

# CCG High Availability Technical Specifications: Legacy protocols (MMTP and Euronext FIX)

May 2009 – V 1.1

### VERSION HISTORY

Version Number	Date	Source Documents Used	Reasons for issuing a New Version	Sections changed
<b>V 1.0</b>	March 2009	CCG high availability overview Customer recovery methods		-
<b>V 1.1</b>	May 2009	CCG high availability overview Customer recovery methods	Adjusting Tag id [Trade Id (tag 9924) and PossResend (tag 97) ] part of Duplicate Detection Method in Euronext FIX Adjusting Euronext Protocol Kinematics Diagrams (replacing IntRef by ClordID - emphasizing the fact that firm must not resend gap) Adjusting MMTP Protocol Kinematics Diagrams (emphasizing the fact that firm must not resend gap)	- Customer Recovery Method - CCG Protocol Kinematics



## **INTRODUCTION**

*Purpose:*

---

The present note provides a technical overview of CCG business continuity features. It describes CCG fail over conditions and recovery guidelines for customer applications.

*Scope*

---

The scope of the present note covers legacy protocols: MMTP and Euronext FIX on both order entry and drop copy services



***Background :***

---

**Our customers requested for low latency order entry which creates challenges for regulated market applications that requires reliability (i.e. no data loss).**

**The low latency challenge can be described as follow : leveraging the adverse effect of low latency technology on CCG failure event.**

**NYSE Euronext addressed the low latency challenge by re-engineering its data delivery model with minimal impact on customer existing implementation. The actual implementation requires two steps:**

- NYSE Euronext implemented necessary changes within CCG.**
- Customers are invited to implement conservatory measures described below.**



## **CCG HIGH AVAILABILITY**

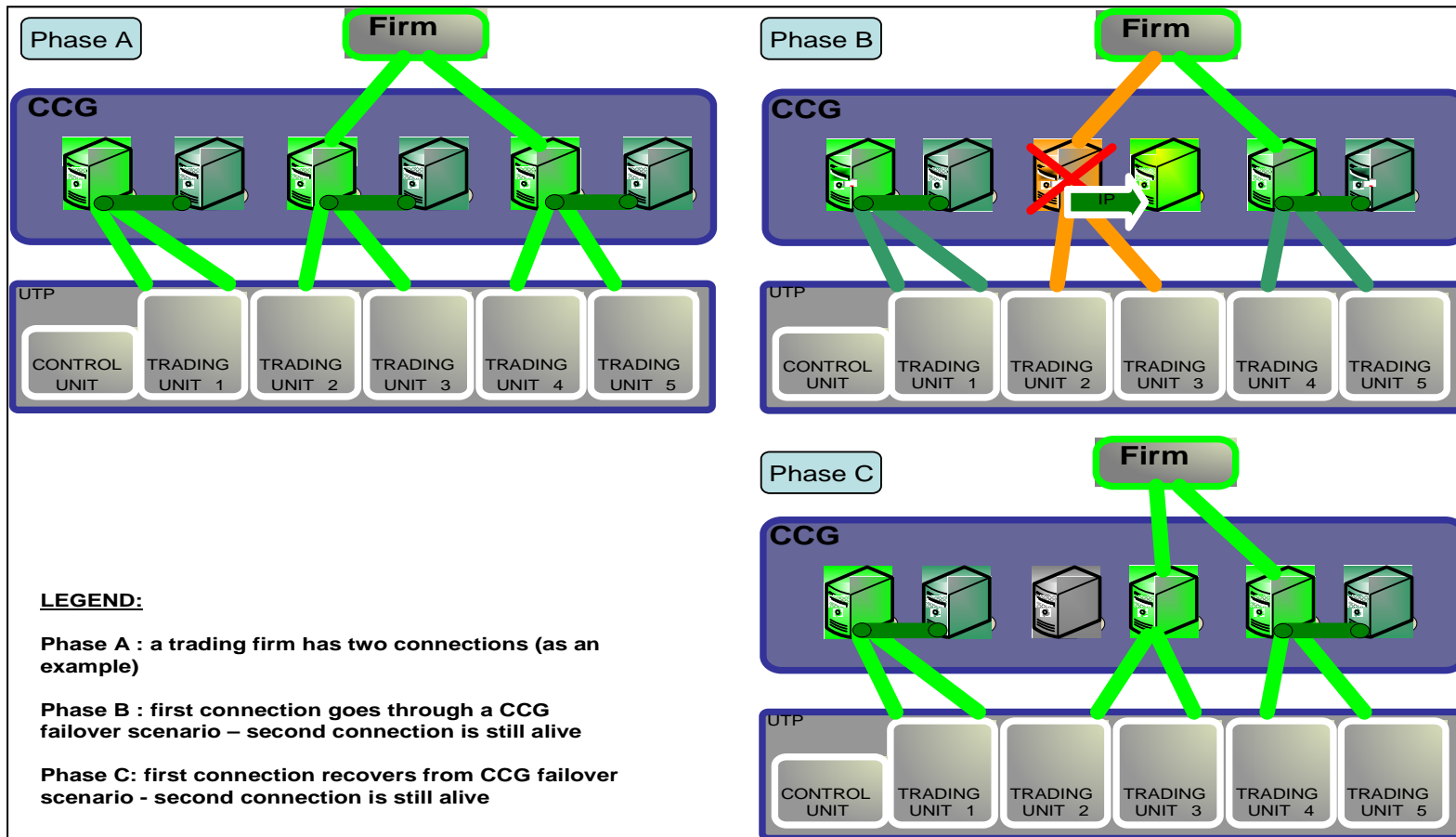
### *failover implementation overview*

---

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

- Customers will be able to reconnect immediately and transparently without changing destination ip address and port (thanks to clustering technology)
- CCG is ensuring no data loss within outbound message delivered to customers by sending duplicate messages if necessary. Yet, duplication will be kept minimal (a minimal amount of duplicate messages - thanks to SAN technology)

The following diagram gives an illustration of CCG fail over scenario.





## Customer recovery method

Customer application will need to implement conservatory measures, to address the event of CCG failover. Basically, Customer application will need to detect the likelihood of outbound messages (messages received by customer) duplicate and to deal with the likelihood of inbound messages (messages sent by customer) gap.

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

- During failover period:
  - Even though the failover period is very short (typically a few seconds), customer may choose to access his previously entered orders from any of his other surviving connections; e.g., customer may choose to issue a bulk cancel by specifying the SLE Identification of the current failing over connection in the related field of bulk cancel command .
- After failover period:
  - Messages **Received: Customer Application must be able to detect any duplicate messages sent by CCG. As mentioned in section 4.1, a minimal amount of duplicates may be sent by CCG to prevent any outbound data loss.**
  - Messages sent: Customer Application must be able to detect any gap depending on protocol :
  - MMTP protocol: Customer application must check [LastMsgId](#) field contained in START-REQ received from CCG during logon phase. **Customer must not resend the gap to avoid the risk of order entry duplicate**

- Euronext FIX protocol: Customer must check BeginSeqNo (Tag 7) and EndSeqNo (Tag 16) contained in ResendRequest, potentially received from CCG during logon phase. **Customer Application must not resend the gap to avoid the risk of order entry duplicate. Instead, Customer Application must send a gapfill and resume emission from the next assigned sequence.**

Regarding outbound messages (messages received by customer) duplicate refers to application level duplicate , i.e. *MMTP field MSG ID and EURONEXT FIX tag 9262 and tag 34 do not garanty application message unicity.* Therefore Customer Application must use the following business level fields, dependant upon message type:

### MMTP protocol:

- Order Fill : Member Code, [Symbol](#), Trade Number,[Side](#)
- Trade Cancellation : Member Code, [Symbol](#), Trade Number,[Side](#)
- All other received messages : Member Code, [Symbol](#), Order Date , [Order](#)Sequence Number.

### Euronext FIX protocol:

- Order Fill : Member Code (tag 9946) , [Symbol](#) (tag 48), Trade ID (tag 9924) ,[Side](#) (tag 54),
- Trade Cancellation : Member Code (tag 9946), [Symbol](#) (tag 48), Trade ID (tag 9924),[Side](#) (tag 54)
- All other received messages : Member Code (tag 9946), [Symbol](#) (tag 48), [Order](#)Sequence Number (tag 37)

Please note that in compliance with FIX4.2 , application level duplicate will be flagged with Possresend (tag 97) set to Yes.

***CCG recovery protocol kinematics***

---

The paragraphs below provide for each CCG protocol an illustrative diagram of protocol and data flow kinematics during CCG fail over. With that respect, Customer Application will need to detect the likelihood of outbound messages (messages received by customer) duplicate and to deal with the likelihood of inbound messages (messages sent by customer) gap.



## EURONEXT FIX LEGACY PROTOCOL RECOVERY KINEMATICS WITHOUT TAG 9262:

Euronext FIX recovery - without MsgID(9262) - with gap fill									
FIRM					CCG				
Seq	Message	Detail			Seq	Message	Detail		
1	Logon								
						Logon			
						Logon	ResetSeqNumFlag=Y		
1	Logon	ResetSeqNumFlag=Y							
2	Order	ClOrdID=ID-001							
3	Order	ClOrdID=ID-002					ClOrdID=ID-001; 9262=C01;		
4	Order	ClOrdID=ID-003							
5	Order	ClOrdID=ID-004							
6	Order	ClOrdID=ID-005							
7	Order	ClOrdID=ID-006					ClOrdID=ID-004; 9262=C02;		
8	Order	ClOrdID=ID-007							
9	Order	ClOrdID=ID-008					ClOrdID=ID-002; 9262=C03;		
10	Order	ClOrdID=ID-009							
<b>CRASH !!</b>									

Firm Persist Files After Crash					CCG Persist Files After Crash + Recovery				
OutBound		Inbound							
Seq	OrderId	Seq	OrderId	9262	Seq	Message	Detail		
2	ID-001	2	ID-001	C01					
3	ID-002	3	ID-004	C02					
4	ID-003	4	ID-002	C03					
5	ID-004				12	ResendReq	BeginSeqNo=5;EndSeqNo=10001		
6	ID-005						BeginSeqNo=5;EndSeqNo=10		
7	ID-006						GapFillFlag=Y;NewSeqNo=10003		
8	ID-007				5	SeqReset	GapFillFlag=Y;NewSeqNo=12		
9	ID-008						ClOrdID=ID-004;9262=C02;PossResend=Y		
10	ID-009								

CCG Persist Files After Crash + Recovery				
Inbound		OutBound		
Seq	OrderId	Seq	OrderId	9262
2	ID-001	2	ID-001	C01
3	ID-002	??	ID-004	??
4	ID-003	??	ID-007	??
		??	ID-002	??
		??	ID-005	??
		??	ID-003	??

Final state		
OrderId	9262	Comment
ID-001	C01	Acked
ID-002	C04	Dup-Acked
ID-003	C06	Acked
ID-004	C02	Dup-Acked
ID-005	C05	Acked
ID-006	-	Gap-Filled
ID-007	C03	Acked
ID-008	-	Gap-Filled
ID-009	-	Gap-Filled
ID-010	?	Pending
ID-011	?	Pending

ClOrdID ID-004 gets an ack sooner than ID-002: this is possible because they went to different trading units

=2 (Last OutBound Seq) + 10000

CCG Resending same message with different MSGID !

### MMTP PROTOCOL RECOVERY KINEMATICS:

**MMTP recovery - Firm must NOT resend gap**

FIRM to CCG			CCG to FIRM		
MsgId	Message	Detail	Detail	Message	MsgId
	START-REQ	MsgId=		START-REQ	
	START-ACK	MsgId=		START-ACK	
F01	DATA-MSG	Order; InternalRef=IR-01			
F02	DATA-MSG	Order; InternalRef=IR-02			
F03	DATA-MSG	Order; InternalRef=IR-03			
F04	DATA-MSG	Order; InternalRef=IR-04			
F05	DATA-MSG	Order; InternalRef=IR-05			
F06	DATA-MSG	Order; InternalRef=IR-06			
F07	DATA-MSG	Order; InternalRef=IR-07			
F08	DATA-MSG	Order; InternalRef=IR-08			
F09	DATA-MSG	Order; InternalRef=IR-09			
<b>CRASH !!</b>					

CIOrdId ID-004 gets an ack sooner than ID-002: this is possible because they went to different trading units

**Firm Persist Files After Crash**

OutBound		Inbound	
MsgId	IntRef	IntRef	MsgId
F01	IR-01	IR-01	C01
F02	IR-02	IR-04	C02
F03	IR-03	IR-02	C03
F04	IR-04		
F05	IR-05		
F06	IR-06		
F07	IR-07		
F08	IR-08		
F09	IR-09		

**FIRM to CCG**

MsgId	Message	Detail
	START-REQ	MsgId=F03
	START-ACK	MsgId=F03
F10	DATA-MSG	Order; InternalRef=IR-10
F11	DATA-MSG	Order; InternalRef=IR-11

**CCG to FIRM**

Detail	Message	MsgId
	START-REQ	MsgId=C03
	START-ACK	MsgId=C03
	Ack; InternalRef=IR-04	DATA-MSG C04
	Ack; InternalRef=IR-07	DATA-MSG C05
	Ack; InternalRef=IR-02	DATA-MSG C06
	Ack; InternalRef=IR-05	DATA-MSG C07
	Ack; InternalRef=IR-03	DATA-MSG C08

**CCG Persist Files After Crash + Recovery**

Inbound		OutBound	
MsgId	IntRef	OrderId	MsgId
F01	IR-01	IR-01	C01
F02	IR-02	IR-04	??
F03	IR-03	IR-07	??
		IR-02	??
		IR-05	??
		IR-03	??

Firm must NOT resend F04 to F09. GapFill in MMTP is as easy as not sending the MSGID you don't want to resend

**Final state**

InternalRef	Comment
IR-01	Acked
IR-02	Dup-Acked
IR-03	Acked
IR-04	Dup-Acked
IR-05	Acked
IR-06	Gap-Filled
IR-07	Acked
IR-08	Gap-Filled
IR-09	Gap-Filled
IR-10	pending
IR-11	pending

CCG Resending same message with different MSGID !