

TQ203 · TECHNICAL SPECIFICATION

Turquoise Equities Drop Copy Gateway (FIX 5.0)

ISSUE 2.0 · 20 February 2013

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Readership	4
1.3	Document Series	5
1.4	Document History	5
1.5	Enquiries	6
2	Service Description	7
2.1	Services Supported by Trading Gateway.....	7
2.2	Connection Configuration.....	7
2.2.1	Real-Time Connections	7
2.2.2	Non-Real Time Connections	7
2.3	Supported Events.....	8
2.4	Open Order Download	8
2.5	Execution Reports.....	9
2.5.1	Order Status	10
2.5.2	Order and Execution Identifiers.....	10
2.5.2.1	Client Order IDs	10
2.5.2.2	Order IDs	10
2.5.2.3	Execution IDs	10
2.5.2.4	Trade Match ID	11
2.5.3	Instrument Identification	11
2.5.4	Party Identification	12
2.6	Timestamps and Dates	12
3	Connectivity	13
3.1	CompIDs	13
3.1.1	Passwords	13
3.2	Sponsored Access - Monitoring Users.....	13
3.3	Production IP Address and Ports	14
3.4	Failover and Recovery.....	14
4	FIX Connections and Sessions	15
4.1	Establishing a FIX Connection	15
4.2	Maintaining a FIX Session	17
4.2.1	Message Sequence Numbers	17
4.2.2	Heartbeats	17
4.2.3	Increasing Expected Sequence Number	17
4.3	Terminating a FIX Session.....	18
4.4	Re-Establishing a FIX Session	18
4.4.1	Resetting Sequence Numbers: Starting a New FIX Session	19
4.4.1.1	Reset Initiated by the Client	19

4.4.1.2	Reset Initiated by the Server	19
5	Recovery	20
5.1	Resend Requests	20
5.2	Possible Duplicates	20
5.3	Possible Resends	20
5.4	Transmission of Missed Messages	21
5.5	Resending Previous Execution Reports.....	21
6	Message Formats	22
6.1	Supported Message Types.....	22
6.1.1	Administrative Messages	22
6.1.2	Application Messages	22
6.1.2.1	Client-Initiated	22
6.1.2.2	Server-Initiated.....	23
6.2	Message Header and Trailer	23
6.2.1	Message Header	23
6.2.2	Message Trailer.....	24
6.3	Administrative Messages	25
6.3.1	Logon.....	25
6.3.2	Logout	26
6.3.3	Heartbeat	26
6.3.4	Test Request	27
6.3.5	Resend Request.....	27
6.3.6	Reject.....	27
6.3.7	Sequence Reset.....	28
6.4	Application Messages: (Client-Initiated).....	28
6.4.1	Order Mass Status Request.....	28
6.5	Application Messages: (Server-Initiated)	29
6.4.5	Execution Report.....	29
6.6	Components of Application Messages	35
6.6.1	Trading Party	35
7	Reject Codes	37
7.1	Reject (35=3)	37
7.2	Execution Report (35=8)	37
7.3	Business Message Reject (35=j)	38
8	APPENDIX A.....	39
8.1	Error & Reject Text Strings.....	39

1 Introduction

The Financial Information Exchange (FIX) protocol enables access to Turquoise using a messaging standard developed for real-time electronic exchange of security transactions.

FIX enables access to the trading services and security information within Turquoise. This specification describes a conceptual overview of the FIX 5.0 SP2 protocol as well as providing technical guidance on adopting FIX 5.0 SP2 to connect to Turquoise.

Turquoise offers a drop copy gateway that will enable member firms to receive additional copies of the [Execution Report](#) messages generated by the matching system. This interface may also be used by clients to download the current status of all their active orders in the event of a failure. The drop copy service cannot be used to submit orders or receive market data.

The interface is a point-to-point service based on the technology and industry standards TCP/IP, FIXT and FIX. The session and application event models and messages are based on versions 1.1 and 5.0 (Service Pack 2) of the FIXT and FIX protocols respectively.

The encryption of messages between the client and server is not supported.

FIX specification: <http://www.fixprotocol.org>

1.1 Purpose

The purpose of this document is to provide a technical description of the drop copy gateway available at Turquoise.

1.2 Readership

This document outlines how to connect to the drop copy gateway and the detailed message types and fields used.

When read in conjunction with the other technical specifications, it is intended that these documents provide all of the details directly connected Turquoise participants require to develop to the trading services.

This document is particularly relevant to technical staff.

1.3 Document Series

This document is part of series of documents providing a holistic view of full trading and information services available from Turquoise.

For reference the full range of documents is outlined below:

- TQ102 - Connectivity Guide
- TQ103 - Trading Technical Parameters
- TQ201 - Trading Gateway (FIX 5.0) Specification
- TQ202 - Post Trade Gateway (FIX 5.0) Specification
- **TQ203 - Drop Copy Gateway (FIX 5.0) Specification (this document)**
- TQ301 - Trading Gateway (Native) Specification
- TQ401 - ITCH Level-2 Market Data Specification
- TQ501 - Guide to Reference Data Services
- TQ601 - Guide to Certification

1.4 Document History

This document has been through the follow iterations:

Issue	Date	Description
R1 1.0	17 Mar 2010	First issue of this document published.
R1 1.1	27 Apr 2010	Second issue of this document published for Release 1 of the Turquoise test platform.
R2 1.0	24 May 2010	First issue of CDS release 2 document published.
R2.1 1.0	09 Jul 2010	First issue of CDS release 2.1 document published.
R2.1 1.2	13 Aug 2010	Issue 1.2, Release 2.1 published.
R2.1 1.3	16 Sep 2010	Issue 1.3, Release 2.1 published.
1.4	01 Apr 2011	Issue 1.4
1.5	31 May 2011	Appended section 3.2 Sponsored Access.
1.6	06 Jul 2011	Updated sections to 4.1 and 4.4 to remove the Test Request message sent at Logon. The Test Request message at Logon will be re-introduced in a later release.
1.7	31 Oct 2011	Support for clearing interoperability.
1.8	27 April 2012	Section 6.4.5 - added exec instruction, added 1094 (PegPriceType)
1.9	31 August 2012	Section 6.4.5 - Added TradeLiquidityIndicator enum of 'C' for Periodic Uncross, added PassiveOnlyOrder field and PriceDifferential field.
2.0	20 Feb 2013	Updated contact details

In subsequent issues, where amendments have been made to the previous version, these changes will be identified using a series of side bars as illustrated opposite.

1.5 Enquiries

Contact Technical Account Management at Turquoise for any functional queries regarding the services outlined in this document. Technical Account Management can be contacted Monday to Friday between 08:00UK and 18:00UK:

- Telephone: +44 (0)20 7797 3939
- Email: londontam@londonstockexchange.com#

2 Service Description

2.1 Services Supported by Trading Gateway

A description of the services (e.g. order types, notification of Market Operations actions, etc.) available via the Trading Gateway is provided in the FIX specification for this interface which vendors are encouraged to read together with this specification.

2.2 Connection Configuration

2.2.1 Real-Time Connections

A real-time client will receive a drop copy of each eligible [Execution Report](#) immediately after it is published.

A member firm connection will be configured to receive a drop copy of all the [Execution Report](#) messages generated for the firm for the events outlined in section [2.3 Supported Events](#). The connection of a service bureau will be configured to receive drop copies for all the firms it serves. If required, a firm or service bureau connection could be configured to only receive drop copies for selected trading mnemonics.

For the purpose of redundancy, the service supports the configuration of multiple drop copy connections to send the same information on the activity of the selected firms/mnemonics.

The identity of the CompID that transmitted the order a particular drop copy relates to will be specified in the header field OnBehalfOfCompID (115).

Refer to sections [5.4 Transmission of Missed Messages](#) and [5.5 Resending Previous Execution Reports](#) for a description of how the [Execution Reports](#) published during the time a real-time client is disconnected from the server may be recovered.

A real-time client may also use the open order download service (outlined in section [2.4 Open Order Download](#)) to recover the status of all active orders in the event of a system failure.

2.2.2 Non-Real Time Connections

[Execution Reports](#) will not be streamed to non-real time clients. Such a client may only connect to the server to use the order download service outlined in section [2.4 Open Order Download](#).

2.3 Supported Events

Clients will receive drop copies of the [Execution Reports](#) generated for the following events:

- (i) Order accepted
- (ii) Order rejected
- (iii) Order executed
- (iv) Order expired
- (v) Order cancelled
- (vi) Order cancel/replaced
- (vii) Order Suspended
- (viii) Trade cancellation

2.4 Open Order Download

Any client may use the [Mass Order Status Request](#) message to download the current status of each active order for a specified trading mnemonic. The total number of [Mass Order Status Requests](#) that a client may submit is limited each day to a configurable value defined by Turquoise. A client may request Turquoise to reset its request count. This feature is intended to help manage an emergency situation and should not be relied upon as a normal practice.

If a request is successful, the server will respond with an [Execution Report](#) for each active order for the specified trading mnemonic. Each such message will include the MassStatus ReqID (584) of the request, an ExecID (17) of "0" and an ExecType (150) of Order Status (I). The last [Execution Report](#) of the partition sent in repose to the request will include a LastRptRequested (912) of Last Message (Y).

The server will transmit a single [Execution Report](#) if the request is rejected or if there are no active orders for the specified trading mnemonic. Such a message will include the MassStatusReqID (584) of the request, an ExecID (17) of "0", an ExecType (150) of Order Status (I) and an OrdStatus (39) of Rejected (8). The message will not include fields that relate to order-specific information (e.g. OrderQty (38), LeavesQty (151), CumQty (14), OrdType (40), etc.). The reason for the rejection will be specified in the field OrdRejReason (103).

2.5 Execution Reports

The [Execution Report](#) message is used to communicate many different events to clients. The events are differentiated by the value in the ExecType (150) field as outlined below.

ExecType	Usage	Ord Status
0	Order Accepted Indicates that a new order has been accepted.	0
8	Order Rejected Indicates that an order has been rejected. The reason for the rejection is specified in the field OrdRejReason (103).	8
F	Order Executed Indicates that an order has been partially or fully filled. The execution details (e.g. price and quantity) are specified.	1, 2
C	Order Expired Indicates that an order has expired in terms of its time qualifier or due to an execution limit.	C
4	Order Cancelled Indicates that an order cancel request has been accepted and successfully processed. This message is also sent if the order was cancelled by Market Operations. In such a scenario the Execution Report will include an ExecRestatementReason (378) of Market Option (8). It will not include an OrigClOrdID (41).	4
5	Order Cancel/Replaced Indicates that an order cancel/replace request has been accepted and successfully processed.	0, 1
D	Order Cancel/Replace by Market Operations Indicates that an order has been amended by Market Operations. The message will include an ExecRestatement Reason (378) of Market Option (8). It will not include an OrigClOrdID (41).	0, 1
H	Trade Cancel Indicates that an execution has been cancelled by Market Operations. An ExecRefID (19) to identify the execution being cancelled will be included.	0, 1, 4, C, E
I	Order Status Reject Indicates that an order mass status request has been rejected.	8

2.5.1 Order Status

As specified in the FIX protocol, the OrdStatus (39) field of an [Execution Report](#) is used to convey the current state of an order. If an order simultaneously exists in more than one order state, the value with highest precedence is reported as the OrdStatus (39). The relevant order statuses are given below from the highest to lowest precedence.

Value	Meaning
2	Filled
4	Cancelled
C	Expired
1	Partially Filled
0	New
8	Rejected

2.5.2 Order and Execution Identifiers

2.5.2.1 Client Order IDs

In the case of orders, the ClOrdID (11) included in each [Execution Report](#) will be that specified when the order was submitted. An order's ClOrdID (11) will be updated each time an Order Cancel/Replace Request or an Order Cancel Request is accepted.

2.5.2.2 Order IDs

The server will use the OrderID (37) field to affix the order identification numbers of the matching system. Order IDs will be unique across trading days.

In terms of the FIX protocol, unlike ClOrdID (11) which requires a chaining through Cancel/Replace Requests and Cancel Requests, the OrderID (37) of an order will remain constant throughout its life.

2.5.2.3 Execution IDs

The server will use the ExecID (17) field to affix the execution identification numbers of the matching system. Execution IDs will be unique across trading days.

2.5.2.4 Trade Match ID

The TradeMatchID (880) matches exactly with the TradeID (1003) on the Trade Capture Report of Post Trade gateway and the TradeMatchID (880) from FIX trading Gateway. This also matches the TradeMatchID field from the Native Trading gateway as well as the ITCH gateway which are in binary format. However this is in base 62 and needs converting to an 8 byte integer for comparison.

Trade Match ID generated for a normal trade being disseminated through each gateway.

FIX Trading	Native Trading	Drop Copy	Post Trade	ITCH
TradeMatchID	TradeMatchID	TradeMatchID	TradeID	Trade MatchID
MOTGqwVehh	297918988148865237	MOTGqwVehh	MOTGqwVehh	297918988148865237

Trade Match ID from Drop Copy gateway mapped to CCP gateway for normal trade

FIX Trading	Native Trading	CCP Gateway
TradeMatchID	TradeMatchID	Exec ID
MOTGqwVehh	297918988148865237	BMOTGqwVehh

Trade Match ID mapped to CCP gateway if the same trade was cancelled.

FIX Trading	Native Trading	CCP Gateway
TradeMatchID	TradeMatchID	Exec ID
MOTGqwVehh	297918988148865237	1BMOTGqwVehh

2.5.3 Instrument Identification

Instruments will be identified using the SecurityID (48) field. It is required to specify SecurityID Source (22) field as well.

2.5.4 Party Identification

ID	Description	Relevant FIX Tags
Member ID	Identifier of the member the interest is submitted under.	PartyRole (452) = 1 PartyID (448)
Trader Group	Identifier of the trader group the interest is submitted under.	PartyRole (452) = 76 PartyID (448)
Trader ID	Identifier of the trader the interest is submitted under.	PartyRole (452) = 12 PartyID (448)
Client Reference	Client reference information applicable to an order	Account (1)

2.6 Timestamps and Dates

The timestamps `SendingTime` (52), `OrigSendingTime` (122) and `TransactTime` (60) are in UTC and in the `YYYYMMDD-HH:MM:SS.sss` format. `ExpireTime` (126) is in UTC and in the `YYYYMMDD-HH:MM:SS` format.

All dates (i.e. `MaturityDate` (541) and `ExpireDate` (432)) are in the `YYYYMMDD` format and specified in the local date for the server (i.e. not in UTC).

3 Connectivity

3.1 ComplIDs

The ComplID of each client must be registered with Turquoise before FIX communications can begin. A single client may have multiple connections to the server (i.e. multiple FIX sessions, each with its own ComplID).

The ComplID of the server will be FGW. The messages sent to the server should contain the ComplID assigned to the client in the field SenderComplID (49) and FGW in the field TargetComplID (56). The messages sent from the server to the client will contain FGW in the field SenderComplID (49) and the ComplID assigned to the client in the field TargetComplID (56).

3.1.1 Passwords

Each new ComplID will be assigned a password on registration. Clients are strongly encouraged to change the password to one of their choosing via the [Logon](#) message. The status of the new password (i.e. whether it is accepted or rejected) will be specified in the SessionStatus (1409) field of the [Logon](#) sent by the server to confirm the establishment of a FIX connection. The new password will, if accepted, be effective for subsequent logins.

In terms of the password policy of Turquoise, the password of each ComplID should be changed. If not, the password will expire and the client will be unable to login to the server. In such a case, the client should contact Turquoise to have its password reset. The SessionStatus (1409) of the server's [Logon](#) message will be 'Password Due to Expire (2)'.

3.2 Sponsored Access - Monitoring Users

In order for a Member firm to be able to constantly monitor their 'Sponsored Users' via a Drop Copy Gateway connection, they will need to register with Turquoise their 'Monitoring User(s)' and advise which Sponsored Users are to be monitored by which 'Monitoring User'. Typically a firm will only have one 'Monitoring User' however it is possible to require more than one should the firm have more than one Drop Copy Gateway.

In order for a 'Sponsored User' to place orders, the firm's assigned 'Monitoring User' will need to have established a successful connection to the Drop Copy Gateway.

Should a Member Firm's 'Monitoring User' lose the ability to monitor their 'Sponsored Users' (e.g. Disconnect or lose connection) and not reconnect within the configurable amount of time, their 'Sponsored Users' will be restricted from submitting new orders, while all their existing orders will be cancelled.

3.3 Production IP Address and Ports

The IP addresses and ports for the post trade gateway will be published in a separate configuration document.

3.4 Failover and Recovery

The system has been designed with fault tolerance and disaster recovery technology that ensures that trading should continue in the unlikely event of a process or site outage.

If the client is unexpectedly disconnected from the server, it should attempt to re-connect to primary site within a few seconds. The client should only attempt to connect to the secondary IP address and port if so requested by Turquoise.

If a service interruption (due to Order Cache Primary failing over to its Mirror or both Order Cache Processes going down) occurs in the Drop Copy Gateway while it is servicing an [Order Mass Status Request](#), Drop Copy Gateway will send an unsolicited [Execution Report](#) with a 'Rejected' state (it would include the MassStatus ReqID (584) of the request, an ExecID (17) of "0", an ExecType (150) of Order Status (I) an OrdStatus (39) of Rejected (8)) and an OrdRejReason (103) of "10005"). When the client receives this, they are expected to try and re-request.

In the unlikely event of a site outage disaster on the Turquoise system, all orders will be cancelled and all unicast and multicast connectivity will be unavailable until the secondary site is invoked.

4 FIX Connections and Sessions

4.1 Establishing a FIX Connection

FIX connections and sessions between the client and server are maintained as specified in the FIX protocol.

Each client will use the assigned IP address and port to establish a TCP/IP session with the server. The client will initiate a FIX session at the start of each trading day by sending the [Logon](#) message. The client will identify itself using the SenderCompID (49) field. The server will validate the CompID, password and IP address of the client.

Once the client is authenticated, the server will respond with a [Logon](#) message. The SessionStatus (1409) of this message will be Session Active (0). If the client's [Logon](#) message included the field NewPassword (925) and the client is authenticated, the SessionStatus (1409) of the [Logon](#) sent by the server will be Session Active (0).

When the client sends a logon with a sequence number higher than expected by the FIX Gateway, the FIX gateway will send a [Resend Request](#) and once the response/s to the [Resend Request](#) is processed by the FIX Gateway, the FIX Gateway would send a [Test Request](#) to make sure both the client and server is in sync before sending out any missed or new application messages.

The client must wait for the server's [Logon](#) before sending additional messages. If additional messages are received from the client before the exchange of [Logon](#) messages, the TCP/IP connection with the client will be disconnected.

If a logon attempt fails because of an invalid SenderCompID, TargetCompID, IP address, invalid password or because the user does not have the appropriate privileges, the server will break the TCP/IP connection with the client without sending a [Logout](#) or [Reject](#) message. If during a logon of a SenderCompID, the server receives a second connection attempt via different TCP/IP connection while a valid FIX session is already underway for that same SenderCompID, the server will break the TCP/IP connection with the second connection without sending a [Logout](#) or [Reject](#) message. As the logon attempt failed, the server will not increment the next inbound message sequence number expected from the client.

If a logon attempt fails because of an invalid or expired password a locked CompID or if logins are not currently permitted, the server will send a [Logout](#) message and then break the TCP/IP connection with the client.

If during a logon of a SenderCompID, the server receives a second connection attempt via the same TCP/IP connection while a valid FIX session is already underway for that same SenderCompID, the server will send a [Reject](#) message and then break the TCP/IP connection with the client. The

server will increment the next inbound message sequence number expected from the client as well as its own outbound message sequence number.

If a logon attempt fails because of a session level failure (e.g. due to invalid EncryptMethod or DefaultApplVerID) the inbound sequence number and the outbound sequence number both will not be incremented. In this scenario the message sequence number 1 will be sent with the [Logout](#) message.

However if a session level failure occurs due to a message sent by a client which contains a sequence number that is less than what is expected and the PossDupFlag (43) not being set to “Y”, then the server will send a Logout message and terminate the FIX connection. In this scenario the inbound sequence number will not be incremented but the outbound sequence number will be incremented.

If during a logon of a SenderCompID, the server receives a second connection attempt via the same TCP/IP connection while a valid FIX session is already underway for that same SenderCompID, the server will send a [Reject](#) message and then break the TCP/IP connection with the client. The server will increment the next inbound message sequence number expected from the client as well as its own outbound message sequence number.

4.2 Maintaining a FIX Session

4.2.1 Message Sequence Numbers

As outlined in the FIX protocol, the client and server will each maintain a separate and independent set of incoming and outgoing message sequence numbers. Sequence numbers should be initialized to 1 (one) at the start of the FIX session and be incremented throughout the session.

Monitoring sequence numbers will enable parties to identify and react to missed messages and to gracefully synchronize applications when reconnecting during a FIX session.

If any message sent by the client contains a sequence number that is less than what is expected and the PossDupFlag (43) is not set to “Y”, the server will send a [Logout](#) message and terminate the FIX connection. The [Logout](#) will contain the next expected sequence number in the Text (58) field.

A FIX session will not continue to the next trading day. The server will initialize its sequence numbers at the start of each day. The client is expected to employ the same logic.

4.2.2 Heartbeats

The client and server will use the [Heartbeat](#) message to exercise the communication line during periods of inactivity and to verify that the interfaces at each end are available. The heartbeat interval will be the HeartBtInt (108) specified in the client’s [Logon](#) message.

The server will send a [Heartbeat](#) anytime it has not transmitted a message for the heartbeat interval. The client is expected to employ the same logic.

If the server detects inactivity for a period longer than the heartbeat interval plus a reasonable transmission time, it will send a [Test Request](#) message to force a [Heartbeat](#) from the client. If inactivity continues for a second heartbeat interval plus a reasonable transmission time, the server will send a [Logout](#) and break the TCP/IP connection with the client. The client is expected to employ similar logic if inactivity is detected on the part of the server.

4.2.3 Increasing Expected Sequence Number

The client or server may use the [Sequence Reset](#) message in Gap Fill mode if it wishes to increase the expected incoming sequence number of the other party.

The client or server may also use the [Sequence Reset](#) message in Sequence Reset mode if it wishes to increase the expected incoming sequence number of the other party.

The Sequence Reset mode should only be used to recover from an emergency situation. It should not be relied upon as a regular practice.

4.3 Terminating a FIX Session

The client is expected to terminate each FIX connection at the end of each trading day before the server shuts down. The client will terminate a connection by sending the [Logout](#) message. The server will respond with a [Logout](#) to confirm the termination. The client will then break the TCP/IP connection with the server.

All open TCP/IP connections will be terminated by the server when it shuts down (a [Logout](#) will not be sent). Under exceptional circumstances the server may initiate the termination of a connection during the trading day by sending the [Logout](#) message.

If, during the exchange of [Logout](#) messages, the client or sever detects a sequence gap, it should send a [Resend Request](#).

4.4 Re-Establishing a FIX Session

If a FIX connection is terminated during the trading day it may be re-established via an exchange of [Logon](#) messages.

Once the client is authenticated, the server will respond with a [Logon](#) message. The SessionStatus (1409) of this message will be Session Active (0). If the client's [Logon](#) message included the field NewPassword (925) and the client is authenticated, the SessionStatus (1409) of the [Logon](#) sent by the server will be Session Active (0).

When the client sends a logon with a sequence number higher than expected by the FIX Gateway, the FIX gateway will send a [Resend Request](#) and once the response/s to the [Resend Request](#) is processed by the FIX Gateway, the FIX Gateway would send a [Test Request](#) to make sure both the client and server is in sync before sending out any missed or new application messages.

The client must wait for the server's [Logon](#) before sending additional messages. If additional messages are received from the client before the exchange of [Logon](#) messages, the TCP/IP connection with the client will be disconnected.

Once the FIX session is re-established successfully, the message sequence numbers will continue from the last message successfully transmitted prior to the termination.

4.4.1 Resetting Sequence Numbers: Starting a New FIX Session

4.4.1.1 Reset Initiated by the Client

If the client requires both parties to initialize (i.e. reset to 1) sequence numbers, it may use the ResetSeqNumFlag (141) field of the [Logon](#) message. The server will respond with a [Logon](#) with the ResetSeqNumFlag (141) field set to “Y” to confirm the initialization of sequence numbers.

A client may also manually inform Market Operations that it would like the server to initialize its sequence numbers prior to the client’s next login attempt.

These features are intended to help a client manage an emergency situation. Initializing sequence numbers on a re-login should not be relied upon as a regular practice.

4.4.1.2 Reset Initiated by the Server

The system has been designed with fault tolerance and disaster recovery technology that should ensure that the server retains its incoming and outgoing message sequence numbers for each client in the unlikely event of an outage.

However, clients are required to support a manual request by Turquoise to initialize sequence numbers prior to the next login attempt.

5 Recovery

5.1 Resend Requests

The client may use the [Resend Request](#) message to recover any lost messages. As outlined in the FIX protocol, this message may be used in one of three modes:

- (i) To request a single message. The BeginSeqNo (7) and EndSeqNo (16) should be the same.
- (ii) To request a specific range of messages. The BeginSeqNo (7) should be the first message of the range and the EndSeqNo (16) should be the last of the range.
- (iii) To request all messages after a particular message. The BeginSeqNo (7) should be the sequence number immediately after that of the last processed message and the EndSeqNo (16) should be zero (0).

5.2 Possible Duplicates

The server handles possible duplicates according to the FIX protocol. The client and server will use the PossDupFlag (43) field to indicate that a message may have been previously transmitted with the same MsgSeqNum (34).

5.3 Possible Resends

The server does not handle possible resends for client-initiated messages and ignores the value in the PossResend (97) field of such messages.

The server may, in the circumstances outlined in sections [5.4 Transmission of Missed Messages](#) and [5.5 Resending Previous Execution Reports](#), use the PossResend (97) field to indicate that an [Execution Report](#) may have already been sent under a different MsgSeqNum (34). The client should validate the ExecID (17) of such a message against those of [Execution Reports](#) already received during the current trading day.

If an [Execution Report](#) with same ExecID (17) had been processed, the resent message should be ignored. If the same ExecID (17) had not been processed, the [Execution Report](#) should be processed.

5.4 Transmission of Missed Messages

The [Execution Reports](#) generated during a period when a client is disconnected from the server will be sent to the client when it next reconnects. In the unlikely event the disconnection was due to an outage of the server, all such messages will include a PossResend (97) of “Y”.

The [Execution Report](#) messages are automatically generated when a client reconnects. Clients are not required to explicitly request for the messages. The resend request applies only when the server has sent messages that a client has not received.

5.5 Resending Previous Execution Reports

A client may manually inform Turquoise that it would like the server to resend all of the [Execution Report](#) messages that were generated for the client during the current trading day when it next logs in. All resent [Execution Report](#) messages will include a PossResend (97) of “Y”.

This feature is intended to help a client manage an emergency situation and it should not be relied upon as a regular practice.

6 Message Formats

This section provides details on the header and trailer, the seven administrative messages and twelve application messages utilized by the server. Any message not included in this section will be ignored by the server.

6.1 Supported Message Types

6.1.1 Administrative Messages

All administrative messages may be initiated by either the client or the server.

Message	MsgType	Usage
Logon	A	Allows the client and server to establish a FIX session.
Logout	5	Allows the client and server to terminate a FIX session.
Heartbeat	0	Allows the client and server to exercise the communication line during periods of inactivity and verify that the interfaces at each end are available.
Test Request	1	Allows the client or server to request a response from the other party if inactivity is detected.
Resend Request	2	Allows for the recovery of messages lost during a malfunction of the communications layers.
Reject	3	Used to reject a message that does not comply with FIXT.
Sequence Reset	4	Allows the client or server to increase the expected incoming sequence number of the other party.

6.1.2 Application Messages

6.1.2.1 Client-Initiated

Message	MsgType	Usage
Order Mass Status Request	AF	Allows the client to request the status of all active orders for a particular trading mnemonic.

6.1.2.2 Server-Initiated

Message	MsgType	Usage
Execution Report	8	Indicates one of the following: (i) Order accepted (ii) Order rejected (iii) Order executed (iv) Order expired (v) Order cancelled (vi) Order cancel/replaced (vii) Trade cancellation (viii) Order mass status request rejected

6.2 Message Header and Trailer

6.2.1 Message Header

Tag	Field Name	Req	Description						
8	BeginString	Y	FIXT.1.1						
9	BodyLength	Y	Number of characters after this field up to and including the delimiter immediately preceding the CheckSum.						
35	MsgType	Y	Message type.						
49	SenderCompID	Y	CompID of the party sending the message.						
56	TargetCompID	Y	CompID of the party the message is sent to. <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>FGW</td> <td>Drop Copy Gateway</td> </tr> </tbody> </table>	Value	Meaning	FGW	Drop Copy Gateway		
Value	Meaning								
FGW	Drop Copy Gateway								
115	OnBehalfOf CompID	N	Required for server-initiated application messages. This will be the CompID of the connection that originated the order referenced in the message being drop copied.						
34	MsgSeqNum	Y	Sequence number of the message.						
43	PossDupFlag	N	Whether the message was previously transmitted under the same MsgSeqNum (34). Absence of this field is interpreted as Original Transmission (N). <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>Y</td> <td>Possible Duplicate</td> </tr> <tr> <td>N</td> <td>Original Transmission</td> </tr> </tbody> </table>	Value	Meaning	Y	Possible Duplicate	N	Original Transmission
Value	Meaning								
Y	Possible Duplicate								
N	Original Transmission								

97	PossResend	N	<p>Whether the message was previously transmitted under a different MsgSeqNum (34). Absence of this field is interpreted as Original Transmission (N).</p> <p>Value Meaning</p> <hr/> <p>Y Possible Resend</p> <hr/> <p>N Original Transmission</p>
52	SendingTime	N	Time the message was transmitted.
122	OrigSendingTime	N	Time the message was originally transmitted. If the original time is not available, this should be the same value as SendingTime (52). Required if PossDupFlag (43) is Possible Duplicate (Y).
1128	ApplVerID	N	<p>Version of FIX used in the message. Required if the message is generated by the server.</p> <p>Value Meaning</p> <hr/> <p>9 FIX50SP2</p>

6.2.2 Message Trailer

Tag	Field Name	Req	Description
10	Checksum	Y	

6.3 Administrative Messages

6.3.1 Logon

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	A = Logon
Message Body			
98	EncryptMethod	Y	Method of encryption. Value Meaning 0 None
108	HeartBtInt	Y	Indicates the heartbeat interval in seconds.
141	ResetSeqNum Flag	N	Indicates whether the client and server should reset sequence numbers. Absence of this field is interpreted as Do Not Reset Sequence Numbers (N). Value Meaning Y Reset Sequence Numbers N Do Not Reset Sequence Numbers
554	Password	Y	Password assigned to the CompID. Required if the message is generated by the client.
925	NewPassword	N	New password for the CompID.
1409	SessionStatus	N	Status of the FIX session or the request to change the password. Required if the message is generated by the server. Value Meaning 0 Session Active 2 Password Due to Expire
1137	DefaultAppVerID	Y	Default version of FIX messages used in this session. Value Meaning 9 FIX50SP2
Standard Trailer			

6.3.2 Logout

Tag	Field Name	Req	Description																		
Standard Header																					
35	MsgType	Y	5 = Logout																		
Message Body																					
1409	SessionStatus	N	<p>Status of the FIX session. Required if the message is generated by the server.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>Session logout complete</td> </tr> <tr> <td>5</td> <td>Invalid password</td> </tr> <tr> <td>6</td> <td>Account locked</td> </tr> <tr> <td>7</td> <td>Logons are not allowed at this time</td> </tr> <tr> <td>8</td> <td>Password expired</td> </tr> <tr> <td>100</td> <td>Other</td> </tr> <tr> <td>101</td> <td>Logout due to session level failure</td> </tr> <tr> <td>102</td> <td>Logout by Market Operations</td> </tr> </tbody> </table>	Value	Meaning	4	Session logout complete	5	Invalid password	6	Account locked	7	Logons are not allowed at this time	8	Password expired	100	Other	101	Logout due to session level failure	102	Logout by Market Operations
Value	Meaning																				
4	Session logout complete																				
5	Invalid password																				
6	Account locked																				
7	Logons are not allowed at this time																				
8	Password expired																				
100	Other																				
101	Logout due to session level failure																				
102	Logout by Market Operations																				
58	Text	N	<p>The field will contain the next expected sequence number if the server terminated the connection after receiving a sequence number that was less than what was expected.</p> <p>In other cases the field will contain the reason for the logout.</p>																		
Standard Trailer																					

6.3.3 Heartbeat

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	0 = Heartbeat
Message Body			
112	TestReqID	N	<p>Required if the heartbeat is a response to a Test Request. The value in this field should echo the TestReqID (112) received in the Test Request.</p>
Standard Trailer			

6.3.4 Test Request

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	1 = Test Request
Message Body			
112	TestReqID	Y	Identifier for the request.
Standard Trailer			

6.3.5 Resend Request

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	2 = Resend Request
Message Body			
7	BeginSeqNo	Y	Sequence number of first message in range.
16	EndSeqNo	Y	Sequence number of last message in range.
Standard Trailer			

6.3.6 Reject

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	3 = Reject
Message Body			
45	RefSeqNum	Y	MsgSeqNum (34) of the rejected message.
372	RefMsgType	N	MsgType (35) of the rejected message.
371	RefTagID	N	If a message is rejected due to an issue with a particular field its tag number will be indicated.
373	SessionReject Reason	N	Code specifying the reason for the reject. Refer to section 7.2.1 Reject for a list of reject codes.
58	Text	N	Text specifying the reason for the rejection.
Standard Trailer			

6.3.7 Sequence Reset

Tag	Field Name	Req	Description						
<u>Standard Header</u>									
35	MsgType	Y	4 = Sequence Reset						
Message Body									
36	NewSeqNo	Y	Sequence number of the next message to be transmitted.						
123	GapFillFlag	N	<p>Mode in which the message is being used. Absence of this field is interpreted as Sequence Reset (N).</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>Y</td> <td>Gap Fill</td> </tr> <tr> <td>N</td> <td>Sequence Reset</td> </tr> </tbody> </table>	Value	Meaning	Y	Gap Fill	N	Sequence Reset
Value	Meaning								
Y	Gap Fill								
N	Sequence Reset								
<u>Standard Trailer</u>									

6.4 Application Messages: (Client-Initiated)

6.4.1 Order Mass Status Request

Tag	Field Name	Req	Description				
<u>Standard Header</u>							
35	MsgType	Y	AF = Order Mass Status Request				
Message Body							
584	MassStatusReqID	Y	Client specified identifier of the mass status request.				
585	MassStatusReqType	Y	<p>Type of mass status request.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>All open orders of specified PartyID</td> </tr> </tbody> </table>	Value	Meaning	8	All open orders of specified PartyID
Value	Meaning						
8	All open orders of specified PartyID						
453	NoPartyIDs	Y	Number of party identifiers. The value in this field can be "1" or "2".				

➔	448	PartyID	Y	Identifier of the trading mnemonic. Set to the MemberFirmID for internalised trade executions else set to the CCP name: EMCF EuroCCP LCH X-Clear
➔	447	PartyID Source	Y	Value Meaning D Proprietary/Custom Code
➔	452	Party Role	Y	Role of the PartyID (448). Value Meaning 76 Trader Group
Standard Trailer				

6.5 Application Messages: (Server-Initiated)

6.4.5 Execution Report

Tag	Field Name	Req	Description
Standard Header			
35	MsgType	Y	8 = Execution Report
Message Body			
17	ExecID	Y	Server specified identifier of the message. Will be "0" if ExecType (150) is Order Status (I).
11	ClOrdID	Y	Client specified identifier of the order.
41	OrigClOrdID	N	OrigClOrdID (41), if any, that was submitted with the order cancel or cancel/replace request.
37	OrderID	Y	Server specified identifier of the order.

198	SecondaryOrderID	Y	Indicates the corresponding Market Data (ITCH) Order ID. This is a 16 character hexadecimal ascii string that needs converting to 8 byte binary for comparison with the ITCH Order ID
584	MassStatus ReqID	N	Client specified identifier of the mass status request. Required is the message in sent in response to such a request.
912	LastRpt Requested	N	Indicates the last message sent in response to a mass order status request. Value Meaning Y Last Message
150	ExecType	Y	Reason the execution report was generated. Value Meaning 0 New 4 Cancelled 5 Replaced 8 Rejected C Expired D Restated F Trade H Trade Cancel
19	ExecRefID	N	Reference to the execution being cancelled. Required if ExecType (150) is Trade Cancel (H).
378	Exec Restatement Reason	N	Reason the order was restated. Required if ExecType (150) is Restated (D). Value Meaning 8 Market Option

39	OrdStatus	Y	Current status of the order. Value Meaning 0 New 1 Partially Filled 2 Filled 4 Cancelled 8 Rejected C Expired
103	OrdRejReason	N	Code specifying the reason for the reject. Required if ExecType (150) is Rejected (8).
58	Text	N	Text specifying the reason for the rejection or expiration
32	LastQty	N	Quantity executed in this fill. Required if ExecType (150) is Trade (F).
31	LastPx	N	Price of this fill. Required if ExecType (150) is Trade (F).
151	LeavesQty	Y	Quantity available for further execution. Will be "0" if OrdStatus (39) is Filled (2), Cancelled (4), Rejected (8) or Expired (C).
14	CumQty	Y	Total cumulative quantity filled.
55	Symbol	N	MTF Common Symbol.
48	SecurityID	Y	Identifier of the instrument.
22	SecurityIDSource	Y	Identifier of the source of the SecurityID (48) value. Value Meaning 4 ISIN
9303	RoutingInst	N	Indicate the liquidity pool Value Meaning I Integrated order book M Dark Midpoint order book
15	Currency	N	Currency Code as per ISO 4217 Currency Code List
207	SecurityExchange	N	Market Identifier Code as per ISO 10383

18	ExecInst	N	<p>Applicable to the Dark Midpoint.</p> <table border="1"> <thead> <tr> <th data-bbox="810 215 911 253">Value</th> <th data-bbox="979 215 1102 253">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="874 271 895 300">x</td> <td data-bbox="979 271 1334 344">Continuous and Periodic un-crossings</td> </tr> <tr> <td data-bbox="874 362 895 392">z</td> <td data-bbox="979 362 1286 436">Periodic un-crossings only</td> </tr> <tr> <td data-bbox="874 454 895 483">y</td> <td data-bbox="979 454 1209 483">Continuous only</td> </tr> </tbody> </table>	Value	Meaning	x	Continuous and Periodic un-crossings	z	Periodic un-crossings only	y	Continuous only
Value	Meaning										
x	Continuous and Periodic un-crossings										
z	Periodic un-crossings only										
y	Continuous only										
9730	TradeLiquidityIndicator	N	<p>Whether the order added or removed liquidity. Required only for messages generated for a trades or trade cancellations. Will be populated for both automatic trades (AT) and auction trades (UT). Possible values are:</p> <table border="1"> <thead> <tr> <th data-bbox="820 797 906 835">Value</th> <th data-bbox="932 797 1050 835">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="852 853 873 882">A</td> <td data-bbox="932 853 1161 882">Added Liquidity</td> </tr> <tr> <td data-bbox="852 900 873 929">R</td> <td data-bbox="932 900 1203 929">Removed Liquidity</td> </tr> <tr> <td data-bbox="852 994 873 1023">C</td> <td data-bbox="932 958 1295 1066">Periodic Uncrossing execution (valid for Dark orders only)</td> </tr> </tbody> </table>	Value	Meaning	A	Added Liquidity	R	Removed Liquidity	C	Periodic Uncrossing execution (valid for Dark orders only)
Value	Meaning										
A	Added Liquidity										
R	Removed Liquidity										
C	Periodic Uncrossing execution (valid for Dark orders only)										
27010	PassiveOnlyOrder	N	Value submitted with the order.								

27011	PriceDifferential	N	<table border="1"> <thead> <tr> <th data-bbox="810 159 922 197">Value</th> <th data-bbox="938 159 1066 197">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="858 219 882 248">A</td> <td data-bbox="938 219 1358 443">Aggressive. Any residual visible quantity is then stamped based on its deviation from the current BBO. Set to 'P' if hidden/dark.</td> </tr> <tr> <td data-bbox="858 454 882 483">B</td> <td data-bbox="938 454 1182 483">New visible BBO.</td> </tr> <tr> <td data-bbox="858 506 882 535">1</td> <td data-bbox="938 506 1182 535">Join visible BBO.</td> </tr> <tr> <td data-bbox="858 557 882 586">2</td> <td data-bbox="938 557 1278 640">Joining/setting 2nd best price point.</td> </tr> <tr> <td data-bbox="858 651 882 680">3</td> <td data-bbox="938 651 1278 734">Joining/setting 3rd best price point.</td> </tr> <tr> <td data-bbox="858 745 882 775">4</td> <td data-bbox="938 745 1278 828">Joining/setting 4th best price point.</td> </tr> <tr> <td data-bbox="858 840 882 869">5</td> <td data-bbox="938 840 1278 922">Joining/setting 5th best price point.</td> </tr> <tr> <td data-bbox="858 934 882 963">6</td> <td data-bbox="938 934 1278 1016">Joining/setting 6th best price point.</td> </tr> <tr> <td data-bbox="858 1028 882 1057">7</td> <td data-bbox="938 1028 1278 1111">Joining/setting 7th best price point.</td> </tr> <tr> <td data-bbox="858 1122 882 1151">8</td> <td data-bbox="938 1122 1278 1205">Joining/setting 8th best price point.</td> </tr> <tr> <td data-bbox="858 1216 882 1245">9</td> <td data-bbox="938 1216 1278 1352">Joining/setting 9th best price point (or joining/setting a worse price point).</td> </tr> <tr> <td data-bbox="858 1373 882 1402">P</td> <td data-bbox="938 1373 1342 1480">Passive. Valid for dark and large in scale hidden orders only.</td> </tr> </tbody> </table>	Value	Meaning	A	Aggressive. Any residual visible quantity is then stamped based on its deviation from the current BBO. Set to 'P' if hidden/dark.	B	New visible BBO.	1	Join visible BBO.	2	Joining/setting 2 nd best price point.	3	Joining/setting 3 rd best price point.	4	Joining/setting 4 th best price point.	5	Joining/setting 5 th best price point.	6	Joining/setting 6 th best price point.	7	Joining/setting 7 th best price point.	8	Joining/setting 8 th best price point.	9	Joining/setting 9 th best price point (or joining/setting a worse price point).	P	Passive. Valid for dark and large in scale hidden orders only.
Value	Meaning																												
A	Aggressive. Any residual visible quantity is then stamped based on its deviation from the current BBO. Set to 'P' if hidden/dark.																												
B	New visible BBO.																												
1	Join visible BBO.																												
2	Joining/setting 2 nd best price point.																												
3	Joining/setting 3 rd best price point.																												
4	Joining/setting 4 th best price point.																												
5	Joining/setting 5 th best price point.																												
6	Joining/setting 6 th best price point.																												
7	Joining/setting 7 th best price point.																												
8	Joining/setting 8 th best price point.																												
9	Joining/setting 9 th best price point (or joining/setting a worse price point).																												
P	Passive. Valid for dark and large in scale hidden orders only.																												
20000	TypeOfTrade	N	<p data-bbox="810 1498 1262 1570">Indicates whether the executed portion is visible or hidden.</p> <table border="1"> <thead> <tr> <th data-bbox="826 1585 922 1624">Value</th> <th data-bbox="938 1585 1066 1624">Meaning</th> </tr> </thead> <tbody> <tr> <td data-bbox="858 1641 882 1671">0</td> <td data-bbox="938 1641 1023 1671">Visible</td> </tr> <tr> <td data-bbox="858 1693 882 1722">1</td> <td data-bbox="938 1693 1023 1722">Hidden</td> </tr> </tbody> </table>	Value	Meaning	0	Visible	1	Hidden																				
Value	Meaning																												
0	Visible																												
1	Hidden																												
Component Block <Trading Party>		Y	Identifier of the trading party.																										
1	Account	N	Client reference information.																										

40	OrdType	Y	Type of the order. Value Meaning 1 Market 2 Limit P Pegged
59	TimeInForce	N	Time qualifier of the order. Absence of this field is interpreted as Day (0). Value Meaning 0 Day 1 Good Till Cancel (GTC) 3 Immediate or Cancel (IOC) 4 Fill or Kill (FOK) 6 Good Till Date (GTD)
126	ExpireTime	N	Time the order expires which must be a time during the current trading day. Required if TimeInForce (59) is GTD (6) and ExpireDate (432) is not specified.
432	ExpireDate	N	Date the order expires. Required if TimeInForce (59) is GTD (6) and ExpireTime (126) is not specified.
54	Side	Y	Side of the order that was executed. Value Meaning 1 Buy 2 Sell
38	OrderQty	Y	Total order quantity.
1138	DisplayQty	N	Quantity currently displayed in the order book.
110	MinQty	N	Value submitted with the order.
44	Price	N	Limit price. Required if OrderType (40) is Limit (2)
581	AccountType	Y	Type of account associated with the order. Value Meaning 1 Client 3 House

528	OrderCapacity	Y	Capacity of the order. Value Meaning A Agency P Principal R Riskless Principal
60	TransactTime	Y	Time the transaction represented by the Execution Report occurred.
526	SecondaryClOrdID	N	A secondary id assigned by the trading party
583	ClOrdLinkID	N	Permits order originators to tie together groups of orders in which trades resulting from orders are associated for a specific purpose. eg. Calculation of average execution price.
1094	PegPriceType	N	Only applicable to Turquoise Dark book, will not be sent for Turquoise lit executions. Value Meaning 0 Midpoint
Standard Trailer			

6.6 Components of Application Messages

6.6.1 Trading Party

Tag	Field Name		Req	Description
453	NoPartyIDs		Y	Number of party identifiers. The value in this field can be "1", "2" or "3".
➔	448	PartyID	Y	Identifier of the party. Set to the MemberFirmID for internalised trade executions else set to the CCP name: EMCF EuroCCP LCH X-Clear

➔	447	PartyID Source	Y	Value Meaning <hr/> D Proprietary/Custom Code
➔	452	Party Role	Y	Role of the specified PartyID (448). Value Meaning <hr/> 12 Trader ID <hr/> 17 Counterparty Firm <hr/> 76 Trader Group

7 Reject Codes

7.1 Reject (35=3)

Session Reject Reason	Meaning
1	Required tag missing
2	Invalid tag
4	Tag specified without a value
5	Value is incorrect (out of range) for this tag
6	Incorrect data format for value
9	CompID problem
11	Invalid MsgType
13	Tag appears more than once
14	Tag specified out of required order
15	Repeating group fields out of order
18	Invalid or unsupported application version
99	Other

7.2 Execution Report (35=8)

OrdRej Reason	Meaning
2	Exchange closed
5	Unknown order
6	Duplicate order (i.e. duplicate ClOrdID)
9	Unknown user (Owner ID)
16	Price exceeds current price band
18	Invalid price increment
99	Other
10000	No open orders for specified Trader Group/Trader
10001	Request limit for day reached
10003	Request not permitted for specified trading mnemonic
10004	Not authorised to request an open order download
10005	Open order download not permitted at this time
10006	Unknown trading party

7.3 Business Message Reject (35=j)

OrdRej Reason	Meaning
0	Other
1	Unknown ID
2	Unknown Security
3	Unsupported Message Type
4	Application Unavailable
5	Conditionally Required Field Missing

8 APPENDIX A

8.1 Error & Reject Text Strings

Text(58)
Instrument Setup Error - Invalid Order Book
Invalid Security ID Source
Invalid Routing Instruction
ISIN/CCY /MIC provided is inconsistent with MTF Symbol
Trader Group not specified on message
Unsupported Message Type
Application Not Available
Unknown Account Type
Unknown Execution Instruction
Unknown ID
Required tag missing
Tag not defined for this message type
Unknown SecurityID
Invalid tag
Tag specified without a value
OrigCLOrdID or OrderID required
PartyID required
PartyIDSource required
PartyRole required
Price unset for limit order
SecurityID required
SecurityIDSource required
StopPx unset for stop order
Value is out of range for this tag
MarketSegmentID required
Value is incorrect (out of range) for this tag
Duplicate Order (i.e. duplicate CLOrdID)
FirmTradeID is not specified
Invalid MsgType
Tag appears more than once
Repeating group fields out of order
Incorrect NumInGroup count for repeating group
Invalid or unsupported application version
Invalid price increment
Tag contains non-numeric character

Copyright © February 2013 Turquoise Global Holdings Limited.
Registered in England and Wales No. 07102717.

Turquoise Global Holdings Limited has used all reasonable efforts to ensure that the information contained in this publication is correct at the time of going to press, but shall not be liable for decisions made in reliance on it.

London Stock Exchange is a registered trade mark of London Stock Exchange plc.
Turquoise is a registered trade mark of Turquoise Trading Limited

Turquoise Services Limited/Turquoise Trading Limited

10 Paternoster Square
London EC4M 7LS
Telephone: +44 (0)20 7797 1000

<http://www.tradeturquoise.com>