- A cross trade ("buy-sell")

3.2.3.2. Message Sending Rules

This message is sent each time a trade takes place on the trading engine (UTP or NSC).

3.2.3.3. Message Structure

The table below describes the body fields of a Equities Trade message, MsgType = '220' Trade creation.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	220 - Trade Creation
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
TradeIDNumber	28	4	Binary Int.	Unique numeric and increasing Identifier of the trade, set by the trading engine.
QuoteLinkID	32	4	Binary Int.	Identifies a unique quote that the Trade executed against. *Used as a "link" to the Quotes Service, as Trade Service and Quote Service are sent in parallel and may be slightly out of sync*
SourceSeqNum	36	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
Price	40	4	Binary Int.	Price of the trade (To be calculated with the PriceScaleCode)
Volume	44	4	Binary Int.	Number of shares transacted in this trade
SystemID	48	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	52	2	Binary Int.	Number of micro seconds. To be combined with SourceTime field.
SmallTradeIndicator	54	1	ASCII Ch.	QATAR EXCHANGE ONLY: This field indicates if the trade is a small trade or not. Valid values: Null Not significant '1' Not a small trade '2' Small trade
TradCond2	55	1	ASCII Ch.	Sequencing related conditions, Last of a series of trades at the same price. Valid values: '0' - Not last of a series of trades at the same price. '1' - Last of a series of trade at the same price
TradCond3	56	1	ASCII Ch.	Trade related conditions, Cross trade indicator. Valid values: <u>For instruments traded on NSC:</u> '0' Trade does not stem from a Cross Order '1' Trade stems from a Cross Order '3' Trade stems from a Basket Cross Order '4' Valuation Trade <u>For instruments traded on UTP:</u> '0' Trade does not stem from a Cross Order '1' Trade stems from a Cross Order '4' Valuation Trade
TradeOrigin	57	1	ASCII Ch.	QATAR EXCHANGE ONLY:

Field Name	Offset	Size (Bytes)	Format	Description
				This field indicates the origin of the trade. Valid values: 'B' Orders from the book 'O' Block trade
OpeningTradeIndicator	58	1	ASCII Ch.	If the trade took place during the opening auction or during the core session. Valid values: 'O' Opening 'S' Core Session
PriceScaleCode	59	1	Binary Int.	Applicable to all prices in the message

3.2.4. Trade Cancellation - 221 Message

3.2.4.1. Message Overview

A Trade cancellation message informs that a trade which was already executed has been cancelled.

3.2.4.2. Message Sending Rules

This message is sent each time a trade executed on the trading engine (UTP or NSC) is cancelled.
In case of modification of the Last Traded Price and/or the Highest and/or the Lowest and/or the First, a message 241 – Price Update- will follow this message.

3.2.4.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '221' Trade cancel.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	221 - Trade Cancel
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
OriginalTradeIDNumber	32	4	Binary Int.	This field refers to the initial TradeIDNumber of the trade concerned (as reported in a 220/240 message) in this cancellation message
SystemID	36	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	40	2	Binary Int.	Number of micro seconds. To be combined with SourceTime field.
Filler	*42*	*2*	*Binary Int.*	*For future use.*

3.2.5. Trade Full Information - 240 Message

3.2.5.1. Message Overview

Trade full information message is sent each time a trade has occurred and a trade message (msg.220) is sent. This specific message provides the detailed information related to the trade.

3.2.5.2. Message Sending Rules

This message is sent each time a trade is taking place on the trading engine (UTP or NSC). As for each trade both a msg.220 and a msg.240 is sent, clients should not process both messages as "reporting of a new trade".

3.2.5.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '240' Trade full information.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	240 - Full Information
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
TradeIDNumber	28	4	Binary Int.	Unique numeric and increasing Identifier of the trade, set by the trading engine.
QuoteLinkID	32	4	Binary Int.	Identifies a unique quote that the Trade executed against. *(used to synchronize between the Quote service and the Trade service)*
SourceSeqNum	36	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
Price	40	4	Binary Int.	Price of the trade(To be calculated with the PriceScaleCode)
Volume	44	4	Binary Int.	Number of shares transacted in this trade
CumulativeQuantity	48	4	Binary Int.	Cumulative number of shares traded on UTP/NSC since the start of the current trading session
HighestPrice	52	4	Binary Int.	Highest Price traded during the day(To be calculated with the PriceScaleCode)
LowestPrice	56	4	Binary Int.	Lowest Price traded during the day(To be calculated with the PriceScaleCode)
VariationLastPrice	60	4	Binary Int. (signed)	Percentage variation of today's price/last reference price(To be calculated with the VariationScaleCode)
SystemID	64	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	68	2	Binary Int.	Number of micro seconds. To be combined with SourceTime field.
Filler	*70*	*2*	*Binary Int.*	*For future use.*
SmallTradeIndicator	72	1	ASCII Ch.	QATAR EXCHANGE ONLY: This field indicates if the trade is a small trade or not. Valid values: Null Not significant '1' Not a small trade '2' Small trade
TradCond2	73	1	ASCII Ch.	Last trade at same price indicator: Valid values: '0' - Not last of a series of trade at the same price. '1' - Last of a series of trade at the same price
TradCond3	74	1	ASCII Ch.	Cross Trade Indicator: Valid values: For instruments traded on NSC: '0' - Trade does not stem from a Cross Order '1' - Trade stems from a Cross Order

Field Name	Offset	Size (Bytes)	Format	Description
				'3' - Trade stems from a Basket Cross Order '4' - Valuation Trade For instruments traded on UTP: '0' Trade does not stem from a Cross Order '1' Trade stems from a Cross Order '4' Valuation Trade
TradeOrigin	75	1	ASCII Ch.	QATAR EXCHANGE ONLY: This field indicates the origin of the trade. Valid values: 'B' Orders from the book 'O' Block trade
TickDirection	76	1	ASCII Ch.	Symbol of the variation of the price versus the previous (traded or reference) price, valid values: '+' - positive '-' - negative '0' - no variation or data field not significant Null - not significant
OpeningTradeIndicator	77	1	ASCII Ch.	If the trade took place during the opening auction or during the core session. Valid values: 'O' Opening 'S' Core Session
VariationScaleCode	78	1	Binary Int.	To be combined with VariationLastPrice
PriceScaleCode	79	1	Binary Int.	Applicable to all prices in the message

3.2.6. Price Update - 241 Message

3.2.6.1. Message Overview

The Price update message contains a bid price or an ask price, or the modification of the last reference price on a given instrument. This message informs of:

- Modifications operated on instrument prices.
- When the message follows the cancellation of a trade, it indicates that an instrument's first, high, low, or most recent price has been modified.
- The updated instrument's last adjusted closing price or the instrument's settlement price when modified.

3.2.6.2. Message Sending Rules

This message is sent:

- When the market operation changes the last adjusted price of the previous day (type of price 34).
- When the market operation cancels a trade which modifies one of the following price: first, high, low and last (type of price 30, 31, 32, 33). There is one msg.241 sent for each modified price.
- When a security cannot quote despite the orders present in the book. In this case, the market operation enters a bid or an ask price on a security depending on the sense of the imbalance (type of price: 02, 03).
- When the security is quoted by a pigeon-hole quotation (type of price 05, 06).
- When a warrant does not quote during a given time after the opening or before the closing auction. In both cases, the market operation enters a valuation price whose type depends on the period (type of price 41, 42). The warrant can quote in the meantime. The first traded price cancels and replaces the opening valuation. Under no circumstances is the closing valuation to replace the last traded price on the warrant.
- When the iNAV fixing time for the Investment Funds groups traded on TCS is reached. The message is only sent for Investment Funds instruments.
- When Euronext Cash Market Operations enters an iNAV on behalf of the fund manager agent after iNAV fixing time for the Investment Funds groups traded on TCS.

3.2.6.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '241' Price Update.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	241 - Price Update
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
Price	32	4	Binary Int.	Last Price of the trading day (To be calculated with the ScaleCode)
HighestPrice	36	4	Binary Int.	Highest Price traded during the day (To be calculated with the PriceScaleCode)
LowestPrice	40	4	Binary Int.	Lowest Price traded during the day (To be calculated with the PriceScaleCode)
VariationLastPrice	44	4	Binary Int. (signed)	Percentage variation of today's price/last reference price (To be calculated with the VariationScaleCode)
SystemID	48	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	52	2	Binary Int.	Number of micro seconds. To be combined with SourceTime field.

Field Name	Offset	Size (Bytes)	Format	Description
TypeOfPrice	54	2	Binary Int.	Market- Code identifying the type of price that is updated. Valid values: 0 - Trade cancellation. 2 - Ask price. 3 - Bid price. 4 - First traded price (one quotation only). 5 - Market imbalance ask price. 6 - Market imbalance bid price. 30 - Change of first price. 31 - New high price. 32 - New low price. 33 - New last price. 34 - New previous day's closing price. 41 - Warrant opening indicative value. 42 - Warrant closing indicative value. 51 - Indicative price.
Filler	*56*	*2*	*Binary Int.*	*For future use.*
PriceScaleCode	58	1	Binary Int.	Applicable to all prices in the message
VariationScaleCode	59	1	Binary Int.	Available for the price variation in the message

3.2.7. TCS Trade - 242 Message

3.2.7.1. Message Overview

A TCS trade message indicates either:

- The automatic matching of two trade declarations outside the central order book (the buyer's and the seller's), or the reception of a single valid declaration made in the names of the buyer and the seller.
- The cancellation of a TCS trade.

3.2.7.2. Message Sending Rules

This message is sent:

- When TCS receives a single valid declaration or two declarations which can be matched automatically (the publication of a trade can be delayed);
- A TCS trade is cancelled
- A digest on TCS activity for the day, built on the basis of TCS statements

3.2.7.3. Message Structure

The table below describes the body fields of a Equities Trade message, MsgType = '242' TCS Trade.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	242 - TCS Trade.
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170.
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
Price	32	4	Binary Int.	Price at which the security was negotiated(To be calculated with the PriceScaleCode).
Volume	36	4	Binary Int.	Number of negotiated securities.
FirstPrice	40	4	Binary Int.	Summary Activity (TCSSubtypeCode='R') – First price(To be calculated with the PriceScaleCode).
LastPrice	44	4	Binary Int.	Summary Activity (TCSSubtypeCode='R') – Last price(To be calculated with the PriceScaleCode).
HighestPrice	48	4	Binary Int.	Summary Activity (TCSSubtypeCode='R') – High price(To be calculated with the PriceScaleCode).
LowestPrice	52	4	Binary Int.	Summary Activity (TCSSubtypeCode='R') – Low price(To be calculated with the PriceScaleCode).
CumulativeCapital	56	4	Binary Int.	Summary Activity (TCSSubtypeCode='R') – Cumulative amount of capital exchanged for a given trading day and instrument (K Euro).
CumulativeNumberSecurities	60	4	Binary Int.	Summary Activity(TCSSubtypeCode='R') –Cumulative quantity for a given trading day and instrument.
SystemID	64	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	68	2	Binary Int.	Number of micro seconds. To be combined with SourceTime field.
TCSTradeID	70	16	ASCII Str.	Unique Identifier of the Trade (European Market).
TradingVenue	86	4	ASCII Str.	Identifies the Euronext market on which a security is traded by its Market Identification Code), valid values: - Initial MIC of instrument if TCS Operation Type = D, E,H, I. - Other Euronext MIC but different of the initial MIC of instrument if TCS Operation Type = R.
DateOriginalDecl	90	8	ASCII Str	Date declarations were matched.

Field Name	Offset	Size (Bytes)	Format	Description
TimeOriginalDecl	98	6	ASCII Str	Time declarations were matched.
StartTimeVwap	104	6	ASCII Str.	Start time for the Volume Weight Average price computation period.
EndTimeVwap	110	6	ASCII Str.	End time for the Volume Weight Average price computation period.
EffectiveDateIndicator	116	1	ASCII Ch.	Declarations making up the trade were introduced on the day. Valid values: '0' - date that the seller's declaration was received. '1' -Trading day preceding day seller's declaration was received.
BlockTradeCode	117	1	ASCII Ch.	Trades relates to a block or a negotiated deal following MiFID rules. Valid values 'B' - Block Trade 'N' - Regular trade or Negotiated deal Null - undefined.
TradeCancelIndicator	118	1	ASCII Ch.	Indicates if the trade specified was cancelled Valid values: '0' - Valid trade '1' - Cancelled trade Null - Summary message
TradeType	119	1	ASCII Ch.	Type of Operation Valid values: 'D' - Delta Neutral Liffe Connect 'E' - Market VWap operation 'H' - Out of Market 'I' - Investment funds 'R' - Secondary listing place.
FinancialMarketCode	120	1	ASCII Ch.	Financial Market as defined the Investment Service Directive. Valid values: 'S' - Paris 'B' - Brussels 'A' - Amsterdam 'P' - Lisbon
TCSSubtypeCode	121	1	ASCII Ch.	Meaning of this message. 'R' - Complete summary 'S' - Trade without summary
PriceScaleCode	122	1	Binary Int.	Applicable to all prices in the message
Filler	*123*	*1*	*Binary Int.*	*For future use.*

3.2.8. Trade Publication - 243 Message

3.2.8.1. Message Overview

A Trade publication message is sent by TCS whenever an OTC trade under the MiFID regulation is sent to TCS. The trade is either executed outside the regulated market by trading members that are MiFID compliant either issued by the matching of 2 trade declarations applied on OTC operations.
Each new trade is stored on TCS. A trade publication message is sent through the Market Data dissemination flows if and at the moment that it is defined for publication.

This message is either sent for a trade creation or cancellation.

3.2.8.2. Message Sending Rules

This message is sent:

- When TCS receives a single valid declaration or two declarations which can be matched automatically (the publication of a trade can be delayed);
- A Trade publication is cancelled

3.2.8.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '243' Trade Publication.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	243 - Trade Publication
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
Price	32	4	Binary Int.	Price at which the instrument was traded, reported to Euronext(To be calculated with the PriceScaleCode).
Quantity	36	4	Binary Int.	Number of traded units
PriceMultiplier	40	4	Binary Int.	Number of units of the financial instrument that are contained in a trading lot(To be calculated with the MultiplierScaleCode)
SystemID	44	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	48	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
filler	*50*	*2*	*Binary Int.*	*For future use.*
TradingDate	52	8	ASCII Str.	Date at which the trade was executed
TradingTime	60	6	ASCII Str.	Time at which the trade was executed
ReportingDate	66	8	ASCII Str.	Date at which the trade was reported to NYSE Euronext Cash markets
ReportingTime	74	6	ASCII Str.	Time at which the trade was reported to NYSE Euronext Cash markets
ISIN	80	12	ASCII Str.	ISIN code of the Instrument.
QuantityNotation	92	4	ASCII Str.	Indication to whether the quantity is a number of units, the nominal value of bonds or a number of contract.
TradeReference	96	30	ASCII Str.	Reference of the trade reported to Euronext.
TCSTradeId	126	16	ASCII Str.	Unique reference of the trade provided by Euronext.
TradeActionIndicator	142	1	ASCII Ch.	Type of update for the trade. Valid values '1' - Insertion, '0' - Deletion
PriceNotation	143	3	ASCII Str.	Code of the currency (ISO 4217-3A)
Venue	146	11	ASCII Str.	Indicates the venue where the trade took place. Possible value: - BIC Code (ISO 9362) - OTC (Over the counter)

Field Name	Offset	Size (Bytes)	Format	Description
				- SI (Systematic Internalizer) - XSMP (SmartPool)
DelayedIndicator	157	1	ASCII Ch.	Subject to a deferred publication or not. '0': - Not Delayed, '1' - Deferred Null - Not available
PriceScaleCode	158	1	Binary Int.	Applicable to all prices in the message
MultiplierScaleCode	159	1	Binary Int.	To be combined with PriceMultiplier

3.2.9. Settlement Price - 244 Message

3.2.9.1. Message Structure

The settlement price message defines the daily clearing price of the underlying instrument for a security lending contract (that means the interest rate of the Lending Borrowing contract that is dealt the day the message is sent).

3.2.9.2. Message Sending Rules

This message is sent every morning before the beginning of the trading session for all the securities eligible to the lending/borrowing market.

3.2.9.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '244' Settlement Price.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	244 - Settlement Price
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the Trade generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
Price	32	4	Binary Int.	Settlement price(To be calculated with the ScaleCode).
SystemID	36	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	40	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
PriceScaleCode	42	1	Binary Int.	Applicable to all prices in the message
Filler	*43*	*1*	*Binary Int.*	*For future use.*

3.2.10. Auction Summary - 245 Message

3.2.10.1. Message Overview

An Auction Summary message summarizes an instrument's opening or auction trades. The Auction Summary message is sent by the trading engine after an instrument auction that has effectively traded, in order to summarize the auction, or after the first trade(s) if it/they occurred during continuous trading.

3.2.10.2. Message Sending Rules

This message indicates, for each security, the cumulative quantity traded during an auction/a first trade.

3.2.10.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '245' Auction Summary.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	245 - Auction Summary
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	Specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Example: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
FirstPrice	32	4	Binary Int.	First traded price for the current trading day (To be calculated with the PriceScaleCode)
LastPrice	36	4	Binary Int.	Most recent traded price for the current trading day (To be calculated with the PriceScaleCode)
HighestPrice	40	4	Binary Int.	Highest traded price for the current trading day (To be calculated with the PriceScaleCode)
LowestPrice	44	4	Binary Int.	Lowest traded price for the current trading day (To be calculated with the PriceScaleCode)
CumulativeQuantity	48	4	Binary Int.	Cumulative quantity traded for the current trading day.
Variation	52	4	Binary Int. (signed)	Percentage variation of the last trade price against the previous day's reference price for the instrument concerned, ie the last known price or the last price indication (if an indication was entered after the last traded price).(To be calculated with the VariationScaleCode).
SystemID	56	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	60	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
TypeOfLastPrice	62	2	Binary Int.	Indicates the message trade type. Valid values: 4 - 1st traded price, 7 - nth traded price (resume after reservation).
TickDirection	64	1	ASCII Ch.	Indicates how the last price has moved as compared to the preceding last traded price. Valid values: '+' - Rising, '-' - Falling, '0' - Unchanged.
InstrumentValuationPrice	65	1	ASCII Ch.	Indicates whether the data item "last price" is a valuation price. Valid values: '0' - Not a valuation price, '1' - Valuation price.
PriceScaleCode	66	1	Binary Int.	Applicable to all prices in the message
VariationScaleCode	67	1	Binary Int.	Available for the price variation in the message

3.2.11. Notice Of Interest - 246 Message

3.2.11.1. Message Overview

This message is specifically for SmartPool members, non members will not be permissioned to receive this message. One message is sent to inform SmartPool members about the presence of orders waiting for matching in different cases (see below).

3.2.11.2. Message Sending Rules

This message is sent:

- When an order (New order or order modification) is entered with order validity type different from FOK (Fill or Kill) and Routing indicator = N or R and Limit Price greater than Mid-Point price (above for buy orders, below for sell orders). The notice indicator flag is set 1.

- When an order is modified with order validity type different from FOK (Fill or Kill) and Routing indicator = N or R and Total Quantity greater than the one defined in the previous order (new order or modified order that generated or not a Notice of Interest message). The notice indicator flag is set 1.

- When the order book empties. The notice indicator flag is set 0.

Field	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary	246 - Notice of interest Message
SymbolIndex	20	4	Binary	Index of the Symbol- Stock representation
SourceTime	24	4	Binary	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
SourceTimeMicroSecs	32	2	Binary	Number of micro seconds. To be combined with SourceTime field.
InterestIndicator	34	1	ASCII Ch.	Indicates if the SmartPool activity is able to generate trades or at the contrary there is no more intention to trade on SmartPool for a given instrument. Permitted values Alphanumeric, always provided. 0 There is no more interest on SmartPool (there was already an interest but the order book is now empty). 1 New interest (there is at least one order able to generate a trade on SmartPool).
Filler	35	1	ASCII Ch.	*For future use.*

3.2.12. VWAP – Closing Price - 247 Message

3.2.12.1. Message Overview

This message is used exclusively at Qatar Exchange.

This message contains the real time Volume-Weighted Average Price for an instrument, along with few statistical information related to the cumulated traded value and volume. It also transports the "current" Closing price, that is, the price at which the instrument would close, would the market stop immediately. (The closing price can be the Last Traded Price or the VWAP).

3.2.12.2. Message Sending Rules

This message is sent in near RealTime, every NN seconds, NN being a global system parameter that cannot be smaller than 15 seconds:

In addition, this message is sent at the End of Day – once the referential system batch has been performed – and provides the Last Adjusted Closing Prices for all securities (taking into account the Corporate Actions that may have taken place).

3.2.12.3. Message Structure

The table below describes the body fields of an Equities Trade message, MsgType = '247' VWAP-Closing Price.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	247 – VWAP / Closing Price Update
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
SendingIndicator	32	2	Binary Int.	Indicate the "sending mode" that can be: 1 - Real Time 2 - Batch mode.
ClosingPriceRule	34	2	Binary Int.	Indicate the Closing Price calculation rule, that can be: 1 - LTP (Last Traded Price) 2 - VWAP (Volume Weighted Average Price)
ClosingPrice	36	4	Binary Int.	Current VWAP/Closing Price (To be calculated with the PriceScaleCode)
AdjustedClosingPrice	40	4	Binary Int.	Adjusted VWAP/Closing Price (To be calculated with the PriceScaleCode)
NbTrades	44	4	Binary Int.	Number of Trades executed on this Instrument since the beginning of the Trading Day
QtyShares	48	4	Binary Int.	Cumulated Quantity traded (number of shares) on this instrument since the beginning of the Trading Day
AmountTraded	52	4	Binary Int.	Total cumulated amount traded on this instrument since the beginning of the Trading day (To be calculated with the PriceScaleCode)
SystemID	56	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
PriceScaleCode	60	1	Binary Int.	Applicable to all prices in the message
Filler	*61*	*3*	*Ascii Str.*	*Reserved for future use*

3.3. *Quotes and BBO10*

3.3.1. Overview

The Equities Quotes service uses the push-based publishing model. This means that data will be published based on its availability. Once a Quote is available, it will be published to clients.

The Equities Quote message reflects a configurable number of highest bids and lowest offers (e.g. BBO1, BBO10 ...) in each traded security.

List of the message types in the Equities Quotes Feed:

- 140 - Quotes
- 141 – WAS
  Not used for Qatar Exchange

3.3.2. Packet Header Format

All messages are preceded by a standard header format. The following table describes the header fields of a Quotes message.

Field	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet including the 16-byte packet header.
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. If all messages within a packet are of the same message type, the packet type will be equal to that message type. If not, the packet type will be set at 994. Possible values: 994 - Generic Quotes Message. 140 - Quotes Message - 68 Bytes 141 - WAS message - 52 Bytes
PacketSeqNum	4	4	Binary Int..	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and monotonically and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in milliseconds indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC.
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream. Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

3.3.3. Quotes - 140 Message

3.3.3.1. Message Overview

A Quotes message indicates the modification of one or more of the N best limits (N being the number of limits disseminated for the given exchange[5]) for an instrument and, during the Pre-Opening phase, the modification of the "market summary" for the instrument.

3.3.3.2. Message Sending Rules

This message is sent:

- Each time one or more of the N best limits for an instrument is modified or when during the call (Pre-Opening) phase, the "market summary" is modified.

- The "market summary" for an instrument is the summary of the orders that would be executed if the auction took place at the moment that this message was sent. It is meaningful only during a Call phase or when the instrument is in Halted mode w/ acceptation of orders, when an IMP is determined. The market summary is not provided if the IMP is equal to the first limit.

3.3.3.3. Message Structure

The table below describes the body fields of an Equities Quotes message, MsgType = '140' Quotes.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int..	140 - Quotes
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
SourceTime	28	4	Binary Int.	This field specifies the Quote generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170.
QuoteLinkID	32	4	Binary Int.	Not used
AskPrice	36	4	Binary Int.	Ask Price for Quote (To be calculated with the ScaleCode) Specific values: FF FF FF FE: Market Order or Market To Limit Order FF FF FF FD At Opening Order (NSC only)
AskSize	40	4	Binary Int.	Total number of shares requested in sell orders at the ask price.
BidPrice	44	4	Binary Int.	Bid Price for Quote (To be calculated with the ScaleCode) Specific values: FF FF FF FE: Market Order or Market To Limit Order FF FF FF FD At Opening Order (NSC only)
BidSize	48	4	Binary Int.	Total number of shares requested in buy orders at the bid price.
SystemID	52	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
NumberAskOrders	56	2	Binary Int.	Number of sell orders at the ask price
NumberBidOrders	58	2	Binary Int.	Number of buy orders at the bid price
SourceTimeMicroSecs	60	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
TypeOfAskPrice	62	1	Binary Int.	Valid values: 0 - Limit order 1 - Market order 2 - At Opening order (NSC only)
TypeOfBidPrice	63	1	Binary Int.	Valid values: 0 - Limit order

[5] For Euronext, the 10 Best Limits are disseminated. For Qatar Exchange, the 10 Best Limits are disseminated.

Field Name	Offset	Size (Bytes)	Format	Description
				1 - Market order 2 - At Opening order (NSC only)
QuoteCondition	64	1	ASCII Ch.	Liquidity provider presence indicator: Valid values: '0' - No Liquidity provider (LP) '1' - LP only on Ask side '2' - LP only on Bid side '3' - LP on Ask and Bid sides Null Not provided/Not applicable
QuoteNumber	65	1	Binary Int.	Indicates the level in the order book for the given aggregate quote, derived from it's price value. Level '0' is dedicated to the market summary (The market summary for an instrument is the summary of the orders that would be executed if the opening of an instrument took place at the moment when this message is sent – applicable only in Call mode when an IMP has been determined).
PriceScaleCode	66	1	Binary Int.	Applicable to all prices in the message
Filler	*67*	*1*	*Ascii Str*	*For future use.*

3.3.4. Weighted Average Spread (WAS) - 141 Message

This message is used only for Euronext markets.

3.3.4.1. Message Overview

The Weighted Average Spread is an evaluation of the buying and selling average prices of an instrument, for a predetermined amount, given by the instrument's attributed Money Amount for calculation of the WAS. This message is mainly provided for institutional investors interested in the Euronext's main instruments to be traded in blocks.

The Weighted Average Spread is also used by the TCS system to check that the declarations of trades outside the central order book comply with Euronext's rule book.

3.3.4.2. Message Sending Rules

This message is sent:

- For opening of an instrument with its group: if (the instrument's Theoretical Opening Price could be determined and belongs to the spread defined by the freezing thresholds) and (the Normal Money Amount of Block attribute is not zero), then a Weighted Average Spread message is sent.
- For opening of an instrument : Same than for Opening of an Instrument with its Group
- For entering and processing an order in continuous market phase:
    - If the instrument's attribute Normal Money Amount of Block is not zero, and if the Weighted Average Spread has fluctuated once or more during the last 30 seconds (due to either the incoming order and / or one the previous orders), a Weighted Average Spread message is sent. (N.B.: the Weighted Average Spread broadcasting delay is a parameter).
    - if the instrument is reserved, if its attribute Normal Money Amount of Block is not zero, a Weighted Average Spread message is sent.

3.3.4.3. Message Structure

The table below describes the body fields of an Equities Quotes WAS message, MsgType = '141' Weighted average spread.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 byte MsgSize field.
MsgType	18	2	Binary Int.	141 - Weighted Average Spread
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically
SourceTime	28	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Example: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170.
BuyingPrice	32	4	Binary Int.	Buying price of the Weighted Average Spread (To be calculated with the ScaleCode).
SellingPrice	36	4	Binary Int.	Selling price of the Weighted Average Spread (To be calculated with the ScaleCode).
MoneyAmount	40	4	Binary Int.	Money amount for calculation of the WAS (To be calculated with the MoneyScaleCode)
SystemID	44	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	48	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
PriceScaleCode	50	1	Binary Int.	Applicable to all prices in the message
MoneyScaleCode	51	1	Binary Int.	To be combined with MoneyAmount.

3.4. *Order Book*

3.4.1. Overview

The Equities OrderBook service uses the push-based publishing model. This means that data will be published based on its availability. Once information is available, it will be published to clients.

List of the messages in the Equities OrderBook:

- 230 - Order update
- 231 - Orderbook retransmission delimiter (used when the Public OrderBook is entirely redisseminated for one or several given instruments)

3.4.2. Packet Header Format

All messages are preceded by a common header format. The table describes the header fields of an OrderBook message.

Field	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet including the 16-bytes packet header.
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. 230 - Order Update message - 80 Bytes. 231 - Orderbook retransmission delimiter message - 32 Bytes.
PacketSeqNum	4	4	Binary Int.	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and monotonically and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in millisecond indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC.
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream. Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

3.4.3. Order Update / Market Sheet - 230 Message

3.4.3.1. Message Overview

The Order update message generated by the trading engines NSC or UTP, indicates the creation or modification of a line in the "market sheet" (a.k.a. "market depth" or "Public Order Book") of normal orders for an instrument. It is also used when the Market Sheet of normal orders is rebroadcast. The deletion of an order from the Market Sheet of normal orders is indicated via a Delete N lines from the Market Sheet message:

- This message takes into account any order type, except Stop Loss and Stop Limit (Stop orders).
- Stop orders are not broadcasted to the market participants until they are triggered.

3.4.3.2. Message Sending Rules

This message is sent:

- In the morning, when NSC or UTP is initialized, to retransmit orders remaining in the book from previous days (taking into account expired orders and orderbook purges). This is known as the 'orderbook retransmission' or 'market sheet retransmission' – action type - Y.
- During the day, each time an order introduced by a member firm modifies the market sheet (for example, when creating, modifying or changing the priority of an order), except during Auction processing.

To be noted:

- **SymbolIndex+ OrderDate+ OrderID uniquely identifies an order**
- Market sheet sequencing:
    - Orders must be arranged according to:
        1. Order type (OrderType): Priorities are first Market orders, second Market to limit and Opening orders and third Limit and Peg orders.
        2. then Order price (Price)
        3. [for Qatar Exchange only: then Order origin]
        4. then Order priority timestamp (OrderPriorityDate+OrderPriorityTime+OrderPriorityMicroSecs)

3.4.3.3. Message Structure

The table below describes the body fields of an Equities OrderBook message, MsgType = '230' Order Update

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 bytes MsgSize field.
MsgType	18	2	Binary Int.	230 - Order Book
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the time when the Order Update is generated. Note, when an order is added, the SourceTime represents the order entry time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	28	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
Price	32	4	Binary Int.	Price (To be calculated with the PriceScaleCode)
AggregatedVolume	36	4	Binary Int.	Total quantity at the price of the current order
Volume	40	4	Binary Int.	Remaining displayed quantity of the order.
LinkID	44	4	Binary Int.	The LinkID identifies a unique transaction in the matching and allows you to correlate execution reports and quotes to the last sale. This field is populated only when an execution occurs. (optional).
OrderID	48	4	Binary Int.	Identifies the uniqueness of the order, when combined with OrderDate.
SystemID	52	4	Binary Int.	The ID of the originating Exchange/System of the message. Refer to appendix E for possible values.
SourceTimeMicroSecs	56	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
NumberOrders	58	2	Binary Int.	Number of orders in the Book at the price of the current order
Side	60	1	ASCII Ch.	Indicates the side of the order.

				Valid values: 'B': Buy 'S': Sell Null: Not provided
OrderType	61	1	ASCII Ch.	Type of Order. Valid values: For instruments traded on NSC: 'O' At Opening Price Order 'L' Limit Order, 'X' Market Order. For instruments traded on UTP: '1' Market order '2' Limit order 'K' Market to limit order 'P' Peg order
ActionType	62	1	ASCII Ch.	This field identifies why the volume (size) at the (current order) price was modified. Values reserved for future use (US markets): 'O', 'C', 'E', 'X', 'Z' Valid values: 'A' - New order 'M' - Modification of existing order (case of partial fill) 'D' - Deletion of order identified by OrderID 'P' - Deletion of order identified by OrderID and all preceding orders 'F' - Deletion of all orders for the given instrument (depending on the side. If side is not provided, it means both) 'Y' - Retransmission of all orders for the given instrument
PriceScaleCode	63	1	Binary Int.	Applicable to all prices in the message
				The following fields will be removed in the future for Euronext
OrderDate	64	4	Binary Int.	Date of order (YYYYMMDD). To be combined with OrderId.
OrderPriorityDate	68	4	Binary Int.	Date giving the priority of the order (YYYYMMDD)
OrderPriorityTime	72	4	Binary Int.	Time giving the priority of the order (HHMMSSsss)
OrderPriorityMicroSecs	76	2	Binary Int.	Number of micro seconds in the current millisecond
OrderOrigin	78	1	ASCII Ch.	QATAR EXCHANGE ONLY: Order Origin (so that the FIFO-Origin priority can be determined). Valid values: 1 Client 2 House 6 Liquidity Provider 8 QE or Broker Staff 9 Board Member
Filler	79	1	*Binary Int.*	

3.4.4. Orderbook Retransmission Delimiter - 231 Message

3.4.4.1. Message Overview

The Orderbook retransmission delimiter message is set to clean then update the market sheet with the orders from previous days still remaining in the order book.

3.4.4.2. Message Sending Rules

This message is sent at the beginning of the trading session, and allows member to clean then update the market sheet with the orders remaining in the orderbook

3.4.4.3. Message Structure

The table below describes the body fields of an Equities OrderBook message, MsgType = '231' Orderbook retransmission delimiter.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body, excluding the 2 bytes MsgSize field.
MsgType	18	2	Binary Int.	231 - Orderbook retransmission delimiter Message
SourceTime	20	4	Binary Int.	This field specifies the time when the Order Update is generated. Note, when an order is added, the SourceTime represents the order entry time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
SourceSeqNum	24	4	Binary Int.	This field specifies the sequence number assigned by the source system to this message. Please note that while the sequence number increases serially, it does not increase monotonically.
TradingEngineID	28	2	ASCII Str.	Code identifying the trading engine. Valid values: C1: UTP for Equities
InstanceID	30	1	Binary Int.	Indicates the number of instances for a given trading engine rebroadcasting orderbooks.
RetransmissionIndicator	31	1	ASCII Ch.	Indicates the status of the retransmission for the instance of the trading engine: 'B' Beginning of the retransmission 'E' End of the retransmission

3.5. *Indices*

3.5.1. Overview

The Equities Indices service uses the push-based publishing model. This means that data will be published based on its availability. Once information is available, it will be published to clients.

List of the message types in the Equities Indices feed:

- 542 - Real Time Index
- 543 - Index Summary
- 544 - Index Composition

3.5.2. Packet Header Format

All messages are preceded by a common packet header format. The table below describes the header fields of an Indices messages.

Field Name	Offset	Size (Bytes)	Format	Description
PacketLength	0	2	Binary Int.	Length of the packet including the 16-byte packet header.
PacketType	2	2	Binary Int.	Identifier for the type of data contained in the packet. If all messages within a packet are of the same message type, the packet type will be equal to that message type. If not, the packet type will be set at 996. Possible values: 996 - Generic Indices Message. 542 - Real-time index – 68 Bytes 543 - Index summary – 108 Bytes 544 - Index composition – 68 Bytes
PacketSeqNum	4	4	Binary Int.	This field contains the packet sequence number. It is unique for each broadcast stream (multicast group) and is used for gap detection. It increases serially and monotonically and is reset to 1 at the beginning of each trading day. The PackSeqNum is unique for packets containing market data only. Heartbeats inherit their sequence number from the last market data packet or packet sequence number reset packet.
SendTime	8	4	Binary Int.	Timestamp in millisecond indicating the packet broadcast time. The number represents the number of milliseconds since midnight of the last Sunday 00:00 UTC.
ServiceID	12	2	Binary Int.	Numeric value identifying the broadcast stream. Possible values are described in Feed Configuration descriptions
DeliveryFlag	14	1	Binary Int.	Indicates delivery method. 0 Real Time message (Uncompressed) 2 Retransmission message (Uncompressed) 17 Refresh message (ZliB Compressed)
NumberMsgEntries	15	1	Binary Int.	The number of messages that are contained within the packet.

3.5.3. Real time Index - 542 Message

3.5.3.1. Message Overview

The real time index message handles the real-time characteristics of an index: the level (a.k.a. value) of the index (or of the forerunner), the variations in the index level, the type of index level (first, real-time, forerunner...), and various indicators for the instruments that make up the index. This message is sent for both:

- Stock indices
- Net Asset Value (NAV) of an ETF

3.5.3.2. Message Sending Rules

The sending of these messages for a given index is conditioned by a flag configured at the index level. These conditions and the nature of these messages that are sent for each index are dependent on two factors:

- The broadcast mode of the index
- The current calculation phase of the index. These two factors are explained below.

There are three broadcasting modes for msg.542:

- Continuous: Calculated index levels are broadcast periodically, at a frequency that can be configured for each index. The frequency must be a multiple of the base calculation frequency (15 seconds).
- Discontinuous: A single broadcast before the provisional closing phase, occurring at a time (a "fixed time") that can be configured for each index.
- At closing only: no broadcast before the provisional closing phase.

Euronext uses the 3 modes, depending on the index.

Qatar Exchange will use only the "Continuous" mode. Also, Qatar Exchange will not use the "forerunner" type.

3.5.3.3. Message Structure

The table below describes the body fields of an Equities Indices message, MsgType = ‘542’ Real Time Index.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body. MsgSize excluded.
MsgType	18	2	Binary Int.	542 - Real Time Index
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170.
IndexLevel	28	4	Binary Int.	The value of the last level for the index that is the subject of this message To be calculated with the LevelScaleCode).
ForerunnerLevel	32	4	Binary Int. (signed)	The percentage of variation of the last Forerunner level for the index that is the subject of this message, if any (To be calculated with the LevelScaleCode)
SessionHigh	36	4	Binary Int.	Highest level of the day (To be calculated with the LevelScaleCode)
SessionLow	40	4	Binary Int.	Lowest level of the day (To be calculated with the LevelScaleCode)
PercentageOfCapitalization	44	4	Binary Int.	Percentage of capitalization for the active instruments in the index (To be calculated with the PercentageScaleCode)
VariationFromPreviousDaysPrice	48	4	Binary Int. (signed)	Percentage of variation for day's index / Previous day's reference (To be calculated with the VariationScaleCode)

Field Name	Offset	Size (Bytes)	Format	Description
SystemID	52	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	56	2	Binary Int.	Number of micro seconds. To be combined with SourceTime..
NumOfSecuritiesQuoted	58	2	Binary Int.	Number of instruments in the index that have actually traded (active instruments)
IndexLevelCode	60	1	ASCII Ch.	Index Level Code. Possible values for Euronext: '0' - Indicative index '1' - First index '2' - Real-time index '3' - Forerunner (éclaireur) index '4' - Not used '5' - Closing Reference index '6' - Options clearing index '7' - Options liquidation index '8' - Pre-Opening forerunner index '9' - Opening forerunner index 'A' - Opening index 'B' - Temporary reference for opening index 'C' - Opening reference index. Possible values for Qatar Exchange: '1' - First index '2' - Real-time index '5' - Closing Reference index
TypeOfLevel	61	1	ASCII Ch	'1' - Price level '2' - Gross profitability level '3' - Net profitability level
LevelScaleCode	62	1	Binary Int.	To be combined with IndexLevel, ForerunnerLevel, SessionHigh, SessionLow
PercentageScaleCode	63	1	Binary Int.	To be combined with PercentageOfCapitalization
VariationScaleCode	64	1	Binary Int.	To be combined with VariationFromPreviousDaysPrice
RebroadcastIndicator	*65*	*1*	*Ascii Str*	On occasion the 542 message carrying the closing level of the previous day is rebroadcast at the beginning of the trading day. This indicator should be used to validate if this message applies to the previous days or current days index level. Possible values are as follows: 0 – Current Day 1 – Previous Day *Please note the definition of this field as described in chapter 7.1 is added for future use.*
Filler	*66*	*2*	*Ascii Str*	*For future use.*

3.5.4. Index Summary - 543 Message

3.5.4.1. Message Overview

The index summary message provides a summary of the data calculated in a stock index for a given day - the opening level, the closing level, the high and the low that were broadcast for the index.

3.5.4.2. Message Sending Rules

This message is sent:

- Every end of trading day, for each index of the type "stock index" (this rules excludes Bond Indices, ETF - Index Estimate Indices and ETF - NAV indices) whose index summary broadcast flag is set to 'Yes', two types of index summary messages are sent:
    - The first is sent when the index enters the provisional closing phase.
    - The second is sent when the index enters the final closing phase.
- When manual entry of an index summary generates and sends such a message.
- For index levels extracted from a third-party data flow and rebroadcast via UTP-MD: When the time of end of rebroadcasting configured for the index-type instrument has been reached, It is generated and sent one Index summary message.

3.5.4.3. Message Structure

The table below describes the body fields of an Equities Indices message, MsgType = '543' Index summary.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body. MsgSize excluded.
MsgType	18	2	Binary Int.	543 - Index Summary
SymbolIndex	20	4	Binary Int.	UTP-MD proprietary identification of the instrument
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
PreliminaryOpeningLevel	28	4	Binary Int.	Preliminary Opening index level(To be calculated with the LevelScaleCode).
PreliminaryOpeningTime	32	4	Binary Int.	Time of provisional opening index level. Number of milliseconds since midnight
FirstLevel	36	4	Binary Int.	First index level broadcast (To be calculated with the LevelScaleCode).
FirstTime	40	4	Binary Int.	Time of provisional first index level. Number of milliseconds since midnight
OpeningReferenceLevel	44	4	Binary Int..	Reference opening index level (To be calculated with the LevelScaleCode).
OpeningReferenceTime	48	4	Binary Int.	Time of provisional opening reference index level. Number of milliseconds since midnight
ClosingReferenceLevel	52	4	Binary Int.	Reference closing index level (To be calculated with the LevelScaleCode).
ClosingReferenceTime	56	4	Binary Int.	Time of provisional closing reference index level. Number of milliseconds since midnight
PercentVariationPrevClos	60	4	Binary Int. (signed)	Percentage of variation for last index level vs. previous day's closing (To be calculated with the VariationScaleCode).
HighLevel	64	4	Binary Int.	Highest index level (To be calculated with the LevelScaleCode).
HighTime	68	4	Binary Int.	Time of provisional highest index level. Number of milliseconds since midnight
LowLevel	72	4	Binary Int.	Lowest index level (To be calculated with the LevelScaleCode).
LowTime	76	4	Binary Int.	Time of provisional lowest index level. Number of milliseconds since midnight

ClearingLevel	80	4	Binary Int.	Reference daily settlement index level (To be calculated with the LevelScaleCode).
ClearingTime	84	4	Binary Int.	Time of provisional daily settlement index level. Number of milliseconds since midnight
LiquidationLevel	88	4	Binary Int.	Reference at-expiration settlement index level (To be calculated with the LevelScaleCode).
LiquidationTime	92	4	Binary Int.	Time of provisional expiation settlement index level. Number of milliseconds since midnight
SystemID	96	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	100	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
TypeOfLevel	102	1	ASCII Ch	'1' -Price level '2' - Gross profitability level '3' - Net profitability level
LevelScaleCode	103	1	Binary Int.	To be combined with all level fields
VariationScaleCode	104	1	Binary Int.	To be combined with PercentVariationPrevClos
Filler	*105*	*3*	*Ascii Str*	*For future use.*

3.5.5. Index Composition - 544 Message

3.5.5.1. Message Overview

An Index Composition message transmits all or part of the list of instruments making up an index:

- Instrument IDs (long ID and mnemonic),
- Percentage of each instrument in the index capitalization,
- Adjustment coefficient for the capitalization of each instrument,
- Total number of instruments making up the index.

3.5.5.2. Message Sending Rules

This message is sent at the beginning of each trading day, when the index platform is initialized.

For each index, there is one message per component.

3.5.5.3. Message Structure

The table below describes the body fields of an equities Indices message, MsgType = '544' Index Composition.

Field Name	Offset	Size (Bytes)	Format	Description
MsgSize	16	2	Binary Int.	Length of the message body. MsgSize excluded.
MsgType	18	2	Binary Int..	544 - Index Composition
SymbolIndex	20	4	Binary Int..	UTP-MD proprietary identification of the index itself
SourceTime	24	4	Binary Int.	This field specifies the message generation time. The number in this field represents the number of milliseconds since midnight of the same day. Ex: If SourceTime = 13:12:56 secs, 170ms and 30 microsecs, this field will contain 47576170
InstrumentWeight	28	4	Binary Int.	Instrument's weight (in %) in previous day's reference capitalization of the index (To be calculated with the PriceScaleCode)
InstrumentFactor	32	4	Binary Int.	Capitalization adjustment coefficient of an instrument in the index (To be calculated with the PriceScaleCode)
PreviousClose	36	4	Binary Int.	Previous day's reference closing level for the index (To be calculated with the LevelScaleCode).
SystemID	40	4	Binary Int.	The ID of the originating Exchange/System of the message
SourceTimeMicroSecs	44	2	Binary Int.	Number of micro seconds. To be combined with SourceTime.
NumberOfConponents	46	2	Binary Int.	Total number of instruments in the index
InstrumentIDOfComponent	48	12	ASCII Str.	Trading code (if any, else ISIN code) ID of index component
Mnemo	60	5	ASCII Str.	Mnemonic code of index component.
IndexFrequency	65	1	ASCII Ch.	Frequency of index calculation: 'C' - Continuous calculation 'D' - Discontinuous calculation.
PriceScaleCode	66	1	Binary Int.	To be combined with InstrumentWeight and InstrumentFactor
LevelScaleCode	67	1	Binary Int.	To be combined with PreviousClose

4. Production Feed Configuration

4.1. *Introduction*

4.1.1. Data Content

The information supplied in this chapter applies to the UTP-MD Euronext Cash markets services only.

A Similar section will be written for Qatar Exchange.

4.1.2. Data Delivery

- UTP-MD Cash is available via MMBA and via SFTI. The configuration data is the same for both.
- Channel 113 (SmartPool – members only) is currently only available via SFTI.

4.2. *Rendez-Vous Point (all connectivity)*

Production Platform	
RP1 platform Line A	RP2 Line B
S: 156.48.121.2 Associated to: G: 224.0.52.x	S: 156.48.121.55 Associated to: G: 224.0.53.x

4.3. *SFTI / MMBA - Feed Configuration*

4.3.1. Production Feed Configuration

The tables below show the UDP/IP configuration for the real time feed and the refresh feed, including line A and line B, for the Production platform.
In this table the following indicators are used:
PS: Primary Data Centre Source IP
SS: Secondary Data Centre Source IP (only used in the event of a disaster recovery situation)
G = Group
P = UDP Port

4.3.1.1. Real Time Configuration

Service ID	Line A	Line B
'101' Euronext Equities – Referential Data	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.4 P: 21028	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.4 P: 21029
'102' Euronext Equities – Trades	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.5 P: 21030	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.5 P: 21031
'103' Euronext Equities – Quotes	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.6 P: 21032	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.6 P: 21033
'104' Euronext Equities – Orders	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.7 P: 21034	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.7 P: 21035
'105' Euronext Warrants – Trades	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.8 P: 21036	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.8 P: 21037
'106' Euronext Warrants – Quotes	PS: 156.48.126.120/29	PS: 156.48.126.112/29

	SS: 156.48.125.32/29 G: 224.0.52.9 P: 21038	SS: 156.48.125.40/29 G: 224.0.53.9 P: 21039
'107' Indices – Composition and Values	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.10 P: 21040	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.10 P: 21041
'108' European Stocks – Referential Data (Off Exchange Trade Reporting)	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.11 P: 21042	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.11 P: 21043
'109' European Stocks – Trade Reporting (Off Exchange Trade Reporting)	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.12 P: 21044	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.12 P: 21045
'110' Luxembourg Stock Exchange	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.13 P: 21046	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.13 P: 21047
'111' SmartPool – Referential Data	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.14 P: 21048	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.14 P: 21049
'112' SmartPool – Public Data	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.15 P: 21050	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.15 P: 21051
'113' SmartPool – Members (only available via SFTI)	PS: 156.48.126.120/29 SS: 156.48.125.32/29 G: 224.0.52.16 P: 21052	PS: 156.48.126.112/29 SS: 156.48.125.40/29 G: 224.0.53.16 P: 21053

4.3.1.2. Refresh Configuration

Service ID	Line A	Line B
'201' Euronext Equities – Referential Data	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.4 P: 22028	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.4 P: 22029
'202' Euronext Equities – Trades / Instrument+Group State changes	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.5 P: 22030	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.5 P: 22031
'203' Euronext Equities – Quotes	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.6 P: 22032	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.6 P: 22033
'204' Euronext Equities – Orders	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.7 P: 22034	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.7 P: 22035
'208' European Stocks – Referential Data (Off Exchange Trade Reporting)	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.11 P: 22042	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.11 P: 22043
'210' Luxembourg Stock Exchange	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.13	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.13

	P: 22046	P: 22047
'211' SmartPool – Referential Data	PS: 156.48.126.176/29 SS: 156.48.125.48/29 G: 224.0.52.14 P: 22048	PS: 156.48.126.184/29 SS: 156.48.125.56/29 G: 224.0.53.14 P: 22049

4.4. *Packet Type, Bandwidth and Market Places per Service ID*

4.4.1. SFTI Lines

Table content described below is effective since 1st June 2009

Packet type	Enxt Equities Ref.	Enxt Equities Trades	Enxt Equities Quotes	Exnt Equities Orders	Enxt Warrants Trades	Enxt Warrants Quotes	Indices	European Stocks Ref.	European Stocks Trades	Luxembourg SE	SmartPool Ref.	SmartPool Public	SmartPool Members
	101	102	103	104	105	106	107	108	109	110	111	112	113
140 - Quotes			•			•				•			
141 - WAS			•										
220 - Trade - Creation		•			•					•			
221 - Trade - Cancel		•			•					•			
230 - Order Update				•						•			
231 - Order Retransmission				•						•			
240 - Trade - Full information		•			•					•			
241 - Trade - Price update		•			•					•			
242 - Trade - TCS trade		•			•					•			
243 - Trade - Trade publication									•				
244 - Trade - Settlement price		•											
245 - Trade - Auction summary		•			•					•			
246 - Trade - Notice of interest													•
505 - Stock state change		•			•					•		•	
513 - Euro & Interbank rates		•								•			
516 - Group State Change		•			•					•		•	
523 - Mail		•			•					•		•	
530 - TOP		•			•					•			•
531 - Market Imbalance													
534 - Authorized out of session limits		•			•					•			
535 - TCS State change		•								•			
537 - Threshold		•			•					•			
539 - Session timetable		•			•					•		•	
540 - Display bid & ask		•			•					•			
541 - Daily Summary										•			
542 - Real-time index							•			•			
543 - Index Summary							•			•			
544 - Index composition							•			•			
550 - Start referential	•				•		•	•		•	•		
551 - End Referential	•				•		•	•		•	•		
555 - Referential	•				•		•	•		•	•		
Bandwidth Size (Kb)	1000	1000	14000	8000	1000	14000	1000	256	256	1000	512	256	256
Population *	EE	EE	EE	EE	EW	EW	EI	ES	ES	LU	SP	SP	SP
									EW				
									EE				

* Population acronym:

- *EE: Euronext Equities*
- *EW: Euronext Warrants*
- *EI: Euronext Indices*

- *ES: European Stocks*
- *LU: Luxembourg instruments*
- *SP: Smartpool*

Bandwidth size (kb) explanation: each UTP-MD multicast channel capacity has been sized / optimized to cope with the expected amount of data to be disseminated on this channel and avoid any pick, overrun, bottleneck or packet loss.

4.4.2. MMBA Lines

Table content described below is effective since 1st June 2009

Packet type	Enxt Equities Ref.	Enxt Equities Trades	Enxt Equities Quotes	Exnt Equities Orders	Enxt Warrants Trades	Enxt Warrants Quotes	Indices	European Stocks Ref.	European Stocks Trades	Luxembourg SE	SmartPool Ref.	SmartPool Public
	101	102	103	104	105	106	107	108	109	110	111	112
140 - Quotes			•			•				•		
141 - WAS			•									
220 - Trade - Creation		•			•					•		
221 - Trade - Cancel		•			•					•		
230 - Order Update				•						•		
231 - Order Retransmission				•						•		
240 - Trade - Full information		•			•					•		
241 - Trade - Price update		•			•					•		
242 - Trade - TCS trade		•			•					•		
243 - Trade - Trade publication									•			
244 - Trade - Settlement price		•										
245 - Trade - Auction summary		•			•					•		
505 - Stock state change message		•			•					•		•
513 - Euro & Interbank rates		•								•		
516 - Group State Change		•			•					•		•
523 - Mail		•			•					•		•
530 - TOP		•			•					•		
531 - Market Imbalance												
534 - Authorized Out of Session Limits		•			•					•		
535 - TCS State Change		•								•		
537 - Threshold		•			•					•		
539 - Session timetable		•			•					•		•
540 - Display bid & ask		•			•					•		
541 - Daily Summary										•		
542 - Real-time index							•			•		
543 - Index Summary							•			•		
544 - Index Composition							•			•		
550 - Start Referential	•				•		•	•		•	•	
551 - End Referential	•				•		•	•		•	•	
555 - Referential	•				•		•	•		•	•	
Bandwidth Size (Kb)	1000	1000	14000	8000	1000	14000	1000	256	256	1000	512	256
Population	EE	EE	EE	EE	EW	EW	EI	ES	ES	LU	SP	SP
									EW			
									EE			

* Population acronym:

- *EE: Euronext Equities*
- *EW: Euronext Warrants*
- *EI: Euronext Indices*
- *ES: European Stocks*
- *LU: Luxembourg instruments*

- *SP: Smartpool*

Bandwidth size (kb) explanation: each UTP-MD multicast channel capacity has been sized / optimized to cope with the expected amount of data to be disseminated on this channel and avoid any pick, overrun, bottleneck or packet loss.

4.5. *Refresh Contents*

The table below describes the Service ID where the refresh functionality will be available in the production environment.

Production Service ID eligible for Refresh functionality	Feed Contents
101 (201 refresh)	Euronext Equities - Referential Data.
102 (202 refresh)	Euronext Equities – Trades The refresh server will provide a snapshot of the last trade (message 240) and the associated instrument state and group state at a given point in time for each instrument.
103 (203 refresh)	Euronext Equities – Quotes The refresh server will provide a snapshot of the best ten limits orderbook (message 140) as of a single point in time.
104 (204 refresh)	Euronext Equities – Orders. The refresh server will provide a snapshot of the order book (message 230) as of a single point time e.g. the book containing the whole active orders (remaining in the book) at this single point of time.
105 –No refresh provided	*Euronext Warrants – Trades*
106 –No refresh provided	*Euronext Warrants – Quotes*
107 –No refresh provided	*Euronext Indices – Composition and Values*
108 (208 refresh)	European Stocks (Off Exchange Trade Reporting) – Referential Data.
109 - No refresh provided	*European Stocks (Off Exchange Trade Reporting) – Trade Reporting*
110 (210 refresh)	Luxembourg Stock Exchange – Referencial Data, Instrument and Group State, Quotes, Trades and Orders
111 (211 refresh)	SmartPool data –Referential Data
112 –No refresh provided	*SmartPool data*
113 –No refresh provided	*Smartpool data*

4.6. *Production Timetable*

The following table is the overview of the main daily event generating activity and indicative time on the Universal Trading Platform Market Data Feed.

Event	Time (CET)	Comment
Application start-up	~ 06:10	
Referential data Sent	~ 06:11	
Orderbook Retransmission	~ 06:20	Retransmission of outstanding orders and associated messages from previous day
Open (all European markets)	~ 09:00	
Close (Paris, Amsterdam, Brussels equities/warrants)	17:40	To include closing auction and TAL phases.
Close (Lisbon Warrants)	18:30	
Close (all indices)	~ 20:00	Manual action
Referential data Sent	Between 20:00 and 21:00	Manual action
Application close down	23:00	

4.7. *Retransmission and Refresh Configuration*

4.7.1. Retransmissions TCP/IP Settings

The table below shows the primary and secondary TCP/IP retransmission address assignments for the UTP-MD Production platform.

- Primary site (Active)

Production Server	IP	Port
RTS Address on Primary Server	156.48.126.43	21150
RTS Address on Secondary Server	156.48.126.171	21150

- Secondary site (Disaster Recovery Site)

Production Server	IP	Port
RTS Address on Primary Server	156.48.125.20	21150
RTS Address on Secondary Server	156.48.125.29	21150

4.7.2. Refresh TCP/IP Settings

The table below shows the primary and secondary TCP/IP refresh address assignments for the UTP-MD Production platform.

- Primary site (Active)

Production Server	IP	Port
RFS Address on Primary Server	156.48.126.43	21600
RFS Address on Secondary Server	156.48.126.171	21600

- Secondary site (Disaster Recovery Site)

Production Platform	IP	Port
RFS Address on Primary Server	156.48.125.20	21600
RFS Address on Secondary Server	156.48.125.29	21600

4.7.3. High Availability Retransmission Behavior

As per the above tables, in production there are two redundant retransmission servers in the exchange. Clients should monitor the connection to the retransmission server to determine if there is an outage, and have the ability to switch to connect to the 'secondary' retransmission server in the event of a failover. The secondary retransmission server will maintain the same cache of packets as the primary therefore providing redundancy. Clients should monitor the availability of the primary / secondary retransmissions server by the following means:

- For clients remaining connected to the retransmission server throughout the day, a disconnection from the retransmission server should trigger a failover to the secondary retransmissions server.

- Clients may choose to only connect to the retransmissions server if the application requires packets to be serviced. In this instance, clients should fail over to the secondary retransmission server if they cannot establish a connection with the primary.

4.7.4. High Availability Refresh Behavior

As per the above tables, in production there are two redundant refresh servers in the exchange. Clients should monitor the TCP/IP connection to the refresh server to determine if there is an outage, and have the ability to switch to connect to the 'secondary' refresh server in the event of a failover. The secondary refresh server will maintain the same cache of packets as the primary therefore providing redundancy. Clients should monitor the availability of the primary / secondary refresh server by the following means:

- For clients remaining connected to the refresh server throughout the day, a disconnection from the refresh server should trigger a failover to the secondary server.

- Clients may choose to only connect to the refresh server if the application requires packets to be serviced. In this instance, clients should fail over to the secondary refresh server if they cannot establish a connection with the primary.

Clients should note that unlike the secondary retransmission server, the secondary refresh server will not service requests while it is in secondary mode. Clients sending a request to the secondary refresh server when there is no outage and the primary server is available, will receive a response with error code 7 - Refresh request rejected as sent to incorrect server (secondary instead of primary).

4.7.5. Source ID

The Source ID allows clients to perform retransmission and refresh requests. Please note that the Source IDs for retransmissions and refresh are identical. NYSE Euronext will provide a default of 4 Source IDs for Production. If more Source IDs are required:

Email: ccc@euronext.com

Tel: +33 (0) 1 4927 5080.

Clients should clearly state for which environment (EUA or Production) a Source ID is requested.

Each Source ID may only be logged on to a server once at a given time.

4.8. *Retransmission Request Limitations*

The below recommendations apply to production and EUA.

4.8.1. Heartbeat mechanism

The heartbeat messages frequency is set to 30 seconds.

A heartbeat message response has to be sent within 5 seconds to stay connected to the server.

4.8.2. Number of Source IDs

Clients are provided with 4 SourceIDs by default. In the event more Source IDs are required, please contact the Universal Trading Platform Support Desk.

4.8.3. Parallel Sessions

Clients may file several concurrent requests on the server at the same time with the same SourceID; there is no need to wait for the active retransmission to be closed to ask for another one.

Responses to these requests are sent in the same order as the initial requests.

4.8.4. Maximum Number of Requests

Maximum number of Retransmission Requests per Source ID per day has been set at 1,000.

4.8.5. Maximum Number of Packets per Request

Maximum number of packets that can be requested in one Retransmission Request has been set at 1,000.

4.8.6. Maximum Number of Packets Stored in the Retransmission Cache

The maximum number of packets that are cached for retransmission request has been set at 500,000 per Service ID. A packet out of cache cannot be retransmitted (A dedicated response “Rejected – requested packets are not available” will be sent in this case).

4.9. *Refresh Request Limitations*

The below recommendations apply to production and EUA.

4.9.1. Heartbeat mechanism

The heartbeat messages frequency is set to 30 seconds.

A heartbeat message response has to be sent within 5 seconds to stay connected to the server.

4.9.2. Number of Source IDs

The Source IDs for the refresh are identical to the retransmissions server. They should be reused with the refresh server.

Clients are provided with 4 Source IDs by default. In the event more Source IDs are required, please contact the Universal Trading Platform Support Desk.

4.9.3. Maximum Number of Requests

Maximum number of Refresh Requests per Source ID per day has been set at 100.

5. External User Acceptance Feed Configuration

5.1. *Introduction*

5.1.1. Data Content

The information supplied in this chapter applies to the UTP-MD Euronext Cash markets EUA service only.

A Similar section will be written for Qatar Exchange.

5.1.2. Data Delivery

UTP-MD Cash EUA is available via MMBA and SFTI. Configuration settings for MMBA and SFTI are identical.

5.2. *Rendez-Vous Point (all connectivity)*

External User Acceptance Platform	
RP1 platform Line A	RP2 Line B
S: 156.48.121.2 Associated to: G: 224.0.52.x	S: 156.48.121.55 Associated to: G: 224.0.53.x

5.3. *SFTI / MMBA Feed Configuration*

5.3.1. EUA Feed Configuration

The table below shows the UDP/IP configuration for both line A and line B, for the UTP-MD External User Acceptance platform.
In this table the following indicators are used:
PS: Primary Data Centre Source IP
G = Group
P = UDP Port

5.3.1.1. Real Time Configuration

Service ID	Line A	Line B
'1' – Euronext Equities / Luxembourg - Orders	PS: 156.48.126.104/29 G: 224.0.52.0 P: 21020	PS: 156.48.126.96/29 G: 224.0.53.0 P: 21021
'2'– Euronext Equities / Luxembourg – Others	PS: 156.48.126.104/29 G: 224.0.52.2 P : 21024	PS: 156.48.126.96/29 G: 224.0.53.2 P: 21025
'3' – Euronext – Warrants	PS: 156.48.126.104/29 G: 224.0.52.17 P : 21054	PS: 156.48.126.96/29 G: 224.0.53.17 P: 21055
'4' – SmartPool	PS: 156.48.126.104/29 G: 224.0.52.19 P : 21058	PS: 156.48.126.96/29 G: 224.0.53.19 P: 21059
'5' - NYSE Arca Europe	PS: 156.48.126.104/29 G: 224.0.52.28 P: 21076	PS: 156.48.126.96/29 G: 224.0.53.28 P: 21077

5.3.1.2. Refresh Configuration

Service ID	Line A	Line B
'11' – Euronext Equities - Orders	PS: 156.48.126.192/29 G: 224.0.52.0	PS: 156.48.126.200/29 G: 224.0.53.0

	P: 22020	P: 22021
'12' – Euronext Equities / Luxembourg - Others	PS: 156.48.126.192/29 G: 224.0.52.2 P: 22024	PS: 156.48.126.200/29 G: 224.0.53.2 P: 22025
'14' – Smartpool	PS: 156.48.126.192/29 G: 224.0.52.19 P: 22058	PS: 156.48.126.200/29 G: 224.0.53.19 P: 22059
'15' - NYSE Arca Europe	PS: 156.48.126.192/29 G: 224.0.52.28 P: 22076	PS: 156.48.126.200/29 G: 224.0.53.28 P: 22077

5.4. *Packet Type, Bandwidth and Market Places per Service ID*

5.4.1. SFTI - EUA

Table content described below is effective since 25th May 2009

Packet type	Exnt Equities Orders	Enxt Equities	Enxt Warrants	SmartPool	NYSE Arca Europe
	1	2	3	4	5
140 - Quotes message		•	•		•
141 - WAS message		•			
220 - Trade message - Creation		•	•		•
221 - Trade message - Cancel		•	•		•
230 - Order Update	•				•
231 - Order Retransmission	•				•
240 - Trade Message - Full information		•	•		•
241 - Trade Message - Price update		•	•		•
242 - Trade Message - TCS trade		•	•		
243 - Trade Message - Trade publication		•			
244 - Trade Message - Settlement price		•			
245 - Trade Message - Auction summary		•	•		
246 - Trade message - Notice of interest				•	
505 - Stock state change message		•	•	•	•
513 - Euro & Interbank rates		•			
516 - Group State Change		•	•	•	•
523 - Mail		•	•	•	
530 - TOP		•	•	•	•
531 - Market Imbalance					
534 - Authorized Out of Session Limits		•	•		
535 - TCS State Change		•			
537 - Threshold		•	•		•
539 - Session timetable		•	•	•	•
540 - Display bid & ask		•	•		
541 - Daily Summary		•			
542 - Real-time index		•			
543 - Index Summary		•			
544 - Index Composition		•			
550 - Start Referential		•	•	•	•
551 - End Referential		•	•	•	•
555 - Referential		•	•	•	•
Bandwidth Size (Kb)	256	784	128	64	256
Population	EE	EE	EW	SP	AE
		EI			
		ES			
		LU			

* Population acronym:

- *EE: Euronext Equities*
- *EW: Euronext Warrants*
- *EI: Euronext Indices*

- *ES: European Stocks*
- *LU: Luxembourg Instruments*
- *AE: NYSE Arca Europe*
- *SP: Smartpool*

Bandwidth size (kb) explanation: each UTP-MD multicast channel capacity has been sized / optimized to cope with the expected amount of data to be disseminated on this channel and avoid any pick, overrun, bottleneck or packet loss.

5.4.2. MMBA - EUA

The table below describes the content of each service in EUA over MMBA lines:

Packet type	Exnt Equities Orders	Enxt Equities	Enxt Warrants	SmartPool	NYSE Arca Europe
	1	2	3	4	5
140 - Quotes message		•	•		•
141 - WAS message		•			
220 - Trade message - Creation		•	•		•
221 - Trade message - Cancel		•	•		•
230 - Order Update	•				•
231 - Order Retransmission	•				•
240 - Trade Message - Full information		•	•		•
241 - Trade Message - Price update		•	•		•
242 - Trade Message - TCS trade		•	•		
243 - Trade Message - Trade publication		•			
244 - Trade Message - Settlement price		•			
245 - Trade Message - Auction summary		•	•		
246 - Trade message - Notice of interest				•	
505 - Stock state change message		•	•	•	•
513 - Euro & Interbank rates		•			
516 - Group State Change		•	•	•	•
523 - Mail		•	•	•	
530 - TOP		•	•	•	•
531 - Market Imbalance					
534 - Authorized Out of Session Limits		•	•		
535 - TCS State Change		•			
537 - Threshold		•	•		•
539 - Session timetable		•	•	•	•
540 - Display bid & ask		•	•		
541 - Daily Summary		•			
542 - Real-time index		•			
543 - Index Summary		•			
544 - Index Composition		•			
550 - Start Referential		•	•	•	•
551 - End Referential		•	•	•	•
555 - Referential		•	•	•	•
Bandwidth Size (Kb)	512	512	512	256	256
Population	EE	EE	EW	SP	AE
		EI			
		ES			
		LU			

* Population acronym:

- *EE: Euronext Equities*
- *EW: Euronext Warrants*

- *EI: Euronext Indices*
- *ES: European Stocks*
- *LU: Luxembourg instruments*
- *AE: NYSE Arca Europe*
- *SP: Smartpool*

Bandwidth size (kb) explanation: each UTP-MD multicast channel capacity has been sized / optimized to cope with the expected amount of data to be disseminated on this channel and avoid any pick, overrun, bottleneck or packet loss.

5.5. *Refresh Contents*

The table below describes the Service ID where the refresh functionality will be available in the EUA environment.

EUA Service ID eligible for Refresh functionality	**Feed Contents.**
1 (Refresh 11)	European Cash Markets – Equities Orders (Orderbook refresh)
2 (Refresh 12)	European Cash Markets – Reference Data and Equities Others (Trade refresh, quote refresh, reference data refresh, stock and instrument state change) Luxembourg – Reference Data, Trade refresh, quote refresh, orders refresh, stock and instrument state change.
4 (Refresh 14)	Smartpool – Reference Data
5 (Refresh 15)	NYSE Arca Europe - Reference Data, Trade refresh, quote refresh, orders refresh, stock and instrument state change.

5.6. *EUA Timetable*

The starting time for the UTP-MD EUA feed is ~ 10 minutes delayed in comparison to the production feed.

The closing time for the UTP-MD EUA feed varies daily depending on which testing is scheduled (afternoon upgrades, after hours testing etc).

5.7. *Retransmission and Refresh Configuration*

5.7.1. Retransmission TCP/IP Settings

The table below shows the primary and secondary TCP/IP retransmission address assignments for the UTP-MD External User Acceptance platform.

External User Acceptance Server	IP	Port
RTS Address on Primary Server	156.48.126.92	21150
RTS Address on Secondary Server	156.48.126.93	21150

5.7.2. Refresh TCP/IP Settings

The table below shows the primary and secondary TCP/IP refresh address assignments for the UTP-MD External User Acceptance platform.

External User Acceptance Server	IP	Port
RFS Address on Primary Server	156.48.126.92	21600
RFS Address on Secondary Server	156.48.126.93	21600

5.7.3. High Availability Retransmission Behavior

As per the above table, in EUA there are two redundant retransmission servers in the exchange. Clients should monitor the connection to the retransmission server to determine if there is an outage, and have the ability to switch to connect to the 'secondary' retransmission server in the event of a failover. The secondary retransmission server will maintain the same cache of packets as the primary therefore providing redundancy. Clients should monitor the availability of the primary / secondary retransmissions server by the following means:

- For clients remaining connected to the retransmission server throughout the day, a disconnection from the retransmission server should trigger a failover to the secondary retransmissions server.

- Clients may choose to only connect to the retransmissions server if the application requires packets to be serviced. In this instance, clients should fail over to the secondary retransmission server if they cannot establish a connection with the primary.

5.7.4. High Availability Refresh Behavior

As per the above tables, in EUA there are two redundant refresh servers in the exchange. Clients should monitor the TCP/IP connection to the refresh server to determine if there is an outage, and have the ability to switch to connect to the 'secondary' refresh server in the event of a failover. The secondary refresh server will maintain the same cache of packets as the primary therefore providing redundancy. Clients should monitor the availability of the primary / secondary refresh server by the following means:

- For clients remaining connected to the refresh server throughout the day, a disconnection from the refresh server should trigger a failover to the secondary server.

- Clients may choose to only connect to the refresh server if the application requires packets to be serviced. In this instance, clients should fail over to the secondary refresh server if they cannot establish a connection with the primary.

Clients should note that unlike the secondary retransmission server, the secondary refresh server will not service requests while it is in secondary mode. Clients sending a request to the secondary refresh server when there is no outage and the primary server is available, will receive a response with error code 7 - Refresh request rejected as sent to incorrect server (secondary instead of primary).

5.7.5. Source ID

The Source ID allows clients to perform retransmission and refresh requests. Please note that the Source IDs for retransmissions and refresh are identical. NYSE Euronext will provide a default of 4 Source IDs for EUA. If more Source IDs are required:

Email: ccc@euronext.com

Tel: +33 (0) 1 4927 5080

Clients should clearly state for which environment (EUA or Production) a Source ID is requested.

Each Source ID may only be logged on to a server once at a given time.

6. Appendices

A – Help Desks

Euronext Cash Markets Help Desk:

Telephone: +33 1 49 27 50 90

Email: utp@nyx.com

Qatar Exchange Help Desk:

Telephone: <TBD>

Email: <TBD>

A - Market Feed Code

Market	Code	Content
	00	None Applicable
	CC	Technical message
	62	Strategic indices
Euronext Paris	01	CAC40 stocks (equities) that support traded options + Euronext 100 index and Euronext 150 index
	02	CAC40 stocks that do not support traded options.
	03	Non-CAC40 stocks that support traded options.
	04	Non-CAC40 stocks that do not support traded options.
	05	Non-CAC40 stocks that support traded options (same as 03)
	06	Non-CAC40 stocks that do not support traded options.
	07	Bonds.
	10	Free market (“Marché libre”)
	11	Securities in process of being introduced (SHIVA).
	12	Gold.
	13	Foreign exchange.
	15	Centralised lending market
	16	Commercial paper
	60	ETF Securities (Exchange traded fund)
Euronext Brussels	20	Stock listed in Brussels
	21	Bonds listed in Brussels
	22	Brussels Index constituents
	23	Warrants listed in Brussels
Euronext Amsterdam	30	Stocks listed in Amsterdam
	31	Bonds listed in Amsterdam
	32	Amsterdam Index constituents
	33	Warrants listed in Amsterdam
Euronext Lisbon	40	Equities, Rights and Investment Funds listed in Lisbon
	41	Bonds and Certificates listed in Lisbon
	42	PSI-20 equities listed in Lisbon
	43	Warrants listed in Lisbon
European instruments	65	European stocks (Trade publication purpose)
	79	SmartPool
Qatar Exchange	XX	TO BE DEFINED

B - Financial Market Code

Code	Content
000	None applicable
025	Paris
277	Centralised lending market
278	Brussels
279	Amsterdam
290	Lisbon
295	Luxembourg
299	Europe
300	Commercial paper
301	SmartPool
NNN	Qatar

C - Stock Exchange Code

Code	Content
000	Not applicable
006	BRUXELLES
025	PARIS
027	LYON
028	MARSEILLE
029	NANCY
030	BORDEAUX
031	NANTES
032	LILLE
038	AMSTERDAM
047	LUXEMBOURG
051	LISBON
675	SMARTPOOL
991	EUROPEAN STOCK
NNN	QATAR

D - Stock Types

Code	Stock type
008	Participating bond-Cum-warrant
009	Participating bond-Ex-warrant
010	Participating share
011	Subordinated bond
013	Interest strip
014	Principal strip
015	Perpetual
017	Bunny bond
018	ORT (France)
019	OAT Fungible government bond
021	Convertible bond
022	Exchangeable bond
023	Participating bond
024	Indexed bond
025	Ordinary bond or note
026	Lottery bond
027	Savings bond

Code	Stock type
028	Indemnity bond
030	Bond warrant
032	Bond Cum-warrant
033	Bond Ex-warrant
035	Right to indemnity security
038	Bond option
039	Emprunt notionnel (France)
040	Founder's share
041	Ordinary share
042	Bonus share
043	Preferred share
044	Saving share
045	Certificate
046	AFV share (Belgium)
047	Accumulating right
048	Allotment right
049	Subscription right
050	Preferred stock
051	VVPR share (Belgium)
052	Certificate of deposit
053	Cash note
054	Allocation right
055	Option right
056	Share-Cum warrant
057	Share-Ex warrant
058	Preference share
059	Preference
060	Gold
061	Unit of international investment trust
062	Unit of unit trust
063	Mortgage warrant
064	Bank note
065	Devise commerciale (France)
066	Devise titre (France)
067	Commodities
068	Index
069	Unit
070	Investment fund share
071	Miscellaneous products-Warrant
072	Share warrant
075	Miscellaneous
076	Listed call option
077	Listed put option
078	Call money, average BD rate
080	Founder's stock
081	Partnership interest
082	Part de réserve (Belgium)
084	Deferred share
085	Regional development company share
086	Venture capital company
087	Real estate company share
090	Mortgage bond
100	Participation certificate
105	Index warrant
106	Currency warrant

Code	Stock type
108	Warrant of a warrant
110	Participating share-Warrant
111	Subordinated convertible bond
121	Convertible bond-Cum warrant
123	Convertible participating bond
139	Matif 90 days treasury bond
140	Part de réserve AFV (Belgium)
141	Convertible ordinary share
142	Dividend right certificate
144	Convertible saving share
145	Investment certificate-Warrant
150	Convertible preference share
162	MBO share
200	Participative certificate-Warrant
221	Convertible bond-Ex warrant
240	AFV company's share (Belgium)
241	VVPR's company's share (Belgium)
242	Accumulation fund share
243	Distribution fund share
244	Unit-Futures and Options market investment fund
245	Accumulation fund share
246	Distribution fund share
247	Certificate of guaranteed value
248	Share warrant
249	BTAN (France)
250	OAT (France)-Interest certificate
251	OAT (France)-Principal certificate
252	Indexed OAT (France)
253	Indexed OAT (France)-Principal certificate
254	Indexed OAT (France)-Interest certificate
255	Euro medium term note-EMTN
256	BTF (France)
257	Indexed certificate
258	BMTN (France)
259	Real estate bond
260	Convertible EMTN
261	Indexed EMTN
262	Indexed certificate-EMTN
263	Exchange Traded fund-ETF
264	Venture Cap. mutual fund share
265	Mutual fund for innovation share
266	Medium Term Note
267	Medium Term Note-Floating rate
268	Accumulating share
269	Distribution share
270	Ordinary bond-Interest certificate
271	Ordinary bond-Principal certificate
272	'Beneficial interest' share
273	Redeemable cumulative preferred share
274	Convertible redeemable cumulative preferred share
275	Convertible cumulative preferred share
276	Cumulative preferred share
277	Redeemable preferred share
278	Foreign treasury note
280	Strip VVPR

Code	Stock type
281	Mortgage Bonds
303	Exotic warrant
304	Certificate of deposit
305	Commercial paper
306	ETF – Closed ended Fund
307	ETF – Open ended Fund

E - System ID

ID	Application
1	Not provided
11 to 19	Euronext - UTP for Equities - Regulated markets
31 to 39	Euronext - UTP for Warrants
41 to 49	Euronext - UTP for Smartpool
101 to 109	Qatar Exchange – UTP for Cash markets

F – Services & Multicast Groups

- **Euronext Cash Markets**
  - Production

Service ID	Multicast content
'101'	Euronext Equities – Referential Data
'102'	Euronext Equities – Trades
'103'	Euronext Equities – Quotes
'104'	Euronext Equities – Orders
'105'	Euronext Warrants – Trades
'106'	Euronext Warrants – Quotes
'107'	Euronext Indices – Composition and Values
'108'	European Stocks (Off Exchange Trade Reporting) – Referential Data
'109'	European Stocks (Off Exchange Trade Reporting) – Trade Reporting
'110'	Luxembourg Stock Exchange
'111'	SmartPool – Referential Data
'112'	SmartPool – Public Data
'113'	SmartPool – Members

  - External User Acceptance

Service ID	Multicast Content
'1'	Euronext Cash Markets – Equities Orders (Production 104)
'2'	Euronext Cash Markets – Referential Data + Equities / Indices / Luxembourg (Production 101, 102, 103, 107, 108, 109, 110)
'3'	Euronext Cash Markets – Warrants (Production 105, 106)
'4'	SmartPool (Production 111, 112, 113)
'5'	NYSE Arca Europe (Production 114 in London)

- **Qatar Exchange Cash Markets**

These settings should be confirmed or modified by Qatar Exchange

  - Production

Service ID	Multicast content
'201'	Qatar Exchange Equities – All Data
~~'202'~~	~~Qatar Exchange Equities – Trades & Quotes~~
~~'204'~~	~~Qatar Exchange Equities – Orders~~
~~'207'~~	~~Qatar Exchange Indices – Composition and Values~~
~~'208'~~	~~Qatar Exchange (Off Exchange Trade Reporting) – Referential Data~~
~~'209'~~	~~Qatar Exchange (Off Exchange Trade Reporting) – Trade Reporting~~

- External User Acceptance

Service ID	Multicast Content
'1'	Qatar Exchange Cash Markets – Equities All Data (Production 201)
~~'2'~~	~~Qatar Exchange Cash Markets – Referential Data + Equities / Indices (Production 201, 202, 207~~

7. Document History

7.1. *Version 2.0*

7.1.1. Document Style and Structure

The most significant change of the version 2.0 specifications was the act of merging all prior 'appendices' and configuration documents into one specification document. The table below shows the 1 to 1 mapping between the old documents and the chapters within version 2.0.

Please note that chapter 2 in the version 2.0 specifications applies to all sections of chapter 3. This was included in all separate appendices before, but is now only documented once.

Old Document Name	Chapter in New Specification
Generic Customer Specifications 1.3	1 Cash Markets Processing Information
Trade Appendix 1.4	3.2 Trades
Quotes & BBO10 Appendix 1.4	3.3 Quotes and BBO10
OrderBook Appendix 1.4	3.4 OrderBook
Indices Appendix 1.4	3.5 Indicies
Market Information Appendix 1.4	3.1 Market Information
Production Feed Configuration 1.6	4 Production Feed Configuration
External User Acceptance Feed Configuration 1.6	5 External User Acceptance Feed Configuration
UTP-MD Production Services Content	4.4.1 – SFTI / 4.4.2 - MMBA
UTP-MD EUA Services Content	5.3.2.1 – SFTI / 5.3.2.2 – MMBA
UTP-MD Notice Of Interest (SmartPool Members)	3.1.11 Notice Of Interest – 246 Message

The change history from the previous 1.x documentation has been removed. If this is required please contact the UTP Helpdesk. Certain textual changes have been made in order to improve the descriptions in sections of the document. Where this did not impact anything other than the cosmetics of the specifications it has not been highlighted.

7.1.2. Refresh Functionality

The following chapters have been added / edited to define the UTP-MD refresh functionality:

Chapter Added / Edited	Description	Impact / Live Date Details
1.2.3	Refresh Functionality (Key Principles)	Refresh functionality live date to be confirmed
1.3.2, 2.2, 2.10.1, 3.1.2, 3.2.2, 3.3.2, 3.4.2, 3.5.2	Inclusion of DeliveryFlag 16 for Real Time ZliB Compressed packets	Refresh functionality live date to be confirmed
2.4.1 / 2.4.2	Refresh Heartbeat Processing Notes	Refresh functionality live date to be confirmed

2.8	Refresh Request	Refresh functionality live date to be confirmed
2.9	Refresh Response	Refresh functionality live date to be confirmed
2.10.1	Refresh Compression	Refresh functionality live date to be confirmed
2.10.2	Refresh Packet Type	Refresh functionality live date to be confirmed
2.10.3	Start Refresh Message	Refresh functionality live date to be confirmed
2.10.4	End Refresh Message	Refresh functionality live date to be confirmed
4.3.1.2	Production Refresh Feed Channel Definitions	Refresh functionality live date to be confirmed
4.5	Refresh Contents (production)	Refresh functionality live date to be confirmed
4.7.2	Production Refresh TCP/IP Settings	Refresh functionality live date to be confirmed
4.7.4	Production High Availability Refresh Behavior	Refresh functionality live date to be confirmed
4.7.5 / 5.7.5	Refresh Source ID (Production and EUA) – Same as retransmissions Source ID.	Refresh functionality live date to be confirmed
4.9	Refresh Request Limitations	Refresh functionality live date to be confirmed
5.3.1.2	EUA Refresh Feed Channel Definitions	Refresh functionality live date to be confirmed
5.7.2	EUA Refresh TCP/IP Settings	Refresh functionality live date to be confirmed
5.7.4	EUA High Availability Refresh Behavior	Refresh functionality live date to be confirmed
5.5	Refresh Contents (EUA)	Refresh functionality live date to be confirmed

A release schedule for refresh will be defined imminently.

7.1.3. Channel Definitions

Channel definitions have been added to the specification for the following:

Chapter Added / Edited	Description	Impact / Live Date Details
4.3.1	Production Disaster Recovery Source IP's (note 'SS' in the definition table)	Already live in production. New information provided.

4.3.1	Production Refresh Feed Channel Definitions (as per above note)	Refresh functionality live date to be confirmed
5.3.1	EUA Refresh Feed Channel Definitions (as per above note)	Refresh functionality live date to be confirmed.
4.7.1	Disaster Recovery Retransmissions IP details	Already live in production. New information provided.

7.1.4. Message Changes

The following changes have been made to specific messages throughout the specification. All of these changes are already live on the data feed.

Document Chapter	Message Number	Details of Change	Impact / Live Date Details
1.5.2, 2.2, 3.1.2, 3.2.2, 3.3.2, 3.4.2, 3.5.2	Packet Header	The FAST compression delivery flags 8 and 10 have been removed from these sections	Removal of Information (N / A for live date)
3.1.3.3	505	Additional values 0 and null added to the InstrumentState field of the 505 message	Already live in production
3.1.3.3	505	Additional value null added to the InstrumentTradingStatus field of the 505 message	Already live in production
3.1.3.3	505	In the InstrumentTradingStatus field, values S and R added for Indicies	Already live in production
3.1.5.3	516	Changed the name of the field Phase to SessionType	Cosmetic Change
3.1.12.3	539	Changed the name of the field Phase to SessionType	Cosmetic Change
3.1.17.3	553	Additional value of 9 added to the TaxCode field. Values 1, 2 and 4 removed.	Already live in production
3.1.17.3	553	The StrikeScaleCode field has been defined in position 455 of the 553 message. This was previously defined as a filler which was an error in the previous specifications.	Already live in production
3.2.3.3	220	Changed the name of the field Phase to OpeningTradeIndicator	Cosmetic Change
3.2.5.3	240	Changed the name of the field Phase to OpeningTradeIndicator	Cosmetic Change
3.4.4.3	231	Removal of the value VE from the TradingEngineID field	Already live in production
3.4.4.3	231	Addition of the value CO to the TradingEngineID field	Already live in production
3.5.3.3	542	The values in the TypeOfLevel field for levels 2 and 3 were inverted in the previous	Already live in

		specifications. This has now been corrected	production
3.5.4.3	543	The values in the TypeOfLevel field for levels 2 and 3 were inverted in the previous specifications. This has now been corrected	Already live in production

7.1.5. Miscellaneous Changes

The following miscellaneous changes have been made throughout the specification:

Chapter Added / Edited	Description	Impact / Live Date Details
1.3.4.3	Improved diagram to describe retransmission behavior	For information purposes.
1.4.1.2	New section added for the High Availability System Failure	Already live in production. New information provided.
1.4.1.3	New section added for Disaster Recovery System Failure	Already live in production. New information provided.
4.6	New section added to describe the production feed timetable	For information purposes.
5.6	New section added to describe the EUA feed timetable	For information purposes.
Appendix D	Addition of two new ETF Stock Types	Already live in production. New information provided.
Appendix E	New appendix added to describe the SystemID. The SystemID field in every message now refers to this appendix	For information purposes.

7.1.6. Message Overview and Message Sending Rules

To improve our documentation the ‘Message Overview’ and ‘Message Sending Rules’ sections have been added to each of the defined message types. This should be reviewed for added value and information.

7.1.7. Messages Not Live – For Information

Two new messages have been included in the specifications that are not yet live:

Chapter Added / Edited	Description	Impact / Live Date Details
3.1.8	Message 531 ‘Market Imbalance’ added	Live date to be communicated, estimated late 2009
3.1.14	Message 541 ‘Daily Summary ’ added	Live date to be communicated, estimated late 2009

A release schedule for these messages will be communicated towards the end of 2009.

7.2. *Version 2.1*

7.2.1. Refresh Server Updates

Chapter Added / Edited	Description	Impact / Live Date Details
1.3.2, 2.2, 2.10.1, 3.1.2, 3.2.2, 3.3.2, 3.4.2, 3.5.2	**Change to DeliveryFlag packet header values of refresh based on client feedback.**	EUA: 28th September 2009 PRD: 26th October 2009
2.9	New refresh response added (7) to cater for clients sending refresh requests to the secondary server.	EUA: 28th September 2009 PRD: 26th October 2009
4.3.1.2 / 5.3.1.2	Added production and EUA multicast channel definitions for the refresh server	EUA: 28th September 2009 PRD: 26th October 2009
4.7.2 / 5.7.2	Added production and EUA TCP/IP channel definitions for refresh server	EUA: 28th September 2009 PRD: 26th October 2009
4.3.1.2 / 5.3.1.2	Added Luxembourg to the production and EUA multicast channel definitions for the refresh server	EUA: 28th September 2009 PRD: 26th October 2009
4.5 / 5.5	Added Luxembourg to production and EUA refresh contents table Addition of 516 messages in channel 102 refresh	EUA: 28th September 2009 PRD: 26th October 2009
4.7.4 / 5.7.4	Added refresh server High Availability behavior	EUA: 28th September 2009 PRD: 26th October 2009
5.3.1.2	Added NAE to EUA refresh configuration, **and updated EUA refresh service IDs.**	EUA: 28th September 2009 PRD: 26th October 2009

7.2.2. Miscellaneous Changes

Chapter Added / Edited	Description	Impact / Live Date Details
3.5.3.3	Addition of a new field in the 542 message 'RebroadcastIndicator'. This field DOES NOT affect the previous message structure and uses one of three characters previously occupied by a filler. The default value of this field is currently 0. The date this field will be activated will be provided via the UTP-MD Issue List.	The date this field will be activated will be provided via the UTP-MD Issue List.
5.3.1.1	IP range details provided for EUA instead of dedicated IP addresses. This is consistent with	Already live. New information provided.

	production.	
5.8	Removed the EUA Internet VPN details as this service has been discontinued.	EUA VPN service has already been terminated.

7.2.3. Warrants and Smartpool UTP Migrations

The following changes have been made to the specification in anticipation of both Warrants and Smartpool being migrated to the Universal Trading Platform trading engines.

Chapter Added / Edited	Description	Impact / Live Date Details
3.1.3.3	Addition of new value 'R' to the HaltReason field in the 505 message Stock State Change.	UTP Warrants migration live date to be confirmed.
Appendix E	Addition of UTP for Smartpool system IDs.	UTP Smartpool migration live date to be confirmed.

7.2.4. Removal of NYSE Arca Europe

In light of NYSE Arca Europe migrating to the London data centre, and an independent market data publisher, a separate specification has been created for this now fully separate market. It should be noted that NYSE Arca Europe EUA is still accessed via the Paris Euronext Cash EUA environment. For that reason NYSE Arca Europe configuration details for EUA have remained in this specification.

The following has been removed from this document:

Chapter Added / Edited	Description	Impact / Live Date Details
1.5.1	Removed NAE channel 114 from the environments section	Cosmetic documentation update
3.1.17.3	Removed reference to NAE in the following fields in the reference data where this information was noted as not being relevant to NAE: 'LastAdjPrice' / 'PrevVolumeTraded' / 'DateOfLastTrade'	Cosmetic documentation update
3.1.17.3	Removal of 'XHFT NYSE ARCA EUROPE' from the 'MIC' field in the reference data	Field value no longer provided
3.4.4.3	Removal of value C0 for UTP NAE in the 'TradingEngineID' field	Field value no longer provided
4.3.1.1	Removal of channel 114 for NAE from production channel definitions	This channel is now defined in the new NAE specification
4.4.1 / 4.4.2	Removal of NAE channel 114 for SFTI and MMBA production	This channel is now defined in the new NAE specification
4.5	Removed reference to NAE in the production refresh contents	This channel is now defined in the new NAE specification

7.3. *Version 2.2*

Miscellaneous corrections and adjustments.

Separation of Euronext specificities and Qatar Exchange specificities.

Addition of Qatar Exchange specificities, that is:

- Addition of message 545 "Global Market Activity Summary"
- Addition of message 247 "VWAP – Closing Price"
- Addition of the OrderOrigin field in the message 230 "Market Sheet" (due to the fact that Qatar Exchange uses a FIFO-Origin priority rule)
- Move of the "List of Services ID and Multicast Groups" in an appendix F

7.4. *Version 2.3*

Miscellaneous corrections and adjustments.

Qatar Exchange specificities:

- Message Global market Activity Summary is message 545 (was 542 in Version 2.2)
- Message Order Book Update / Market Sheet – 230: field OrderOrigin is in position 78
- Messages Trade Creation – 220 and Trade Full Information – 240: Field TradeCond1 is renamed SmallTradeIndicator and broadcast the small trade indicator (small trade, not a small trade) and field TradeCond4 is renamed TradeOrigin and broadcast the origin of the trade (orders from the book, Block trade)

7.5. *Version 2.4*

Qatar Exchange specificities for message 553:

- Addition of the MarketSegment field
- Addition of the Algo field
- Addition of LocalName field
- Addition of the SmallTrade field

7.6. *Version 2.5*

Qatar Exchange specificities for message 553:

- Message 553 is renamed into message 555
- Filed LocalName length is extended to 30 bytes

7.7. *Version 2.6*

Correction of 2 errors in message 545:

- SourceSeqNum and SourceTime have been permuted;
- SystemID and SourceTimeMicroSecs were missing and have been added.

Appendix 6.F: only one channel will be used at Qatar Exchange.

Addition of field CollarType in message 537.

Addition of field SystemID in message 247.

Message 555: last filler size is changed to 12