Universal Trading Platform for Cash Markets

Error List

Version 1.5 - 25 May 2010

DOCUMENT HISTORY

Version	Date	Author	Description	Status
0.4	5 Nov. 2008	NYSE Technologies	Enriched initial version, after new analysis done on an extract from code of all error messages (extract done the October 10th).	Deliverable
0.5	18 Dec. 2008	NYSE Technologies	Superficial Checks on Requests section corrected. Order Modification: Price, Validity & Quantity qualifier compatibility corrected. Price Input: PriceInputType / Price / External parameters combinations updated. New table: Cancel / Replace and Cancel: OrderID/OrigClOrdID values. Added: new error message "OrderID not found" on C/R request in the case of confirmation after Collar rejection. Document layout reformatted.	
0.6	20 Feb. 2009	NYSE Technologies	4.5 Price Input table updated: "Cannot trade through BBO" error message removed, correction for "Parameter do not allow this function". Price Input: PriceInputType / Price / External parameters combinations table updated. Conditions for "No shares available", "No shares available (IOC)", "No shares available (FOK)", "No shares available (minQty)" error texts updated in 2.1 New Order Request (D) and 2.2 Cancel Order Request (F).	
0.7	12 May 2009	NYSE Technologies	Added an associated code (see "Error Code" column in tables) to each error listed in this document. Some rewording.	Delivered
0.8	1 Jun. 2009	NYSE Technologies	Removed *"field (ExternalTime) invalid"* error from 3 Superficial Checks on Requests, as this ExternalTime field is never checked.	Draft
0.9	7 Sep. 2009	NYSE Technologies	Added new error messages dedicated to UTP-W 20500 "LP Quotes crossed" 20501 "LP Quotes not allowed on this instrument" 20502 "Quote Side conflicting w/ Inst. Situation" 20503 "Instrument Situation unchanged" 20504 "LP Quotes must be cancelled first" 20505 "Firm is not a Liquidity Provider" 20506 "Bad trigger price versus Quote Price" 20507 "MinQty Forbidden" Change disclaimer. Change Logo and label. Change name of the document to "UTP for Cash Markets". Added SmartPool specific error messages.	Draft
1.0	11 Dec. 2009	NYSE Technologies	Added SmartPool codes Status set to deliverable.	Deliverable

The registered office of NYSE Technologies SAS is
39 rue Cambon 75039 Paris Cedex 01 France
Telephone: +33 (0)1 49 27 10 00 Fax: +33 (0)1 49 27 56 59

Siège social : 39 rue Cambon 75039 Paris
SAS au capital de 2 869 962 €
RCS Paris 425 100 294 - N°TVA : FR 87 425 100 294

NYSE Euronext is part of the NYSE Euronext Group.
www.nyseeuronext.com

Version	Date	Author	Description	Status
1.01	16 Dec. 2009	NYSE Technologies	Added 21242 & 21243 (Lit do dark routing upon order entry).	Deliverable
1.2	12 Jan. 10	NYSE Technologies	Added UTP-W Error message dedicated to Bulk Quote : 20508 "Invalid quote size" 20509 "Maximum quote size overtaken" 20510 "Invalid number of quotes" 20501 message changed from "LP Quotes not allowed on this instrument" to "Invalid Class"	Deliverable
1.3	9 Feb. 10	NYSE Technologies	20389, 20377, 20380, 20405, 20217, 20379, 20357, 20360, 20141, 20396, 20351, 20343, 20354, 19010, 19011, 11009, 20512 and 30000 error code are listed in Message 7	Deliverable
1.4	11 Mar. 10	NYSE Technologies	Correction: the error code which corresponds to the label "*Dated/timed only on GTD/GTT orders*" is 20149 instead of 20144. Add a new message in Superficial Checks on Request dedicated to CCG disconnection : 1105 "Field CancelGatewayDcnx invalid" Update the matrix in chapter 3 "Superficial Checks on Requests".	
1.5	25 May 10	NYSE Technologies	Add error message 20443 : Passive Order cannot cross opposite side Add error message 22048 : Cannot Cross outside static collar Add error message 22059 : Cannot trade outside static collar Add a new error message 20513 : Invalid Offer quote price in Offer Only situation Modify the disclaimer Reorganization of the different chapters	

TABLE OF CONTENTS

1 PURPOSE

This document provides the list of errors that can be issued from the UTP Trading Engine upon reception of request messages. Indeed, when the Trading Engine receives a request, it performs several checks on this entering request message, and may reject it. When it does, the error is identified in the response message sent back by the Trading Engine by an error code to which is also associated a text and a reason number.

More precisely, the entering request messages checked by the Trading Engine, the associated response messages sent in case of error, and the fields of this response message where to find the error Code, error Text and Reason number identifying the error, are the following ones (see also *UTP for Cash Markets - UTP-X Message Specifications* for the description of these UTP Trading Engine messages and their fields):

- Errors on New order request (message D): Execution Report message (message 8) sent back by the Trading Engine, with the error identified in the `ErrorCode`, `Text` and `OrdRejReason` fields.
- Errors on Cancel/Replace request (message G) or Cancel order request (message F): Cancel Reject message (message 9) sent back by the Trading Engine, with the error identified in the `ErrorCode`, `Text` and `CxlRejReason` fields.
- Errors on Price Input request (message I): Generic Response message (message y) sent back by the Trading Engine, with the error identified in the `ErrorCode` and `Text` fields.
- Errors on Bulk Quote (message S) related in Bulk Quote Acknowledgment (message 7) or Cancel Reject (message 9).
- Errors on Instrument situation changes which cannot be honored which are related in One Side Only Period Acknowledgment (message 6)

For each of these errors listed, the conditions leading to it are described in the following sections.

2 ERRORS ON CUSTOMER REQUESTS

2.1 NEW ORDER REQUEST (D)

All the error message codes described here above are carried by the Execution Report Message (8) in response to a mistaken New Order Request (D)

Error Code	Reason	Text	Conditions
20203	1	*Invalid OnBehalfOfCompID*	Firm [`OnBehalfOfCompID` field] is not empty and refers to a non-existent firm.
20217	1	*Invalid Symbol*	Instrument [`Symbol` field] is not empty and refers to a non-existent instrument.
20397	1	*System full*	The in-memory storage (containing all orders, trades, etc. of the trading day) is full.
20351	1	*Order entry disabled for this class*	Attempt to enter an order on an instrument attached to a class whose [class] order rejection flag is set.
20353	1	*Order entry disabled for this instrument*	Attempt to enter an order on an instrument whose [instrument] order rejection flag is set.
20354	1	*Order entry disabled for this phase*	Attempt to enter an order during a market phase implemented with [phase] order rejection flag set.
20352	1	*Order entry disabled for this firm*	Attempt to enter an order by a Firm whose [firm] order rejection flag set.
11009	1	*Class not authorized for this Firm*	Attempt to enter an order on an instrument by a firm not authorized for the class the instrument is attached to.
19010	1	*Class not authorized for this Firm*	Attempt to enter a buy order on an instrument by a firm not authorized to buy for the class the instrument is attached to.
19011	1	*Class not authorized for this Firm*	Attempt to enter a sell order on an instrument by a firm not authorized to sell for the class the instrument is attached to.
20360	3	*Order Qty invalid against Min/Max Qty*	Order quantity [`OrderQty` field] greater than or equal to **100,000,000**.
20357	3	*Order size should be multiple of LotSize*	Order quantity [`OrderQty` field] not multiple of the instrument lot size.
20211	3	*Invalid PegDifference, must be = 0*	Attempt to enter a primary peg order with a non null peg difference: `OrderType` = **P**, `ExecInst` = **R**, `PegDifference` ≠ **0**.
20196	3	*Invalid ExecInst on peg*	Attempt to enter a peg order whose execution instruction is not primary peg: `OrderType` = **P**, `PegDifference` = **0**, `ExecInst` ≠ **R**.
20327	3	*No price on market*	Attempt to enter a priceless order (Market or Market to limit) with price field populated.
20379	3	*Price needed*	Attempt to enter a priced (limit or stop limit) order without price, e.g. `OrderType` = **2** or **4**, `Price` empty.
20340	3	*No stop price on stop*	Attempt to enter a stop (limit or loss) order without trigger price, e.g. `OrderType` = **3** or **4**, `StopPx` empty.
20393	3	*Stop price only on stop order*	Attempt to enter a non stop order but specifying a trigger price, e.g. `OrderType` ≠ **3** and **4**, `StopPx` not empty.
20241	3	*MinQty should be multiple of Lot Size*	Minimum quantity [`MinQty` field] not multiple of the instrument lot size.
20239	3	*MinQty greater than Order Qty*	Minimum quantity [`MinQty` field] cannot be greater than order quantity [`OrderQty` field].
20229	3	*MaxFloor greater than Order Qty*	Disclosed quantity [`MaxFloor` field] cannot be greater than order quantity [`OrderQty` field].
20233	3	*MaxFloor under LotSize*	Disclosed quantity [`MaxFloor` field] cannot be lower than the instrument lot size.
20232	3	*MaxFloor should be multiple of x Lot Size*	Disclosed quantity [`MaxFloor` field] must be a multiple of the instrument lot size.
20293	1	*No ExecInst on MOC/LOC*	Attempt to enter an market/limit on close order specifying an invalid execution instruction, e.g. `OrderType` = **1** or **2** or **K**, `TimeInForce` = **7**, but `ExecInst` = **R** or **X** or **m** or **o**.
20195	1	*Invalid ExecInst on order*	Attempt to enter an order with an incompatible execution instruction, e.g.: ▪ `ExecInst` = **R** (primary peg) for a non-pegged order [`OrderType` ≠ **P**], or ▪ `ExecInst` = **m** (Market to limit) or **o** (Market on opening) for a different order type [`OrderType` ≠ **K** and **P**[1]], or ▪ `ExecInst` = **X** or **Y** for a non-cross order [`Side` ≠ **8**].
20192	3	*Invalid ExecInst on cross*	Attempt to enter a cross order with an incompatible execution instruction, e.g. `Side` = **8** and `ExecInst` ≠ **X**.

[1] The case `OrderType` = **P**, `ExecInst` = **m** or **o** is covered by the (20196) *Invalid ExecInst on peg* control/message.

Error Code	Reason	Text	Conditions
20193	1	*Invalid ExecInst on Mkt to Limit order*	Attempt to enter a Market to limit order through a legacy access point (CAP) with an incompatible execution instruction, e.g.: `OrderType` = **K**, `OriginMsg` = **N**, `ExecInst` ≠ **m** and **o**
20292	1	*No ExecInst on Mkt to Limit order*	Attempt to enter a Market to limit order through a CCG with an incompatible execution instruction, e.g.: `OrderType` = **K**, `OriginMsg` = **O**, `ExecInst` = **m** or **o**
20417	1	*Type of price invalid for this phase*	Attempt to enter through a legacy access point (CAP) an order whose execution instruction is not compatible with the current phase type: ▪ Either a market to limit order in call phase type, e.g. `OrderType` = **K**, `OriginMsg` = **N**, and `ExecInst` = **m** ▪ Or a market on opening order in continuous phase type, e.g. `OrderType` = **K**, `OriginMsg` = **N**, and `ExecInst` = **o**
20262	3	*Must be Market Maker*	Attempt to enter a market maker order (`Rule80A` = **6**) on an instrument when no market making link exists between the issuing firm [`OnBehalfOfCompID` field] and the instrument concerned [`Symbol` field].
20104	1	*Bad ExpireTimeFlag, not a GTD*	(see Forbidden Validity Field Combinations section)
20105	1	*Bad ExpireTimeFlag, not a GTT*	
20173	1	*GTD needs ExpireTime*	
20177	1	*GTT needs ExpireTime*	
20149	1	*Dated/timed only on GTD/GTT orders*	
20174	1	*GTD needs ExpireTimeFlag D or T*	
20176	1	*GTT : invalid ExpireTime*	
20171	1	*GTD : invalid ExpireTime*	
20178	1	*GTT time must be > current time*	
20377	3	*Price fraction too long*	The number of digits in the decimal part of price is greater than the decimal count defined at the instrument level (`iPxScale`)
20389	3	*Price too large*	The number of digits in the integer part of price is too big as compared to the decimal count defined at the instrument level (e.g. greater than `9 – iPxScale`). For example, if the decimal count is 3, the integer part of the price cannot exceed 6 digits (999999.999).
20380	3	*Price invalid for tick*	The price is not multiple of the price tick.
20405	3	*No tick found*	There is no defined price tick for the instrument.
20377	3	*StopPx: Price fraction too long*	The number of digits in the decimal part of the trigger price of a stop order is greater than the decimal count defined at the instrument level (`iPxScale`)
20389	3	*StopPx: Price too large*	The number of digits in the integer part of the trigger price of a stop order is too big as compared to the decimal count defined at the instrument level (e.g. greater than `9 – iPxScale`). For example, if the decimal count is 3, the integer part of the trigger price cannot exceed 6 digits (999999.999).
20380	3	*StopPx: Price invalid for tick*	The trigger price of a stop order is not multiple of the price tick.
20109	3	*Bad trigger price versus LTP/ACP*	Attempt to enter a stop order whose trigger price is not better than the last traded price (or adjusted closing price if the instrument has not traded yet), e.g. `OrderType` = **3** or **4**, `StopPx` lower than (`Side` = **1**) / greater than (`Side` = **2**) or equal to the current LTP/ACP
20378	3	*Price must be better than Trigger price*	Attempt to enter a stop limit order whose limit price is not better than its trigger price, e.g. `OrderType` = **4**, `Price` strictly lower than (`Side` = **1**) / greater than (`Side` = **2**) `StopPx`
20153	3	*Discretion order forbidden*	The discretion offset [`DiscretionOffset` field] must always be null (**0** or empty)
20141	1	*Class closed*	Attempt to enter an order on an instrument attached to a class in closed state.
20396	4	*Symbol closed*	Attempt to enter an order on an instrument in closed state.
20399	3	*TAL phase – invalid price*	Attempt to enter a priced order [`OrderType` = **2**] during a Trading At Last phase at a price that differs from the last traded price, or to enter a priceless order [`OrderType` = **1** or **K**].
20401	3	*TAL phase – no Stop order*	Attempt to enter a stop order [`OrderType` = **3** or **4**] during a Trading At Last phase.
20400	3	*TAL phase – no Peg order*	Attempt to enter a peg order [`OrderType` = **P**] during a Trading At Last phase.
20428	3	*Rejected for collar*	Order rejected for trade execution attempt outside collars (see *UTP for Equities - Detailed Functional Specification* for collars rejection cases).
20337	3	*No shares available*	Attempt to enter an Immediate-Or-Cancel or Fill-Or-Kill order, or an order with a non-null minimum quantity, when the book conditions does not allow satisfying that condition because there is no matchable quantity at all on the contra side. Examples: ▪ Entering an IOC or FOK order, or an order with a minimum quantity condition, when the contra side of the book is empty; ▪ Entering an IOC or FOK order, or an order with a minimum quantity condition, when order(s) at the contra side have price(s) inside collars but do not cross the entering order's price.

Error Code	Reason	Text	Conditions
20335	3	*No shares available (IOC)*	Immediate-Or-Cancel order entry rejected because contra-side orders have price(s) outside collars.
20334	3	*No shares available (FOK)*	Fill-Or-Kill order entry rejected because there is not enough contra-side quantity that can match inside collars. Examples: ▪ Entry of a FOK order having a quantity Q (>1) with, at the other side, a quantity 1 whose price is inside collars and crosses the entering order's price, and a quantity Q-1 whose price is inside collars but does not cross the entering order's price. ▪ Entry of a FOK order having a quantity Q with, at the other side, a quantity Q whose price is crossing the entering order's price, but outside collars.
20336	3	*No shares available (MinQty)*	Order entry with a minimum quantity condition rejected because there is not enough contra-side quantity that can match inside collars. Examples: ▪ Entry of an order having a minimum quantity of Q (>1) with, at the other side, a quantity 1 whose price is inside collars and crosses the entering order's price, a quantity Q-1 whose price is inside collars but does not cross the entering order's price. ▪ Entry of an order having a minimum quantity of Q with, at the other side, a quantity Q whose price crosses the entering order's price, but outside collars.
20135	1	*Cannot cross through BBO*	A cross order must be in the best limits range and collars range intersection. ▪ If the order price is outside BBO (either within or outside collars), the error message is *Cannot cross through BBO*. ▪ If the order price is inside BBO but outside collars, the error message is *Cannot cross outside collar*.
20134	1	*Cannot cross outside collar*	
20250	3	*Mkt2Lim : Cannot calculate price*	Attempt to enter a Market to Limit order (`OrderType` = **K**) in continuous phase with no counterpart order.
20225	3	*Max amount reached - need confirmation*	Attempt to enter an unconfirmed order on an equity whose amount (`Price` times `OrderQty`) is greater than the maximum amount for unconfirmed orders on equities defined at the class level. The order may be entered again with its confirmation indicator set.
20226	3	*Max qty reached - need confirmation*	Attempt to enter an unconfirmed order on a bond whose quantity (`OrderQty`) is greater than the maximum quantity for unconfirmed orders on bonds defined at the class level. The order may be entered again with its confirmation indicator set.
20404	1	*technical error*	▪ Either attempt to enter an order from a legacy access point (**CAP**: `OriginMsg`=**N**) with an existing `OrderID` (software malfunction). ▪ Or other technical problem.
20273	1	*No ClOrdID*	Attempt to enter an order with an empty `ClOrdID` mandatory identifier.
20142	3	*ClOrdID already exists*	Attempt to enter an order with the `ClOrdID` identifier of an order already existing in the central book.
20143	3	*ClOrdID insert internal error*	Internal error (e.g. `ClOrdID` table full)
20296	1	*No IOC/FOK in call mode*	(see Order Creation section)
20236	3	*Minimum quantity forbidden in call mode*	
20208	1	*Invalid OrdType/TimeInForce combination*	
20200	1	*Invalid mkt phase for Cross*	
20201	3	*Invalid mkt phase for Peg*	
20237	3	*MinQty forbidden for this OrdType*	
20227	3	*MaxFloor forbidden for this OrdType*	
20238	3	*MinQty forbidden for this TimeInForce*	
20228	3	*MaxFloor forbidden for this TimeInForce*	
20307	3	*No NBBO for peg*	Attempt to enter a peg order (`OrderType` = **P**, `ExecInst` = **R**) when there is no limit of its side.
20420	1	*Unsupported OrdType for Cross*	Attempt to enter a cross order (`Side` = **8**) with `OrderType` other than **X**.
20146	1	*Cross needs Account type (Rule80A_2)*	Attempt to enter a cross order without valuating the account type of the sell side (`Rule80A_2` field). Note: if the account type of the buy side (`Rule80A` field) is not valuated, the rejection message is *field (Rule80A) invalid* (superficial control).
20429	1	*Cancelled for LACP change*	Raised for non triggered stop orders cancelled upon Adjusted Closing Price change by Market Operations
20430	1	*Cancelled for LTP change*	Raised for non triggered stop orders cancelled upon Last Traded Price change by Market Operations
20506	3	*Bad trigger price versus Quote Price*	The buy (sell) stop on quote triggering price is lower than or equal to the LP Offer (greater than or equal to the LP Bid)
20507	3	*MinQty forbidden*	Order with Minimum Quantity are forbidden on Warrant New Market Model
21234	0	*Invalid mkt phase for this TIF*	(SmartPool) The specified Time In Force attribute is not authorized in the current market phase
21235	0	*Midpoint peg must have Peg type*	(SmartPool) Order with an execution instruction of Mid-point peg must also have an order type Peg
21236	0	*Type forbidden during this market phase*	(SmartPool) The specified order type cannot be entered during the current market phase
21237	0	*SmartPool : Maxfloor forbidden*	(SmartPool) Maxfloor (disclosed quantity) not authorized on this platform

Error Code	Reason	Text	Conditions
21239	0	*Invalid firm type (GAZER BOX expected)*	Invalid firm type (GatherBox expected)
21240	0	*Min quantity must be >= MIQ*	(SmartPool) When specified, the order minimum quantity must be greater than or equal to the Minimum Instrument Quantity
21506	0	*SmartPool : MaxFloor forbidden*	(SmartPool) Disclosed quantity management not available in SmartPool
19012	0	*Firm not authorized for instr./acc. Type*	(SmartPool) The specified Firm has no authorization for this instrument (cleared by EuroCCP) and this account type
21242	0	*Mid Point pricetoo high*	(Lit to Dark routing) Dark book conditions have changed at the time a routable buy order is routed to SmartPool and the dark matching price is no more better than the lit one.
21243	0	*Mid Point pricetoo low*	(Lit to Dark routing) Dark book conditions have changed at the time a routable sell order is routed to SmartPool and the dark matching price is no more better than the lit one.
20443	1	*Passive Order cannot cross opposite side*	(NAE) When a member tries to sent a passive order which can be executed upon order entry then the order is rejected
22048	1	*Cannot cross outside static collars*	The entered price of the cross is not within static collars

2.2 CANCEL ORDER REQUEST (F)

All the error message codes described here above are carried by the Cancel Reject Message (9) in response to a mistaken Cancel Order Request (F)

Error Code	Reason	Text	Conditions
20397	1	*System full*	The in-memory storage (containing all orders, trades, etc. of the trading day) is full.
20141	0	*Class closed*	Attempt to cancel (an) order(s) on an instrument attached to a class in closed state.
20396	0	*Symbol closed*	Attempt to cancel/replace an order on an instrument in closed state.
20217	1	*Invalid Symbol*	Instrument [`Symbol` field] is not empty and refers to a non-existent instrument.
20247	1	*Missing Symbol*	Instrument [`Symbol` field] is empty.
20203	1	*Invalid OnBehalfOfCOmpID*	Firm [`OnBehalfOfCompID` field] is not empty and refers to a non-existent firm.
20245	1	*Missing OnBehalfOfCompID*	Firm [`OnBehalfOfCompID` field] is empty.
20344	0	*Not same OnBehalfOfCompID (F115)*	Firm [`OnBehalfOfCompID` field] refers to a different firm.
20244	1	*Missing ClOrdID*	Order identifier [`ClOrdID` field] is empty.
20362	0	*OrigClOrdID not found*	(see Cancel/Replace & Cancellation section)
20246	1	*Missing OrigClOrdID*	
20358	0	*OrderID (HON) not found*	
20351	1	*Order entry disabled for this class*	Attempt to cancel an order on an instrument attached to a class whose [class] order rejection flag is set.
20353	1	*Order entry disabled for this instrument*	Attempt to cancel an order on an instrument whose [instrument] order rejection flag is set.
20354	1	*Order entry disabled for this phase*	Attempt to cancel an order during a market phase implemented with [phase] order rejection flag set.
20411	0	*Too late to cancel*	When order cancellation is requested for an order filled or already cancelled.
20185	1	*Invalid Account on bulk cancel*	Account [`Symbol` field] is not empty and refers to a non-existent account.
20217	1	*Invalid Symbol*	(see Order Creation section or the conditions for each of these messages)
20186	1	*Invalid class*	
20203	1	*Invalid OnBehalfOfCOmpID*	
20202	1	*Invalid OnBehalfOfCOmpID on bulk cancel*	
20243	1	*Missing Class/Instrument on bulk cancel*	
21238	0	*Cancel/Replace invalid on routed order*	Modifying a routed (IOC) order is not possible

Error Code	Reason	Text	Conditions
21240	0	*Cannot change MinQty*	Modifying the minimum quantity of a partially matched order is forbidden

2.3 CANCEL/REPLACE ORDER REQUEST (G)

All the error message codes described here above are carried by the Cancel Reject Message (9) in response to a mistaken Cancel/Replace Order Request (G)

Error Code	Reason	Text	Conditions
20397	1	*System full*	The in-memory storage (containing all orders, trades, etc. of the trading day) is full.
20203	1	*Invalid OnBehalfOfCompID*	Firm [`OnBehalfOfCompID` field] is not empty and refers to a non-existent firm.
20344	0	*Not same OnBehalfOfCompID (F115)*	`OnBehalfOfCompID` = *<Not same firm>*
20217	1	*Invalid Symbol*	Instrument [`Symbol` field] is not empty and refers to a non-existent instrument.
20141	0	*Class closed*	Attempt to cancel/replace an order on an instrument attached to a class in closed state.
20396	0	*Symbol closed*	Attempt to cancel/replace an order on an instrument in closed state.
20362	0	*OrigClOrdID not found*	(see Cancel/Replace & Cancellation section)
20246	1	*Missing OrigClOrdID*	
20358	0	*OrderID (HON) not found*	
20359	0	*OrderID not found*	`OrderID` not found on a confirmation after rejection for collars (in this case, `OrderID` and `OrigClOrdID` are required).
11009	1	*Class not authorized for this Firm*	Attempt to cancel an order on an instrument by a firm not authorized for the class the instrument is attached to.
19010	1	*Class not authorized for this Firm*	Attempt to cancel a buy order on an instrument by a firm not authorized to buy for the class the instrument is attached to.
19011	1	*Class not authorized for this Firm*	Attempt to cancel a sell order on an instrument by a firm not authorized to sell for the class the instrument is attached to.
20412	0	*Too late to confirm*	Attempt to confirm an order rejected for collars beyond the authorized confirmation delay.
20413	0	*Too late to modify*	Attempt to replace a filled or cancelled order.
20132	1	*Cannot change side*	Attempt to replace an order with a different side.
20262	1	*Must be Market Maker*	Attempt to replace a market maker order (`Rule80A` = **6**) on an instrument when no market making link exists between the issuing firm [`OnBehalfOfCompID` field] and the instrument concerned [`Symbol` field].
20417	1	*Type of price invalid for this phase*	Attempt to replace through a legacy access point (CAP) an order whose execution instruction is not compatible with the current phase type: ▪ Either a market to limit order in call phase type, e.g. `OrderType` = **K**, `OriginMsg` = **N**, and `ExecInst` = **m** ▪ Or a market on opening order in continuous phase type, e.g. `OrderType` = **K**, `OriginMsg` = **N**, and `ExecInst` = **o**
20195	1	*Invalid ExecInst on order*	Attempt to replace an order with an incompatible execution instruction, e.g.: ▪ `ExecInst` = **R** (primary peg) for a non-pegged order [`OrderType` ≠ **P**], or ▪ `ExecInst` = **m** (Market to limit) or **o** (Market on opening) for a different order type [`OrderType` ≠ **K** and **P**[2]], or ▪ `ExecInst` = **X** or **Y** for a non-cross order [`Side` ≠ **8**].
20196	1	*Invalid ExecInst on peg*	Attempt to replace a peg order with an execution instruction other than primary peg: `OrderType` = **P**, `PegDifference` = **0**, `ExecInst` ≠ **R**.
20211	1	*Invalid PegDifference, must be = 0*	Attempt to replace a primary peg order with a non null peg difference: `OrderType` = **P**, `ExecInst` = **R**, `PegDifference` ≠ **0**.
20193	1	*Invalid ExecInst on Mkt to Limit order*	Attempt to replace a Market to limit order through a legacy access point (CAP) with an incompatible execution instruction, e.g.: `OrderType` = **K**, `OriginMsg` = **N**, `ExecInst` ≠ **m** and **o**
20131	1	*Cannot Change Peg OrdType*	Attempt to replace a peg order with a non-peg order or vice-versa.
20104	1	*Bad ExpireTimeFlag, not a GTD*	(see Cancel/Replace & Cancellation section for the conditions for each of these messages)

[2] The case `OrderType` = **P**, `ExecInst` = **m** or **o** is covered by the (20196) *Invalid ExecInst on peg* control/message.

Error Code	Reason	Text	Conditions
20105	1	*Bad ExpireTimeFlag, not a GTT*	
20173	1	*GTD needs ExpireTime*	
20177	1	*GTT needs ExpireTime*	
20149	1	*Dated/timed only on GTD/GTT orders*	
20174	1	*GTD needs ExpireTimeFlag D or T*	
20176	1	*GTT : invalid ExpireTime*	
20171	1	*GTD : invalid ExpireTime*	
20178	1	*GTT time must be > current time*	
20312	1	*No order size*	Attempt to replace an order created through a CCG without specifying the quantity. Note: this does not apply if replacing an order through a legacy access point (CAP) as the quantity is optional in that case (quantity unchanged if not specified).
20357	1	*Order size should be multiple of Lot Size*	Order quantity [`OrderQty` field] not multiple of the instrument lot size.
20268	0	*New size already filled*	Attempt to replace an order with a new quantity smaller than the already filled quantity.
20379	1	*Price needed*	Attempt to replace a priced (limit or stop limit) order without price, e.g. `OrderType` = **2** or **4**, `Price` empty.
20327	1	*No price on market*	Attempt to replace a priceless order (Market or Market to limit) with price field populated.
20377	3	*Price fraction too long*	The number of digits in the decimal part of price is greater than the decimal count defined at the instrument level (`iPxScale`)
20388	3	*Price too large*	The number of digits in the integer part of price is too big as compared to the decimal count defined at the instrument level (e.g. greater than `9 – iPxScale`). For example, if the decimal count is 3, the integer part of the price cannot exceed 6 digits (999999.999).
20389	3		
20380	3	*Price invalid for tick*	The price is not multiple of the price tick.
20405	3	*No tick found*	There is no defined price tick for the instrument.
20377	3	*StopPx: Price fraction too long*	The number of digits in the decimal part of the trigger price of a stop order is greater than the decimal count defined at the instrument level (`iPxScale`)
20389	3	*StopPx: Price too large*	The number of digits in the integer part of the trigger price of a stop order is too big as compared to the decimal count defined at the instrument level (e.g. greater than `9 – iPxScale`). For example, if the decimal count is 3, the integer part of the trigger price cannot exceed 6 digits (999999.999).
20380	3	*StopPx: Price invalid for tick*	The trigger price of a stop order is not multiple of the price tick.
20168	3	*StopPx: No tick found*	No defined tick for the instrument.
20340	3	*No stop price on stop*	Attempt to replace a stop (limit or loss) order without trigger price, e.g. `OrderType` = **3** or **4**, `StopPx` empty.
20393	3	*Stop price only on stop order*	Attempt to replace a non stop order but specifying a trigger price, e.g. `OrderType` ≠ **3** and **4**, `StopPx` not empty.
20109	3	*Bad trigger price versus LTP/ACP*	Attempt to replace a stop order with a trigger price not better than the last traded price (or adjusted closing price if the instrument has not traded yet), e.g. `OrderType` = **3** or **4**, `StopPx` lower than (`Side` = **1**) / greater than (`Side` = **2**) or equal to the current LTP/ACP
20378	3	*Price must be better than Trigger price*	Attempt to replace a stop limit order with a limit price not better than its trigger price, e.g. `OrderType` = **4**, `Price` strictly lower than (`Side` = **1**) / greater than (`Side` = **2**) `StopPx`
20117	1	*Can't change to Discretion*	Attempt to replace a non-discretion order with a discretion order (`DiscretionOffset` field not null)
20250	3	*Mkt2Lim : Cannot calculate price*	Attempt to replace a Market to Limit order (`OrderType` = **K**) in continuous phase with no counterpart order.
20229	1	*MaxFloor greater than OrderQty*	Disclosed quantity [`MaxFloor` field] cannot be greater than order quantity [`OrderQty` field].
20233	1	*MaxFloor under Lot Size*	Disclosed quantity [`MaxFloor` field] cannot be lower than the instrument lot size.
20232	1	*MaxFloor should be multiple of x Lot Size*	Disclosed quantity [`MaxFloor` field] must be a multiple of the instrument lot size.
20116	1	*Can't change Side*	Attempt to replace an order with a different side.
20239	1	*MinQty greater than OrderQty*	Minimum quantity [`MinQty` field] cannot be greater than order quantity [`OrderQty` field].
20144	1	*Confirmed price should be the same*	An order rejected for collars is confirmed with a different price.
20145	1	*Confirmed quantity should be the same*	An order rejected for collars is confirmed with a different quantity from the one for which the order has been rejected (remaining quantity when rejected). Note: for an order whose origin is NSC, the confirmation must be done with the "remaining quantity value" in the Order quantity Field. For other orders, the confirmation must be done with the "original total order quantity value" in the Order quantity Field.
20225	1	*Max amount reached - need confirmation*	Attempt to enter an unconfirmed order on an equity whose amount (`Price` times `OrderQty`) is greater than the maximum amount for unconfirmed orders on equities defined at the class level. The order may be entered again with its confirmation indicator set.
20226	1	*Max qty reached - need confirmation*	Attempt to enter an unconfirmed order on a bond whose quantity (`OrderQty`) is greater than the maximum quantity for unconfirmed orders on bonds defined at the class level. The order may be entered again with its confirmation indicator set.

Error Code	Reason	Text	Conditions
20273	1	*No ClOrdID*	Attempt to replace an order with an empty `ClOrdID` mandatory identifier.
20142	1	*ClOrdID already exists*	Attempt to replace an order with the `ClOrdID` identifier of an order already existing in the central book.
20143	1	*ClOrdID insert internal error*	Internal error (e.g. `ClOrdID` table full)
20404	1	*Technical error*	▪ Either attempt to replace an order from a legacy access point (**CAP**: `OriginMsg`=**N**) with an existing `OrderID` (software malfunction). ▪ Or other technical problem.
20360	3	*OrderQty invalid against Min/Max Qty*	Order quantity [`OrderQty` field] greater than or equal to **100,000,000**.
20343	0	*Not same ClOrdID (F11-F41)*	C/R request with `OriginMsg` = **0**, `OrderID` value corresponding to an existing `OrderID` but `OrigClOrdID` value not corresponding to any existing `ClOrdID`.
20351	1	*Order entry disabled for this class*	Attempt to cancel an order on an instrument attached to a class whose [class] order rejection flag is set.
20354	1	*Order entry disabled for this phase*	Attempt to cancel an order during a market phase implemented with [phase] order rejection flag set.
20353	1	*Order entry disabled for this instrument*	Attempt to cancel an order on an instrument whose [instrument] order rejection flag is set.
20352	1	*Order entry disabled for this firm*	Attempt to cancel an order by a Firm whose [firm] order rejection flag set.
20428	3	*Rejected for collar*	Order rejected for trade execution attempt outside collars (see *UTP for Equities - Detailed Functional Specification* for collars rejection cases).
20337	3	*No shares available*	Attempt to enter an Immediate-Or-Cancel or Fill-Or-Kill order, or an order with a non-null minimum quantity, when the book conditions does not allow satisfying that condition because there is no matchable quantity at all on the contra side. Examples: ▪ Entering an IOC or FOK order, or an order with a minimum quantity condition, when the contra side of the book is empty; ▪ Entering an IOC or FOK order, or an order with a minimum quantity condition, when order(s) at the contra side have price(s) inside collars but do not cross the entering order's price.
20335	3	*No shares available (IOC)*	Immediate-Or-Cancel order entry rejected because contra-side orders have price(s) outside collars.
20334	3	*No shares available (FOK)*	Fill-Or-Kill order entry rejected because there is not enough contra-side quantity that can match inside collars. Examples: ▪ Entry of a FOK order having a quantity Q (>1) with, at the other side, a quantity **1** whose price is inside collars and crosses the entering order's price, and a quantity Q-1 whose price is inside collars but does not cross the entering order's price. ▪ Entry of a FOK order having a quantity Q with, at the other side, a quantity Q whose price is crossing the entering order's price, but outside collars.
20399	3	*TAL phase – invalid price*	Attempt to replace an order with a price during a Trading At Last phase that differs from the last traded price.
20401	3	*TAL phase – no Stop order*	Attempt to replace a stop order [`OrderType` = **3** or **4**] during a Trading At Last phase.
20296	1	*No IOC/FOK in call mode*	(see Order Creation section for the conditions for each of these messages)
	1	*Field (MinQty) invalid*	
20228	3	*MaxFloor forbidden for this TimeInForce*	
20227	3	*MaxFloor forbidden for this OrdType*	
20208	1	*Invalid OrdType/TimeInForce combination*	
20201	3	*Invalid Mkt phase for peg*	
20132	1	***Cannot change side***	
20506	1	*Bad trigger price versus Quote Price*	The buy (sell) stop on quote triggering price is lower than or equal to the LP Offer (greater than or equal to the LP Bid)
21235	0	*Midpoint peg must have Peg type*	(SmartPool) Order with an execution instruction of Mid-point peg must also have an order type Peg
21236	0	*Type forbidden during this market phase*	(SmartPool) The specified order type cannot be entered during the current market phase
21237	0	*Maxfloor forbidden on this market*	(SmartPool) Maxfloor (disclosed quantity) not authorized on this platform
21240	0	*Min quantity must be >= MIQ*	(SmartPool) When specified, the order minimum quantity must be greater than or equal to the Minimum Instrument Quantity
19012	0	*Firm not authorized for instr./acc. Type*	(SmartPool) The specified Firm has no authorization for this instrument (cleared by EuroCCP) and this account type
20443	3	*Passive Order cannot cross opposite side*	(NAE) When a member tries to modify an order in a passive order which can be executed upon order modification then the order modification is rejected

2.4 PRICE INPUT REQUEST (I)

All the error message codes described here above are carried by the message Generic Response Message (y) in response to a mistaken Price Input Request (I)

Error Code	Comments	Text	Conditions
20217		*Invalid Symbol*	Instrument [`Symbol` field] is not empty and refers to a non-existent instrument.
20431		*Invalid OnBehalfOfCompID*	Firm [`OnBehalfOfCompID` field] is empty or refers to a non-existent firm.
11009		*Class not authorized for this Firm*	Attempt to enter a price on an instrument by a firm not authorized for the class the instrument is attached to.
19010		*Class not authorized for this Firm*	Attempt to enter a price on an instrument by a firm not authorized to buy for the class the instrument is attached to.
19011		*Class not authorized for this Firm*	Attempt to enter a price on an instrument by a firm not authorized to sell for the class the instrument is attached to.
20436		*Invalid PriceType*	Price type empty or invalid.
20213	Price validity checks and associated errors (see also Price Input section) ->	*Invalid Price*	`Price` field is valuated while it should not, or vice-versa.
20377		*Price fraction too long*	The number of digits in the decimal part of price is greater than the decimal count defined at the instrument level (`iPxScale`)
20389		*Price too large*	The number of digits in the integer part of price is too big as compared to the decimal count defined at the instrument level (e.g. greater than `9 – iPxScale`). For example, if the decimal count is 3, the integer part of the price cannot exceed 6 digits (999999.999).
20380		*Price invalid for tick*	The price is not multiple of the price tick.
20405		*No tick found*	There is no defined price tick for the instrument.
20213		*Invalid Price*	(see Price Input section)
20432		*Parameters does not allow this function*	
20434		*Cannot trade outside collar*	
20433		*Cannot trade through BBO*	
20435		*Cannot create Valuation trade*	
22059		*Cannot trade outside static collar*	When the price indicated in the Price Input Request is outside the static collars

2.5 BULK QUOTE (S)

All the error message codes described here above are carried by the Bulk Quote Ack. message (7) in response to a mistaken Bulk Quote message (S)

Error Code	Comments	Text	Conditions
11009		*Class not authorized for this Firm*	Attempt to cancel an order on an instrument by a firm not authorized for the class the instrument is attached to.
19010		*Class not authorized for this Firm*	Attempt to enter a buy order on an instrument by a firm not authorized to buy for the class the instrument is attached to.
19011		*Class not authorized for this Firm*	Attempt to enter a sell order on an instrument by a firm not authorized to sell for the class the instrument is attached to.
20141		*Class closed*	Attempt to enter an order on an instrument attached to a class in closed state.
20217		*Invalid Symbol*	Instrument [`Symbol` field] is not empty and refers to a non-existent instrument.

Error Code	Comments	Text	Conditions
20351		*Order entry disabled for this class*	Attempt to enter an order on an instrument attached to a class whose [class] order rejection flag is set.
20353		*Order entry disabled for this instrument*	Attempt to enter an order on an instrument whose [instrument] order rejection flag is set.
20354		*Order entry disabled for this phase*	Attempt to enter an order during a market phase implemented with [phase] order rejection flag set.
20357		*Order size should be multiple of LotSize*	Order quantity [`OrderQty` field] not multiple of the instrument lot size.
20360		*Order Qty invalid against Min/Max Qty*	Order quantity [`OrderQty` field] greater than or equal to **100,000,000**.
20377		*Price fraction too long*	The number of digits in the decimal part of price is greater than the decimal count defined at the instrument level (`iPxScale`)
20379		*Price needed*	Attempt to enter a priced (limit or stop limit) order without price, e.g. `OrderType` = **2** or **4**, `Price` empty.
20380		*Price invalid for tick*	The price is not multiple of the price tick.
20389		*Price too large*	The number of digits in the integer part of price is too big as compared to the decimal count defined at the instrument level (e.g. greater than `9 – iPxScale`). For example, if the decimal count is 3, the integer part of the price cannot exceed 6 digits (999999.999).
20396		*Symbol closed*	Attempt to enter an order on an instrument in closed state.
20405		*No tick found*	There is no defined price tick for the instrument.
20500		*LP Quotes Crossed*	The LP tries to send a Quote with Bid and Offer crossed.
20501		*Class Invalid*	The LP tries to send a Quote on an instrument which is not supposed to receive Quotes.
20502		*Quote Side conflicting w/ Inst. Situation*	The LP tries to send a Bid (Offer) Quote on an instrument in a Offer (Bid) Only Situation.
20505		*Firm is not a Liquidity Provider*	A Member Firm tries to send a Quote on an instrument not animated by this given Firm.
20512		*Duplicated Symbol*	When an Isin code is duplicated in a Bulk Quote, the whole is rejected.
30000*		*Unable to route*	* When the first Isin code met in a Bulk Quote message is wrong, the CCG is not able to route the B message onto the appropriate Trading Unit. This error code is sent by CCG and carried in the J message and is not issued by message 7.
20513		*Invalid Offer quote price in Offer Only situation*	In Offer Only situation, when the LP tries to make an Offer with a price greater than "minimum tick number"* "instrument tick size".

.

All the error message codes described here above are carried by the Cancel Reject message (9) in response to a mistaken Bulk Quote message (S)

Error Code	Comments	Text	Conditions
20508		*Invalid Quote size*	The LP tries to send a Bulk Quote message whose number of expressed quotes is bigger than 150
20509		*Maximum Quote size overtaken*	The LP tries to send a Bulk Quote message whose "NoQuoteEntries" field is bigger than 150
20510		*Invalid number of quotes*	The LP tries to send a Bulk Quote message whose number of expressed quotes do not match the field "NoQuoteEntries"

2.6 ONE SIDE ONLY PERIOD (P)

All the error message codes described here above are carried by the One Side Only Period Ack. message (6) in response to a mistaken One Side Only Period message (P).

Error Code	Comments	Text	Conditions
20503		*Instrument Situation unchanged*	The LP or the Market Control tries to set an Instrument Situation already existing.
20504		*LP Quotes must be cancelled first*	The LP or the Market Control tries to switch in Bid/Offer Only Situation with a Quote still in the book

3 SUPERFICIAL CHECKS ON REQUESTS

Error Code	Reason	Text	<msgID>
1027	1	*field (MaxFloor) invalid*	D, G
1028	1	*field (OnBehalfOfCompID) invalid*	D, F, G
1029	1	*field (TechOrig) invalid*	D, F, G
1030	1	*field (ICfmOm) invalid*	D, G
1032	1	*field (StopPx) invalid*	D, G
1034	1	*field (Rule80A) invalid*	D, F, G
1035	1	*field (Account) invalid*	D, F, G
1036	1	*field (AccountID) invalid*	D
1037	1	*field (ClientId) invalid*	D, G
1041	1	*field (Rule80A_2) invalid*	D
1042	1	*field (Account_2) invalid*	D
1044	1	*field (ClientId_2) invalid*	D
1055	1	*field (ClearingFirm) invalid*	D, G
1056	1	*field (NSCFillerD) invalid*	D
1056	1	*field (NSCFillerF) invalid*	F
1056	1	*field (NSCFillerG) invalid*	G
1057	1	*field (FreeText) invalid*	D, G
1058	1	*field (OpenClose) invalid*	D, G
1059	1	*field (ClearingHandlingType) invalid*	D, G
1060	1	*field (ClearingFirm_2) invalid*	D
1062	1	*field (FreeText_2) invalid*	D
1063	1	*field (OpenClose_2) invalid*	D
1064	1	*field (ClearingHandlingType_2) invalid*	D
1070	1	*field (SeqNum) invalid*	D, F, G
1071	1	*field (SenderCompID) invalid*	D, F, G
1072	1	*field (TargetCompID) invalid*	D, F, G
1073	1	*field (SenderSubID) invalid*	D, F, G
1074	1	*field (ExecInst) invalid*	D, G
1075	1	*field (DiscretionInst) invalid*	D, G
1076	1	*field (DiscretionOffset) invalid*	D, G
1077	1	*field (PossResend) invalid*	D
1078	1	*field (SettleInstCode) invalid*	D
1079	1	*field (PegDifference) invalid*	D, G
1080	1	*field (TradingSessionID) invalid*	D, G
1081	1	*field (ExpireTimeFlag) invalid*	D, G
1082	1	*field (ProactiveDiscretionIndicator) invalid*	D, G
1083	1	*field (DispRange) invalid*	D, G
1084	1	*field (PartyID) invalid*	D
1085	1	*field (PartySubID) invalid*	D
1086	1	*field (AttributedQuote) invalid*	D
1087	1	*field (ProactiveIfLocked) invalid*	D
1088	1	*field (ExecBroker) invalid*	D
1089	1	*field (ProspectusDelivery) invalid*	D
1090	1	*field (ISO) invalid*	D
1091	1	*field (CorpActInst) invalid*	D, G
1092	1	*field (ExtendedExecInst) invalid*	D
1093	1	*field (ExtendedPNP) invalid*	D
1095	1	*field (OriginMsg) invalid*	D, G
1096	1	*field (DeliverToCompID) invalid*	D, F, G
1097	1	*field (OnBehalfOfLocationID) invalid*	D, F, G
1098	1	*field (OnBehalfOfSubID) invalid*	D, F, G
1099	1	*field (OrderID) invalid*	F, G
1100	1	*field (SymbolMic) invalid*	D, F, G
1101	1	*field(SubscriberID) invalid*	F
1102	1	*Field(NoQuoteSets) invalid*	S
1103	1	*Field(NoticeOfInterest) invalid*	D,G
1104	1	*Field(Routing) invalid*	D
1105	1	*Field (CancelGatewayDcnx) invalid*	D,G,S
1011	1	*field (filler1) invalid*	D, F, G
1011	1	*field (filler2) invalid*	D, F, G
2026	1	*field (ExpireTime) invalid*	D, G
3458	1	*field(ClassShortName) invalid*	F
9015	1	*field (ClOrdID) invalid*	D, F, G
9016	1	*field (Head) invalid*	D, F, G
9018	1	*field (OrigClOrdID) invalid*	G
9019	1	*field (Symbol) invalid*	D, F, G
9020	1	*field (Side) invalid*	D, F, G
9021	1	*field (OrderQty) invalid*	D, F, G
9022	1	*field (OrdType) invalid*	D, F, G
9023	1	*field (Price) invalid*	D, G
9024	1	*field (TimeInForce) invalid*	D, G
9026	1	*field (MinQty) invalid*	D, G

4 ERRORS ON ATTRIBUTE COMBINATIONS

Caution: the SmartPool and the Warrant market model may have combinations rules that slightly differ from those described in this section. Please refer to UTP-W or UTP-SP Detailled Specifications for dedicated order authorizations.

4.1 ORDER CREATION

		Limit		Market To Limit		Market Order		Stop Loss		Stop Limit		Peg		Cross	
		Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type
Immediate Or Cancel (IOC)	Standard qty	(1)		(1)		(1)		(7)	(7)	(7)	(7)	(7)	(7)	(8)	
	Minimum qty	(2)		(2)		(3)	(3)	(7)	(7)	(7)	(7)	(7)	(7)	(3)	(3)
	Disclosed qty	(6)	(6)	(5)	(5)	(5)	(5)	(7)	(7)	(7)	(7)	(7)	(7)	(5)	(5)
Fill Or Kill (FOK)	Standard qty	(1)		(1)		(1)		(7)	(7)	(7)	(7)	(7)	(7)	(8)	
	Minimum qty	(4)	(4)	(4)	(4)	(4)	(4)	(7)	(7)	(7)	(7)	(7)	(7)	(4)	(4)
	Disclosed qty	(6)	(6)	(5)	(5)	(5)	(5)	(7)	(7)	(7)	(7)	(7)	(7)	(5)	(5)
Day, GTD, GTC	Standard qty											(9)		(8)	
	Minimum qty	(2)		(2)		(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)
	Disclosed qty			(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)
Valid For Auction / Closing	Standard qty							(7)	(7)	(7)	(7)	(7)	(7)	(8)	
	Minimum qty	(4)	(4)	(4)	(4)	(3)	(3)	(7)	(7)	(7)	(7)	(7)	(7)	(3)	(3)
	Disclosed qty			(5)	(5)	(5)	(5)	(7)	(7)	(7)	(7)	(7)	(7)	(5)	(5)

(1) No IOC/FOK in call mode
(2) Minimum quantity forbidden in call mode
(4) MinQty forbidden for this TimeInForce
(5) MaxFloor forbidden for this OrdType
(7) Invalid OrdType/TimeInForce combination
(8) Invalid mkt phase for Cross

(3) MinQty forbidden for this OrdType	*(6) MaxFloor forbidden for this TimeInForce*	*(9) Invalid mkt phase for Peg*

4.2 ORDER MODIFICATION

		Limit		Market To Limit		Market		Stop Loss		Stop Limit		Peg		Cross	
		Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type	Call Type	Continuous Type
Immediate Or Cancel (IOC)	Standard qty	(1)		(1)		(1)		(5)	(5)	(5)	(5)	*See **Cannot Change Peg OrdType** order modification rejection message for a modification from a non-peg order to a peg order.* *Else (modification of a Peg order), N/A as Peg orders are not allowed during Call phase type.*	(5)	*Modification to Cross rejected by superficial control (**Field (ExecInst) invalid**)*	
	Minimum qty	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)		(2)		
	Disclosed qty	(4)	(4)	(3)	(3)	(3)	(3)	(5)	(5)	(5)	(5)		(5)		
Fill Or Kill (FOK)	Standard qty	(1)		(1)		(1)		(5)	(5)	(5)	(5)		(5)		
	Minimum qty	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)		(2)		
	Disclosed qty	(4)	(4)	(3)	(3)	(3)	(3)	(5)	(5)	(5)	(5)		(5)		
Day, GTD, GTC	Standard qty														
	Minimum qty	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)		(2)		
	Disclosed qty			(3)	(3)	(3)	(3)	(3)	(3)	(3)	(3)		(3)		
Valid For Auction / Closing	Standard qty							(5)	(5)	(5)	(5)		(5)		
	Minimum qty	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)	(2)		(2)		
	Disclosed qty			(3)	(3)	(3)	(3)	(5)	(5)	(5)	(5)		(5)		

(1) No IOC/FOK in call mode	*(4) MaxFloor forbidden for this TimeInForce*
(2) Field (MinQty) invalid	*(5) Invalid OrdType/TimeInForce combination*
(3) MaxFloor forbidden for this OrdType	

4.3 FORBIDDEN VALIDITY FIELD COMBINATIONS

	TimeInForce value	ExpireTimeFlag value	ExpireTime value	Error message
[TIF / ExpireTimeFlag / ExpireTime] combinations and associated error messages	= 6	= *<empty>*	= *<whatever value>*	*GTD needs ExpireTimeFlag D or T*
		= D	= *<empty>*	*GTD needs ExpireTime*
			= <invalid date according to ExpireTime superficial control>	*Field (ExpireTime) invalid*
			= <invalid date, for example '9999'>	*GTD: invalid ExpireTime*
			= <valid date>	*No error message whatever is the <valid date>. Indeed, if <valid date> is earlier than the current date, this date is taken into account but is for the next year.*
		= T	= *<empty>*	*GTT needs ExpireTime*
			= <invalid time according to ExpireTime superficial control>	*Field (ExpireTime) invalid*
			= <invalid time, for example '999999'>	*GTT: invalid ExpireTime*
			= <valid time>, with this <valid time> earlier than the current time	*GTT time must be > current time*
	≠ 6	= *<empty>*	= *<not empty>*	*Dated/timed only on GTD/GTT orders*
		= D	= *<whatever value>*	*Bad ExpireTimeFlag, not a GTD*
		= T	= *<whatever value>*	*Bad ExpireTimeFlag, not a GTT*

4.4 BULK CANCELLATION

	SenderCompID value	Symbol value	ClassID value	Error message
Attributes combinations and associated error messages	= *<empty>*	= *<empty>*	= *<empty>*	*Invalid OnBehalfOfCompID on bulk cancel*
			= <not existing ClassID>	*Invalid Class*
			= <valid ClassID>	*Invalid OnBehalfOfCompID on bulk cancel*
		= <not existing Symbol>	= *<whatever value>*	*Invalid Symbol*
		= <valid Symbol>	= *<empty>*	*Invalid OnBehalfOfCompID on bulk cancel*
			= <not existing ClassID>	*Invalid Class*
			= <valid ClassID>	*Invalid OnBehalfOfCompID on bulk cancel*
	= <not existing Firm>	= *<whatever value>*	= *<whatever value>*	*Invalid OnBehalfOfCompID*
	= <valid Firm>	= *<empty>*	= *<empty>*	*Missing Class/Instrument on bulk cancel*
			= <not existing ClassID>	*Invalid Class*
			= <valid ClassID>	*No error message: Bulk cancel OK*

		= <not existing Symbol>	= *<whatever value>*	***Invalid Symbol***
		= <valid Symbol>	= *<empty>*	*No error message: Bulk cancel OK*
			= <not existing ClassID>	***Invalid Class***
			= <valid ClassID>	*No error message: Bulk cancel OK*

4.5 PRICE INPUT

PriceInputType check	Class Valuation Type check (in case PriceInputType = V)	Class Indicative Of Interest Type check (in case PriceInputType = A)	Firm Connection Type check (in case PriceInputType = R)	Price presence/validity check	Order entry allowed check	Price within collars check	Firm Indicative Price Input authorization check	Firm Market Maker check	Class phase check	ERROR MESSAGE
PriceInputType (field of PriceInput message) must have one of its possible values (V,A,R)	Class Valuation Type (Class parameter) must be F or R	Class IOI Type (Class parameter) must be F	Firm Connection Type (Firm parameter) must be Z	Pricemust: - Be valuated in case [(PriceInputType=V & Class Valuation Type=F) or (PriceInputType=A & ClassAIPType=F) or (PriceInputType=R & FirmConnectionType=Z)] - Not be valuated in case PriceInputType=V & Class Valuation Type=R. If valuated, Price must be valid (refer to Price validity checks and associated errors)	Order entry must be allowed for theClass (Class parameter) AND the Firm (Firm parameter) AND the Instrument (Instrument parameter)	Price (field of PriceInput message) must be within collars	Firm IPI authorization (Firm parameter) must be “ValuationTrades” (in case PriceInputType=V) or “AIP” (in case PriceInputType=A)	The Firm authorization (Firm parameter) must be defined as a Market Maker for the instrument	The current Class phase (Market parameter) must be Core or Closing Call	↓
X										*Invalid PriceType*
√ (V)	X									*Parameters do not allow this function*
	√ (F)			X						*For the different errors and their associated texts concerning the Price validity checks, refer to* *Price validity checks and associated errors*
				√	X					*Parameters do not allow this function*
					√	X				*Cannot trade outside collars*
						√	X			*Parameters do not allow this function*
							√	X		*Parameters do not allow this function*

								√	X	*Parameters do not allow this function*
									√	*No error message: request OK*
	√ (R)			X						*For the different errors and their associated texts concerning the Price validity checks, refer to Price validity checks and associated errors*
				√	X					*Parameters do not allow this function*
					√	X				*Cannot trade outside collars*
						√	X			*Parameters do not allow this function*
							√	X		*Parameters do not allow this function*
								√	X	*Parameters do not allow this function*
									√	*No error message: request OK*
√ (A)		X		*≠ <empty>*						*Parameters do not allow this function*
				= <empty>						*Invalid price*
		√ (F)		X						*For the different errors and their associated texts concerning the Price validity checks, refer to Price validity checks and associated errors*
				√	X					*Parameters do not allow this function*
					√	X				*Cannot trade outside collars*
						√	X			*Parameters do not allow this function*
							√			*No error message: request OK*
√ (R)			X							*Parameters do not allow this function*
			√	X						*Invalid price*
			√	√						*No error message: request OK*

Legend

The 2 first lines give a list (refer to Price Input Request (I) section for the list of all Price Input checks performed) of successive checks performed when a Price Input request is received, with a title of the check in the 1st line and a check definition/description in the 2nd line.

√ (value)	The concerned check (see the 2 first lines of the same column) is OK. A value can also be indicated to be more specific about the case described.
X	The concerned check (see the 2 first lines of the same column) is wrong and implies an error, whose associated text is given in the last column of this line.
	This parameter is no more taken into account as the current case is already an ERROR case, whose appearance conditions are to be found in the previous column, in the same line, having a red cross (see also just above), and whose associated text is given in the last column of this line.

	This parameter is not taken into account in the current case (defined by the previous columns values of the same line).
No error message: request OK	No error text: the request is accepted in the current case.

4.6 CANCEL/REPLACE & CANCELLATION

`OrderID` value in request	`OrigClOrdID` value in request	Error message
= *<empty>*	= *<empty>*	*Missing OrigClOrdID*
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value> AND ≠ <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value>	*OrigClOrdID not found*
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value> AND = <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value>	
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value> AND = <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value>	*No error message: request accepted in this case, the order having the OrigClOrdID value of the request is cancelled or modified.*
≠ *<empty>* AND ≠ <`OrderID` value of an order having 'O' as `OriginMsg` value> AND ≠ <`OrderID` value of an order having 'N' as `OriginMsg` value> AND having the <date+NSO> format	= *<empty>*	*OrderID (HON) not found*
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value> AND ≠ <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value>	
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value> AND = <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value>	
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value> AND = <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value>	
≠ *<empty>* AND ≠ <`OrderID` value of an order having 'O' as `OriginMsg` value> AND ≠ <`OrderID` value of an order having 'N' as `OriginMsg` value> AND not having the <date+NSO> format	= *<empty>*	*Missing OrigClOrdID*
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value> AND ≠ <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value>	*OrigClOrdID not found*
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value> AND = <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value>	
	≠ *<empty>* AND ≠ <`OrigClOrdID` value of an order having 'N' as `OriginMsg` value> AND = <`OrigClOrdID` value of an order having 'O' as `OriginMsg` value>	*No error message: request accepted in this case, the order having the OrigClOrdID value of the request is cancelled or modified.*
≠ *<empty>* AND	= *<empty>*	*No error message: request accepted in this case, the order having the OrderID*

OrderID value in request	OrigClOrdID value in request	Error message
≠ <OrderID value of an order having 'O' as OriginMsg value> AND = <OrderID value of an order having 'N' as OriginMsg value>	≠ <*empty*> AND ≠ <OrigClOrdID value of an order having 'O' as OriginMsg value> AND ≠ <OrigClOrdID value of an order having 'N' as OriginMsg value>	*value of the request is cancelled or modified.*
	≠ <*empty*> AND ≠ <OrigClOrdID value of an order having 'O' as OriginMsg value> AND = <OrigClOrdID value of an order having 'N' as OriginMsg value, but not the order having the OrderID value of the request>	
	≠ <*empty*> AND ≠ <OrigClOrdID value of an order having 'N' as OriginMsg value> AND = <OrigClOrdID value of an order having 'O' as OriginMsg value>	
≠ <*empty*> AND ≠ <OrderID value of an order having 'N' as OriginMsg value> AND = <OrderID value of an order having 'O' as OriginMsg value>	= <*empty*>	*Missing OrigClOrdID*
	≠ <*empty*> AND ≠ <OrigClOrdID value of an order having 'O' as OriginMsg value> AND ≠ <OrigClOrdID value of an order having 'N' as OriginMsg value>	*OrigClOrdID not found*
	≠ <*empty*> AND ≠ <OrigClOrdID value of an order having 'O' as OriginMsg value> AND = <OrigClOrdID value of an order having 'N' as OriginMsg value>	
	≠ <*empty*> AND ≠ <OrigClOrdID value of an order having 'N' as OriginMsg value> AND = <OrigClOrdID value of an order having 'O' as OriginMsg value, but not the order having the OrderID value of the request>	*No error message: request accepted in this case, the order having the OrigClOrdID value of the request is cancelled or modified.*