EURONEXT

Document title

EURONEXT CASH MARKET - UTP TO OPTIQ™ MDG KINEMATICS SPECIFICATIONS

Version number

1.1.0

Date

28 Sep 2016

Number of pages

65

SBE Template Version

1.14.1

PREFACE

PURPOSE

The Optiq Programme will replace the current Euronext core trading applications (UTP/CCG/XDP) for the Cash (Regulated Market and Warrants) and Derivative markets.

Following customer feedback, the migration to Optiq will start with the Market Data module for Cash and Derivatives. As such, the new Optiq MDG will be connected to the Euronext Cash and Derivatives UTP matching engines in phase 1, and then to the Optiq matching engines for Cash (phase 2) and then Derivatives (phase 3).

Optiq MDG will be linked to UTP-C during phase 1 of the migration, and this document describes the message kinematics between Optiq MDG and UTP-C.

Important note : *For the transition phases with Optiq MDG still connected to the Cash and Derivatives UTP matching engines, the trading functionalities remain the same, behaviors are not altered. Thus, in terms of Market Data, the message sending rules and message sequencing are driven by UTP matching engine. Those rules are not necessarily those that will be delivered on the full Optiq platform, in phase 2 for Cash markets and phase 3 for derivatives markets.*

TARGET AUDIENCE

The intended audience of this document is any client consuming Euronext Cash market data.

SCOPE

The scope of this document is listed below (✔ In scope, ✖ Out of scope):

Products	
Equities	✔
Exchange Traded Funds	✔
Fixed Income	✔
Warrants & Certificates	✔
Derivatives – Options	✖
Derivatives – Futures	✖
Commodities	✖

WHAT'S NEW?

The following lists only the most recent modification made to this revision/version. For the Document History table, see the Appendix.

VERSION NO.	DATE	CHANGE DESCRIPTION
1.1.0	28 Sep 2016	-Addition of the SBE Template Version that this document refers (located on the first page). -Clarification on the conventions used in the document. -TradingMode change in Market Status Changes section. -Addition of Valuation Price & AIP kinematics.

ASSOCIATED DOCUMENTS

The following documents should be read in conjunction with these documents:

- Euronext Cash and Derivatives Markets - Optiq™ MDG Client Specifications.
- Euronext Cash and Derivatives Markets - Change Highlights - XDP To Optiq™ MDG Migration.

CONTENTS

1. OVERVIEW

1.1   Introduction

This document provides an overview of messages exchanged between the Universal Trading Platform (UTP), order entry gateways (CCG's) and the Market Data Gateway (MDG). It includes:

■ Typical trading situations and the corresponding public and private messages that are sent during the trading day.

■ The names of the messages that are sent.

■ The events that trigger the transmission of messages.

The messages that are sent between trading members and UTP are referred to as private messages, and the messages that are sent between UTP and the market data feed are referred to as public messages.

These messages are sent in response to events that affect the state of an instrument, an order or the market.

Private messages are confidential messages sent via order entry gateways to the UTP matching engine, to request information from the system or to issue a command (e.g. enter an order). Private messages are also sent by UTP back to the order entry gateways to provide the information requested, or confirm that a command has been successfully executed (or not), as well as to notify trades or market events, etc.

Public messages are sent by UTP via MDG to provide the market with anonymous trading data, such as orders entered, best limits, executed trades, market events, etc.

The diagrams in this document express representative examples of message sequences and do not provide an exhaustive overview of all possible message transmissions. The detail of the message contents may vary depending on the example.

For a complete description of the private messages and their fields, please refer to:

■ Euronext Cash Regulated Markets – CCG Client Specifications – Binary Interface.

■ Euronext Cash Regulated Markets – CCG Client Specifications – FIX Interface.

1.2 Message Codes And Names

1.2.1 Private Messages

1.2.1.1 FIX Protocol

FIX message IDs are provided throughout the message kinematics as shown below, in this example for the Exec Report message:

8

Below is a list of the names and corresponding codes of FIX Private messages.

FIX APPLICATION MESSAGES			
Incoming Messages		**Outgoing Messages**	
Code	**Description**	**Code**	**Description**
D	New Order Single	8	Execution Report
F	Order Cancel Request	8	Execution Report
		9	Cancel Reject
G	Cancel/Replace Order	8	Execution Report
q	Order Mass Cancel	r	Order Mass Cancel Report
UB	Bulk Quote	UJ	Bulk Quote Acknowledgement
		UN	End of a One Side Only Period
UI	Price Input	Uy	Request Acknowledgement message
UO	One Side Only Period	UP	One Side Only Period Acknowledgement
UZ	Liquidity Provider Command	Uy	Request Acknowledgement message

1.2.1.2   Binary Protocol

Binary message IDs are provided in brackets throughout the message kinematics section as shown below, in this example for an order acknowledgement:

(a)

The list of Binary messages is hereafter provided:

BINARY APPLICATION MESSAGES			
Incoming Messages		**Outgoing Messages**	
Code	**Description**	**Code**	**Description**
D	New Order Single	a	Order Acknowledgement
		2	Order Fill
		8	Order Rejection
		4	Order Killed
F	Order Cancel Request	6	Cancel Request Acknowledgement
		8	Order Cancel/Replace Reject
		K	Bulk Cancel Ack/Report
G	Cancel/Replace Order	E	Cancel/Replace Request Ack
		5	Order Replaced
B	Bulk Quote	J	Bulk Quote Ack
		N	End of a One Side Only Period
I	Price Input	y	Generic Response
O	One Side Only Period	P	One Side Only Period Acknowledgement
Z	Liquidity Provider Command	y	Generic Response

1.2.2   Public Messages (Market Data Gateway messages)

MDG message identifiers are provided throughout the message kinematics section as shown below, in this example for a Market Update:

1001   **MarketUpdate**

Message Code	Description
1001	Market Update
1002	Order Update
1003	Price Update
1004	Trade Full Information
1005	Market Status Change
1006	Timetable
1007	Standing Data
1009	Statistics
1010	Exchange Announcement

1.2.3   MDG messages Mapping with former XDP messages

A mapping of MDG messages with the former XDP protocol is provided hereafter:

UTP Messages		Optiq™ MDG Messages	
XDP Code	**Description**	**Binary Code**	**Description**
140	Quotes	**1001**	Market Update
146	Retail Matching Facility (RMF) Quotes	**1001**	Market Update
221	Trade Cancel	**1001**	Market Update
230	Order Update / Market Sheet	**1002**	Order Update
240	Trade Full Information	**1004**	Full Trade Information
241	Price Update	**1003**	Price Update
243	Trade Publication	**1004**	Full Trade Information
244	Settlement Price	**1003**	Price Update
245	Auction Summary	**1009**	Statistics
505	Stock State Change	**1005**	Market Status Change
516	Class State Change	**1005**	Market Status Change
523	Mail	**1010**	Exchange Announcement
530	Indicative Matching Price	**1003**	Price Update
534	Authorized Out of Session Limits	**1009**	Statistics
537	Collars	**1003**	Price Update
539	Session Timetable	**1006**	Timetable

UTP Messages		Optiq™ MDG Messages	
XDP Code	**Description**	**Binary Code**	**Description**
550	Start Reference Data	**1007**	Standing Data
551	End Reference Data	**1007**	Standing Data
553	Reference Data	**1007**	Standing Data

1.2.4 Conventions used in this document

1.2.4.1 Full Trade Information generation

A Public **FullTradeInformation** message is sent each time a Market Update (following a trade) is disseminated to the Market, but for readability purposes it is not shown on the kinematics diagrams.

Every time there is change in reference price, the dynamic collars are disseminated in the **MarketUpdate** message with the trade.

Example:

The Last Traded Price of an instrument was 100, low dynamic collar was 95, and high dynamic collar was 105.

If a trade occurs and modifies the LTP to 101 (low dynamic collar is updated to 96, and high dynamic collar is updated to 106) then the following sequence will be sent:

1001   **MarketUpdate (Trade + Limits + Collars)** ➡

1004   **FullTradeInformation** ➡

Whenever trades occur and cause an update of the dynamic collars, for simplification purposes, only the following message will be represented:

1001   **MarketUpdate (Trade + Limits)** ➡

1.2.4.2 Order Update and Market Update generation

All kinematics described in this document are common across Regulated Markets and Warrants Markets (when it is not specified in brackets, as it is the case for Warrants kinematics), the following convention has been adopted:

- For all Cash Regulated Markets (Equities, Bonds, ETFs) excluding RMF and SI facilities, and also excluding Quote Driven Warrant Market Models & Certificates, Optiq MDG will provide **OrderUpdate** messages and **MarketUpdate** messages (containing only BBOs and not the depth of the order book, i.e. Market Update Type = 1 (Best Bid) or 2 (Best Offer)).

  The **MarketUpdate** message always follows an **OrderUpdate** message when notifying a limit except when it notifies a trade in which case it is disseminated before the **OrderUpdate** message.

          1.1.0

- For RMF (Retail Matching Facility) and SI (Systematic Internalizer) facilities available on Cash Regulated Markets, Optiq MDG will **not** provide **OrderUpdate** messages at all: it will provide **MarketUpdate** messages only for the full depth of the order book.

  RMF Market Update Types are 16 (New Bid RLP), 17 (New Offer RLP), 18 (Updated Bid RLP) and 19 (Updated Offer RLP).
  SI Market Update Types are 20 (New Bid SI), 21 (New Offer SI), 22 (Updated Bid SI) and 23 (Updated Offer SI)

- For Quote Driven Warrant Market Models & Certificates, Optiq MDG will **not** provide **OrderUpdate** messages at all: it will provide **MarketUpdate** messages for BBOs and for the full depth of the order book.

  Warrant Market Update Types are 1 (Best Bid), 2 (Best Offer), 3 (New Bid), 4 (New Offer), 5 (Updated Bid), 6 (Updated Offer), 55 (New Bid with LP), 56 (New Offer with LP), 57 (Updated Bid with LP), 58 (Updated Offer with LP), 59 (Updated Bid and Offer with LP) and 60 (New Bid and Offer with LP).

Example:

After a trade takes place, the following messages are displayed in the kinematics diagram:

1001   **MarketUpdate** (Trade + Limits) ⟶

1002   **OrderUpdate** (Modify bid and sell orders) ⟶

According to the convention used in this document, this scenario is different for Regulated Markets, RMF and SI facilities, and Quote Driven Warrant Market Models Markets kinematics:

For Regulated Markets (except RMF and SI markets), it should be read as followed:

1001   **MarketUpdate** (Trade) ⟶

1002   **OrderUpdate** (Updated Bid) ⟶

1001   **MarketUpdate** (Best Bid)   (If any change in the Best Bid) ⟶

1002   **OrderUpdate** (Updated Offer) ⟶

1001   **MarketUpdate** (Best Offer)   (If any change in the Best Offer) ⟶

Remark: There is **no** dissemination of the **market depth** in the MDG **MarketUpdate** message.

<u>For RMF (Retail Matching Facility) and SI (Systematic Internalizer) facilities available on Cash Regulated Markets, the scenario is as follows:</u>

1001   **MarketUpdate** (Limits)

Remark: There is **no** MDG **OrderUpdate** message dissemination for RMF and SI Market Models.

<u>For Quote Driven Warrant Market Model Markets, the scenario is as follows:</u>

1001   **MarketUpdate** (Trade)

1001   **MarketUpdate** (Limits + BBOs)

Remark: There is **no** MDG **OrderUpdate** message dissemination for Quote Driven Warrant Market Model Markets.

1.2.4.3   Convention example on a concrete trading case – Phase 1 (UTP to MDG)

Assume that the following bids and offers are present in the order book at Continuous Phase:

Bid			Offer		
Time	Qty	Price	Price	Qty	Time
Order 3	300	10	10	100	Order 1
			10	150	Order 2

When Order 3 enters the order book, two trades are generated and are presented as followed in the present specifications document:

1001   **MarketUpdate** (Trade for 100 + Limits)

1002   **OrderUpdate** (Remove sell order)

1001   **MarketUpdate** (Trade for 150 + Limits)

1002   **OrderUpdate** (Remove sell order, Add buy order)

For Regulated Markets (except RMF and SI facilities), it should be read as followed:

1001 **MarketUpdate** (Trade for 100)

1002 **OrderUpdate** (Delete Order1)

1001 **MarketUpdate** (Best Offer for 150)

1001 **MarketUpdate** (Trade for 150)

1002 **OrderUpdate** (Delete Order2)

1001 **MarketUpdate** (Best Offer for 0)

1001 **MarketUpdate** (Low Collar, High Collar) (2 repeating sections in a single message)

1002 **OrderUpdate** (Add Order3)

1001 **MarketUpdate** (Best Bid for 50)
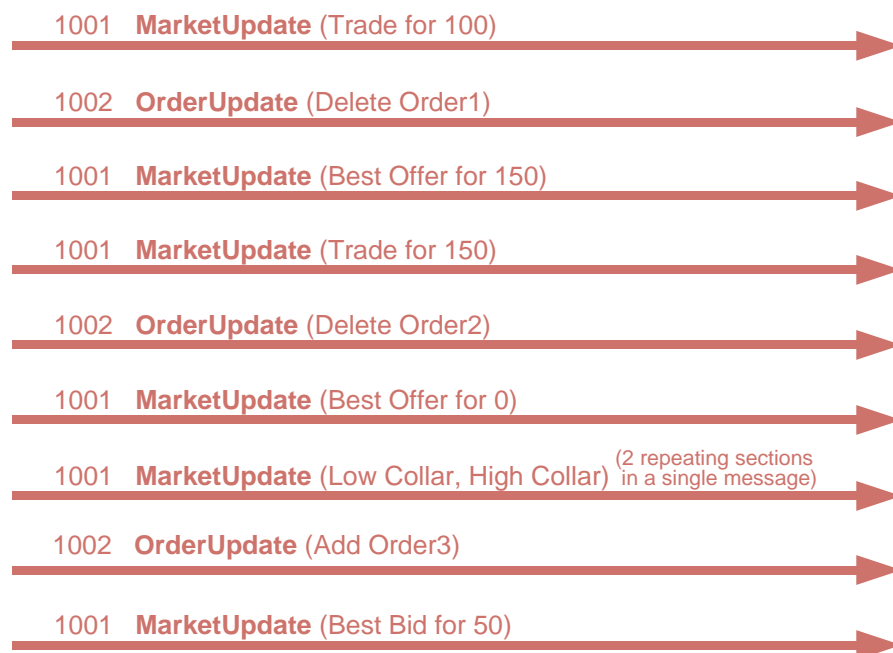
 For RMF and SI, only limits are provided:

1001 **MarketUpdate** (Updated Offer RLP - or SI - for 150)

1001 **MarketUpdate** (Updated Offer RLP - or SI - for 0)

1001 **MarketUpdate** (Updated Bid RLP - or SI - for 50)

For Warrants markets, it should be read as followed:

1001 **MarketUpdate** (Trade for 100)

1001 **MarketUpdate** (Updated Offer for 0, Best Offer for 150) (2 repeating sections in a single message)

1001 **MarketUpdate** (Trade for 150)

1001 **MarketUpdate** (Updated Offer for 0, Best Offer for 0) (2 repeating sections in a single message)

1001 **MarketUpdate** (New Bid for 50, Best Bid for 50) (2 repeating sections in a single message)

1.2.4.4   Convention example on a concrete trading case – Phase 2 (Optiq to MDG)

Let's take the same scenario (previous example) in a full Optiq world.

Bid			Offer		
Time	Qty	Price	Price	Qty	Time
Order 3	300	10	10	100	Order 1
			10	150	Order 2

When Order 3 enters the order book, two trades are generated and the following MDG messages are sent to the Market:

1001 **MarketUpdate** (Trade for 100, Trade for 150, Low Collar, High Collar) (4 repeating sections in a single message)

1002 **OrderUpdate** (Delete Order 1, Delete Order 2, Add Order 3) (3 repeating sections in a single message)

1001 **MarketUpdate** (Delete Best Offer, New Best Bid for 50) (2 repeating sections in a single message)

The following table gives a comparison between the number of messages generated on Phase 1 and those generated on Phase 2:

Phase 1 (RM)	Phase 2
1001 **MarketUpdate** (Trade for 100)	1001 **MarketUpdate** (Trade for 100, Trade for 150, Low Collar, High Collar) (4 repeating sections in a single message)
1002 **OrderUpdate** (Delete Order1)	1002 **OrderUpdate** (Delete Order 1, Delete Order 2, Add Order 3) (3 repeating sections in a single message)
1001 **MarketUpdate** (Best Offer for 150)	1001 **MarketUpdate** (Delete Best Offer, New Best Bid for 50) (2 repeating sections in a single message)
1001 **MarketUpdate** (Trade for 150)	
1002 **OrderUpdate** (Delete Order2)	
1001 **MarketUpdate** (Best Offer for 0)	
1001 **MarketUpdate** (Low Collar, High Collar) (2 repeating sections in a single message)	
1002 **OrderUpdate** (Add Order3)	
1001 **MarketUpdate** (Best Bid for 50)	

The full Optiq target (Phase 2) will reduce the number of messages and improve the bandwidth usage.

Important Note: The full Optiq target (Phase 2) kinematics will be part of another dedicated document.

1.2.4.5　Quote Driven Warrant Example 1 (Phase 1)

Assume that the following orders enter a book on the Warrant Market at Continuous Phase:

Bid			Offer			
Time	Qty	Price	Price	Qty	Time	
Order 2	40	90	90	50	Order 3	Incoming order triggering trading
LPQuote 1	100	90	95	100	LPQuote 1	

The entry of the LP Quote generates the following MDG sequence:

1005   **MarketStatusChange** (Automatic Unhalting by ME)

1001   **MarketUpdate** (New Bid with LP for 100, Best Bid for 100)   (2 repeating sections in a single message)

1001   **MarketUpdate** (New Offer with LP for 100, Best Offer for 100)   (2 repeating sections in a single message)

The entry of the Non-LP order (Order 2) generates the following MDG message:

1001   **MarketUpdate** (Updated Bid with LP for 140, Best Bid for 140)   (2 repeating sections in a single message)

When Order3 enters the order book, one trade is generated:

1001   **MarketUpdate** (Trade for 50)

1001   **MarketUpdate** (Updated Bid with LP for 50)
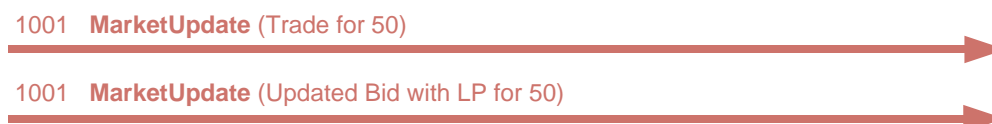
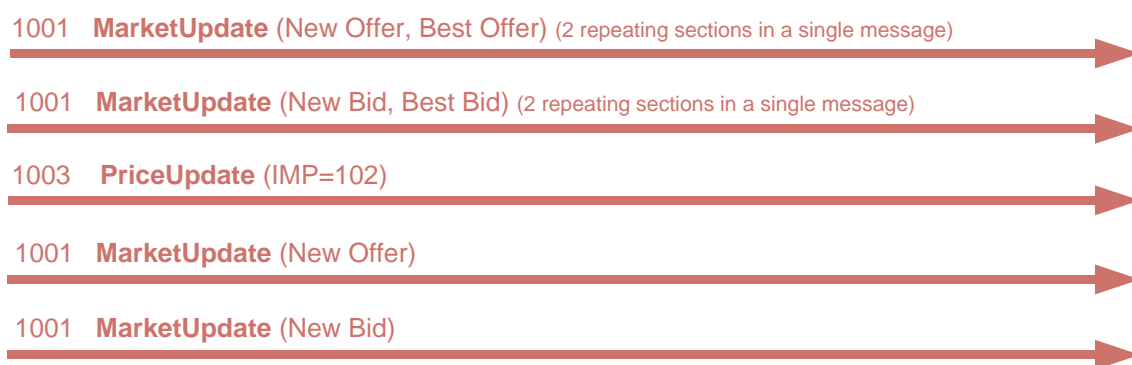### 1.2.4.6   Warrant Example 2 (Phase 1)

Assume that the following bid and offers enters a book on the Warrant Market at Continuous Phase:

Bid			Offer		
Time	Qty	Price	Price	Qty	Time
Order 2	15	104	100	5	Order 1
Order 4	60	102	102	30	Order 3
LPQuote 5	1000	98	102	1000	LPQuote 5

The instrument is halted with reason "No LP" and the order book is empty. The entry of the non-LP orders generates the following MDG sequence:

1001   **MarketUpdate** (New Offer, Best Offer) (2 repeating sections in a single message)

1001   **MarketUpdate** (New Bid, Best Bid) (2 repeating sections in a single message)

1003   **PriceUpdate** (IMP=102)

1001   **MarketUpdate** (New Offer)

1001   **MarketUpdate** (New Bid)

The LP Quotes submission generates the following MDG sequence:

1005   **MarketStatusChange** (Automatic Unhalting by ME, Uncrossing)   (2 repeating sections in a single message)

1003   **PriceUpdate** (Updated IMP)

1001   **MarketUpdate** (Trade for 5)

1001   **MarketUpdate** (Updated Bid, Updated Offer, Best Offer)   (3 repeating sections in a single message)

1001   **MarketUpdate** (Trade for 10)

1001   **MarketUpdate** (Updated Bid, Best Bid, Updated Offer with LP) (3 repeating sections in a single message)

1001   **MarketUpdate** (Trade for 20)

1001   **MarketUpdate** (Updated Bid, Updated Offer with LP)   (2 repeating sections in a single message)

1001   **MarketUpdate** (Trade for 40)

1001   **MarketUpdate** (Updated Bid with LP, Best Bid, Updated Offer with LP) (3 repeating sections in a single message)

2.   COMMON KINEMATICS

The following public messages contains repeating sections:

- PriceUpdate;
- OrderUpdate;
- MarketUpdate;
- MarketStatusChange.

Detailed information regarding repeating sections can be found in the document **Euronext Cash and Derivatives Markets – MDG Client Specifications**.

2.1   Trading Session Management

2.1.1   Initialisation of a New Trading Day
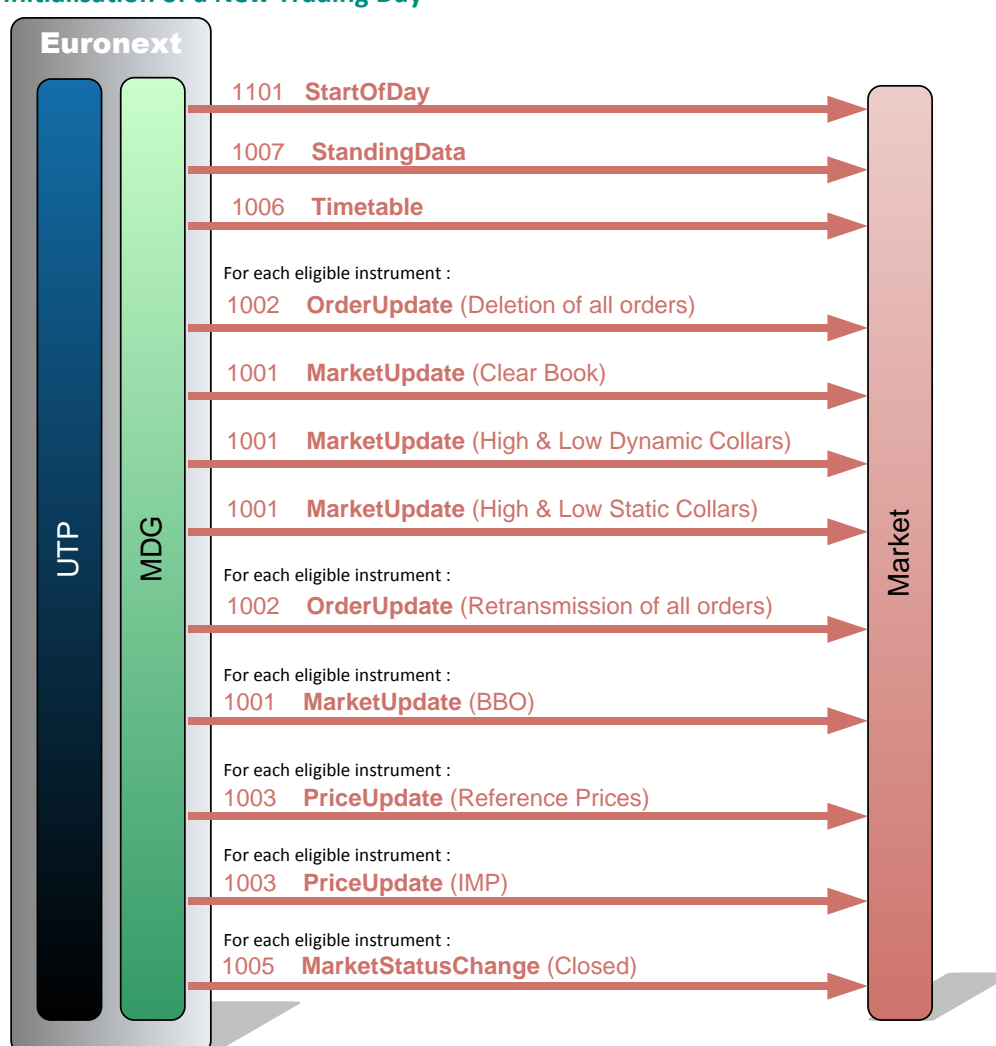
At the start of each new trading day, the following public messages are sent:

- A **StartOfDay** message is always the first message of the day, indicating the date of the trading session.

- A **StandingData** message provides all of the required instrument characteristics for the trading day.

- A **Timetable** message provides all trading patterns that are used across all instruments.

- Then all active orders are sent to clear the order book of each eligible instrument:
    - An **OrderUpdate** message deletes of all orders for the given instrument.
    - A **MarketUpdate** message clears the book of the given instrument.
    - A **MarketUpdate** message provides the Dynamic Collars.
    - A **MarketUpdate** message provides the Static Collars.

- Then there is the retransmission of the order book for each eligible instrument:
    - An **OrderUpdate** message broadcasts all active orders to the Market.

- A **MarketUpdate** message provides the BBOs for all eligible instruments.

- A **PriceUpdate** message provides all updated reference prices complementary to the BBO for trading: Closing Price (YDSP), Uncrossing Price, Valuation Prices, Min/Max Out of Session Trade Prices, Net Asset value for eligible instrument(s) the Official Settlement Price for all eligible instruments.

- A **MarketUpdate** message provides the IMP for all eligible instruments.

- Then the last messages being sent at the Start of the Day are **MarketStatusChange** messages. For all eligible instruments, an instrument state **Closed** is sent to the Market (before any scheduled instrument change like a class opening).

2.1.2 End Of Day
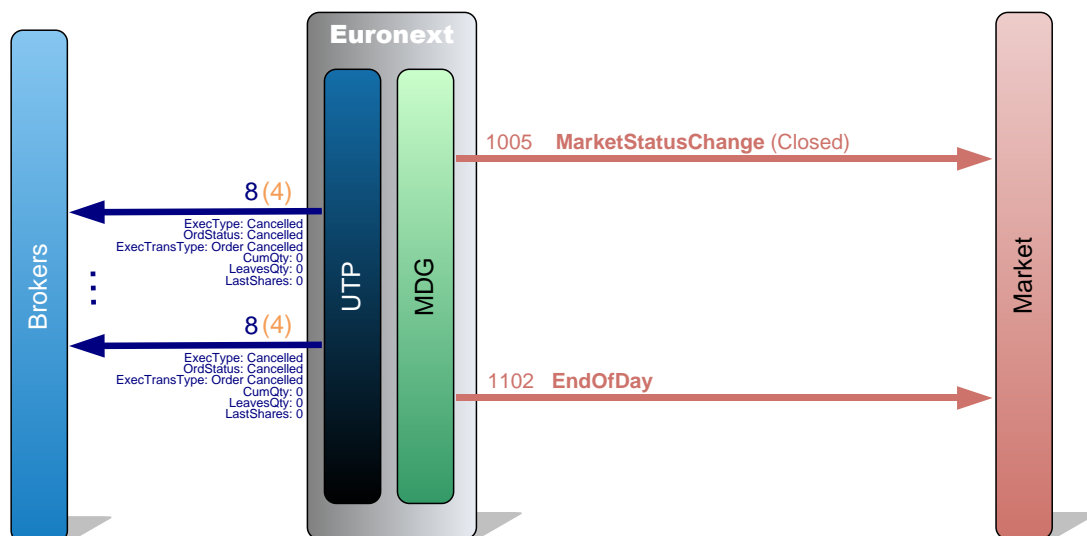
Expired orders (Day orders, GTD orders reaching expiry date, expired Bulk Quotes) are cancelled at the end of the trading day. A public **1005 (MarketStatusChange)** message will be sent for each instrument.

The public **1102 (EndOfDay)** message is always the last message sent, and it notifies that the platform and its network are now closed.
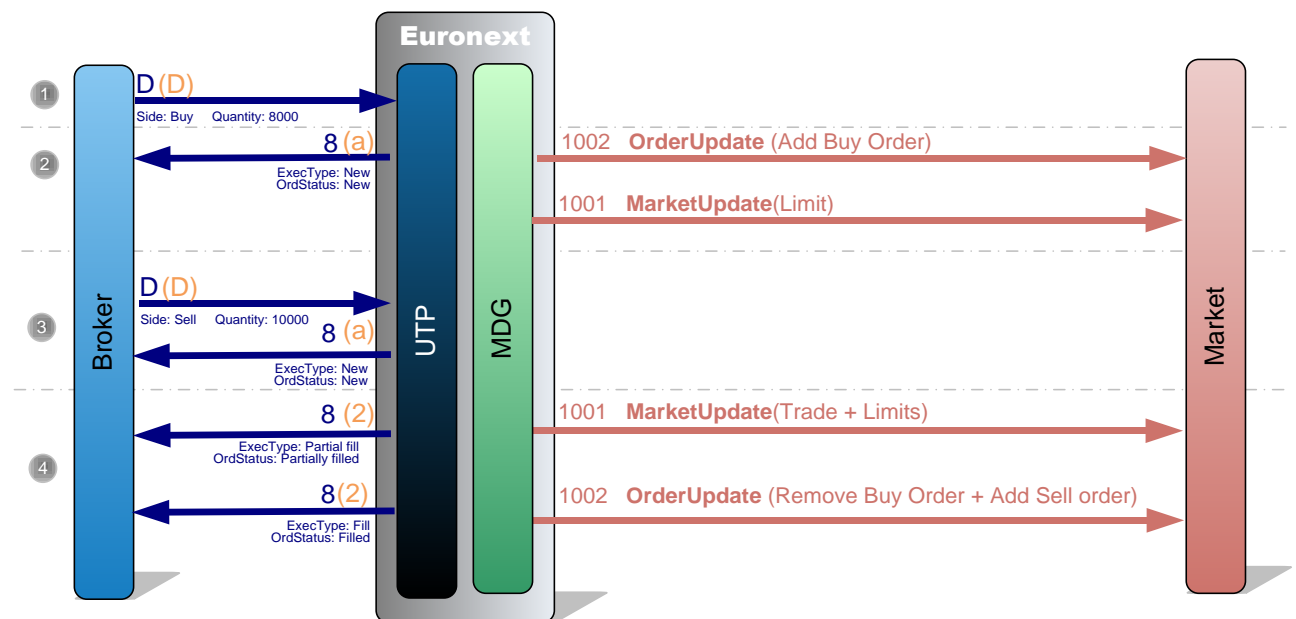
## 2.2   Entering An Order

2.2.1   Incoming Order Fully Matched
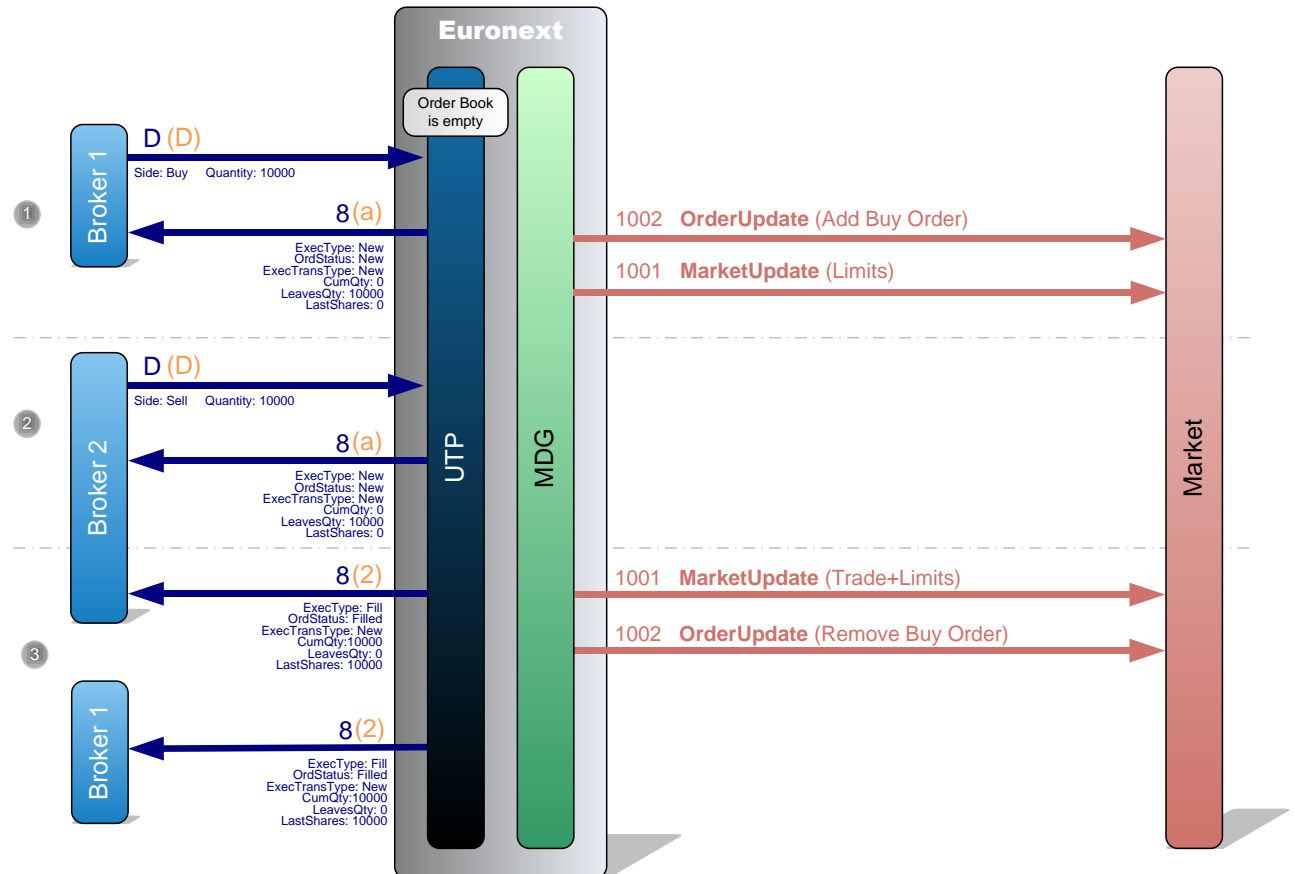
①   A broker sends a private **D (8)** message to enter a new buy order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement. The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order whilst a public **1001 (MarketUpdate)** message is sent to the market to update the limits.

②   Another broker sends a private **D (8)** message to enter a new sell order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

③   The order immediately matches the first order and two private **8 (2)** messages are sent back to each broker to notify the execution, whilst a public **1001 (MarketUpdate)** message is sent to the market to notify the trade and to update the limits.

A public **1002 (OrderUpdate)** message is sent to remove the buy order from the order book .

Note : There is no removal of the sell order in the last public **1002 (OrderUpdate)** message as it is immediately matched and thus never enters the book.

2.2.2   Incoming Order Partially Matched

① A Broker sends a private **D (D)** message to enter a new buy order with a quantity of 8,000 shares.

② UTP sends back a private **8 (a)** message to confirm the order acknowledgement. The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

③ Another Broker sends a private **D (D)** message to enter a new sell order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The two orders match.

④ UTP sends back a private **8 (2)** message to each broker to notify the trade execution whilst a public **1001 (MarketUpdate)** message is sent to the market for the trade and update of limits, along with a public **1002 (OrderUpdate)** message to remove the buy order and to add the sell order (for the remaining quantity of 2,000 shares after the trade).

2.2.3   Cross Order (New Order with a Cross side)
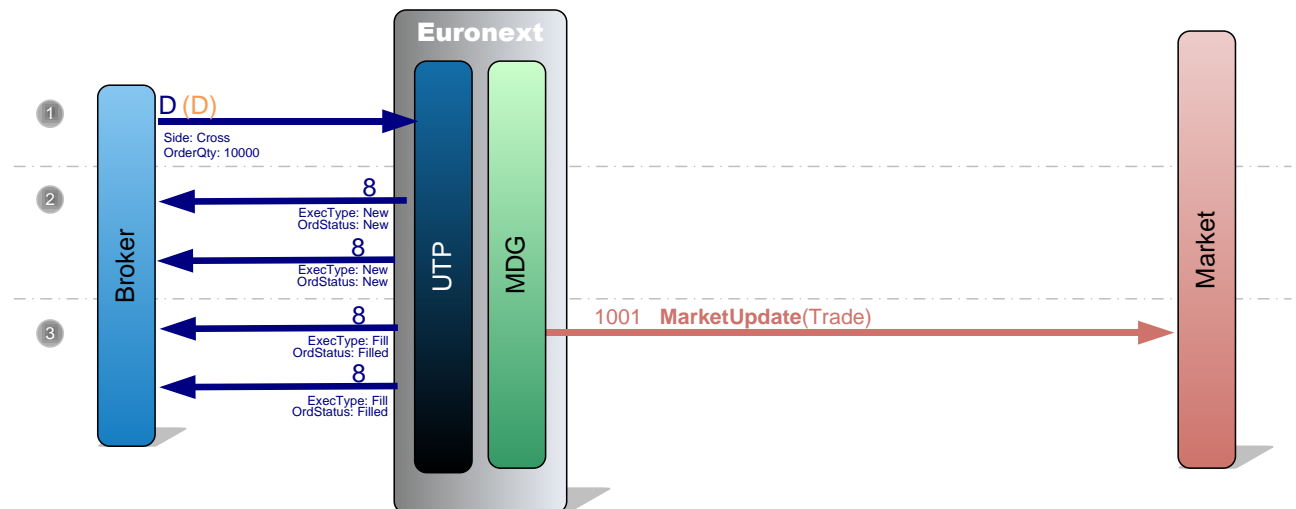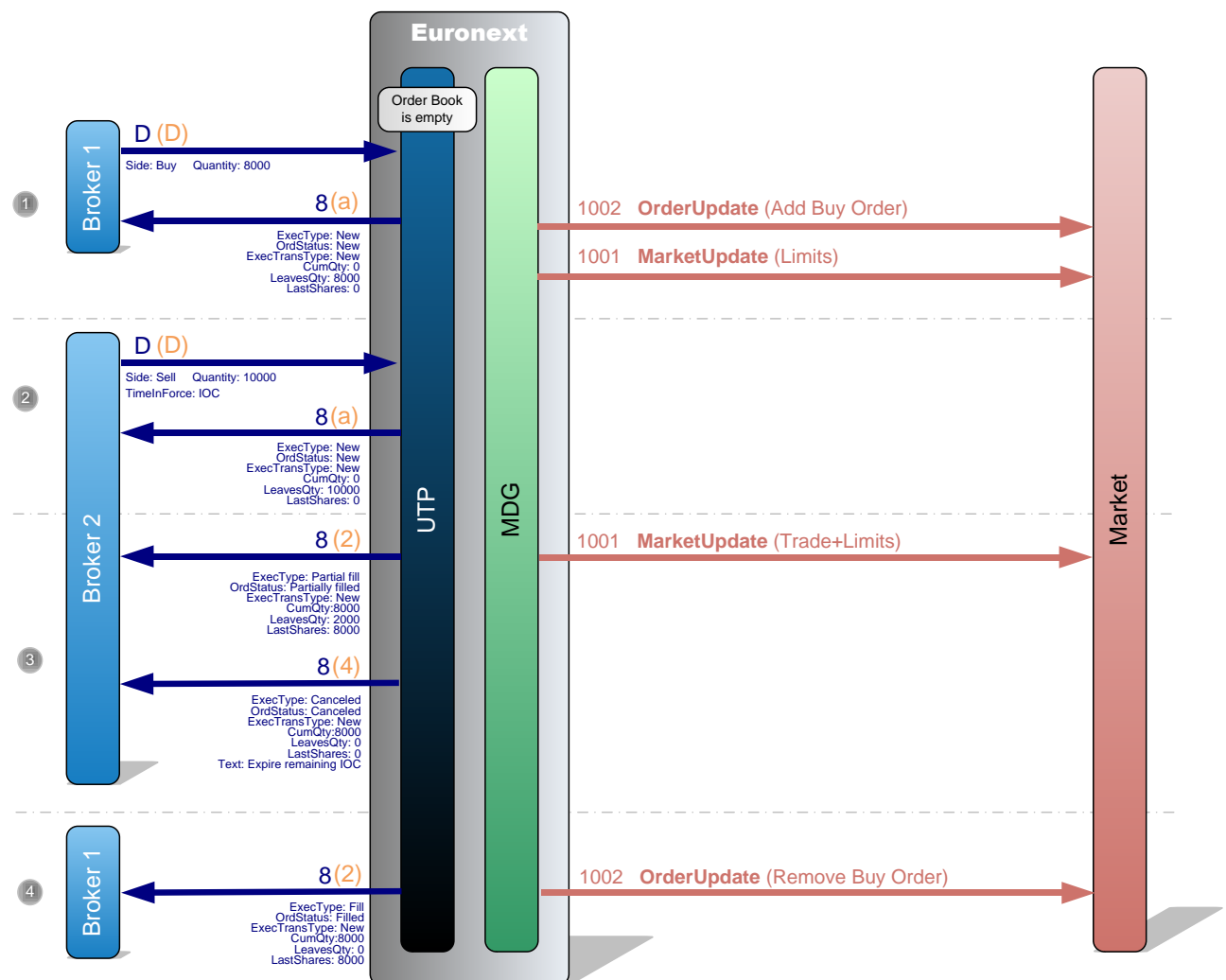

① A Broker sends a private **D (D)** message to enter a new order with a quantity of 10,000 shares and a cross side.

② UTP sends back two private **8 (a)** messages (buy and sell) to confirm the cross order acknowledgement.

③ The entering cross order is immediately filled for its total quantity of 10,000 and two 8 （2） messages are sent back to notify the broker of this full execution while the corresponding public **1001 (MarketUpdate)** message is sent to the market for the trade (and this is the only message being sent).

2.2.4   New Order Rejected
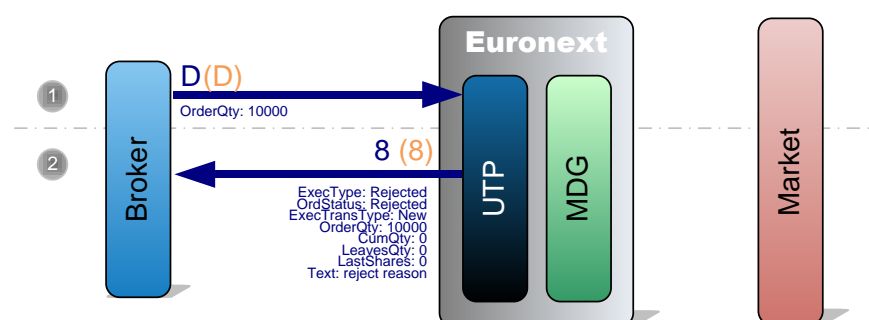
① The Broker sends a private **D (D)** message to enter a new order with a quantity of 10,000 shares.

② If, for whatever reason, the order is rejected, UTP sends back a private **8 (8)** message with a text providing the reason for the rejection (Example: an IOC order received during a call phase is rejected with reason "Mkt/Symbol not open", an order request received with an already existing order identifier is rejected with reason "ClOrdId already exists", a Fill-Or-Kill order request that cannot be immediately filled is rejected with reason "No shares available (FOK)", etc).

2.2.5　Immediate Or Cancel Order Partially Filled

① A broker sends a private **D (8)** message to enter a new buy order with a quantity of 8,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order whilst a public **1001 (MarketUpdate)** message is sent to the market to update the limits.

② Another broker sends a private **D (8)** message to enter a new sell order with a quantity of 10,000 shares and a validity condition of Immediate or Cancel (IOC). UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

③ The entering order immediately matches for a quantity of 8,000 and a private **8 (2)** message is sent back to Broker 2 to notify its execution whilst public **1001 (MarketUpdate)** message is sent to the market for the trade and the limits. As the remaining quantity (2000) cannot be immediately filled, UTP sends back a private **8 (4)** message to cancel it.

④ A private **8 (2)** message is sent back to Broker 1 to notify its execution, and a public **1002 (OrderUpdate)** message to remove the fully matched buy order from the order book.

2.2.6   Market to Limit Order Partially Filled
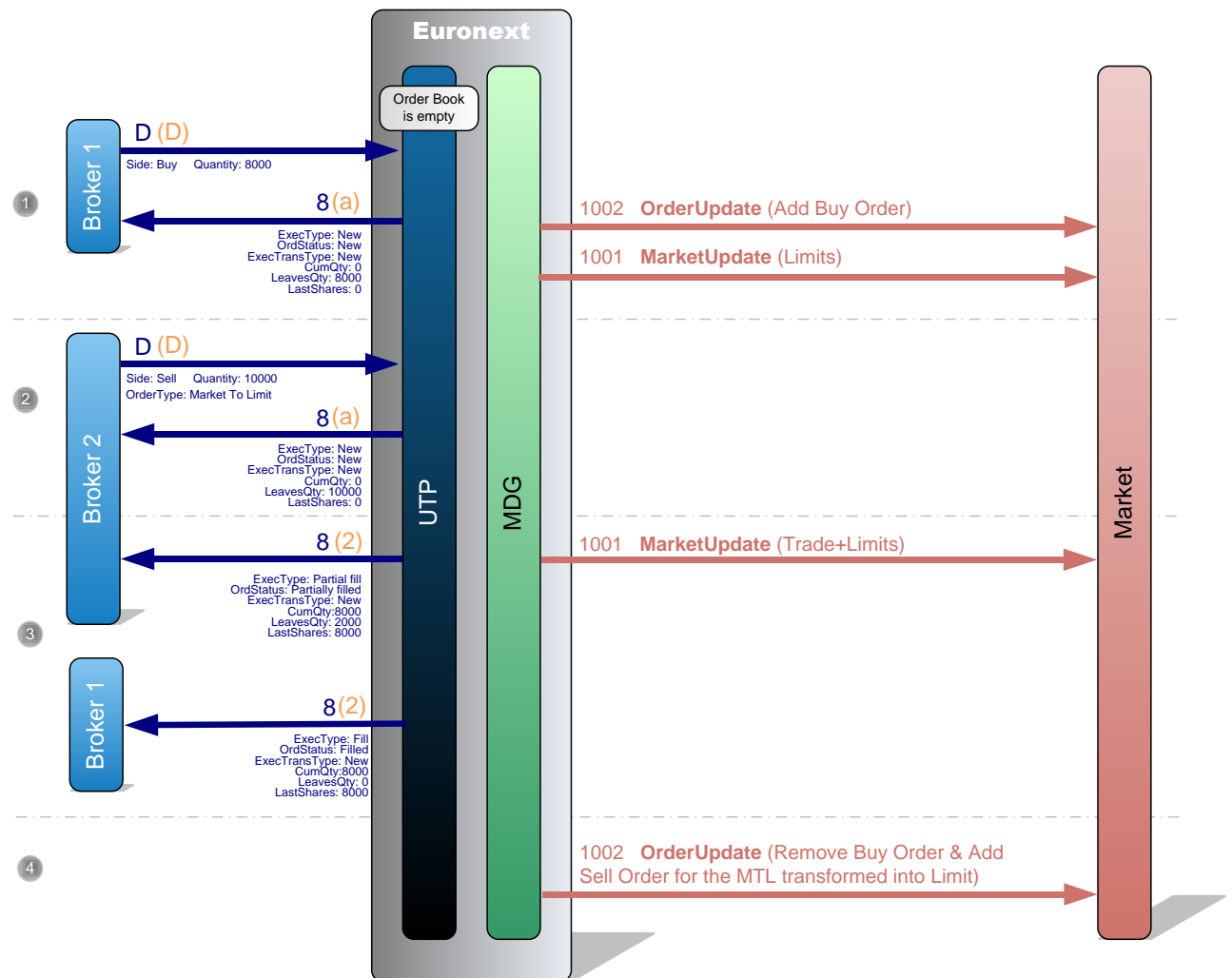
① A broker sends a private **D (8)** message to enter a new buy order with a quantity of 8,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order whilst a public **1001 (MarketUpdate)** message is sent to the market to update the limits.

② Another Broker sends a private **D (D)** message to enter a new <u>Market to Limit</u> order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

③ The entering order immediately matches for a quantity of 8,000 and a private **8 (2)** message is sent back to Broker 2 to notify this partial execution, whilst public **1001 (MarketUpdate)** message is sent to the market for the trade and to update the limits.

A private **8 (2)** message is sent back to Broker 1 to notify the full execution of the passive buy order.

④ A public **1002 (OrderUpdate)** message is disseminated to the market to:

■ Remove the passive buy order;

■ Add the remaining quantity (from the MTL order) transformed into a Limit order.

2.2.7 Triggered Stop Limit Orders
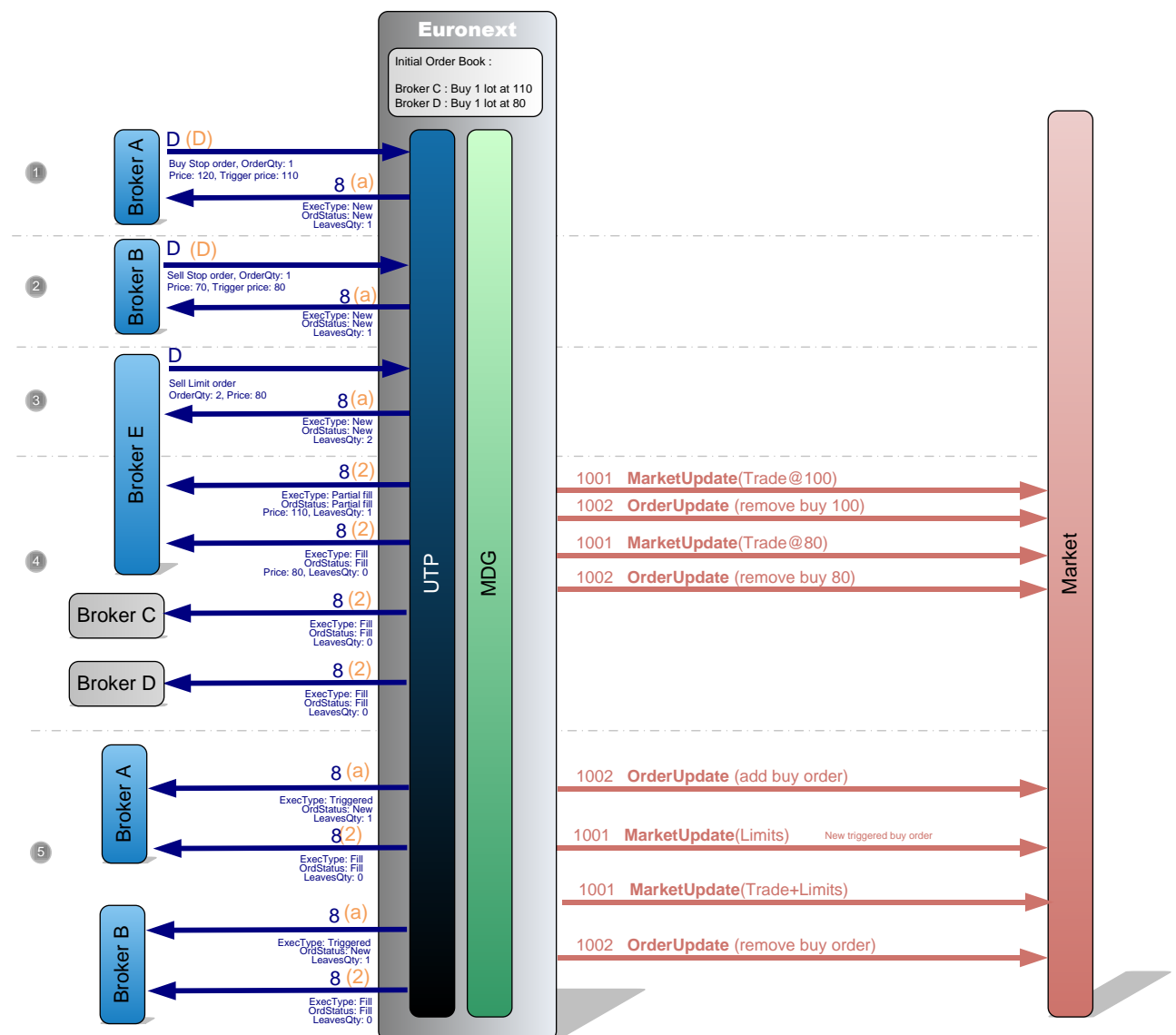
There are already two limit orders in the order book, one buy at price=110 and quantity=1 and another buy at price=80 and quantity=1.

① Broker 'A' sends a private **D (D)** message to enter a new stop limit buy order with Trigger=110 and Price=120. UTP sends back a private **8 (a)** message to confirm the order acknowledgement. No public message is generated.

② Broker 'B' sends a private **D (D)** message to enter a new stop limit sell order with Trigger=80 and Price=70. UTP sends back a private **8 (a)** message to confirm the order acknowledgement but no public message is generated.

③ Another Broker 'C' sends a private **D (D)** message to enter a new sell limit order with a quantity of 2 and a price of 80. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

④ This order matches the two buy orders that were in the order book (price=110 and price=80), so UTP generates four private **8 (2)** messages, two for the buy orders and two for the sell order and this triggers the two Stop Limit Orders.

Consequently two public **1001 (MarketUpdate)** messages are sent to the market for the trades along with two public **1002 (OrderUpdate)** messages to remove the two buy orders from the order book.

⑤ UTP sends a private **8 (a)** message to Broker 'A' to confirm the trigger of the Stop Limit buy order.

The triggered stop limit buy order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order (price=120) along with a public **1001 (MarketUpdate)** message to update the limits.

UTP sends a private **8 (a)** message to Broker 'B' to confirm the trigger of the Stop Limit sell order.

The entering order immediately matches the first order and UTP sends back a private **8 (2)** message to each broker ('A' & 'B') to notify the trade execution at price=120.

Consequently a public **1001 (MarketUpdate)** message is sent to the market for the trade to update the limits, followed by a public **1002 (OrderUpdate)** message to remove the first order (buy) from the order book.
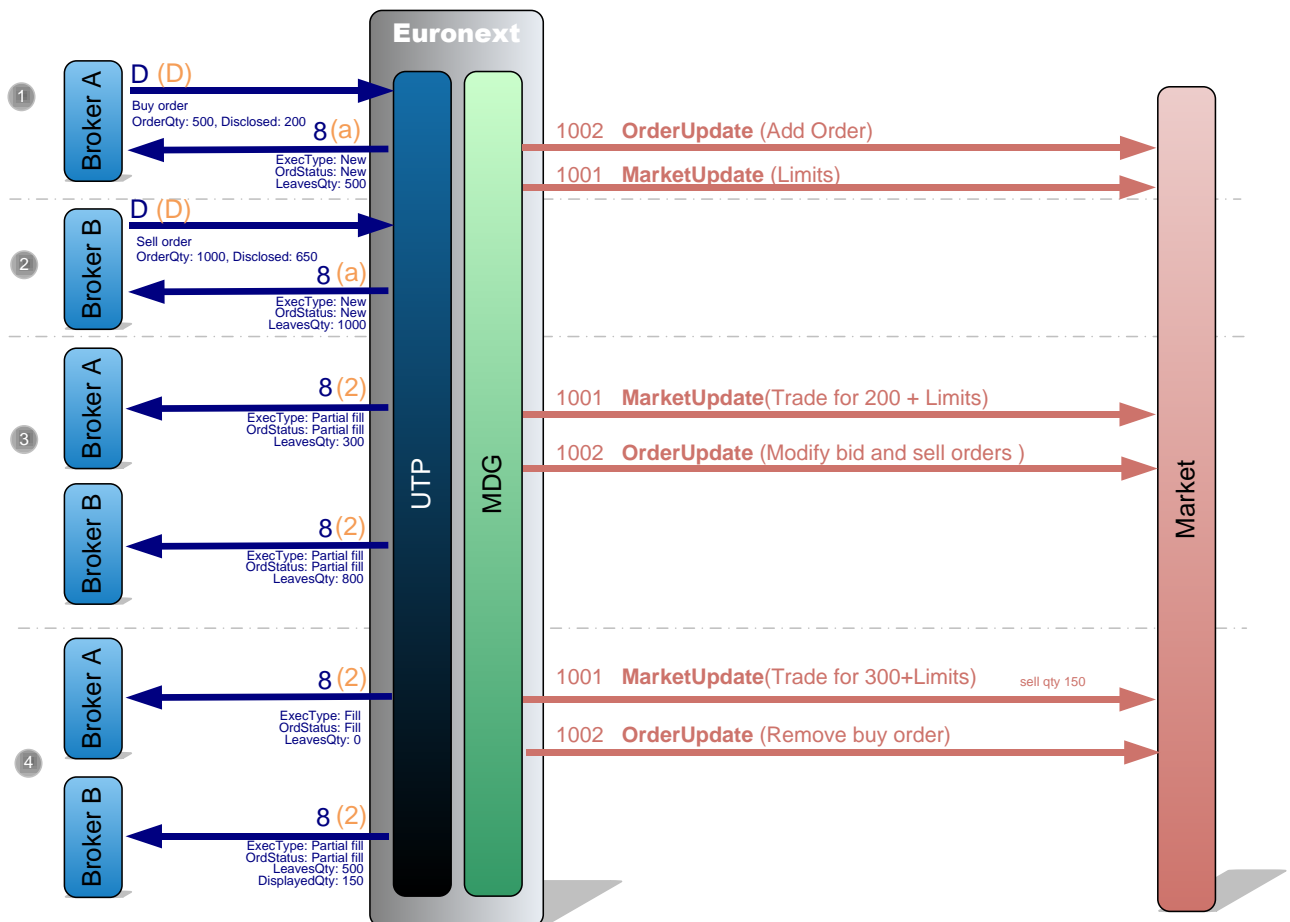
Note: In the Binary protocol, there is currently no way to see the difference between the first acknowledgement (for the new stop order) and the second acknowledgement for the triggered stop order).

In FIX, there are two distinct messages:

> Stop Entry: Exec Type = 0 (New)
> Stop Triggering: Exec Type = L (Stop Triggered)
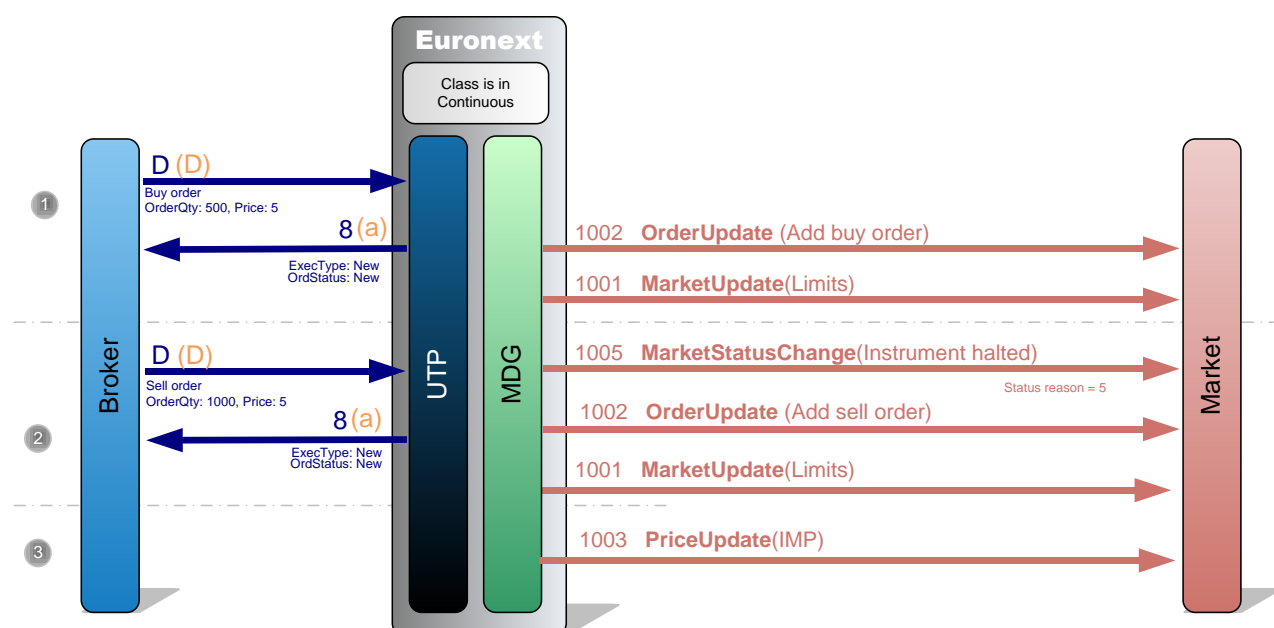
2.2.8 Disclosed Quantity Partially Filled

① **Broker A** sends a private **D (D)** message to enter a Buy order with a total quantity of 500 shares and a disclosed quantity of 200. UTP sends back a private **8 (a)** message to confirm the order acknowledgement and a public **1002 (OrderUpdate)** message is sent to the market along with a public **1001 (MarketUpdate)** message for the new limit.

② **Broker B** sends a private **D (D)** message to enter a Sell order with a total quantity of 1000 shares and a disclosed quantity of 650. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The two orders match for an initial quantity of 200 (the smallest quantity of the two disclosed).

③ UTP sends back a private **8 (2)** message to each broker to notify a partial execution and public **1001 (MarketUpdate)** message is sent to the market for the trade and to update the limits.

④ UTP sends a private **8 (2)** message to each broker (Fill for the Buy and Fill for the Sell) to notify the trade. A public **1001 (MarketUpdate)** message is sent to the market, as well as a public **1002 (OrderUpdate)** message to modify the quantity of the Buy order (300) that remains in the order book. The two orders then match for 300 (the remaining quantity of the buy order) and UTP sends a private **8 (2)** message to each broker to notify the trade execution. Whilst a public **1001 (MarketUpdate)** message is sent to the market for the trade and to remove the first iceberg order (buy) from the order book and to modify the displayed quantity to 150 for the second sell iceberg order. Then a public **1002 (OrderUpdate)** message is sent to remove the first order (buy) from the order book.
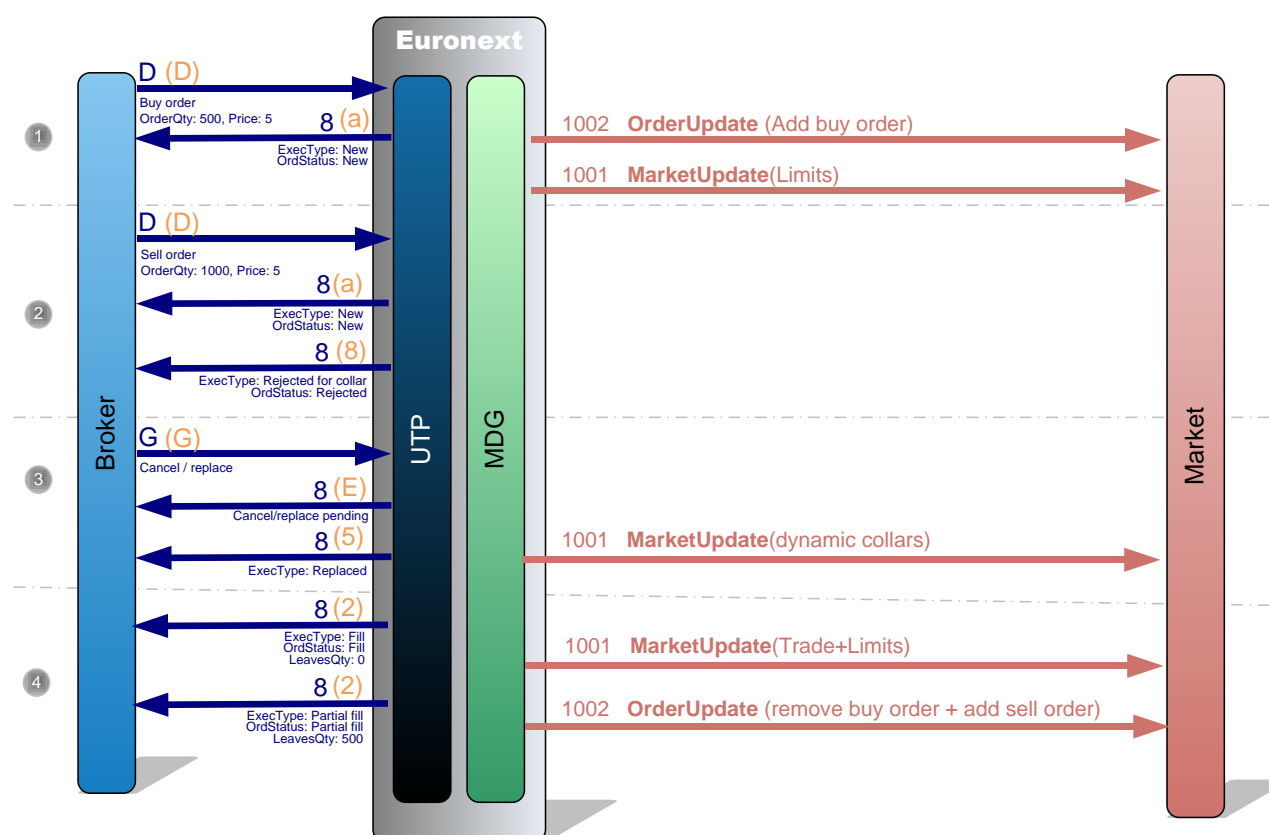
2.2.9   Breaching a Static Collar

① A Broker sends a private **D (D)** message to enter a Buy order with a total quantity of 500 shares at a price of 5, which is lower than the low static collar. UTP sends back a private **8 (a)** message to confirm the order acknowledgement and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

② Another Broker sends a private **D (D)** message to enter a Sell order with a total quantity of 1,000 shares at a price of 5. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

 1.1.0

The instrument is automatically Halted and a public **1005 (MarketStatusChange)** message is sent to the market as well as a public **1002 (OrderUpdate)** message to add the order (sell) in the order book and a public **1001 (MarketUpdate)** message to update the limit.

③ As the instrument is halted, an IMP is calculated and broadcasted to the market via a public **1003 (PriceUpdate)** message.

2.2.10 Breaching a Dynamic Collar (Collar Logic)

In the following example, the collar logic applied to 'Blue Chips' is demonstrated:

① A Broker sends a private **D (D)** message to enter a Buy order with a total quantity of 500 shares at a price of 5, which is lower than the low dynamic collar. UTP sends back a private **8 (a)** message to confirm the order acknowledgement and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

② Another Broker sends a private **D (D)** message to enter a Sell order with a total quantity of 1,000 shares and a price of 5. UTP sends back a private **8 (a)** message to confirm the order reception.

UTP then sends a second private **8 (8)** message to reject/request confirmation for the order breaching the dynamic collar, and thus would match outside it.
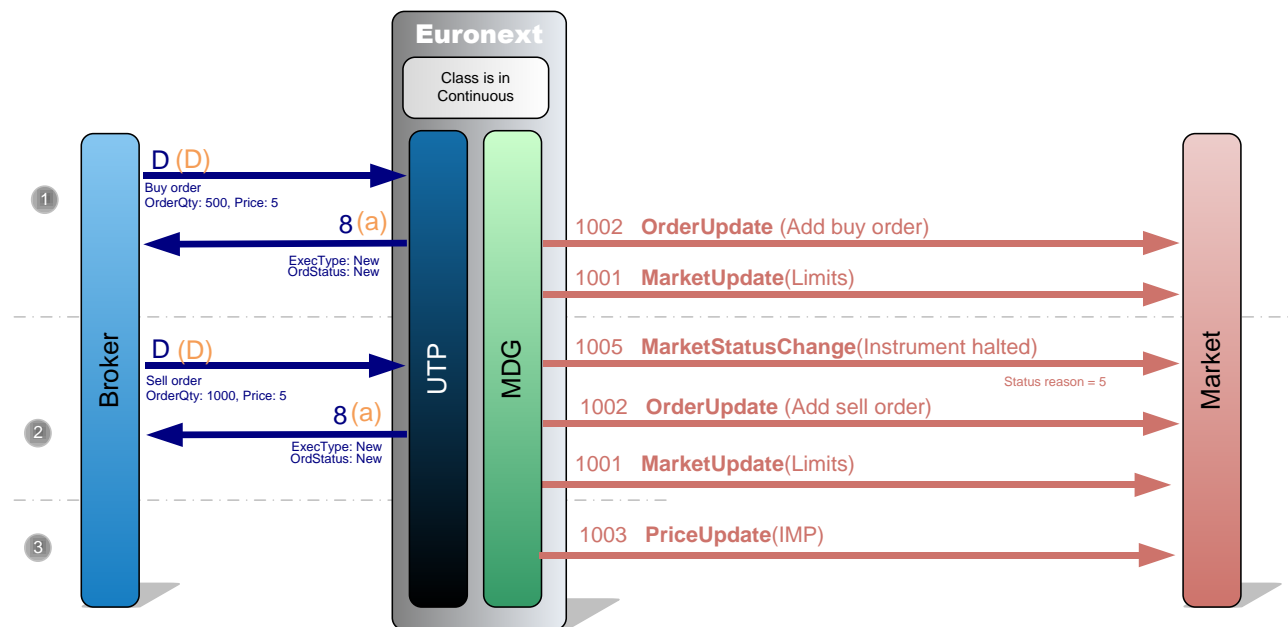
                   1.1.0

③ The Broker sends back a private **G (G)** message to confirm the rejected order and UTP sends back a private **8 (E)** message followed by a private **8 (5)** message to confirm the order acknowledgement.

The collars are updated correspondingly around the low collar and a public **1001 (MarketUpdate)** message is sent to the market to broadcast the new dynamic collars.

④ The two orders match and UTP sends a private **8 (2)** message to each broker (Fill for the Buy and Partial fill for the Sell) to notify the trade. A public **1001 (MarketUpdate)** message is also sent to the market for the trade and to update the limits and the updated collars around the last traded price, followed by a public **1002 (OrderUpdate)** message to remove the Buy order and to add the sell order (with the remaining quantity).

2.2.11 Breaching a Dynamic Collar (Non Collar Logic)

In the following example, the collar logic is <u>not</u> applied.

① A Broker sends a private **D (D)** message to enter a Buy order with a total quantity of 500 shares at a price of 5, which is lower than the low dynamic collar. UTP sends back a private **8 (a)** message to confirm the order acknowledgement and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

② Another Broker sends a private **D (D)** message to enter a Sell order with a total quantity of 1,000 shares at a price of 5. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

         1.1.0

The instrument is automatically Halted and a public **1005 (MarketStatusChange)** message is sent to the market as well as a public **1002 (OrderUpdate)** message to add the order (sell) in the order book and a public **1001 (MarketUpdate)** message to update the limit.

③ As the instrument is halted, an IMP is calculated and broadcasted to the market via a public **1003 (PriceUpdate)** message.

Note: For non-Blue Chips stocks for example, a dynamic collar breach halts the instrument and automatically program a reopening time with newly calculated dynamic collars. The new reference price is the breached collar.

2.2.12 Peg Order
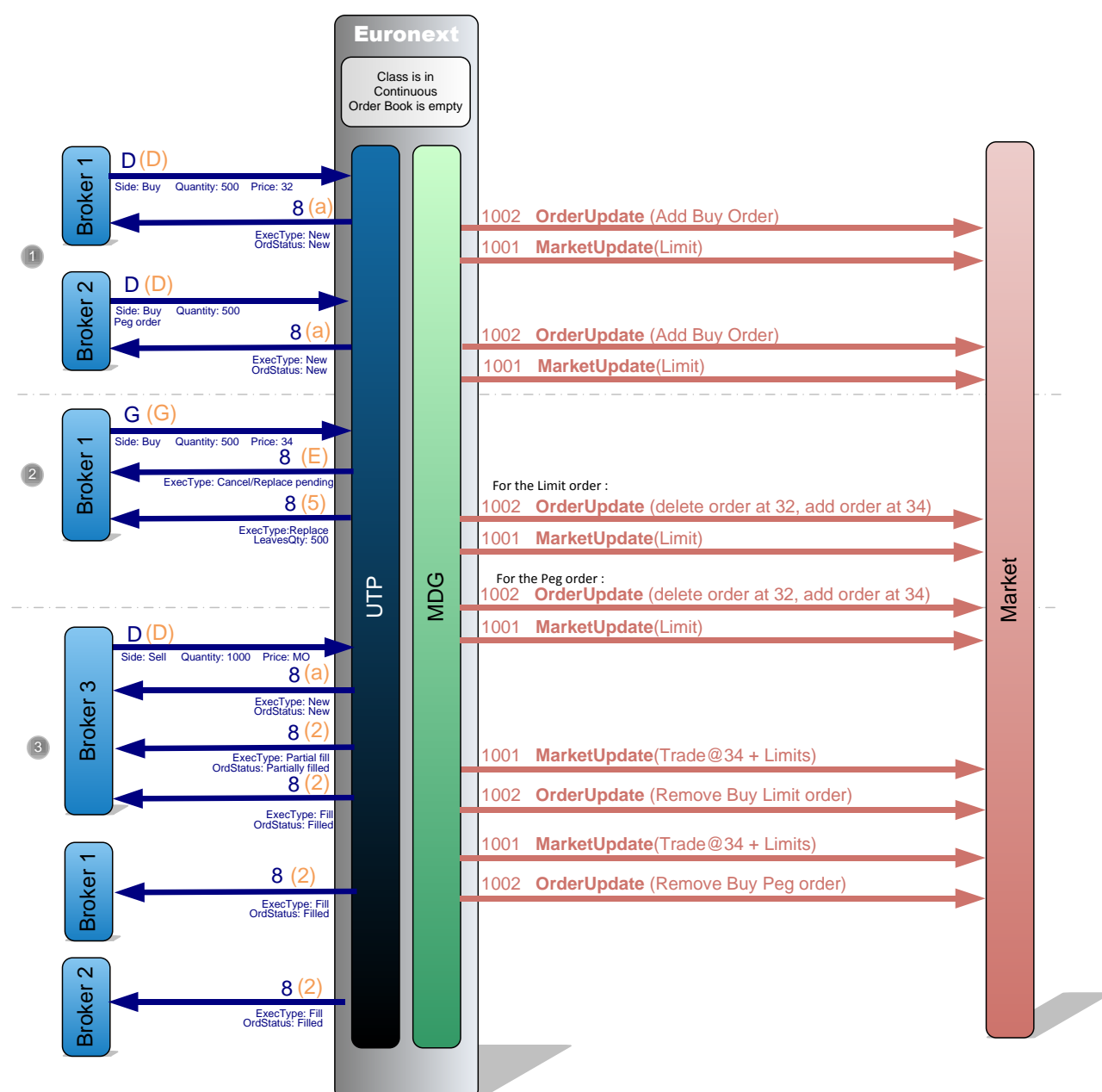
①   On the Normal Market segment, Broker A sends a private **D (D)** message to enter a Limit Buy order with a total quantity of 500 shares at a price of 32. **UTP** sends back a private **8 (a)** message to confirm the order acknowledgement. A public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.
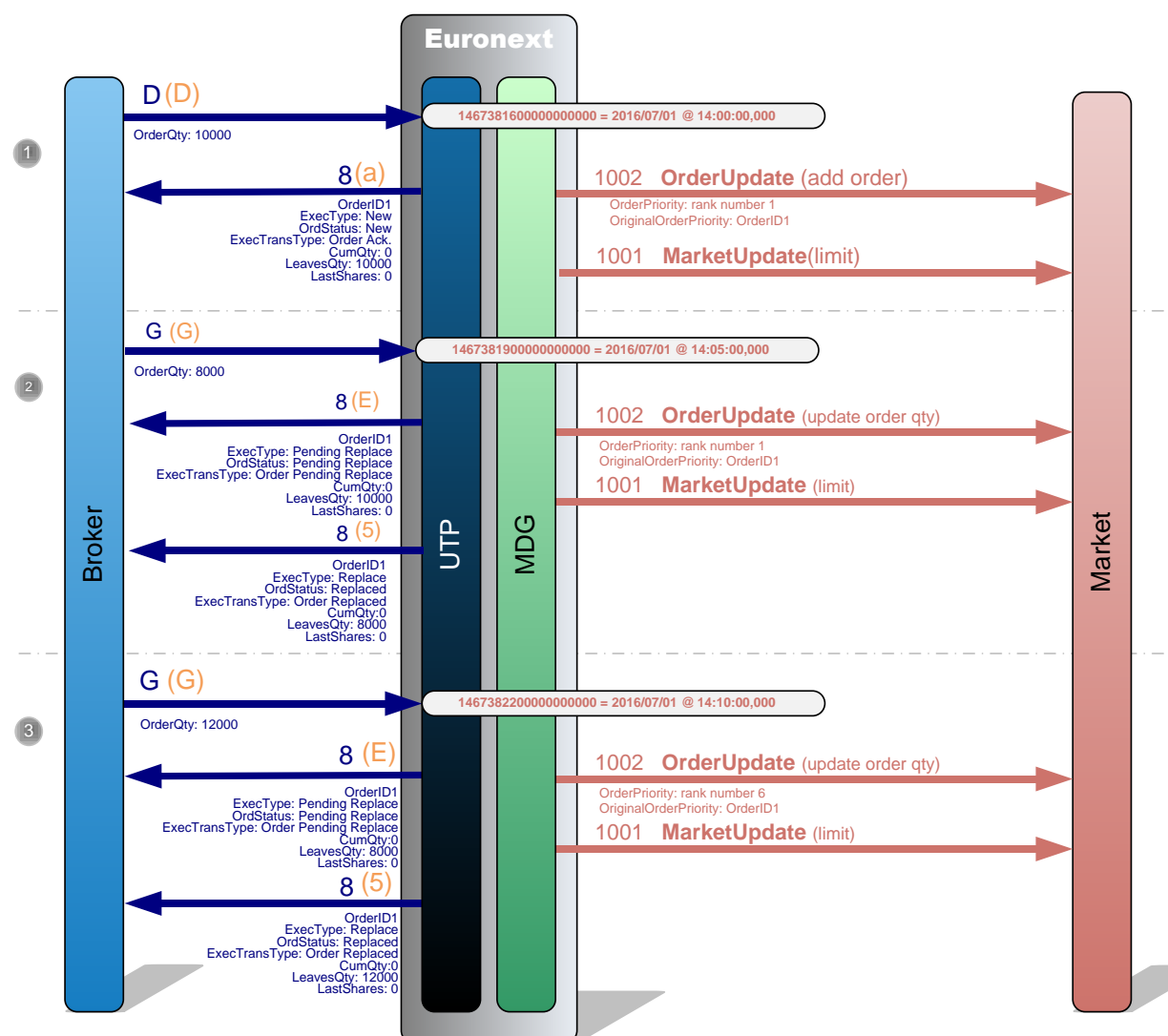
      Broker B sends a private **D (D)** message to enter a Peg Buy order with a total quantity of 500 shares **UTP** sends back a private **8 (a)** message to confirm the order acknowledgement. A public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

②   Later, Broker A sends a private **G (G)** message to cancel/replace the order by changing the price to 34. A public **1002 (OrderUpdate)** message is sent to the market to remove the order at 32 and add the order at 34, and a public **1001 (MarketUpdate)** message to update the limit. The peg order follows this price: the same couple of public messages are sent to the market: a public **1002 (OrderUpdate)** message is sent to remove the order at 32 and add the order at 34, and a public **1001 (MarketUpdate)** message to update the limit

③   Broker C sends a private **D (D)** message to enter a Market Order to sell for a quantity of 1000. **UTP** sends back a private **8 (a)** message to confirm the order acknowledgement. This order immediately matches with the order from Broker A for a quantity of 500 lots at a price of 34 and a private **8 (2)** message is sent back to Broker C to notify this first partial execution, and a private **8 (2)** message is sent back to Broker A to notify the full execution. A public **1001 (MarketUpdate)** message is sent to the market to notify the trade and to update the limits.

      The Market Order to sell matches then for its remaining quantity of 500 with the peg order from Broker B at a price of 34. Two private **8 (2)** messages are sent to Broker C and Broker B to notify this second execution and a public **1001 (MarketUpdate)** message is sent to the market to publish the trade and to update the limits. The order is now completely filled as reflected by the order status in the second execution report. A public **1002 (OrderUpdate)** message is sent to the market to remove the order from the order book.

2.3   Modifying an Order Modifying an Unmatched Order (with and without loss of priority)

① At 14:00, a Broker sends a private **D (D)** message to enter a new order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order to the book, with a rank number giving the order priority (rank number #1 in the example), and an OriginalOrderPriority equal to the trading engine order id (OrderID1).

Then a public **1001 (MarketUpdate)** message is sent to update the limit.

② At 14:05, the same Broker sends a private **G (G)** message to cancel/replace the order by decreasing the quantity to 8,000 shares. UTP sends back a private **8 (E)** message to confirm that the broker's request has been accepted and that the cancel/replace operation is pending.

UTP sends back another private **8 (5)** message to confirm that the cancel/replace operation has been performed. UTP sends a public **1002 (OrderUpdate)** message to the market to

     1.1.0

modify the previous order but without loss of order priority (still rank number #1), and an OriginalOrderPriority remaining equal to the trading engine order id (OrderID1).

A public **1001 (MarketUpdate)** message is sent to update the limit.

③ At 14:10, the same Broker sends a private **G (G)** message to cancel/replace the order by increasing the quantity to 12,000 shares. UTP sends back a private **8 (E)** message to confirm that the broker's request has been accepted and that the cancel/replace operation is pending.
UTP sends back another private **8 (5)** message to confirm that the cancel/replace operation has been performed. MDG sends a public **1002 (OrderUpdate)** message to the market to modify the previous order with loss of priority (for example four orders have been entered meanwhile, giving a new order priority with a rank number #6) and an OriginalOrderPriority remaining equals to the trading engine order id (OrderID1).

A public **1001 (MarketUpdate)** message is sent to update the limit.

2.3.2   Modifying a Partially Matched Order
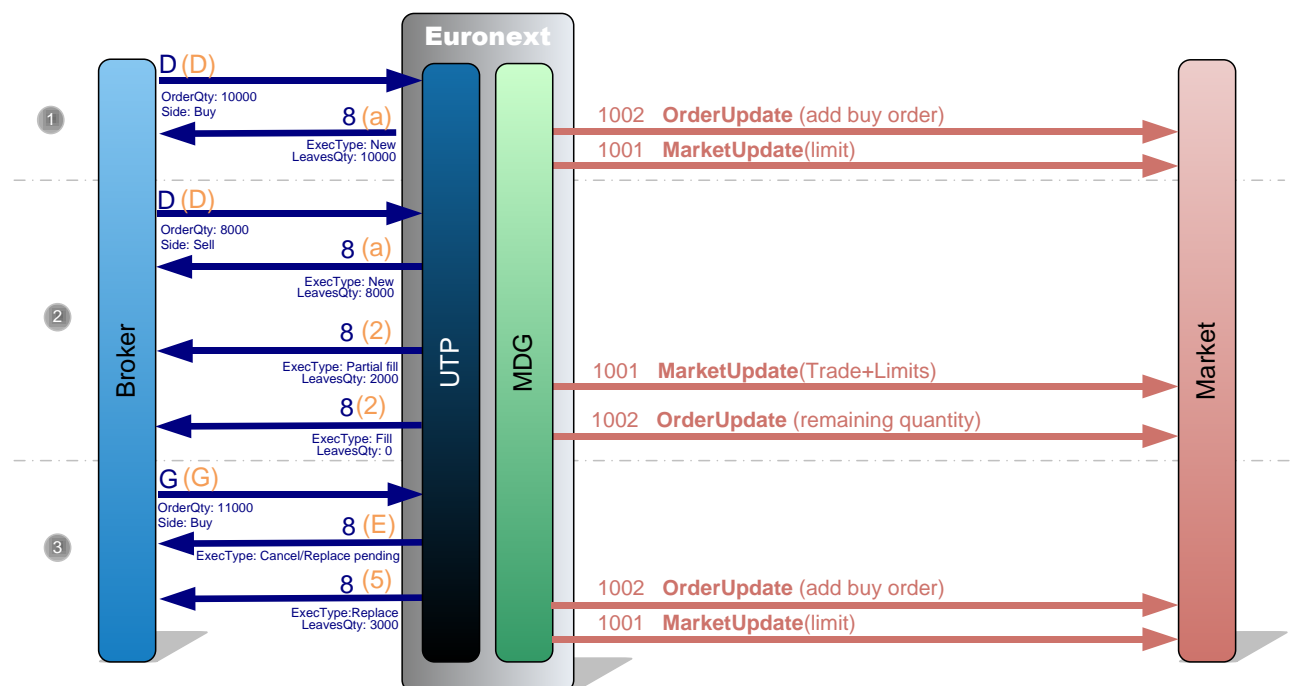
① A Broker sends a private **D (D)** message to enter a new order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

② Another Broker sends a private **D (D)** message to enter a matching Sell order with a quantity of 8,000 shares.

UTP sends back a first private **8 (a)** message to confirm reception of the order and then two more private **8 (2)** messages to indicate to each broker that the orders have been partially filled and filled.
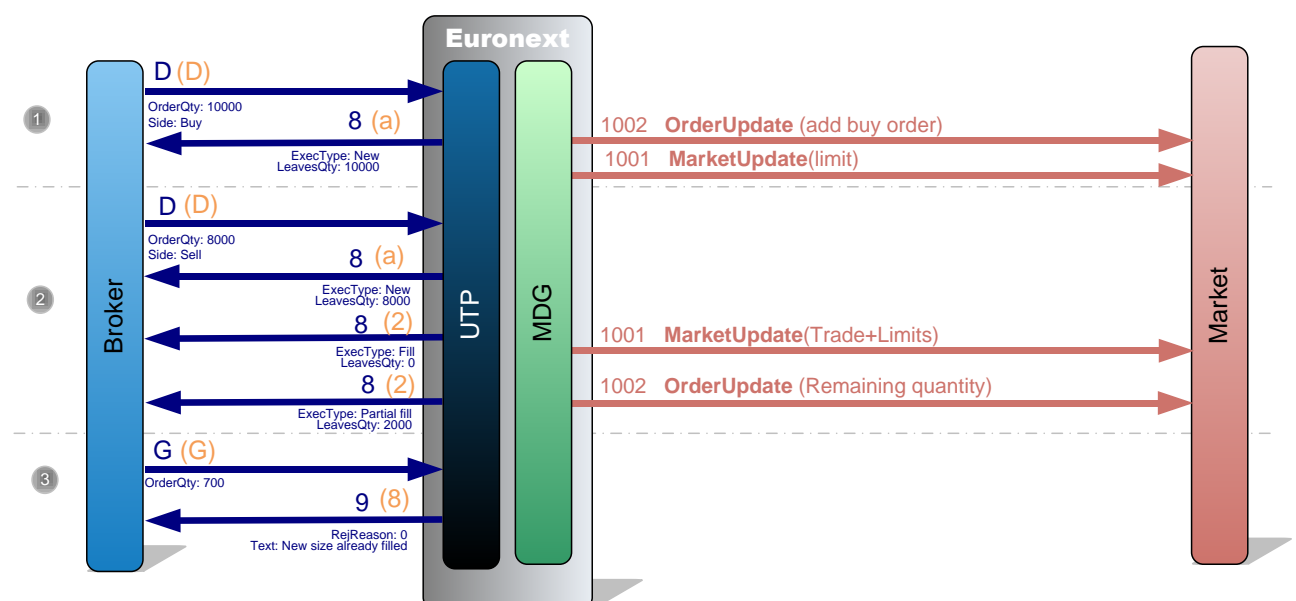
Consequently a public **1001 (MarketUpdate)** message is sent to the market for the trade and a message to update the limit, along with a public **1002 (OrderUpdate)** message to update the buy order (remaining quantity = 2,000) from the order book.

③ Later, the first Broker sends a private **G (G)** message to cancel/replace the quantity of the Buy order. As the modified quantity is to be equal to 3,000, the broker indicates a quantity of 11,000 (as 8,000 have already matched).

UTP sends back a private **8 (E)** message to confirm that the broker's request has been accepted and that the cancel/replace operation is pending. UTP then sends back another private **8 (5)** message to confirm that the cancel/replace operation has been successful.

The order with the increased quantity replaces the modified order in the order book and a public **1002 (OrderUpdate)** message to update the quantity (3000) along with a public **1001 (MarketUpdate)** to update the limit.

2.3.3   Rejected Modification

① A Broker sends a private **D (D)** message to enter a new order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The order enters in the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

② Another Broker sends a private **D (D)** message to enter a matching Sell order with a quantity of 8,000 shares.

UTP sends back a first private **8 (a)** message to confirm reception of the order and then two more private **8 (2)** messages indicating to each broker that the orders have been partially filled and filled.

Consequently a public **1001 (MarketUpdate)** message is sent to the market for the trade and a message to update the limit, along with a public **1002 (OrderUpdate)** message to update the Buy order (remaining quantity = 2,000) in the order book.

③ Later, the first Broker sends a private **G (G)** message to cancel/replace the quantity of the Buy order. The Broker indicates a quantity of 700.

UTP sends back a private **9 (8)** message to reject the cancel/replace operation as the quantity to be modified is no longer available. So the remaining quantity of 2,000 stays in the order book. No message is sent to the Market.

Note: If the broker attempts to change the quantity of an order to a value less than or equal to the quantity of the order already traded, the order modification will be rejected. In this example, new quantity=8,000 shares will be rejected.

2.4   Cancelling an Order

2.4.1   Cancelling an Unmatched Order
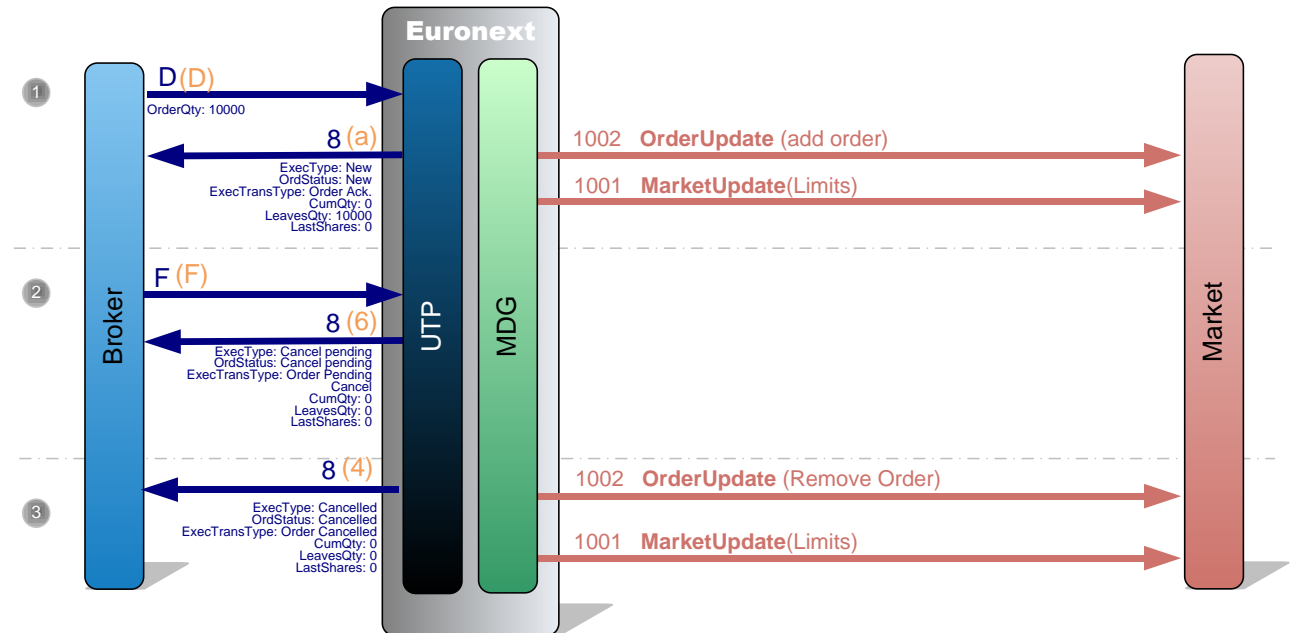
① The Broker sends a private **D (D)** message to enter a new order with a quantity of 10,000 shares. UTP sends back a private **8 (a)** message to confirm the order acknowledgement.

The order enters the order book without matching and a public **1002 (OrderUpdate)** message is sent to the market to add the order and a public **1001 (MarketUpdate)** message to update the limit.

② Later, the Broker sends a private **F (F)** message to cancel the previously entered order. UTP sends back a private **8 (6)** message to confirm that the broker's request has been accepted and that the cancel operation is pending.

③ UTP sends back a private **8 (4)** message to confirm that the order has been cancelled. Consequently MDG sends a public **1002 (OrderUpdate)** message to remove the order along with a public **1001 (MarketUpdate)** message to update the limit.

2.4.2   Attempt to Cancel an Already Matched Order
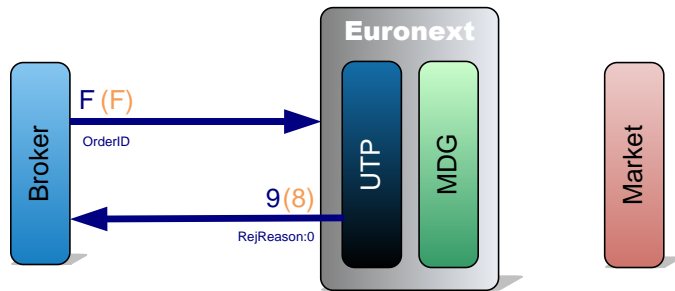
① The Broker sends a private **F (F)** message to cancel an order that has already matched.

② UTP sends back a private **9 (8)** message to reject the cancellation. No public message is disseminated.

2.4.3   Bulk Order Cancellation
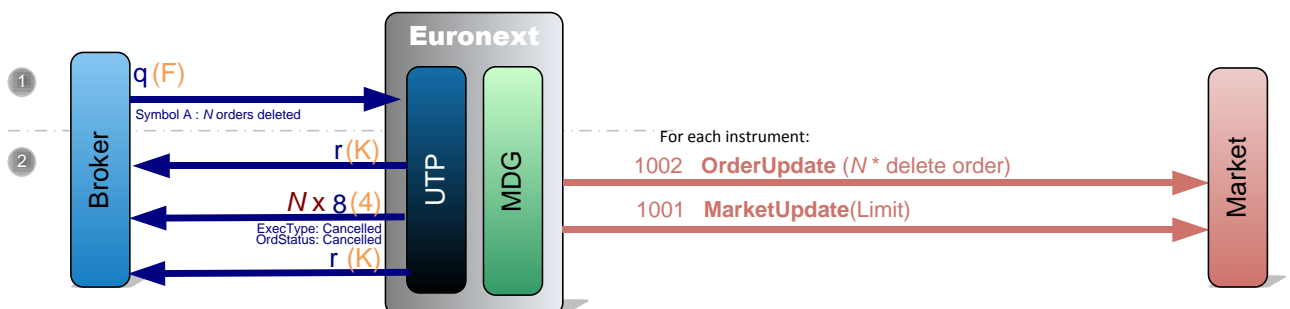
① The Broker sends a private **q (F)** message to cancel some of his orders matching specific criteria.

② UTP sends back a private bulk order cancellation acknowledgement **r (K)** message followed by a private **8 (4)** message for each cancelled order.

   For each instrument involved in the bulk order cancellation, a public **1002 (OrderUpdate)** is sent to the market containing all deleted orders for this instrument and a public **1001 (MarketUpdate)** to update the limit.

Note: A Bulk Order Cancellation is never rejected. When the Bulk Order contains an order that was already fully matched, the number of deleted orders in message r is not incremented.

2.5   Openings

2.5.1   Market To Limit on Opening
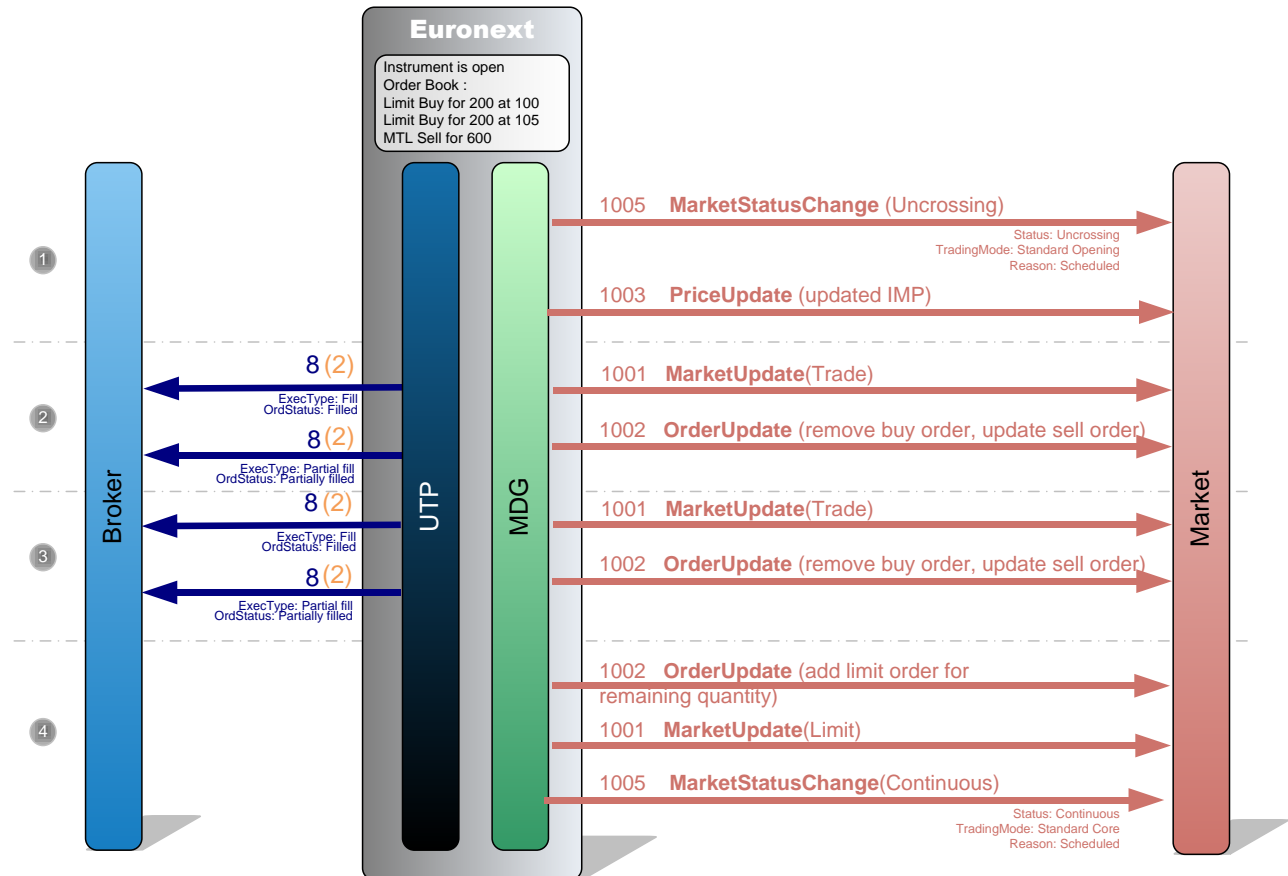
①   An auction occurs for the instrument. A first public message **1005 (MarketStatusChange)** message is sent to the market to notify the auction along with a public **1003 (PriceUpdate)** message to broadcast the Indicative Matching Price (IMP).

②   The Sell order matches the first Buy order for 200 and UTP sends back a private **8(2)** message to each member to notify the trade execution.

Consequently a public **1001 (MarketUpdate)** message is sent to the market for the trade and a public **1002 (OrderUpdate)** message to remove the first Buy order from the order book and update the quantity of the Sell order.

③   The Sell order matches again with the second Buy order for 200 and UTP sends back a private **8(2)** message to each member to notify the trade execution.

Consequently a public Trade (1004) message is sent to the market for the trade and a public **1002 (OrderUpdate)** message to remove the second Buy order from the order book and to update the quantity of the Sell order again.

④ A public **1002 (OrderUpdate)** message is sent to update the Sell order as a limit order with the remaining quantity (200) and price equal to the TOP (100). A private message **8(2)** is sent to notify the Broker, with Ack Type = MTL Second Ack, Order Price=100, Order Quantity=600 and Order Filled Quantity=400.

A public **1001 (MarketUpdate)** message is sent to the market to update the limit and a second public **1005 (MarketStatusChange)** message to indicate that the instrument is now open.

Note 1: There is only one public **1001 (MarketUpdate)** message to update the limits at the end of the uncrossing.

Note 2: For randomized auctions, the **TradingMode** value in the public **1005 (MarketStatusChange)** message is equal to 2 for "Random Uncrossing".

2.6   Indicative Price Inputs

Indicative price inputs include:

- Valuation Price.
- Alternative Indicative Price.

2.6.1   Automatic Valuation Price (ETF Kinematic)
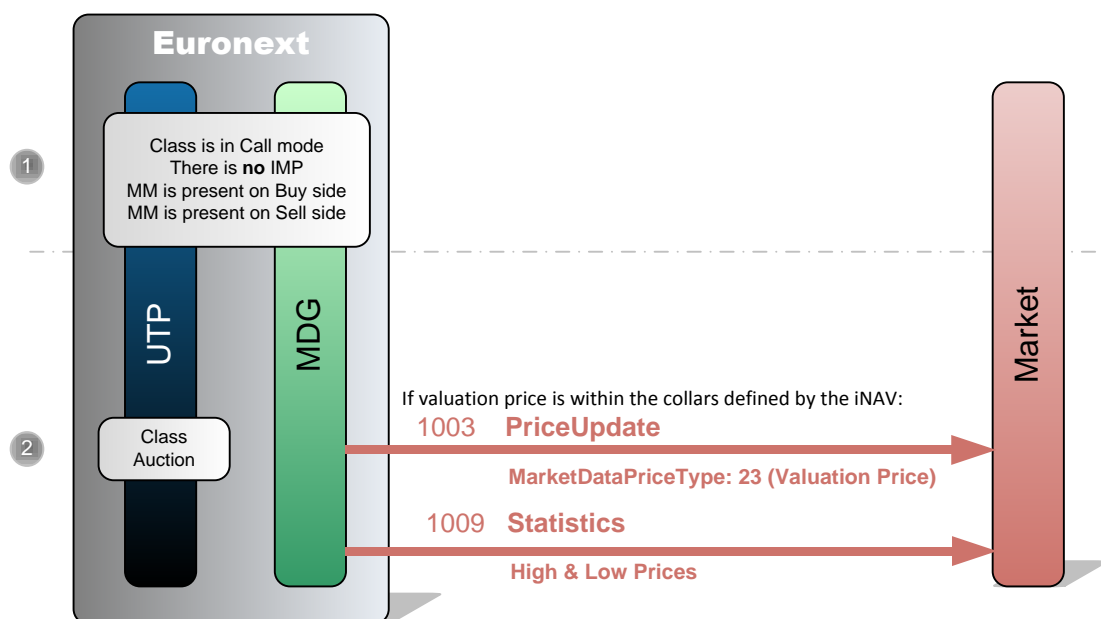
① Exchange Traded Funds (ETFs) instruments with no Auction price are automatically valuated by **UTP** at the Class Auction time and under the following Market conditions:

- There is no trade during the auction phase;
- At least one Market Maker order is present on the buy side of the Central Order Book;
- At least one Market Maker order is present on the sell side of the Central Order Book;
- The valuation price is within the collars defined by the iNAV.

② Twice a day at the end of opening and closing auction phase, a public **1003 (PriceUpdate)** message is sent to the Market to broadcast the Valuation Price, along with a public **1009 (Statistics)** message when applicable.

Note : A Valuation Price never impacts the cumulated volume of the Instrument.

### 2.6.2 Automatic Valuation Price (Quote Driven Warrant Market Model Kinematic)
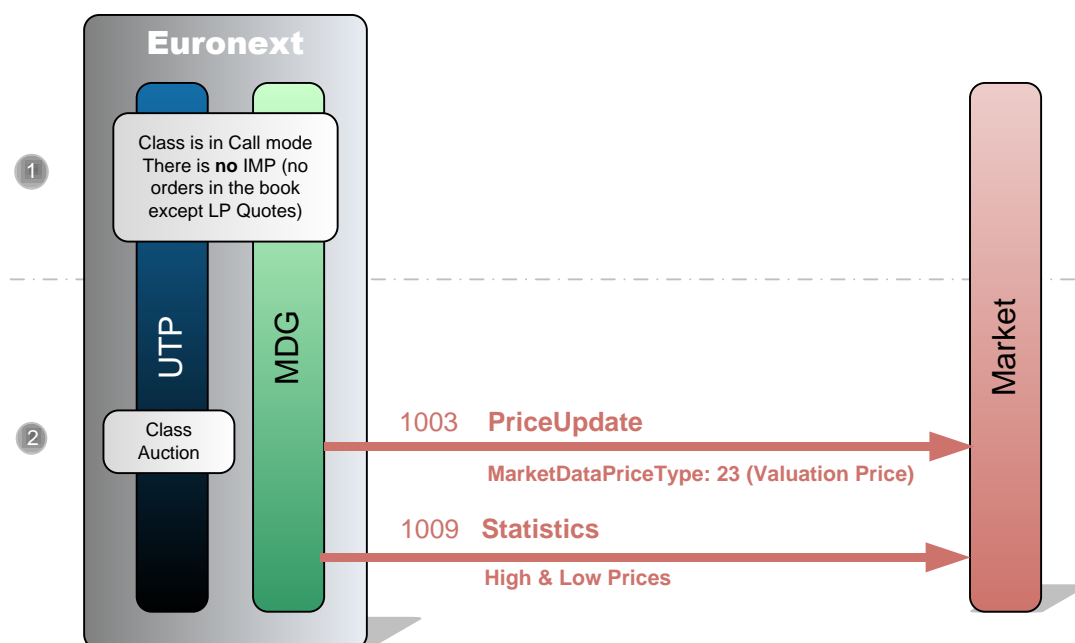
① Some specific illiquid Instruments can be parameterized to be automatically valuated by **UTP** at the Class Auction time and under certain Market conditions (instrument has no IMP at the opening, an instrument is in an inherited state after the opening, instrument is not in an Offer Only situation).

② At the Class auction time, a public **1003 (PriceUpdate)** message is sent to the Market to broadcast the Valuation Price which can take one of the following values:

- The BBO midpoint price if the LP was present on both sides;
- The Bid price if the instrument is in a Bid Only situation and a Bid LP quote is present;
- The Adjusted Closing Price if no LP is present in the order book.

A public **1009 (Statistics)** message is sent to the market to update High and Low Prices when applicable.

Note : A Valuation Price never impacts the cumulated volume of the Instrument.

2.6.3　AIP (Alternative Indicative Price)
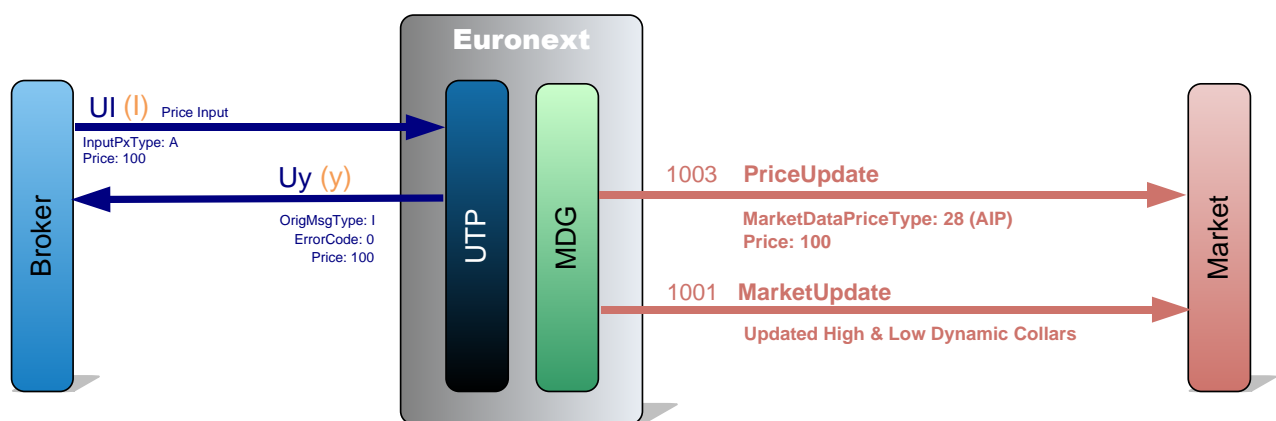
A price information message can be disseminated to Market Participants in order to valuate some illiquid bonds instruments through an Alternative Indicative Price (AIP).

The Broker sends private **UI (I)** message (with an AIP type).

The Alternative Indicative Price is then generated and disseminated to the Market participants through a public **1003 (PriceUpdate)** message with a price being equal to the price contained in the Price Input message.

The Dynamic Collars of the Instrument are updated around the price specified in the Price Input message and a public **1001 (MarketUpdate)** message is sent to the market to disseminate the new collars.

3. Unsollicited Messages

## 3.1 Asynchronous Messages

3.1.1 News and Announcements

The public news and announcements message **1010 (ExchangeAnnouncement)** is a message sent to inform markets participants about trading information like technical problems, regulatory decisions, etc. The contents of the message are purely informative and can be sent at any time during the day.

3.1.2 Statistics Message
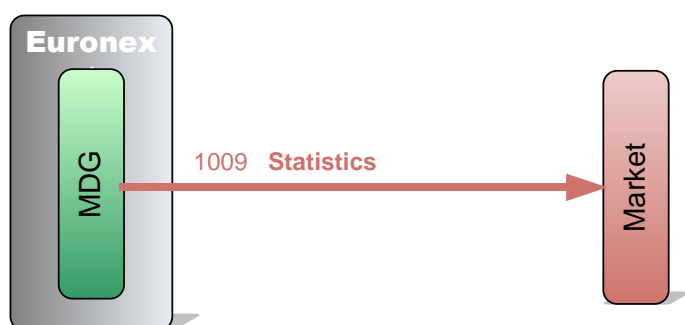
The public **1009 (Statistics)** message is periodically sent to the market only if the information needs to be updated. It includes minimum and maximum traded prices for daily, yearly and lifetime periods along with the cumulative volume since the start of the trading day and the percentage of variation of the traded price versus the last reference price.

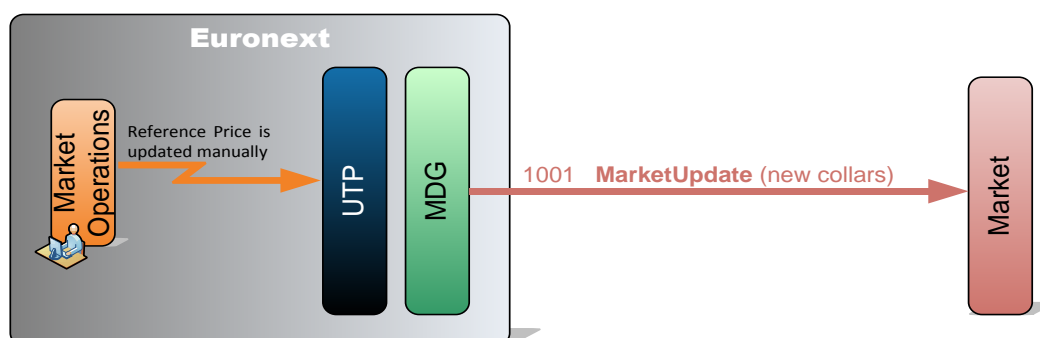   1.1.0

3.1.3   Automatic IMP Calculation
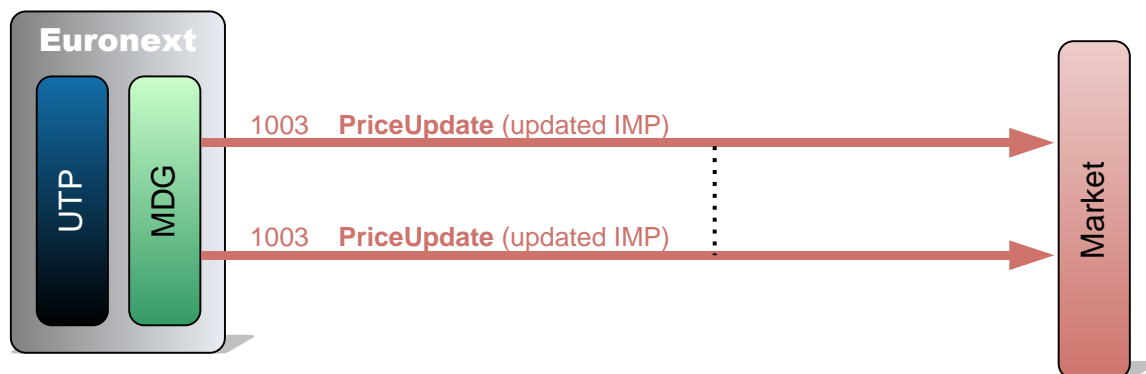
When an instrument is in Call phase, a public **1001 (MarketUpdate)** message is sent to the market each time there is a change in the Indicative Matching Price (IMP).

3.2   Actions Performed By Market Operations

### 3.2.1   Reference Price Update

A manual update by Market Operations of the reference price on a given instrument:

■ Dynamic Collar Reference Price modification by Market Operations,

■ Last Traded Price modification by Market Operations command (if the Dynamic Collar Reference Source is not "External").
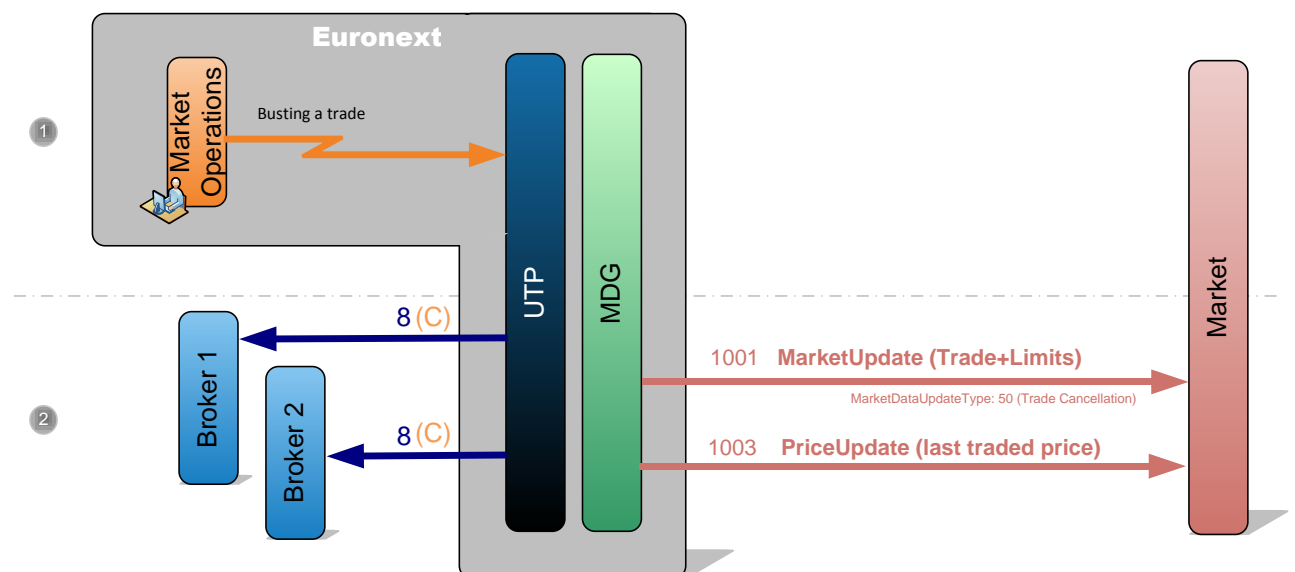
UTP sends a public **1001 (MarketUpdate)** to broadcast the new collars.

3.2.2 Bulk Order Cancellation
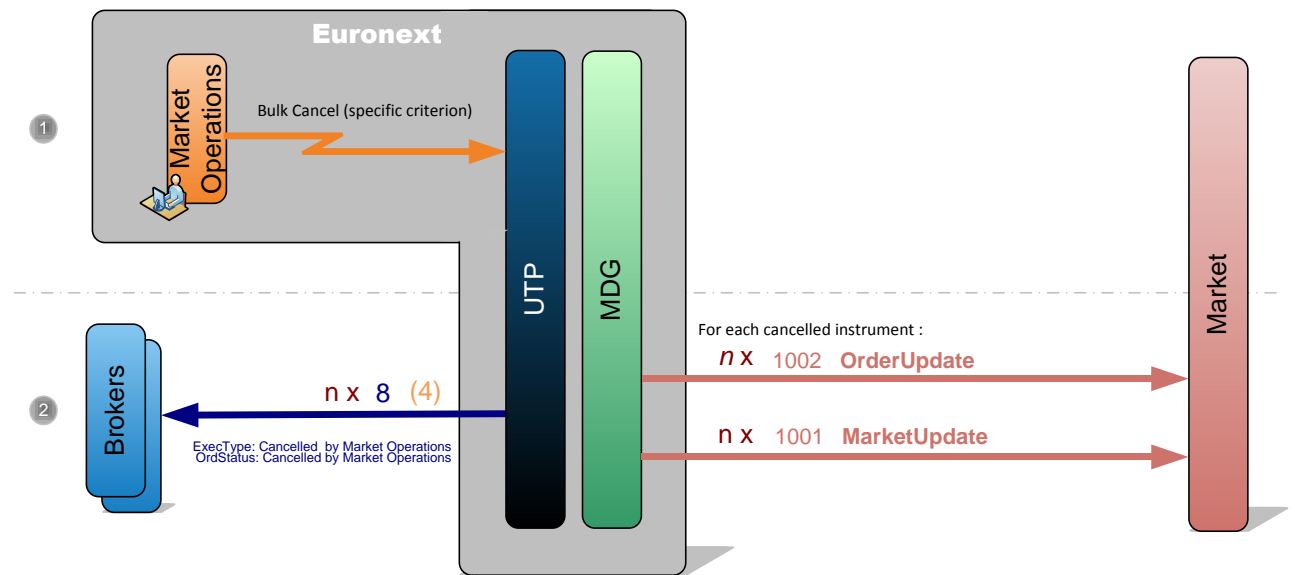
① Market Operations cancels orders matching specified criteria.

② UTP sends a private **8 (4)** message for each cancelled order to the member who entered the orders and all necessary public **1002 (OrderUpdate)** messages to remove the cancelled orders, along with a public **1001 (MarketUpdate)** message to update the limits.

3.2.3 Trade Cancellation

①   Market Operations cancels a trade on behalf of two members.

②   UTP sends a private **8 (C)** message to each member to indicate that the trade has been cancelled and a public **1001 (MarketUpdate)** message to market participants to update the limits, followed by a public **1003 (PriceUpdate)** message with the last traded price - if it is modified by the trade cancellation.

3.2.4   Resending Key Market Data
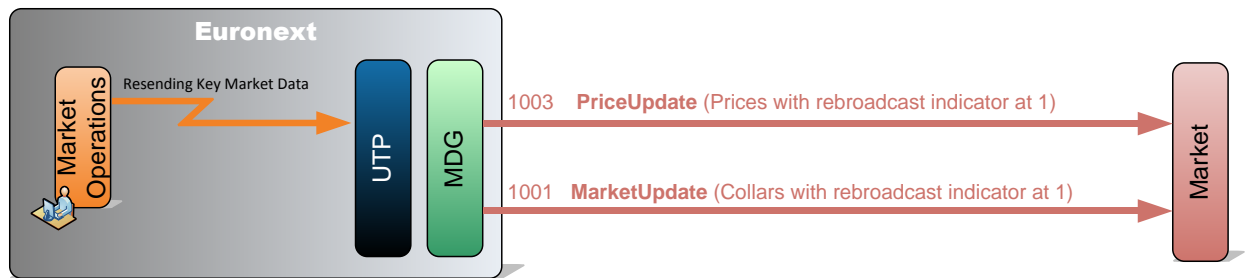
In the event of a technical failure leading to a loss of data (software or network problem), Market Operations has the possibility to resend key market data for several instruments.

<u>For each instrument</u> affected by the resending command, a public **1003 (PriceUpdate)** message is sent to disseminate the reference prices (YDSP Settlement Price, New Last Price, Last Trading Price), followed by a public **1001 (MarketUpdate)** message to update the collars.

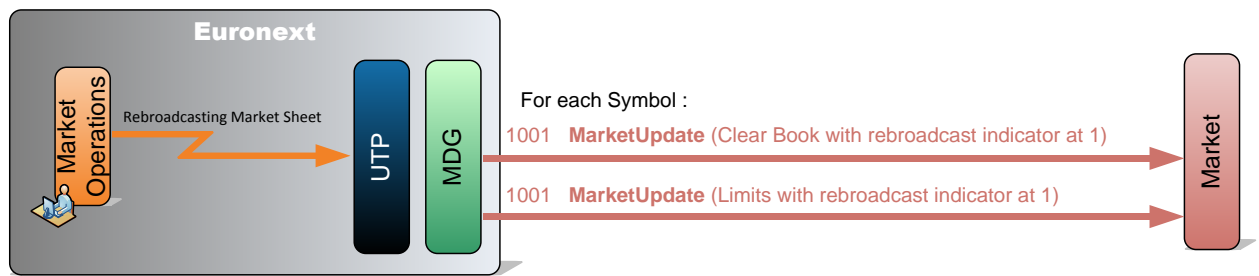### 3.2.5   Rebroadcasting Market Sheet
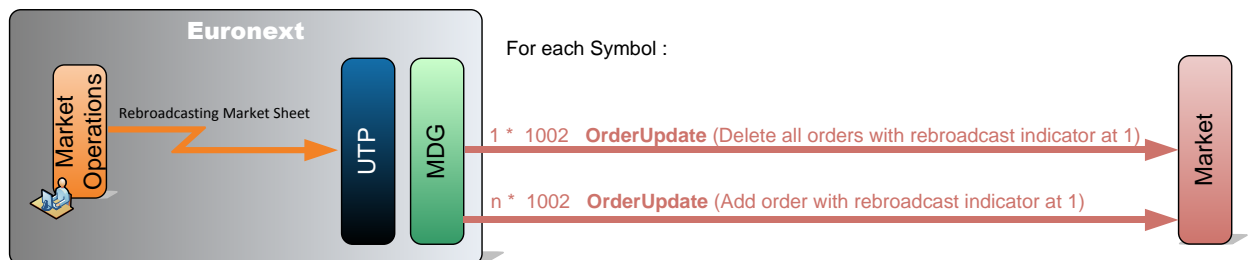
Market Operations has the possibility to rebroadcast the Market Sheet for several instruments.

<u>For each instrument</u> affected by the rebroadcasting command:

- a public **1002 (OrderUpdate)** message is sent to delete all the orders (with Market Data Action Type = 3);
- followed by at least one public **1002 (OrderUpdate)** message to resend all the orders (with Market Data Action Type = 5).

3.2.6 Rebroadcasting Market Sheet (Quote Driven Warrant Market Model Kinematics)

On the Quote Driven Warrants Market Model, when Market Operations rebroadcast the Market Sheet of several instruments, or for a whole class of instruments, for each instrument affected by the rebroadcasting command, first a public **1001 (MarketUpdate)** message is sent to clear the order book, followed by at least one public **1001 (MarketUpdate)** message to update the limits.

                             1.1.0

4.   Market Status Changes

This section is dedicated to all market publications that deal with changes to the Market Status on Euronext Cash markets, which are communicated via the Market Status Change (1005) message.

The Optiq MDG Market Status Change (1005) is common across all Euronext Markets, including Cash, Warrants and off-book data publication. The Market Status of an instrument can be determined using the following fields:

- **Instrument State**: Market State of the Contract
- **Status Reason**: Instrument State origin
- **Trading Mode**: Specifics during a trading phase that do not impact the Instrument State
- **Trading Period**: indicates the trading period
- **Trading Side**: indicates the side of a One-Side Only period for Quote Driven Warrant Market Model
- **Order Entry Qualifier**: Describes whether order entry is allowed for the instrument, and will depend on the Instrument State, Trading Mode and global availability
- **Scheduled Event**: Market Event notification
- **Scheduled Time**: Scheduled Event associated time if required

The possible Market Status values on Euronext Cash are as follows:
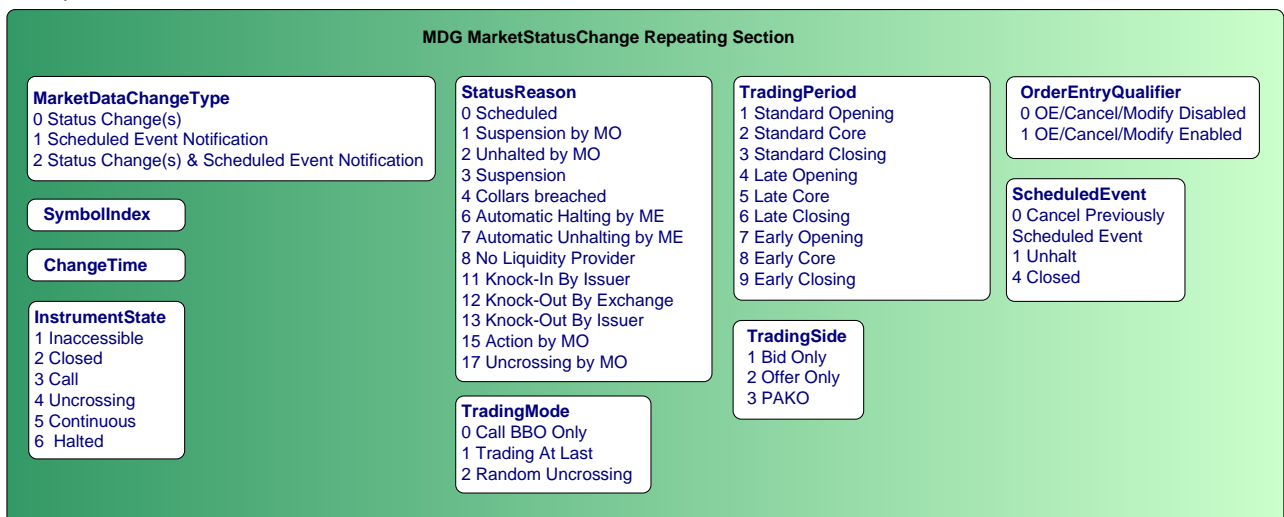
One of the main improvements of this message is that it provides the full state of the instrument. The updated instrument state values will appear in the updated fields, the unchanged values from the previous state will persist. Using this method, no interpretation is required as information is provided each time the message is sent. Note that all fields are always populated, except the Change Time, TradingSide, Scheduled Event and Schedule Event Time, as they are notifications.

In the following Market Status change example, an instrument is manually suspended by Market Operations with Order entry disabled:

1005 **MarketStatusChange**

InstrumentState: 6 (Halted)
StatusReason: 1 (Suspension)
TradingPeriod: Current Value
OrderEntryQualifier: 0 (OE/Cancel/Modification disabled)
ScheduledEvent: N/A

         1.1.0

4.1   Market Status Changes Due To Manual Intervention

4.1.1   Market Operations Toggle Order Entry for a Specific Class (Class Level)
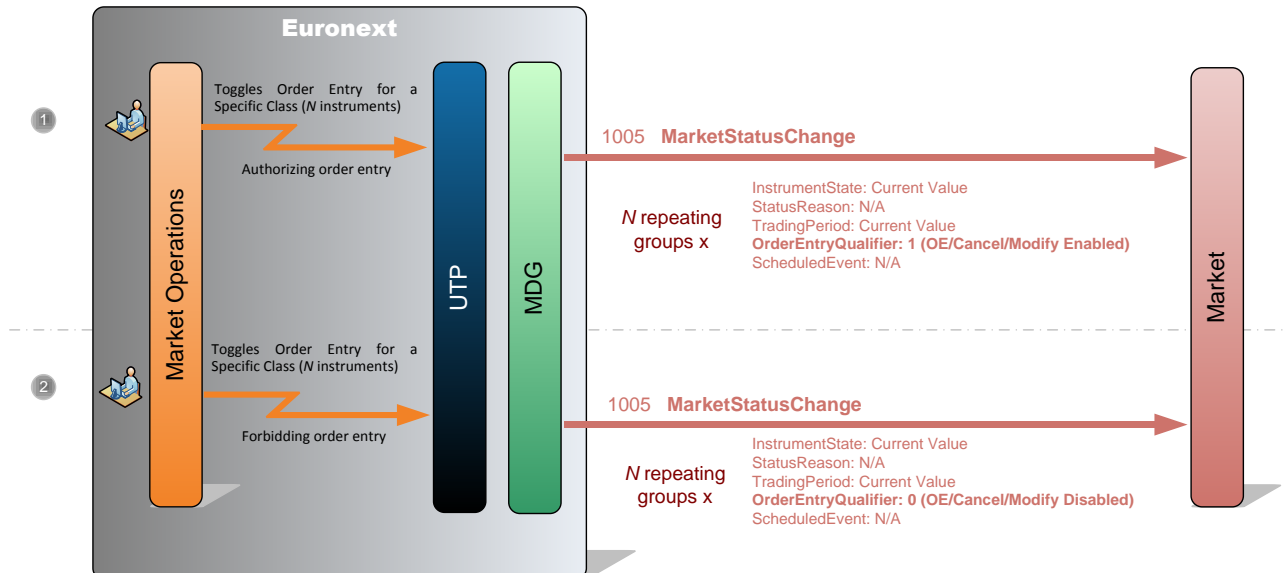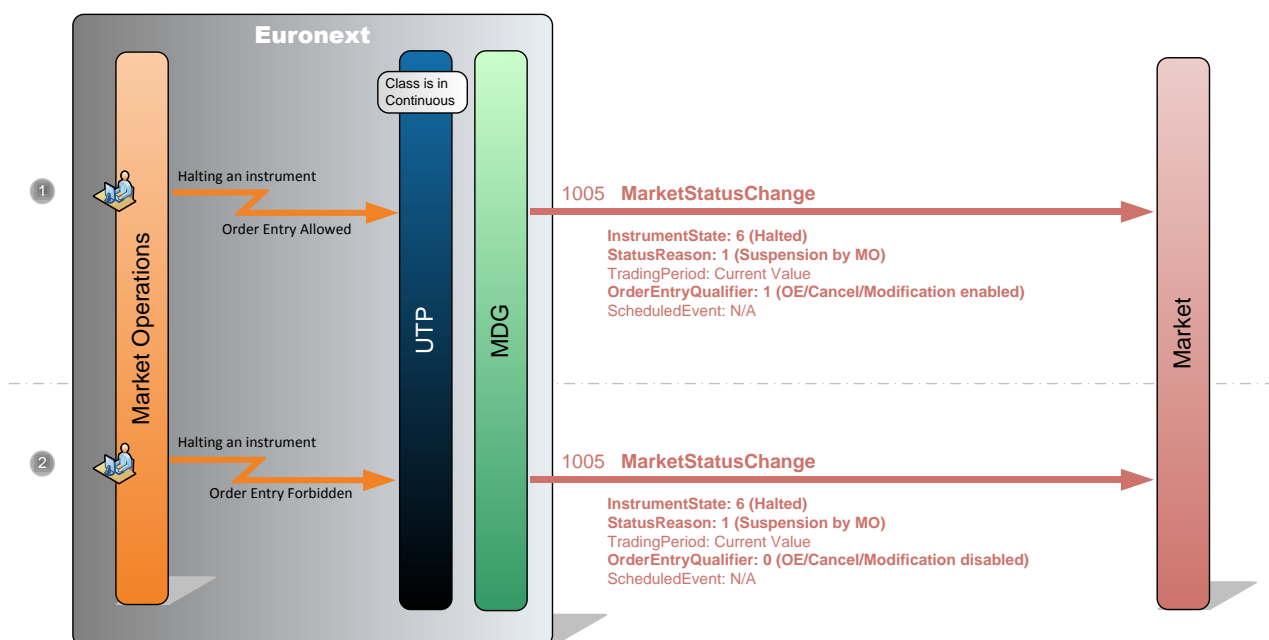

①   Market Operations toggles order entry authorization for a specific class of instruments. UTP sends **at least one*** public **1005 (MarketStatusChange)** message to the market to indicate order entry is authorized for all the instruments belonging to this class (*N* instruments).

*** At least** means that if there are more instruments in the class than the maximum number of repeating sections possible (i.e. 34), then another Market Status message is sent for the other instruments of the class. Each time the maximum number of repeating section is reached, another Market Status is disseminated.

②   Market Operations toggles order entry to disabled for a specific class.

UTP sends at least one public **1005 (MarketStatusChange)** message to the market to indicate order entry is disabled for all the instruments belonging to this class (*N* instruments).

4.1.2 Instrument Suspended by Market Operations (Order Entry Allowed or Forbidden) (Instrument Level)

①   Market Operations suspends an instrument with order entry enabled.

UTP sends a public **1005 (MarketStatusChange)** message to the market to indicate that the instrument has been suspended by Market Operations and order entry is authorized.

Note: if the class was in Call phase, then the IMP is still disseminated if any changes occur.

②   Market Operations suspends an instrument with order entry disabled.

UTP sends a public **1005 (MarketStatusChange)** message to the market to indicate that the instrument has been halted due to intervention by Market Operations and that order entry is disabled.

Note: The IMP is no longer disseminated if the class was in Call phase, and no trading is possible in Continuous phase.

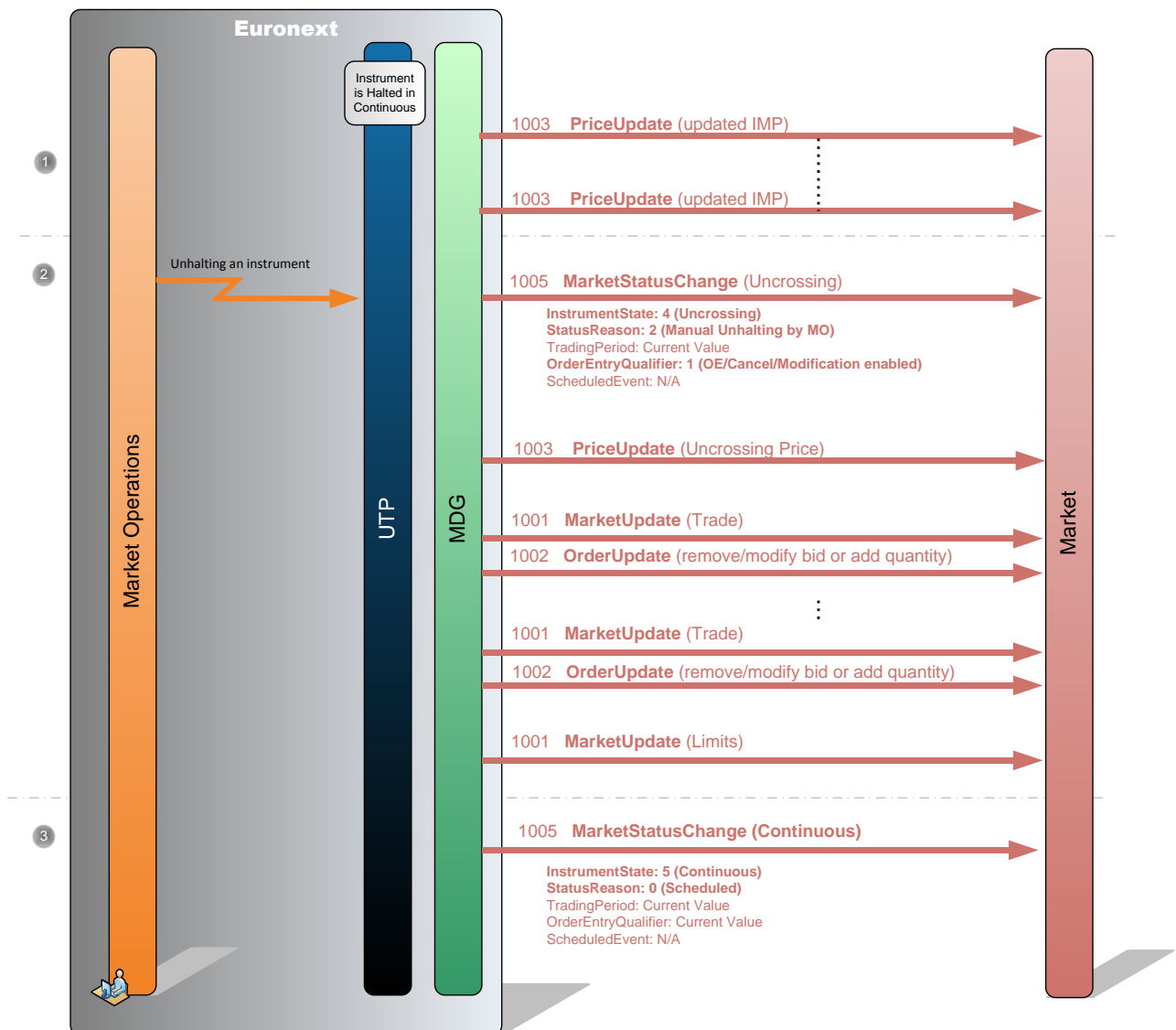### 4.1.3 Instrument Unhalted in Call phase (Instrument Level)
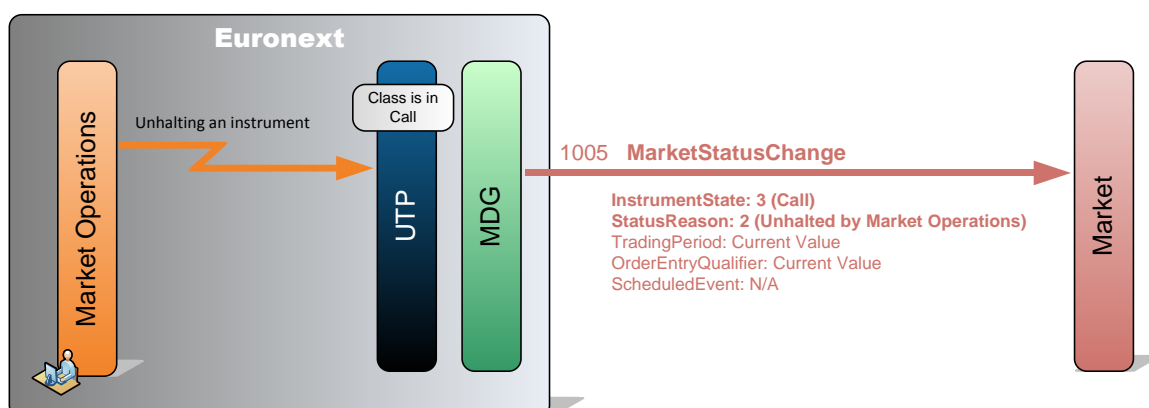
Market Operations unhalts a specific instrument at Call phase.

UTP sends a public **1005 (MarketStatusChange)** message to the market to indicate that the instrument has resumed the original scheduled phase (Call in this case).

Note: The IMP dissemination remains active regardless of the instrument's state during a Call phase.

4.1.4 Instrument Unhalted in Continuous phase (Instrument Level)

①   An instrument is halted in Continuous phase. A public **1003 (PriceUpdate)** message is sent to broadcast the Indicative Matching Price if available (and each time there is an IMP change).

②   The Market Operations unhalts the instrument. A first public message **1005 (MarketStatusChange)** message is sent to the market to notify the resumption of trading on the instrument along with a public **1003 (PriceUpdate)** message to broadcast the Uncrossing Price.

For each trade generated at the uncrossing, **UTP** sends a public **1001 (MarketUpdate)** message for the trade followed by a public **1002 (OrderUpdate)** message to update the limit and quantity from

the order book. At the end of trade matching, i.e. at the end of the uncrossing process, a last public **1001 (MarketUpdate)** message is send to update the values of each limit that has changed as a result of the uncrossing.

③   **UTP** sends a public **1005 (MarketStatusChange)** message to the market to indicate that the instrument has changed to the Continuous phase.


4.1.5   Beginning of a Bid Only Situation (Quote Driven Warrant Market Model Kinematic) (Instrument Level)
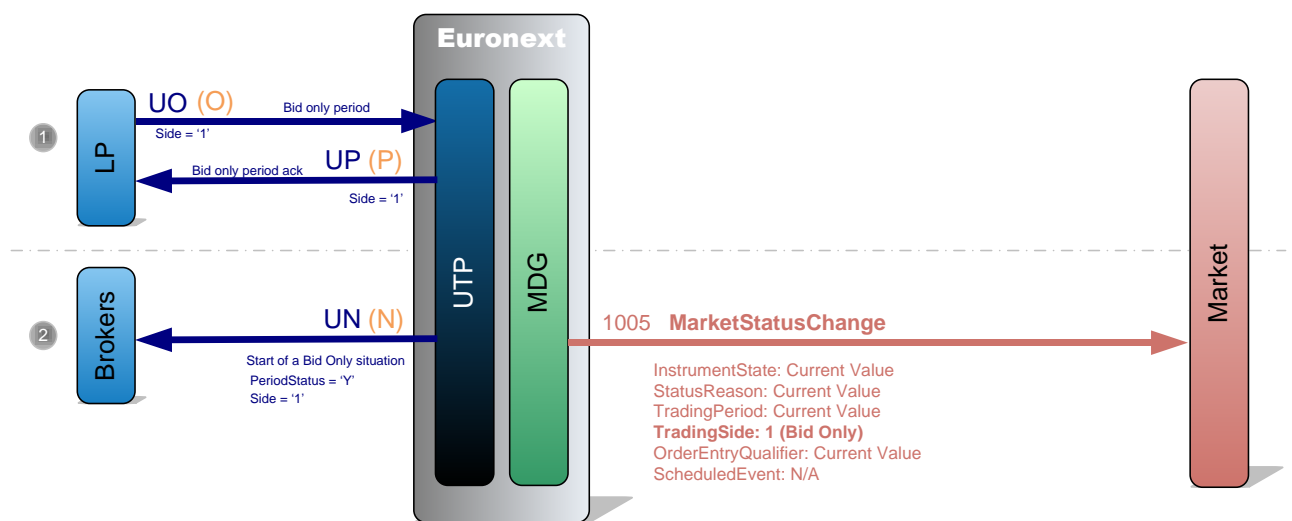
When a Liquidity Provider animating an instrument has no more instruments to sell, he moves to a Bid Only situation by entering the appropriate command. This command can be entered only if the Liquidity Provider has no more Quotes on the Instrument.

① The Liquidity Provider submits a private **One Side Only Period UO (O)** message to change the instrument state from normal to Bid Only. **UTP** sends back a private **UP (P)** message to the Liquidity Provider to acknowledge the one side only period.

② The information that a Bid Only Period has started is then broadcast to all Market Participants via a private **UN (N)** message and a public **1005 (MarketStatusChange)** message.

4.1.6   Beginning of an Offer Only Situation (Quote Driven Warrant Market Model Kinematic) (Instrument Level)
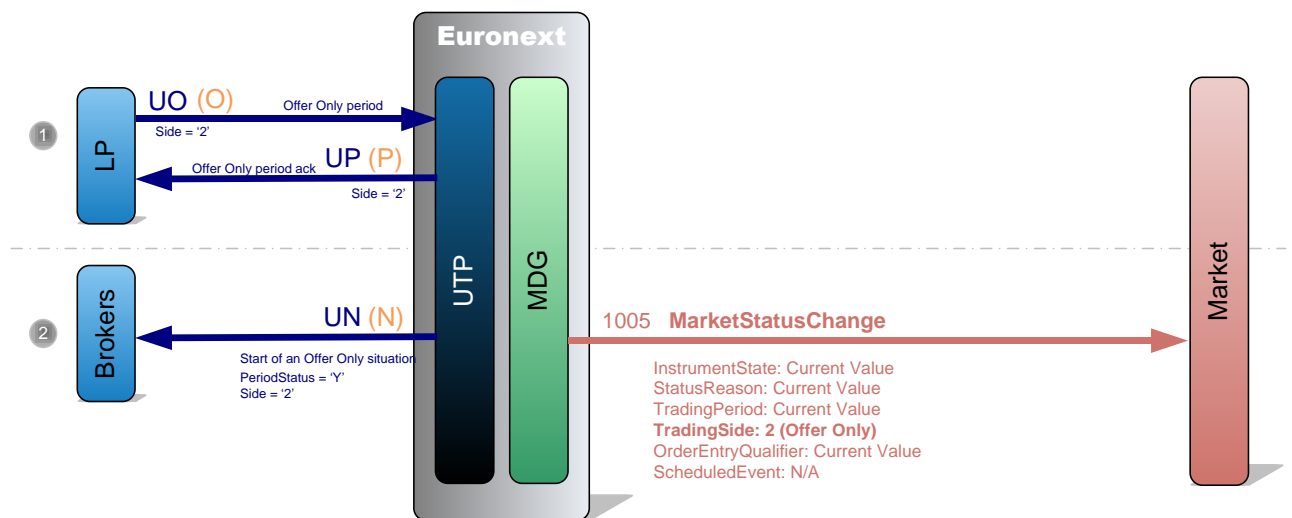
 When a Liquidity Provider animating an instrument values his product at a bid price lower than one tick, he moves to an Offer Only situation by entering the appropriate command. This command can be entered only if the Liquidity Provider has no more Quotes on the Instrument.

① The Liquidity Provider submits a private **One Side Only Period UO (O)** message to change the instrument situation from normal to Offer Only.

 UTP sends back a private **UP (P)** message to the Liquidity Provider to acknowledge the one side only period.

② The information that an Offer Only Period has started is then broadcast to all Market Participants via a private **UN (N)** message and a public **1005 (MarketStatusChange)** message.

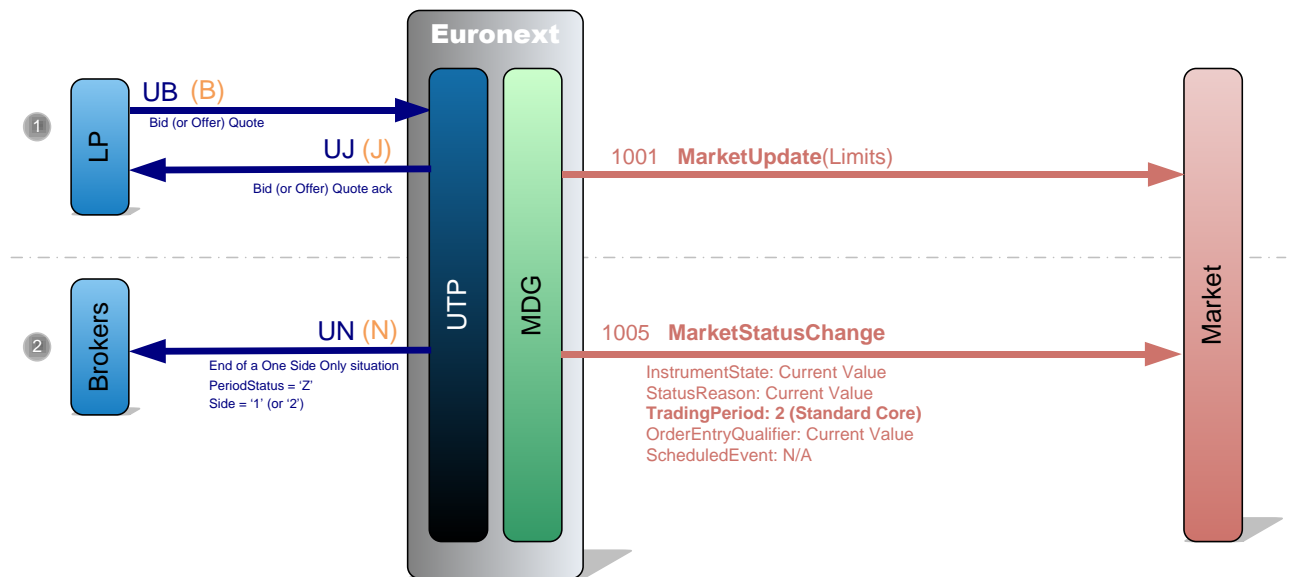### 4.1.7 End of a One Side Only Situation (Quote Driven Warrant Market Model Kinematic) (Instrument Level)

① An instrument is in a Bid Only (or Offer Only) state. The Liquidity Provider submits a private **UB (B)** Bid (or Offer) Quote message to end the Bid (or Offer) Only period. **UTP** sends back a private **UJ (J)** to acknowledge the LP Quote.

Updated quotes limits are disseminated to the market participants using a public **1001 (MarketUpdate)** message.

② **UTP** sends back a private **UN (N)** message to notify market participants that the Bid (or Offer) Only state has ended.

A public **1005 (MarketStatusChange)** message is sent to the Market.

4.1.8   Knock-In by Issuers (Quote Driven Warrant Market Model Kinematic) (Instrument Level)
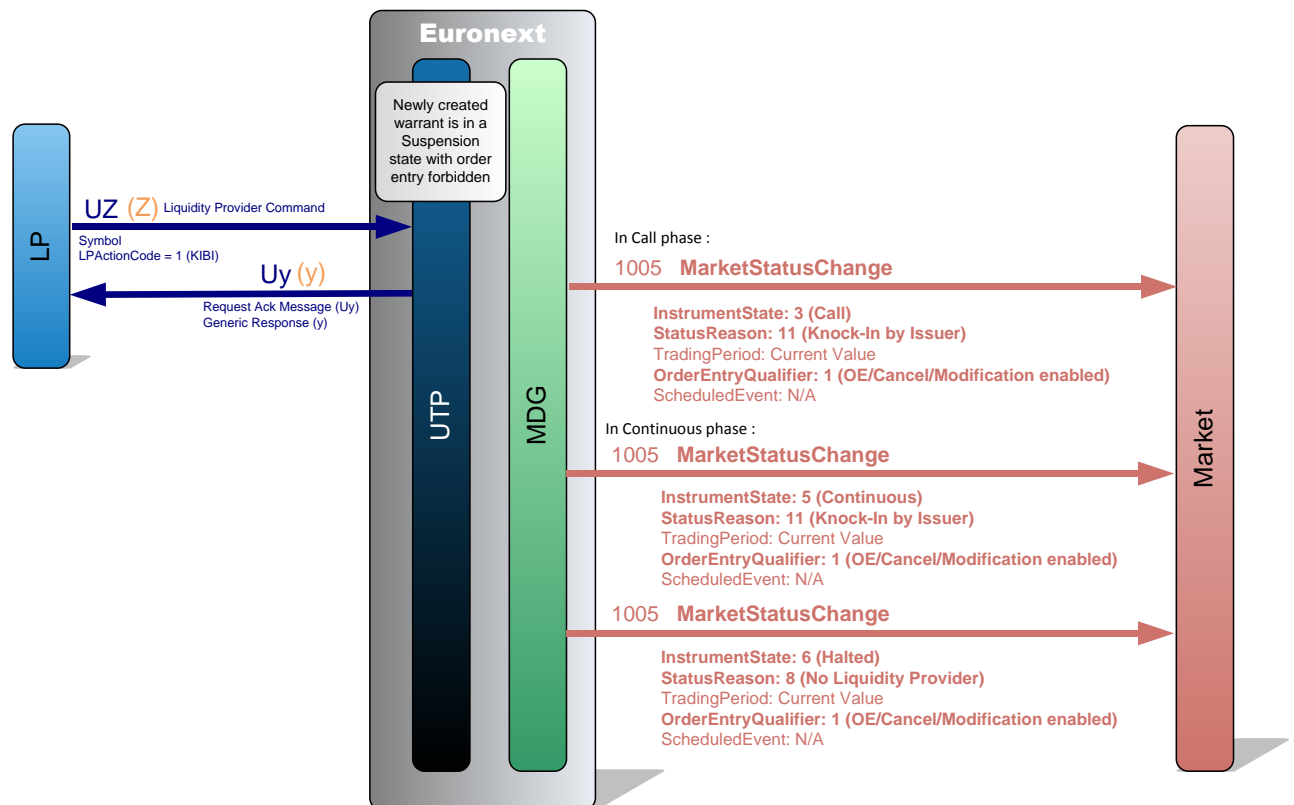
A Liquidity Provider has the ability to unhalt a newly created warrant instrument by sending a private **UZ (Z)** Liquidity Provider Command (Knock-In request). **UTP** sends back a private **Uy (y)** message to acknowledge the request.

If the command is sent at Call, a public **1005 (MarketStatusChange)** message is sent to inform the market that the warrant has been unhalted (with status reason "Knock-In by Issuer").

If the command is sent at Continuous, a first public **1005 (MarketStatusChange)** message is sent to inform the market that the warrant is reopened (with status reason "Knock-In by Issuer"), but since the order book is empty, it is immediately followed by a public **1005 (MarketStatusChange)** message to inform that the warrant is halted (with status reason "No Liquidity Provider").

4.1.9   Knock-Out by Issuers (Quote Driven Warrant Market Model Kinematic) (Instrument Level)
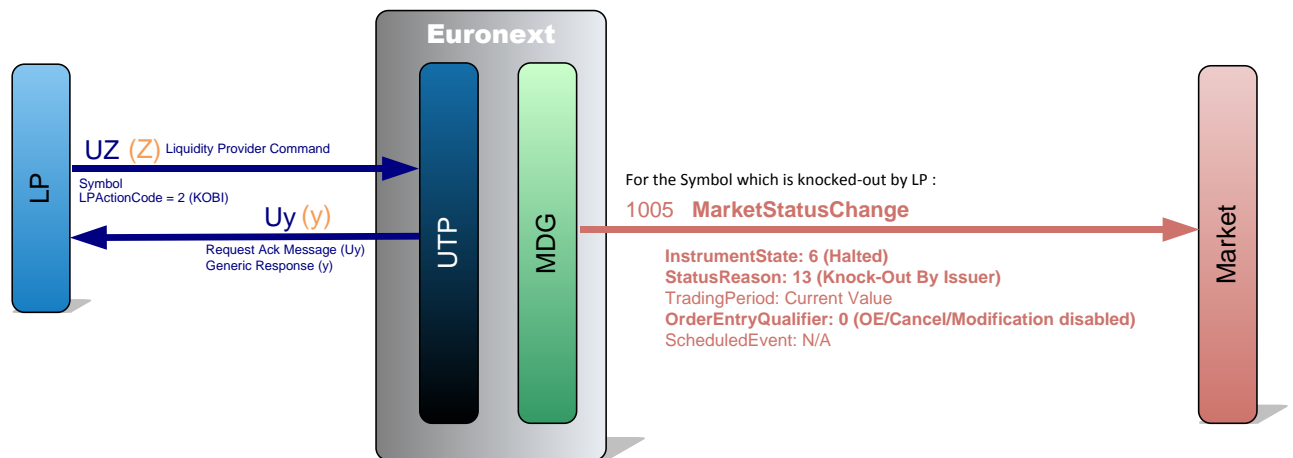
A Liquidity Provider has the ability to knock-out a warrant instrument.

Upon reception of a private **UZ (Z)** Liquidity Provider Command, **UTP** sends back a private **Uy (y)** message to acknowledge the request, and a public **1005 (MarketStatusChange)** message is sent to inform the market that the warrant has been suspended (Status Reason Knock-Out by Issuer).

4.1.10 Underlying Status Change (Warrant Kinematic) (Instrument Level)
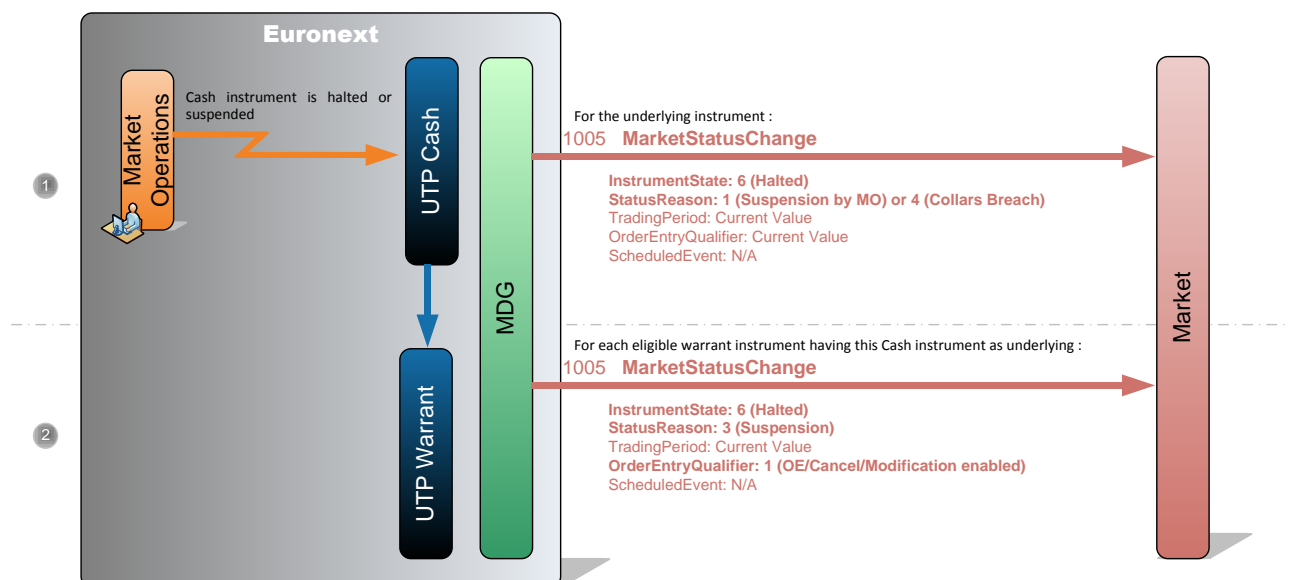
①   The underlying of a warrant is halted for collars breach (or suspended by Market Operations). UTP sends a public **1005 (MarketStatusChange)** message to indicate to the market that the Cash instrument has been halted (or suspended).

②   For each configured eligible warrant having this cash instrument as an underlying, **UTP** sends a public **1005 (MarketStatusChange)** message to inform the market that the warrant has been suspended.

Note:

*The Unhalting underlying kinematic is similar to the suspension scenario: the status of the underlying affects the status of the warrant, i.e. Instrument State returns to its original scheduled phase, the Status Reason is "Automatic Unhalting by Matching Engine" and order entry is enabled.*

Exception to the unhalting warrant kinematics: warrants that were suspended by Market Operations remain in that state.

4.1.11 Reactivating a Warrant (Quote Driven Warrant Market Model Kinematic) (Instrument Level)
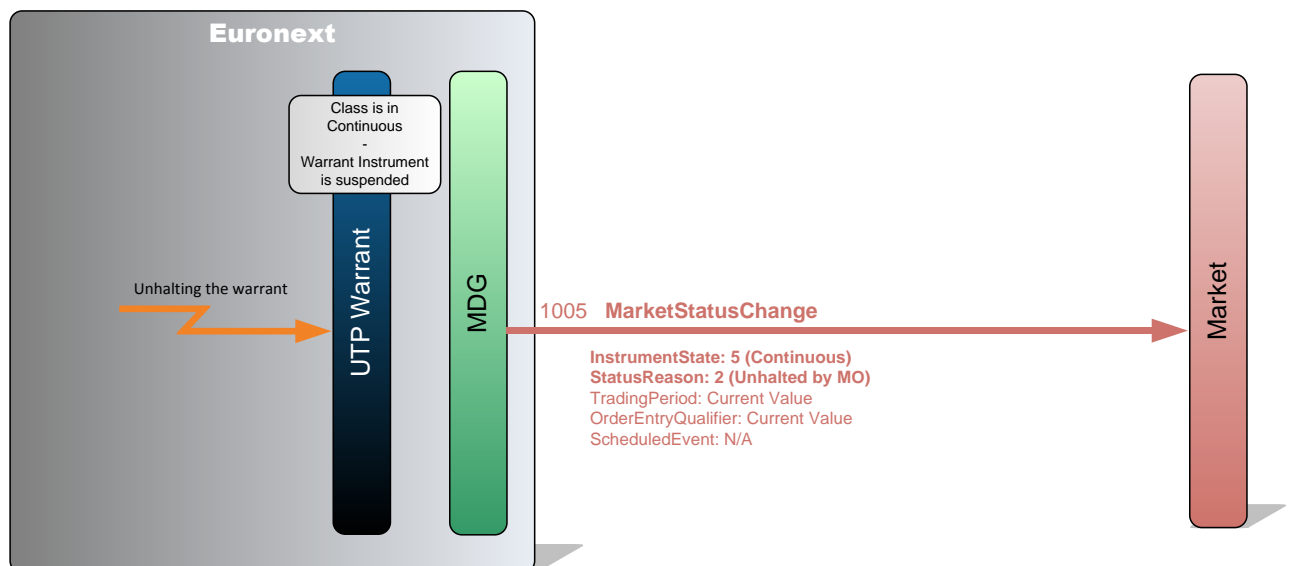
When a warrant erroneously reaches its deactivation barrier by the underlying instrument (eg, deactivation barrier incorrectly entered at the warrant creation for example), the warrant is automatically suspended with order entry disabled. The warrant must be reauthorized for trading.

Market Operations unhalts the warrant, authorizing it to trade again (according to the market phase). **UTP** sends a public **1005 (MarketStatusChange)** message to inform the market that the warrant has been unhalted by Market Operations.

4.2   Automatic Market Status Changes

4.2.1   Class Auction (Class Level)

Euronext

① Class is in Call mode

② Class Auction

For each halted instrument of the class because of static or dynamic collars breached:

1005   **MarketStatusChange**

InstrumentState: 6 (Halted)
StatusReason: 4 (Collars Breached)
TradingPeriod: 1 (Standard Opening)
OrderEntryQualifier: Current Value
ScheduledEvent: N/A

③ For each successful instrument uncrossing:

1005   **MarketStatusChange** (Uncrossing)

InstrumentState: 4 (Uncrossing)
StatusReason: 0 (Scheduled)
TradingPeriod: 1 (Standard Opening)
OrderEntryQualifier: Current Value
ScheduledEvent: N/A

1003   **PriceUpdate** (Uncrossing Price)

1001   **MarketUpdate** (Trade)
1002   **OrderUpdate** (remove/modify bid or sell quantity)

⋮

1001   **MarketUpdate** (Trade)
1002   **OrderUpdate** (remove/modify bid or sell quantity)

1001   **MarketUpdate** (Limits)

1005   **MarketStatusChange** (Continuous)

InstrumentState: 5 (Continuous)
StatusReason: 0 (Scheduled)
TradingPeriod: 2 (Standard Core)
OrderEntryQualifier: Current Value
ScheduledEvent: N/A

UTP

MDG

Market

① Class is in Call mode.

② At the Class Opening:

  ▪ each instrument for which the IMP is not within the static collars (IMP is greater than the High Static Collar for example) is automatically Halted and the Market is informed about this instrument state change via a public **1005 (MarketStatusChange)** message.
  It stays in a Halted status until intervention by Market Operations leading to a new reopening attempt. There is no automated unhalt scheduled on a static collar breach.

  ▪ each Blue Chip instrument for which the IMP is within the static collars but outside the dynamic collars is automatically Halted and the Market is informed about this instrument state change via a public **1005 (MarketStatusChange)** message.

③ At the Class Opening, each instrument for which the IMP lies within the static collars resume trading and the following sequence of messages are generated:

  ▪ A first public message **1005 (MarketStatusChange)** message is sent to the market to notify the resumption of trading on the instrument along with a public **1003 (PriceUpdate)** message to broadcast the Uncrossing Price.

  ▪ For each trade generated at the uncrossing, **UTP** sends a public **1001 (MarketUpdate)** message for the trade followed by a public **1002 (OrderUpdate)** message to update the limit and quantity in the order book.

  ▪ At the end of the uncrossing process, a finals public **1001 (MarketUpdate)** message is send to update the values of each limit that has changed as a result of the uncrossing.

  ▪ **UTP** sends a public **1005 (MarketStatusChange)** message to the market to indicate that the instrument has changed to the Continuous phase.

Note: In the case of a stock auction, the **TradingMode** value is '2' (Random Uncrossing).

### 4.2.2   Halting No LP / LP is back (Quote Driven Warrant Market Model kinematic) (Instrument Level)
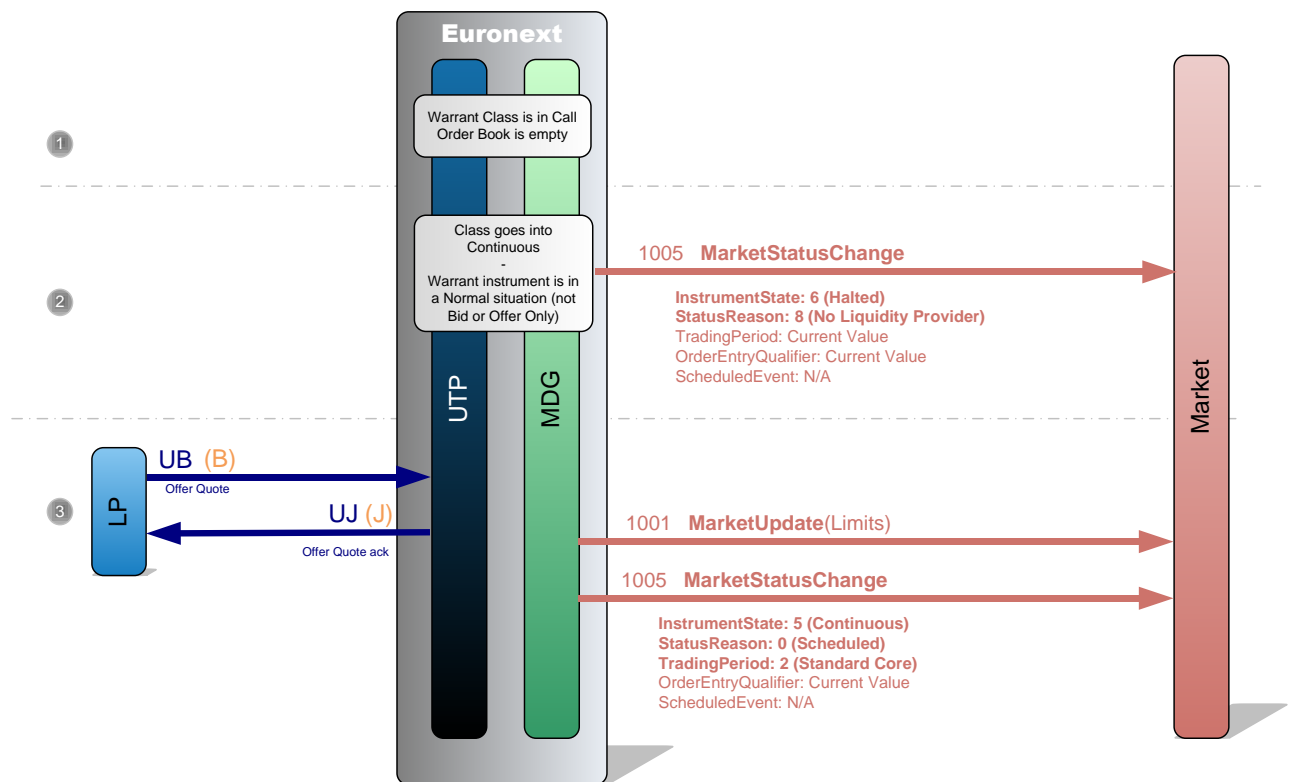
①   When the Liquidity Provider is not present in a warrant instrument that belongs to a Class in the Core Call phase, the instrument is **not** in the "No Liquidity Provider" state, but in a Scheduled Call state (Instrument State = '3' Call and Status Reason = '0' Scheduled).

②   When the LP is not present on the market in Continuous Trading phase, the instrument goes automatically to "No Liquidity Provider" status. A public **1005 (MarketStatusChange)** message is sent to inform the Market that the instrument status has changed.

③   The Liquidity Provider submits a private **UB (B)** Bulk Quote message to resume trading (and stop the AFQ process), **UTP** sends back a private **UJ (J)** to acknowledge the LP Quote and the instrument automatically resumes trading.

A public **1001 (MarketUpdate)** message is broadcast to market participants to update the limits.

A public **1005 (MarketStatusChange)** message is sent to inform the Market that the instrument resumes Continuous trading.

4.2.3   Automatic Knock-Out by Exchange (Quote Driven Warrant Market Model Kinematic) (Instrument Level)
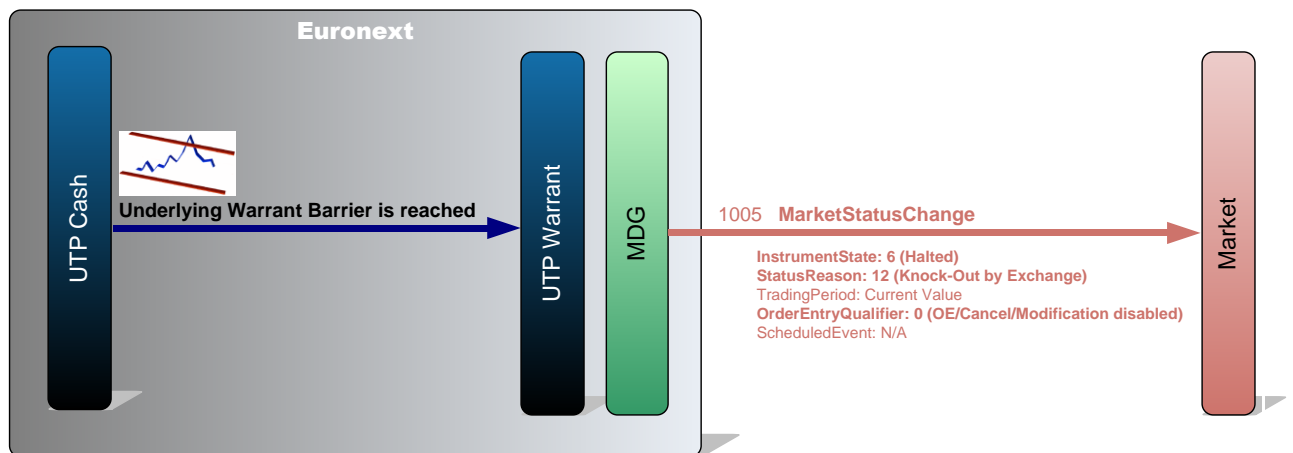
When an underlying price reaches a warrant deactivation barrier, UTP sends a public **1005 (MarketStatusChange)** message to inform the market that the warrant has been suspended (Status Reason "Knock-Out by Exchange").

Order entry is no longer possible.

4.2.4 Starting Payment After a Knock-Out Period (Quote Driven Warrant Market Model kinematic) (Instrument Level)
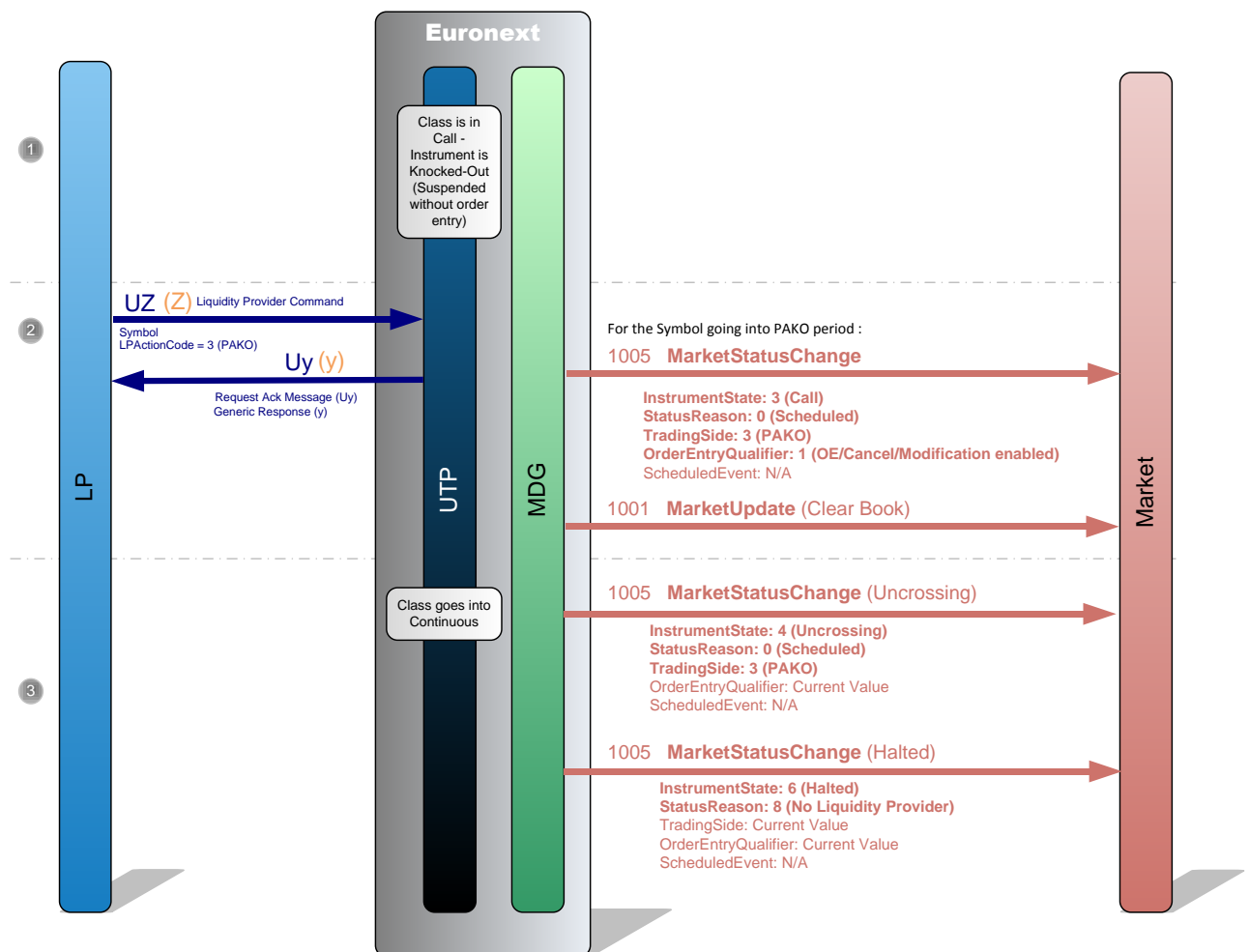
① An instrument has been knocked-out because trading on its underlying has breached the authorized trading barriers and is in a suspended state. The LP wants to start a Payment After a Knock-Out period.

② The LP wants to trade back the residual value of this instrument after knock-out (the tick size changes but no message is sent to the market). The LP then submits a private **UZ (Z)** Liquidity Provider Command. **UTP** acknowledges the request by sending back a private **Uy (y)** message. **UTP** sends a public **1005 (MarketStatusChange)** message to inform the market that the instrument has entered a PAKO (Payment after Knock-Out) period and a public **1001 (MarketUpdate)** message to clear the book.

③ The class to which the instrument belongs goes into an uncrossing state but immediately halts without the presence of the LP. A first public **1005 (MarketStatusChange)** message is sent to notify the uncrossing, followed immediately by a public **(MarketStatusChange)** message to inform that the instrument is halted with **StatusReason** '8' (No Liquidity Provider).

The kinematic describing the return of the Liquidity Provider to the market is explained in section 4.2.2 Halting No LP / LP is back (Quote Driven Warrant Market Model kinematic) (Instrument Level).

Note: During the PAKO period, the **TradingSide** stays at '3' (PAKO).

4.3   UTP Class Status versus Optiq Instrument Status

The mapping between previous UTP Class status and the new Optiq MDG instrument status is described in the following table:

UTP Class Status	Optiq^{TM} Instrument Status		
	Trading Period	**Instrument State**	**Trading Mode**
EAMO Early Monitoring	1 Standard Opening	2 Closed	NA
COCA Core Call	2 Standard Core	3 Call	0 Call BBO Only (if active)
COAU Core Auction	2 Standard Core	4 Uncrossing	2 Random Uncrossing (if active)
COCO Core Continuous	2 Standard Core	5 Continuous	NA
CLCA Closing Call	3 Standard Closing	3 Call	0 Call BBO Only (if active)
CLAU Closing Auction	3 Standard Closing	4 Uncrossing	2 Random Uncrossing (if active)
TAL Trading At Last	6 Late Closing	5 Continuous	1 Trading At Last
LAMO Late Monitoring	NA	2 Closed	NA
CLSD Closed	NA	1 Inaccessible	NA
HALT Halted	Current Mode	6 Halted	NA

APPENDIX A: DOCUMENT

DOCUMENT HISTORY

VERSION NO.	DATE	AUTHOR	CHANGE DESCRIPTION
1.0.0	12 Jul 2016	ITS Solutions - BA Team - FLO	Initial version