



Document title

CCG BINARY APPENDIX

Document type or subject

UNIVERSAL TRADING PLATFORM FOR EURONEXT CASH MARKETS

Revision number/ Version number

Revision Number: 3.2

Date

18 Apr 2016

Number of pages

23

Author

Euronext

This document is for information purposes only. The information and materials contained in this document are provided 'as is' and Euronext does not warrant the accuracy, adequacy or completeness and expressly disclaims liability for any errors or omissions. This document is not intended to be, and shall not constitute in any way a binding or legal agreement, or impose any legal obligation on Euronext. This document and any contents thereof, as well as any prior or subsequent information exchanged with Euronext in relation to the subject matter of this presentation, are confidential and are for the sole attention of the intended recipient. Except as described below, all proprietary rights and interest in or connected with this publication shall vest in Euronext. No part of it may be redistributed or reproduced without the prior written permission of Euronext. Portions of this presentation may contain materials or information copyrighted, trademarked or otherwise owned by a third party. No permission to use these third party materials should be inferred from this presentation.

Euronext refers to Euronext N.V. and its affiliates. Information regarding trademarks and intellectual property rights of Euronext is located at <https://www.euronext.com/terms-use>.

PREFACE

PURPOSE

This document provides additional information related to session management, order management, CCG high availability and Exchange business continuity.

WHAT'S NEW?

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

REVISION NO./ VERSION NO.	DATE	AUTHOR	CHANGE DESCRIPTION
1.0	14/03/2010		Initial document
2.0	06/06/2010		Add the BondMatch intermediated scheme
2.1	05/12/2011		Added new section Trading Technical Interruption Management, which provides guidelines regarding risk management principles in terms of detection and recovery measures available to clients to maintain their firm's order book in sync with the Exchange order book
2.2	18/09/2012		Rebranding
3.0	09/07/2015		Rebranding
3.1	09/03/2016		Rebranding
3.2	18/04/2016		Add a new section "Exchange Trading Engine failover overview" added. Add the available message type via Drop-Copy.

ASSOCIATED DOCUMENTS

The following associated documents should either be read in conjunction with this document, or provide other relevant information for the user (including the description of the fields and messages named in this document), they could be found on our [website](#) :

- UTP for European Cash Markets – CCG Binary Protocol Message Specifications
- UTP for BondMatch – CCG Binary Protocol Message Specifications.

CONTENTS

1. SESSION MANAGEMENT	4
1.1 Architecture	4
1.2 Logon Overview	4
1.3 Logon Handshake.....	4
2. ORDER MANAGEMENT.....	5
2.1 Introduction	5
2.1.1 Client Order Identifier	5
2.1.2 Firm identifier.....	5
2.1.3 Firm Access types	5
2.1.4 Firm Access Connections	5
2.1.5 Firm Trading Application Instances	6
2.1.6 Firm Access/Trading Applications/ Connections architecture overview	6
2.2 Client Order ID Management.....	6
2.2.1 ClOrdID uniqueness - format	6
2.2.2 Mandatory prefix: Service Bureau vs Regular Access.....	7
2.2.3 Recommended Instance prefix.....	7
2.2.4 Application Instance ClOrdID available range overview.....	7
2.2.5 Intermediated buy-side client on BondMatch.....	9
2.2.6 UTP-Direct vs UTP-FIX.4.2 compatibility.....	9
2.2.7 Drop-copy process.....	9
2.3 Kinematics.....	11
2.3.1 Creating an Order	11
2.3.2 Replacing an Order	11
2.3.3 Confirming an Order	13
2.3.4 Cancelling an Order	14
2.3.5 Cancelling Several Orders.....	15
2.4 Order Book Retransmission	16
3. CCG BINARY HIGH AVAILABILITY.....	17
3.1 failover implementation overview.....	17
3.2 Customer recovery method	18
3.3 CCG binary recovery protocol kinematics.....	19
4. EXCHANGE BUSINESS CONTINUITY	20
4.1 Exchange Secondary DC Connectivity	20
4.2 Firm Order Book Integrity	20
5. EXCHANGE TRADING ENGINE FAILOVER OVERVIEW	21
5.1 Standard failover.....	21
5.2 Non-standard failover.....	21
5.3 EXCHANGE TRADING ENGINE Failover detection	22
5.4 Mitigate the risk of an EXCHANGE TRADING ENGINE failover.....	22
6. TRADING TECHNICAL INTERRUPTION MANAGEMENT	23
6.1 Technical Trading interruption risk management overview.....	23
6.2 Recommended method for recovery of Affected orders	23

1. SESSION MANAGEMENT

1.1 ARCHITECTURE

Each client connection sends messages to its assigned IP and port. The CCG Binary Gateway then routes the messages to the appropriate trading platform.



1.2 LOGON OVERVIEW

Clients initiate a TCP/IP session to the UTP-Direct Server, and then initiate a logon. Session Logon is always initiated by the client. Application messages may be exchanged between the client and server after logon is successful. A client has five seconds after they connect to send a logon request, or the server drops the connection.

Messages have a defined number of fixed-length fields, containing both binary and ASCII data. All binary data is sent in network Endian format (Big Endian). All ASCII data is left justified and null padded. Each message contains a message termination field (ETX) that must contain an ASCII Line Feed character '\n' (0x0A ASCII). An empty field must be filled with nulls.

During periods of inactivity, the server and/or client use the Test Request and Heartbeat messages to ensure the connection is up and functioning properly. The client must be able to respond to Test Request messages from the server by sending a Heartbeat message.

1.3 LOGON HANDSHAKE

The UTP-Direct session layer uses sequence numbers to guarantee no message loss. Clients assign sequence numbers to the messages they send the server and the server tracks these numbers. Similarly, the server assigns its own sequence numbers to the order responses that it sends to the client. When clients log on after a disconnection, the Logon message allows the client and server to exchange the sequence number of the last message that they processed from the other party. Each side can then start sending the next message that has not been processed by the other side.

Clients can also choose to have UTP-Direct cancel all remaining orders when a disconnection happens.

2. ORDER MANAGEMENT

2.1 INTRODUCTION

Several entities and concepts are involved in the sending of orders from clients to trading platforms, and in the identification of these orders. They are introduced below.

2.1.1 Client Order Identifier

Each order sent by a client has a Client Order ID, which is the reference identifier for this order. This information is contained in the **ClOrdID** field in the messages sent or received by the client or the trading platform.

2.1.2 Firm identifier

Each client has a Firm ID, which is the reference identifier for the client, “Firm” being the term used for “Client of Exchange”. This information is contained in the **OnBehalfOfCompID** (in Client requests messages) and **DeliverToCompID** (in Trading Platform responses messages) fields in the messages sent or received by the client or the trading platform.

2.1.3 Firm Access types

A Firm Access is an entity allowing the Firm to access the Trading Platform. The two Firm Access types, which can both be used by a given Firm, are **Regular Access** and **Service Bureau Access**, as described below:

- **Regular Access:** when a firm contracts its own and exclusive order entry access means directly with Euronext, the Firm Trading Solution type is Regular Access (or sometimes Direct Access).
- **Service Bureau Access:** when a third-party customer, also called a Service Bureau, contracts order entry access means with Euronext to act as an order carrier on behalf of several firms, the Firm Trading Solution type is Service Bureau Access. On the BondMatch platform, the intermediated buy-side client will also use the **Service Bureau Access** with a specific configuration for the **ClOrdID**.

Note: The term “Firm Trading Solution type” can also be used instead of Firm Access type. The term “Direct access” can also be used instead of “Regular Access”.

Important note: There are two types of Service Bureau, depending on whether they generate ClOrdID, or simply transmit the ClOrdID values directly generated by the Firm on behalf of which the Service Bureau sends messages. If a Service Bureau does not generate ClOrdIDs but simply transmits them, it is called a “Pass Through” Service Bureau. It can be assimilated to a **Regular Access** Trading Solution type, and must therefore feature the Regular Access ClOrdID format presented below. The firms on behalf of which a “Pass Through” Service Bureau sends messages are required to generate ClOrdIDs in the same manner as for Regular Access, as described below.

2.1.4 Firm Access Connections

A Firm Access can establish connections to the Trading Platform by logging onto CCG Binary gateways. These Firm Access connections are established by applications, also called Firm Trading Applications, as described in the next section, each of which can establish several connections.

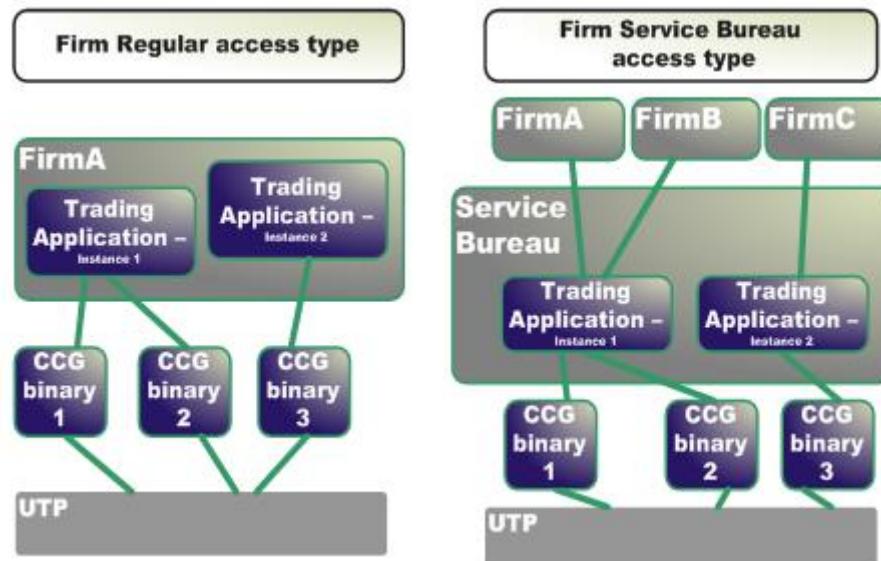
2.1.5 Firm Trading Application Instances

A given Firm can have several Firm Trading Application Instances. A Firm Trading Application is an application allowing a Firm to trade (i.e. to send orders, to receive trade notices, etc.). Firm Trading Application instances may be independent from each other, and each one:

- can use different software solutions (“In-House” or “Software vendor”);
- manages its own pool of CCG connections;
- can run in different branches of the same firm.

2.1.6 Firm Access/Trading Applications/ Connections architecture overview

An overview of the architecture of the set of entities presented in previous sections is given below, according to the Firm Access type.



2.2 CLIENT ORDER ID MANAGEMENT

2.2.1 ClOrdID uniqueness - format

The **ClOrdID** value assigned to any given order must be unique to the Firm across all available connections for the current trading day. This constraint also applies to any orders placed on a previous trading day that still remain on the order book. As an example, the **ClOrdID** assigned to a GTC or GTD order not fully executed on the day of its entry and not yet cancelled, cannot be used for a new order placed on a subsequent day.

Remark: This applies to the following requests: **New Order (D)** and **Cancel/Replace Order (G)** where **ClOrdID** is used as an order reference identifier. However, an **Order Cancel Request (F)** does not need to be unique, as in this case ClOrdID is used as a reference for the request and is optional.

To ensure that there is no conflict between Regular Access and Service Bureau Access for a given Firm, Euronext has a "**mandatory prefix policy**" for Service Bureau Access (see Section **Mandatory prefix: Service Bureau vs Regular** Access for details).

Euronext also highly recommends an optional "**instance prefix policy**" for all Firms (regardless of the access type used), to ensure that there is no conflict between multiple Trading Solution or Connection instances within the Firm (see **Recommended Instance** prefix for details).

The **ClOrdID** length is 8 bytes signed. Its decimal representation (base 10) is therefore 19 digits. The **ClOrdID** decimal representation (base 10) is required to encode prefix policy. Please refer to the sections that follow.

Note: **ClOrdID** format also applies for Warrant bulk quotes. However, unlike orders for which retransmitted orders will be rejected, duplicates will not be checked by Euronext for retransmitted bulk quotes. Therefore, customer connections must not retransmit any bulk quotes following a disconnection. Please refer also to the CCG HA availability section.

2.2.2 Mandatory prefix: Service Bureau vs Regular Access

Clients are required to implement available ranges according to the following general rules:

- **Regular Access** must specify a **ClOrdID** value in the positive number range.
- **Service Bureau Access:**
 - Must specify a **ClOrdID** value in the negative number range.

AND

 - Must start all **ClOrdID** values with the unique 3-digit number prefix assigned to the Service Bureau by the exchange.
- Intermediated Buy-side client:
 - Must start all **ClOrdID** values with the '-' character.
 - The next three characters must be populated with the unique 3-digit number assigned to the Service Bureau by the exchange.

AND

 - A specific value will be given on the BondMatch platform for intermediated buy-side clients (see 2.2.5).

2.2.3 Recommended Instance prefix

It is recommended that “In-house” and “Software Vendor” developers implement a configurable prefix in order to allow firms to integrate several application instances easily to ensure **ClOrdID** uniqueness across the firm’s orders.

The exchange recommends that a 2-character prefix is used. This prefix should be placed in the following locations within the **ClOrdID** value, depending on access type:

- **Regular Access:** The 2 leading digits of decimal (base 10) representation, keeping in mind the “mandatory prefix policy” constraint detailed in **Mandatory prefix: Service Bureau vs Regular Access** section.
- **Service Bureau Access:** The next 2 digits of decimal (base 10) representation after the 2-character Service Bureau prefix (see also **Mandatory prefix: Service Bureau vs Regular Access** section).

“Software Vendors” and “In-house” developers must be able to extend the instance prefix size for a firm which has more instances than the available 2-character prefix combinations can cover.

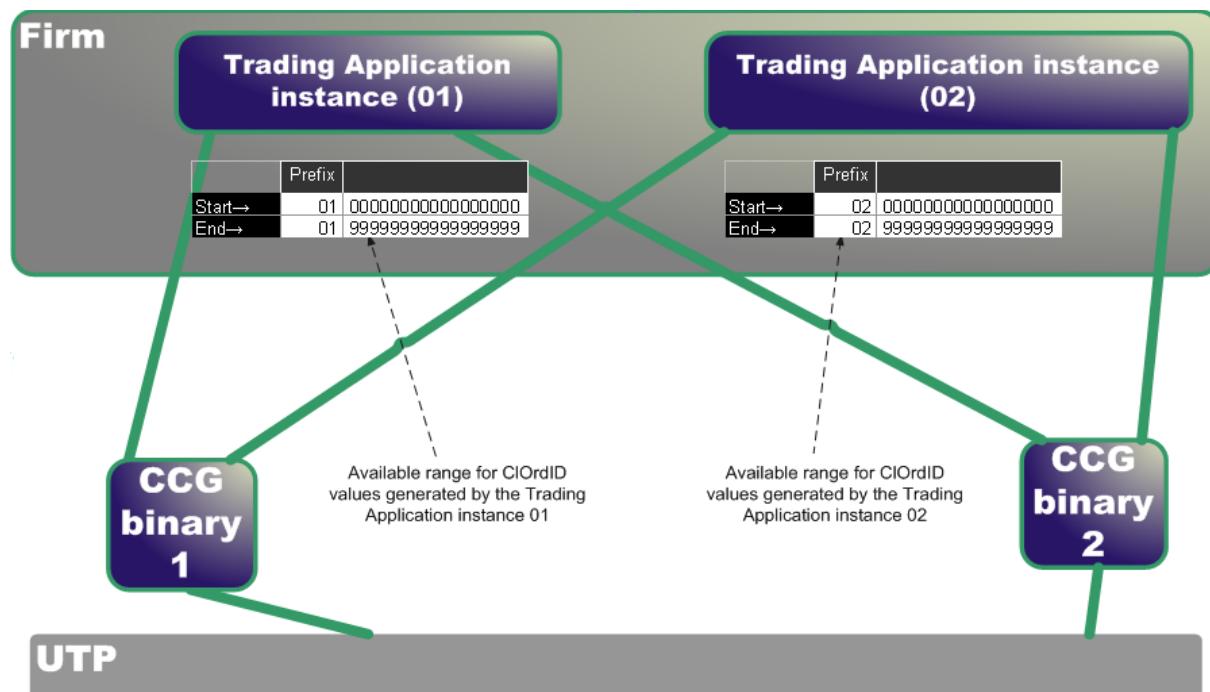
2.2.4 Application Instance ClOrdID available range overview

An overview of the available **ClOrdID** value range, in base 10 format, for a Firm Application Instance, according to the Firm Access type, is presented below.

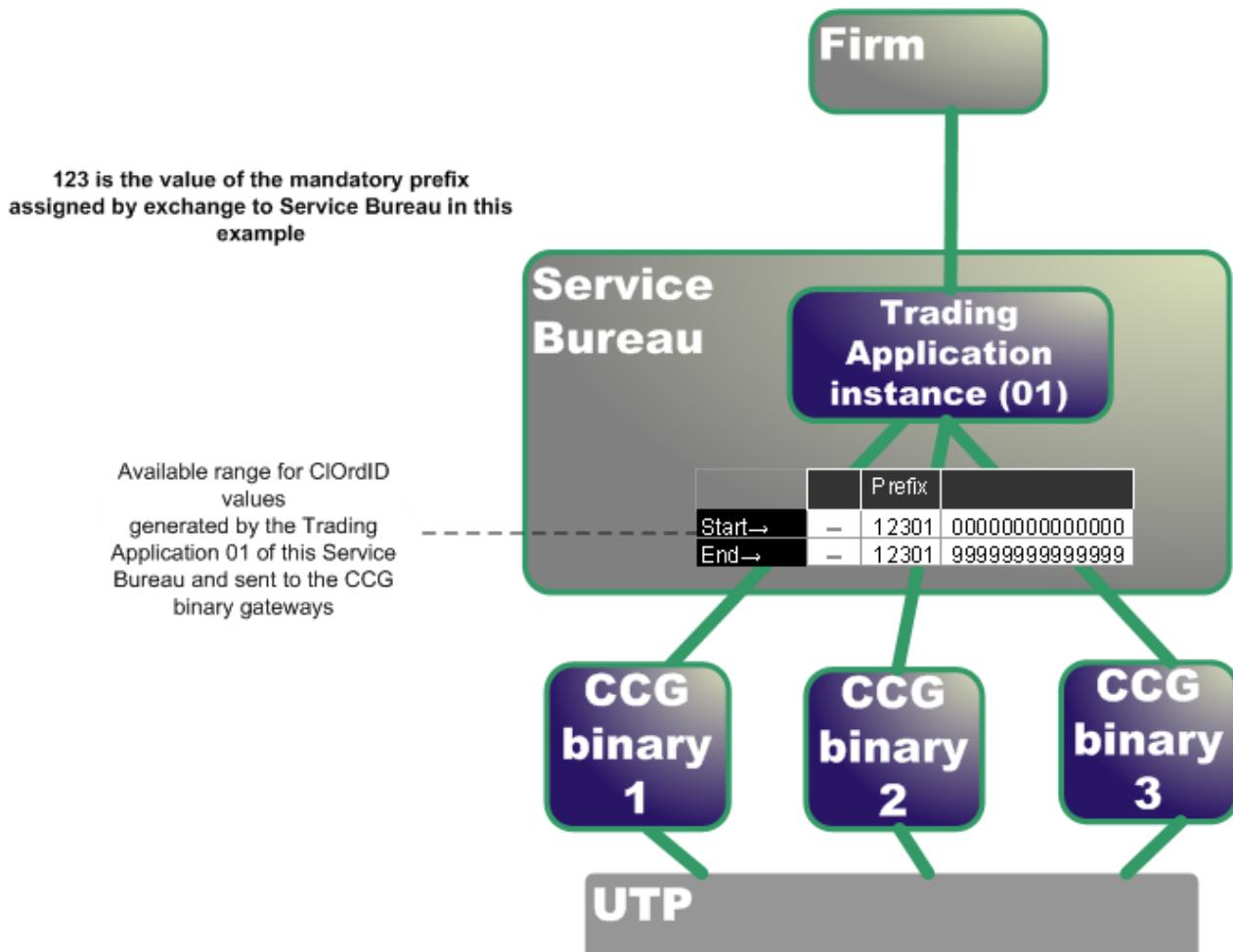
In this section:

- **2** = the Firm Application Instances prefix sizes
- **01** and **02** = the two values associated to the two instances.
- The Service Bureau mandatory prefix will be provided by the exchange.

2.2.4.1 Regular Access

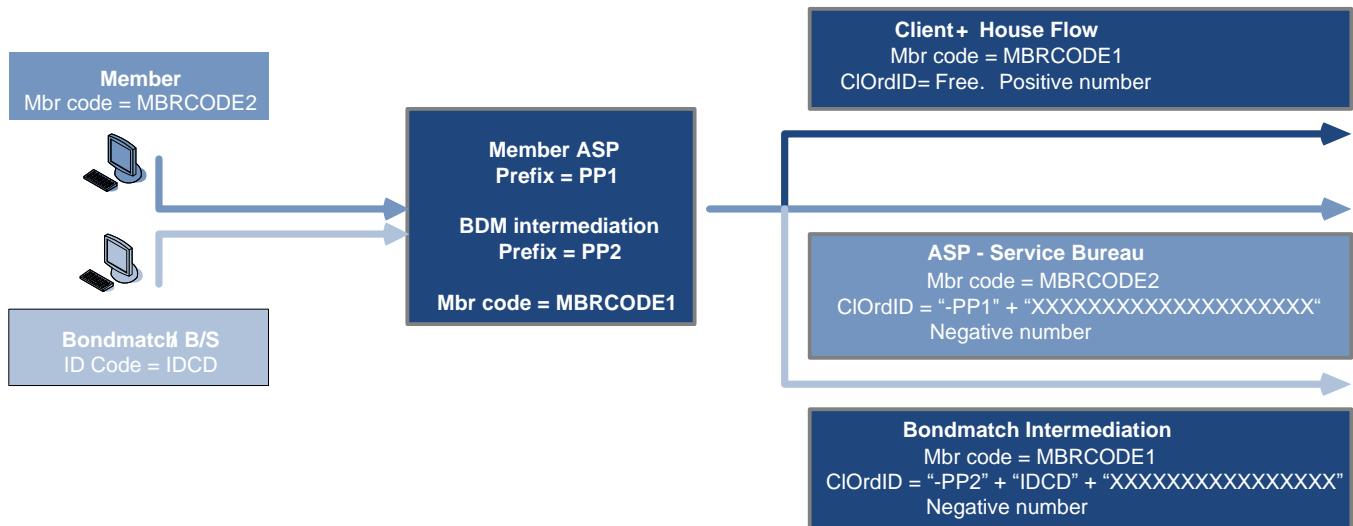


2.2.4.2 Service Bureau Access



2.2.5 Intermediated buy-side client on BondMatch

Immediately after the 3-character Service Bureau prefix, the intermediated buy-side client must enter 4 numeric characters provided by the Exchange.



PP1 / PP2 : Prefix – 3 digits – Provided by Support Team
 MBRCODE1 / MBRCODE2 : Member Code – 8 digits – Provided by Membership
 IDCD : ID Code – 4 digits – Provided by Membership

2.2.6 UTP-Direct vs UTP-FIX.4.2 compatibility

As UTP-Direct can only handle numerical ClientOrderID values, clients needing to ensure UTP-FIX.4.2 / UTP-Direct compatibility must restrict the range of ClientOrderID values used in the UTP-FIX.4.2 protocol to numerical values only, in a range compatible with UTP-Direct protocol.

An order entered through a UTP-FIX.4.2 connection with non-numerical characters in its ClientOrderID value cannot be modified or cancelled through a UTP-Direct connection, and the associated order trade notices cannot be received through a UTP-Direct drop-copy connection.

2.2.7 Drop-copy process

A drop copy session (also named ATR copy) is a dedicated Cash Order Entry Access (SLE) session within order entry is **NOT** allowed. This connection it is used to receive copies of execution or cancellation trades from different sessions for the same member code.

A drop copy subscription can be requested by the member within the Cash CCG Creation Form, [HERE](#).

The subscription or question associated should be sent (asked) via e-mail or phone at:

CAS@euronext.com

+33 1 49 8514 8589

A single drop copy session can receive execution or cancellation trades coming from any of the UTP D or FIX 4.2 protocols.

Each of the followings markets must have a dedicated drop copy session per member code if a member wants to receive execution or cancellation trades:

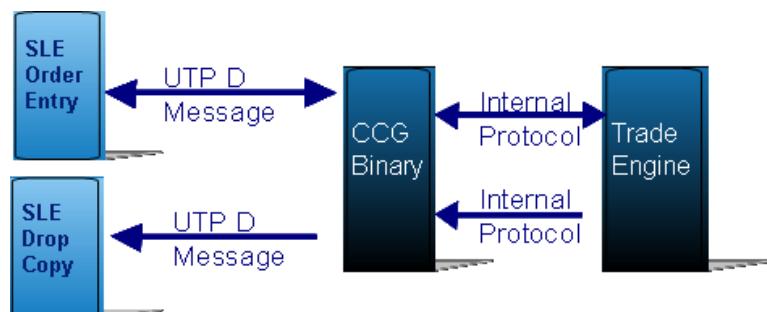
- Regulated Markets.
- Warrants.
- Bondmatch.
- SmartPool.
- TCS.

The following outgoing messages can be received by the members within a drop-copy session (see the diagram bellow):

- Order fills.
- Partial order fills.
- Trade executions.
- Trade cancellations.

The following table displays the message type which can be received via drop-copy connection:

Protocol	Message Type
UTP D	Extended response (x) message
Fix 4.2	Execution reports (8) message



2.3 KINEMATICS

In the case of acknowledgement messages, the ClOrdID field sends back the value provided within the firm request.

In case of acknowledgement of a firm's cancel/replace or cancel request, the OrigClOrdID field identifies the order concerned by the request.

In case of replacement notices sent by the trading engine (**Order Replaced (5)**), the ClOrdID and OrderID fields identify the new order.

In case of cancellation notices (as a response to an **Order Cancel Request (F)**) sent by the trading engine (**Order Killed (4)**), the OrigClOrdID and OrderID fields identify the cancelled order.

In case of elimination notices sent by the trading engine (**Order Killed (4)**), for example a GTT order eliminated at its expiration time, the ClOrdID and OrderID fields identify the eliminated order.

If the case of a reject message, an **Order/Cancel/Replace Reject (8)** message is sent back, providing the client order ID of the request in the ClOrdID field, and the original client order ID in the OrigClOrdID field.

2.3.1 Creating an Order

The firm creates an order using the **New Order (D) request**.

- ◆ If the new order request is accepted, the trading engine answers with an **Order Ack (a)** message. This message contains the **OrderID** field that should be used for subsequent requests regarding the created order (cancel/replace, cancel).



- ◆ If the new order request is rejected, the trading engine answers with an **Order/Cancel/Replace Reject (8)** providing the rejection reason.



Note: a new order request may be acknowledged with an Order Ack (a) message, then rejected with an Order/Cancel/Replace Reject (8) message for collar crossing.

2.3.2 Replacing an Order

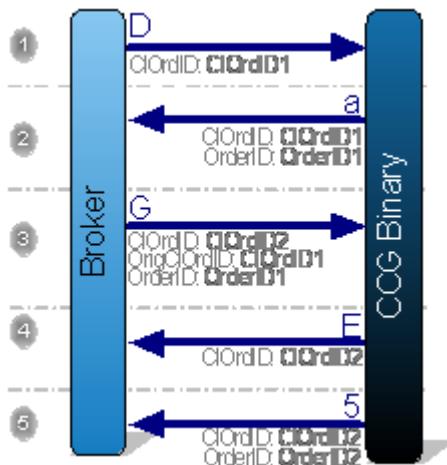
The firm can replace an existing order using the **Cancel/Replace Order (G)** message. This request enables any valid attribute of a live order to be changed (i.e. reduce or increase quantity, change limit price, change instructions, etc.).

The OrderID field or the OrigClOrdID field identifies which order to replace. Using OrderID is recommended (for better performance), and in this case OrigClOrdID is not taken into account even if, for FIX standard compliancy, it must be provided.

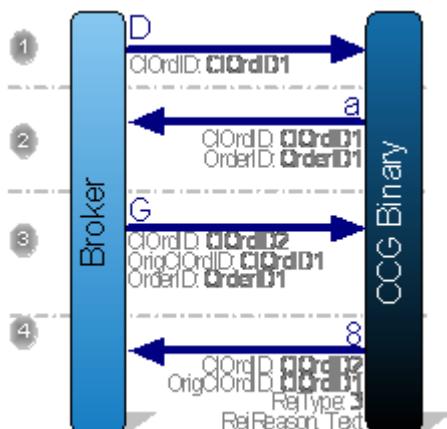
Using OrderID is optional when replacing an order, however, and if not provided, the order is then identified by the OrigClOrdID field (see "[2.3.3-Confirming an Order](#)" for the only functional case where, in

the Cancel/Replace Order (G) message, both OrderID and OrigClOrdID must be provided and are taken into account).

- ◆ If the Cancel/Replace request is accepted, it is acknowledged with a **Cancel/Replace Request Ack (E)** response from the trading engine. Then, if the order replacement is successfully processed, the trading engine sends back an **Order Replaced (5)** confirmation.



- ◆ If the Cancel/Replace request is rejected, an **Order Replaced (5)** message is sent back instead, providing the reason for the rejection.



Note: a cancel/replace request may be acknowledged (**Cancel/Replace Request Ack (E)** message), then rejected (**Order/Cancel/Replace Reject (8)** message) for collar crossing.

2.3.2.1 Order Quantity

To increase or decrease an order quantity, the firm must specify the new total quantity (whatever the already executed quantity of the order to replace).

Example: new order 1 with a total quantity Qtot1=1000 is executed for 700, hence its remaining quantity Qrem1 is 300.

- ◆ If the new total quantity is strictly greater than the difference between the original total quantity and the remaining quantity, the Cancel/Replace request is accepted.

In the example above, order 1 is then replaced by a new order 2 with a total quantity Qtot2 of 800, hence the quantity of order 2 is:

$$Qrem2 = Qrem1 - (Qtot1 - Qtot2) = 30$$

$$0 - (1000 - 800) = 100.$$

- ◆ If the new total quantity is less than or equal to the difference between the original total quantity and the remaining quantity, the Cancel/Replace request is rejected.

In the example above, if the firm attempts to replace order 1 with a new order 2 with a total quantity Qtot2 of 500, the request is rejected as:

$$Qrem1 - (Qtot1 - Qtot2) = 300 - (1000 - 500) <= 0.$$

2.3.3 Confirming an Order

New Order (D) and **Cancel/Replace Order (G)** messages both contain the **ConfirmFlag** field. This indicator is used in the following two situations:

- ◆ Order Size Confirmation

If the order exceeds the amount (equities) or total quantity (bonds) predefined at the class level, it is rejected and needs to be entered once again with the confirmation indicator set (however, firms may choose to avoid rejects for confirmation by systematically setting the confirmation indicator to new order requests in the first place).

■ Collar Crossing Confirmation

If a new order can be executed upon entry, but the matching price hits a collar, the remaining quantity of that order is rejected.

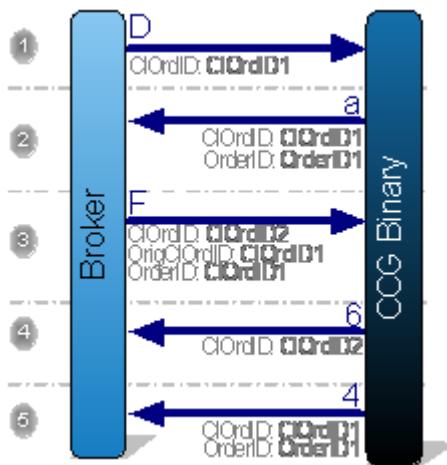
However a firm can force the collar crossing by reactivating the rejected order within a short time period using a **Cancel/Replace Order (G)** with the same price and quantity and the confirmation indicator set. Collars are therefore adjusted around the hit collar before the confirmed order is processed.

In that situation, the firm must provide both the OrigClOrdID and OrderID fields in the **Cancel/Replace Order (G)** message.

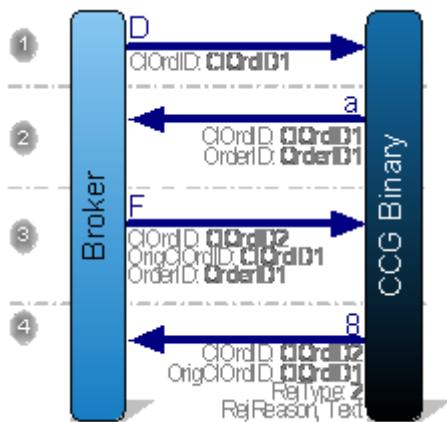
2.3.4 Cancelling an Order

A firm may cancel the remaining quantity of a live order using the **Order Cancel Request (F)** message. The **OrderID** field or the **OrigClOrdID** field identifies which order to cancel. Using OrderID is recommended (for better performance), and in this case OrigClOrdID is not taken into account even if, for FIX standard compliancy, it must be provided. Using OrderID when cancelling an order is optional, however, and if not provided, the order is then identified by the OrigClOrdID field.

- ◆ If the Cancel request is accepted, the trading engine always answers this firm request with a **Cancel Request Ack (6)**. Then, if the order cancellation is committed, an **Order Killed (4)** message is sent back.



- ◆ If the Cancel request is rejected, an **Order/Cancel/Replace Reject (8)** message is sent back instead, providing the reason for the rejection.



2.3.5 Cancelling Several Orders

A firm may cancel the remaining quantity of several live orders using the **Order Cancel Request (F)** message. The following conditions must be met:

- ◆ The **OrderId** field must be set to **-888** and both the **ClOrdID** and **OrigClOrdID** fields must be left empty;
- ◆ Either the **ClassID** field or the **Symbol** field must be populated;
- ◆ Optional additional criteria may be specified with the **CancelByLocationID**, **Side**, **TechnicalOrdType** and **Rule80A** fields. If no optional criteria are specified, all remaining orders of the issuing firm are cancelled for the populated **ClassID** or **Symbol**.

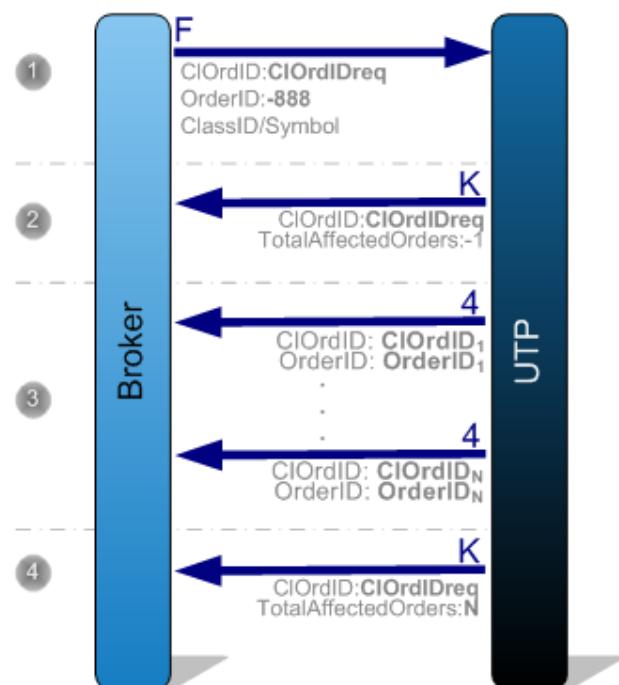
Important Note:

If **CancelByLocationID** criteria are not specified, all remaining orders of all connections of the issuing firm, are cancelled for the populated **ClassID** or **Symbol** along with the other additional criteria.

Following a valid Mass Cancel request, the UTP trading engine will send the following messages:

- ◆ A first message **Bulk Cancel Ack/Report (K)**, sent back to the connection that issued the Mass Cancel request;
- ◆ One message **Order Killed (4)** for each order cancelled, sent to the connection that owns the order;
- ◆ A second and final message **Bulk Cancel Ack/Report (K)**, sent back to the connection that issued the Mass Cancel request.

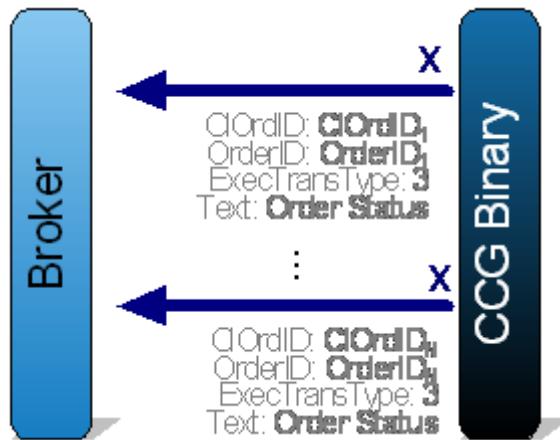
See also the **Order Cancel Request (F)** message.



2.4 ORDER BOOK RETRANSMISSION

For Exchange Disaster Recovery scenario purposes, Order Book Retransmission involves retransmitting all orders present in the central order book for a given firm, as follows:

Extended Response (x) messages (with ExecTransType = 3) will be sent to related client connection, one for each order in the book.



Notes:

- ◆ To request an order book retransmission, firms must contact the Market Desk.
- ◆ Clients working with several firms will need to contact the Market Desk to apply for retransmission requests for all relevant firm identifiers.
- ◆ Firms that apply for Order Book Retransmission must empty their local order book of any remaining orders in order to receive the incoming data with a clean sheet.

3. CCG BINARY HIGH AVAILABILITY

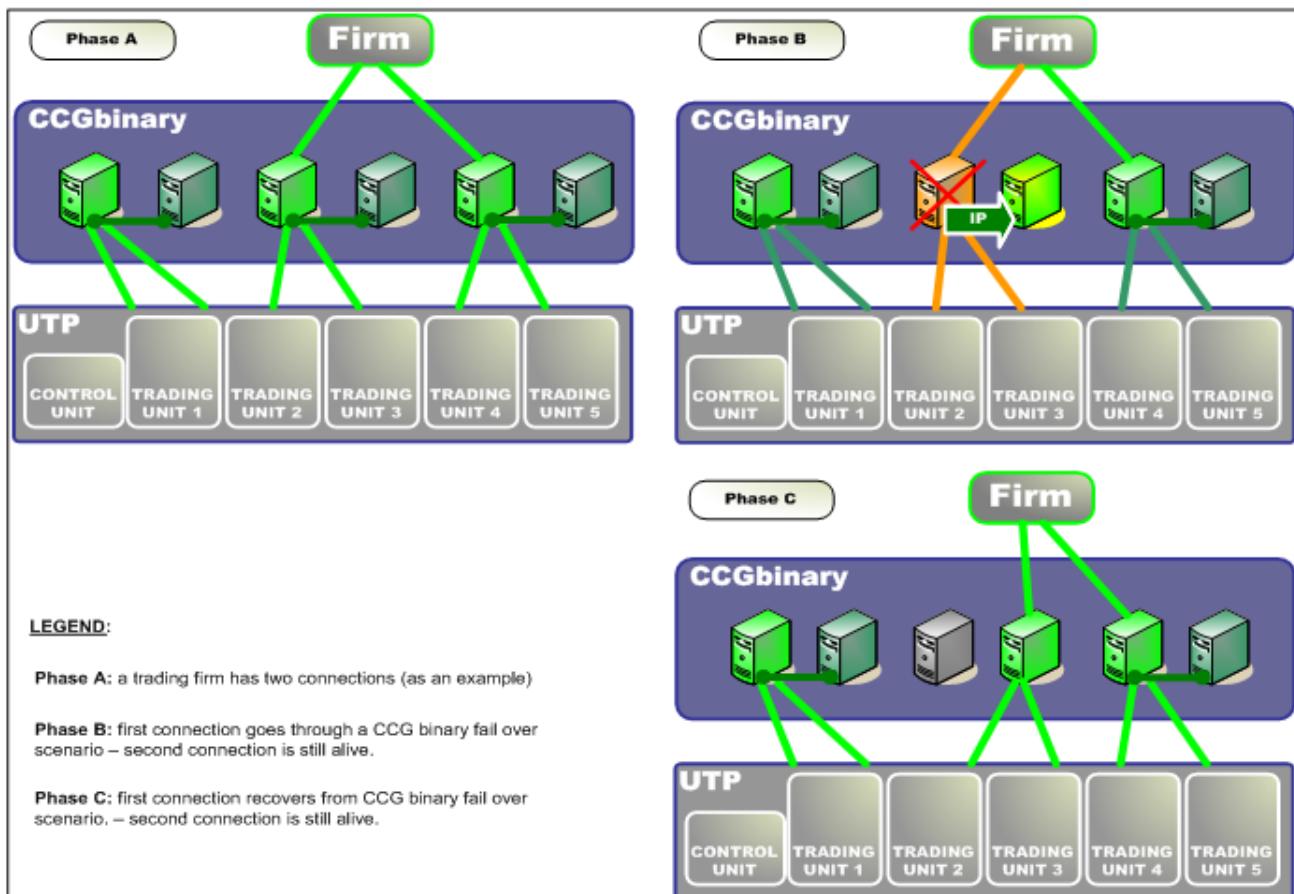
3.1 FAILOVER IMPLEMENTATION OVERVIEW

In the unlikely event of CCG Binary hardware failure, the CCG Binary application will allow customer connection to run a smooth and 100 % safe recovery as follows:

- ◆ Customers will be able to reconnect immediately and transparently without changing destination IP address and port (due to clustering technology);
- ◆ CCG Binary will ensure that there is no data loss within outbound message delivered to customers by sending duplicate messages if necessary. However, duplication will be kept to a minimum (i.e., a minimal number of duplicate messages will be sent), due to the use of SAN technology.

The diagram below provides an illustration of a CCG binary failover scenario.

CCG binary failover scenario



3.2 CUSTOMER RECOVERY METHOD

In the event of CCG Binary failover, customer applications will need to implement conservational measures. A customer application will need to detect the likelihood of duplicate outbound messages (messages received by the customer) and to deal with the likelihood of gaps in inbound message (messages sent by the customer).

From the customer perspective, these conservational measures are as follows:

- During failover period:
 - Even though the failover period is very short (typically a few seconds), a customer may choose to access his previously entered orders from any of his other surviving connections; e.g., a customer may choose to issue a bulk cancel by specifying the username of the current failing-over connection with field **CancelByLocationID**.
- After the failover period:
 - **Messages Received:** The customer application must be able to detect any duplicate messages sent by CCG Binary. As mentioned in section 4.1, a minimal number of duplicates will be sent by CCG Binary to prevent any outbound data loss.
 - **Messages sent:** The customer application must be able to detect any gaps by checking **LastMsgSeqNum** contained in the logon response from CCG Binary. Any gaps must then be dealt with according to customer policy, ie, deciding whether or not to resend messages. If the customer chooses to resend, UTP will reject any duplicates, based on **CIOrId** uniqueness.

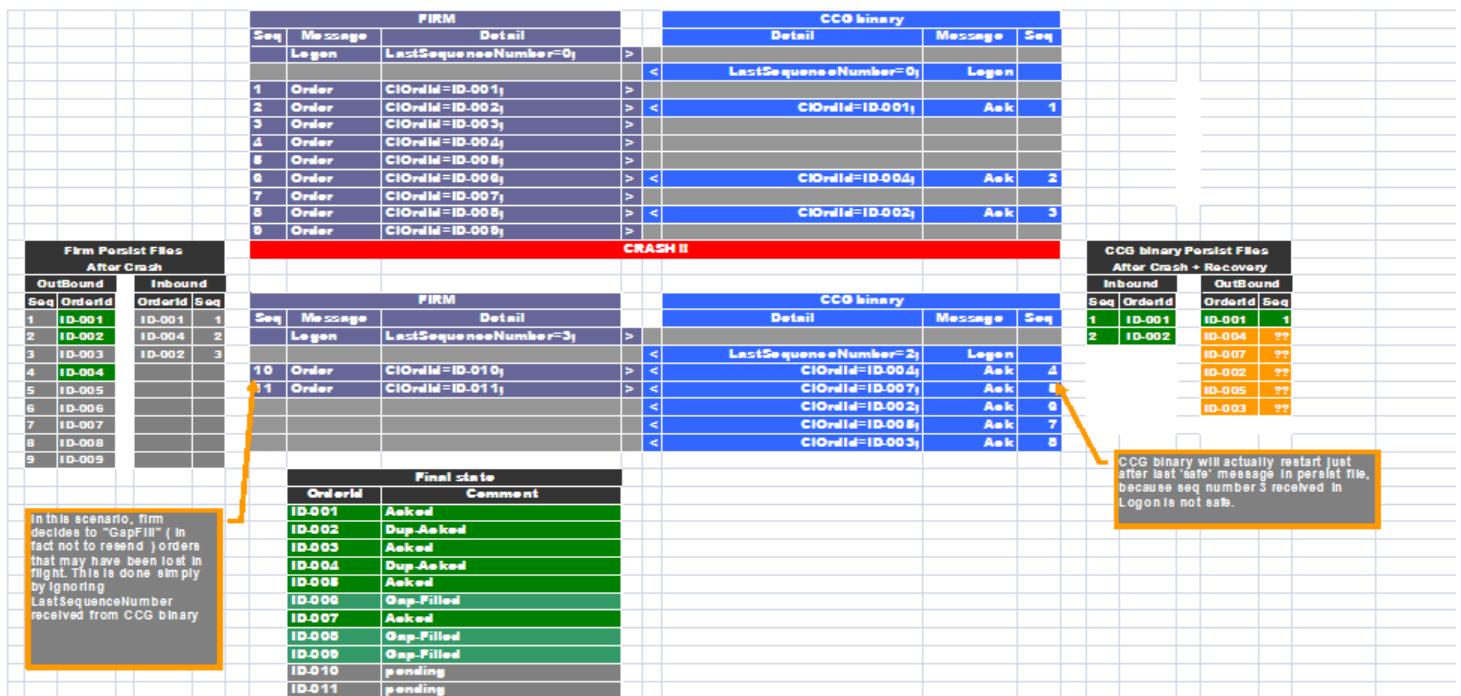
Note: Unlike orders for which retransmitted orders will be rejected, duplicates will not be checked by Euronext for retransmitted bulk quotes. Therefore, customer connections must not retransmit any bulk quotes following a disconnection.

For the duplicate detection method for outbound messages (messages received by a customer), the customer application will need to use the following fields, dependent upon message type:

- | | |
|--------------------------------|--|
| - Order Fill (2): | DeliverToCompID, Symbol, UTPExID, Side |
| - Bust Correct (C): | DeliverToCompID, Symbol, UTPExID |
| - Generic Response (y): | DeliverToCompID, Symbol, OrigMsgType,
OrigMsgSeqNum |
| - All other received messages: | DeliverToCompID, Symbol, CIOrId,
OrderID. |

3.3 CCG BINARY RECOVERY PROTOCOL KINEMATICS

The diagram below provides an illustration of the CCG Binary protocol and data flow kinematics during CCG binary failover. In this context, the Customer Application will need to detect the likelihood of duplicate outbound messages (messages received by customer) and deal with the likelihood of gaps in inbound messages (messages sent by customer).



4. EXCHANGE BUSINESS CONTINUITY

Describe project-related details regarding change management tools and process (defects, issues etc.

Please note that this paragraph does not apply for BondMatch.

In the specific case of a disaster event affecting the Exchange's primary data centre, this section provides guidelines regarding business continuity principles in terms of Exchange secondary data centre connectivity and firm order book integrity.

4.1 EXCHANGE SECONDARY DC CONNECTIVITY

Prior to connecting to Euronext's secondary data centre (DC), the customer application must be restarted as though it were the beginning of a new trading day. Accordingly, the last sequence number must be reset to zero.

4.2 FIRM ORDER BOOK INTEGRITY

Once connected to -Euronext's secondary DC, the customer application must be able to recover the firm's order book from Euronext's secondary DC. Please refer to section [Order Book Retransmission \(page 16\)](#) for more information.

5. EXCHANGE TRADING ENGINE FAILOVER OVERVIEW

Exchange trading engine failover is a technical interruption of trading with two possible scenarios: Standard and Non-Standard failover.

5.1 STANDARD FAILOVER

In the event of an EXCHANGE TRADING ENGINE failover the backup EXCHANGE TRADING ENGINE will automatically take over all the functionalities of the defective EXCHANGE TRADING ENGINE. The trading will continue without interruption.

The instruments are not halted.

Sequence of events:

- ◆ During the failover (backup EXCHANGE TRADING ENGINE node replace the main EXCHANGE TRADING ENGINE node and the CCG to EXCHANGE TRADING ENGINE connection is being re-established) – this should be a short period of time - the CCG's will reject any order sent by the client with the error message “system unavailable”
- ◆ The EXCHANGE TRADING ENGINE application starts with all the instruments in active mode (unhalted)
- ◆ The trading engine will cancel all current day submitted orders for all the Order Entry Gateways (SLE's) configured with “Cancel on disconnect” feature.
- ◆ The trading unit will send via the **Order Entry Gateway (SLE) feed** a Class Event (Q) message which confirms the failover and provides a timestamp until which messages received from the client are correctly guaranteed / processed.
- ◆ Order Entry is allowed.

5.2 NON-STANDARD FAILOVER

In the event that the failover will not be successful a manual intervention is required, and a backup EXCHANGE TRADING ENGINE will be manually dedicated to replace the defective EXCHANGE TRADING ENGINE.

In this specific manual failover situation the backup EXCHANGE TRADING ENGINE will be started in a suspended trading state, where all the instruments are halted and order entry is not allowed.

Sequence of events:

- ◆ During the failover (backup EXCHANGE TRADING ENGINE node replace the main EXCHANGE TRADING ENGINE node, and the CCG to EXCHANGE TRADING ENGINE connection is being re-established) the CCG's will reject any order sent by the client with the error message “system unavailable”.
- ◆ The EXCHANGE TRADING ENGINE application starts with all the instruments in the halted state.
- ◆ The trading engine will cancel all day current submitted orders for all the **Order Entry Gateway (SLE's)** configured with “Cancel on disconnect” feature.

- ◆ The trading unit will send via the **Order Entry Gateway (SLE) feed** a Class Event (Q) message which confirms the failover and provides a timestamp until which messages received from the client are correctly guaranteed / processed.
- ◆ The trading unit will send via **the Market data feed** the Class Halted notice via 516 message.
- ◆ Until the instruments are being unhaltered any order sent by the client on those instruments will be rejected

5.3 EXCHANGE TRADING ENGINE FAILOVER DETECTION

The client has the following options to detect an EXCHANGE TRADING ENGINE failover:

- ◆ Upon reception of Order Entry Gateway (CCG) rejection message “System unavailable” the client has a pre-indication of an EXCHANGE TRADING ENGINE failover, therefore risk mitigation measures should be prepared.
- ◆ Please note this is just a pre-indication, the confirmation is needed via the Class Event (Q) message
- ◆ Upon reception of the Class Event (Q) message the client has the confirmation that an EXCHANGE TRADING ENGINE failover has occurred.

5.4 MITIGATE THE RISK OF AN EXCHANGE TRADING ENGINE FAILOVER

Once the Class Event (Q) message is being received the client has the opportunity to take either one of the following corrective actions:

For order book resynchronization:

- ◆ Automatic self-reconciliation as detailed in the section 6 of the current document.
- ◆ Cancel current day orders entered on the defective Exchange Trading Engine instruments

For trades resynchronization:

- ◆ Contact market surveillance to get the list of lost trades.

6. TRADING TECHNICAL INTERRUPTION MANAGEMENT

In the specific case of an Exchange Trading Engine interruption, as explained in section 5, on one or several classes, this section provides guidelines regarding risk management principles in terms of detective and recovery measures offered to clients to maintain their firm order book in sync with Euronext's order book.

6.1 TECHNICAL TRADING INTERRUPTION RISK MANAGEMENT OVERVIEW

In the unlikely event of an outage at the Exchange, clients may be required to proceed with a partial order book refresh for one or more classes. In such a scenario, clients can use the following CCG API messages to proceed with automatic recovery:

- ◆ **Class Event CCG API message (Q)**: unsolicited notice that will inform clients of a trading engine interruption and will also indicate the timestamp **[field: Timestamp]** from which order response / execution report information cannot be guaranteed by Euronext.
- ◆ **Order Status Request CCG API message (H)**: request that can be sent by the client to Euronext for each single order response / execution report message received that cannot be guaranteed, in order to receive a status report **[Extended response (x)]**. The client will then be able to proceed with the necessary checks and update his order book if required.

6.2 RECOMMENDED METHOD FOR RECOVERY OF AFFECTED ORDERS

When the client receives a **Class Event ('Q')** message with **ClassEvent = 'F'** (Failover of a trading unit), he must check the gap between his local order book and the current Exchange order book and make the following adjustments, as follows:

- ◆ **Identify the scope of affected orders:**
For each class impacted, all response messages (**acks** and/or **fills**) pertaining to symbols that belong to this class and were received before the Class Event message, but after the time provided in the Class Event message (in the **Timestamp** field) up to which all order response information is guaranteed, and related orders should be checked in detail.
- ◆ **Recover the correct status of affected orders:**
For each message that is identified as being affected as detailed above, use the **client order ID** and/or **order ID, symbol** and **firm** fields within the **Order Status Request (H)** in order to establish the actual status of the related order(s). **Extended Response (x)** message(s) are sent in return of the function. If the order status is rejected with reason **OrderId (HON not found or ClOrdId not found)**, the client must handle it correctly in his order book. Please note that the client can only send order status requests once the preopening phase has begun for the related class(es).

CONTACT INFORMATION

For any questions about this document please contact Euronext Customer Technical Support Group (CTSG):

- Email: CTSG@euronext.com
- Tel.: +33 (0)1 8514 8588