FINANCIAL INFORMATION EXCHANGE (FIX)

FIX APPLICATION LAYER

Order State Changes

FIX Latest

As of EP97, August 2019

FIX Global Technical Committee

Table of Contents

DISCLAIMER

THE INFORMATION CONTAINED HEREIN AND THE FINANCIAL INFORMATION EXCHANGE PROTOCOL (COLLECTIVELY, THE “FIX PROTOCOL”) ARE PROVIDED “AS IS” AND NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL MAKES ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, AS TO THE FIX PROTOCOL (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF) OR ANY OTHER MATTER AND EACH SUCH PERSON AND ENTITY SPECIFICALLY DISCLAIMS ANY WARRANTY OF ORIGINALITY, ACCURACY, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SUCH PERSONS AND ENTITIES DO NOT WARRANT THAT THE FIX PROTOCOL WILL CONFORM TO ANY DESCRIPTION THEREOF OR BE FREE OF ERRORS. THE ENTIRE RISK OF ANY USE OF THE FIX PROTOCOL IS ASSUMED BY THE USER.

NO PERSON OR ENTITY ASSOCIATED WITH THE FIX PROTOCOL SHALL HAVE ANY LIABILITY FOR DAMAGES OF ANY KIND ARISING IN ANY MANNER OUT OF OR IN CONNECTION WITH ANY USER'S USE OF (OR ANY INABILITY TO USE) THE FIX PROTOCOL, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL (INCLUDING, WITHOUT LIMITATION, LOSS OF DATA, LOSS OF USE, CLAIMS OF THIRD PARTIES OR LOST PROFITS OR REVENUES OR OTHER ECONOMIC LOSS), WHETHER IN TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), CONTRACT OR OTHERWISE, WHETHER OR NOT ANY SUCH PERSON OR ENTITY HAS BEEN ADVISED OF, OR OTHERWISE MIGHT HAVE ANTICIPATED THE POSSIBILITY OF, SUCH DAMAGES.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein), except as expressly set out in FIX Protocol Limited’s Copyright and Acceptable Use Policy.

REPRODUCTION

FIX Protocol Limited grants permission to print in hard copy form or reproduce the FIX Protocol specification in its entirety provided that the duplicated pages retain the “Copyright FIX Protocol Limited” statement at the bottom of the page.

Portions of the FIX Protocol specification may be extracted or cited in other documents (such as a document which describes one’s implementation of the FIX Protocol) provided that one reference the origin of the FIX Protocol specification (http://www.fixtrading.org) and that the specification itself is “Copyright FIX Protocol Limited”.

FIX Protocol Limited claims no intellectual property over one’s implementation (programming code) of an application which implements the behavior and details from the FIX Protocol specification.

1 Introduction

This document defines the semantics for the key order handling paradigm of the FIX application layer related to the ExecutionReport(35=8) message. The semantics are mainly driven by only two FIX fields as follows:

- ExecType(150) conveys the reason for the ExecutionReport(35=8) message to have been sent.
- OrdStatus(39) conveys the current status of an order, i.e. after the event causing the message to be sent.

The various scenarios are grouped as follows and primarily reflect the different events that could occur during the lifetime of an order.

A	Order executions (aka Vanilla)
B	Order cancellations
C	Order modifications
D	Order sequencing and chaining
E	Order restatements
F	Order rejections
G	Order status information
H	Multi-day orders (GTC)
I	Non-resting orders (IOC, FOK)
J	Execution corrections and cancellations
K	Trading halts
L	Miscellaneous

Many of the events change the quantity fields of an order. The main fields shown in the scenarios are as follows:

- OrderQty(38) – total executable quantity
- CumQty(14) – executed quantity
- LastQty(32) – quantity of last execution
- LeavesQty(151) – quantity remaining for execution

An order can be in more than one state since the owner was last informed. The precedence rules governing the setting of OrdStatus(39) are defined in Chapter *2 Order State Precedences*. The first set of scenarios set out in Chapter *3 General Order State Change Matrices* are of general nature and are not limited to a specific environment or asset class. A much smaller set of scenarios is defined in Chapter *4 Order State Change Matrices for Exchanges*. They mirror general scenarios but define a different workflow specific to exchanges and other electronic trading venues. The key difference is the reduced verbosity of workflows, i.e. less messages are used to convey the same amount of information on an order.

2 Order State Precedences

In an ExecutionReport(35=8) message OrdStatus(39) is used to convey the current state of the order. If an order simultaneously exists in more than one order state, the value with highest precedence is the value that is reported in OrdStatus(39). The order statuses are as follows (in highest to lowest precedence):

Precedence	***OrdStatus***	***Description***
11	Pending Cancel	Order with a pending cancel request, used to confirm receipt of an OrderCancelRequest(35=F) message. DOES NOT INDICATE THAT THE ORDER HAS BEEN CANCELED.
10	Pending Replace	Order with a pending replacement request, used to confirm receipt of an OrderCancelReplaceRequest(35=G) message. DOES NOT INDICATE THAT THE ORDER HAS BEEN REPLACED.
9	Done for Day	Order not, or partially, filled; no further executions forthcoming for the trading day
8	Calculated	Order has been completed for the day (either filled or done for day). Commission or currency settlement details have been calculated and reported in this execution message
7	Filled	Order completely filled, no remaining quantity
6	Stopped	Order has been stopped at the exchange. Used when guaranteeing or protecting a price and quantity
5	Suspended	Order has been placed in suspended state at the request of the client.
4	Canceled	Canceled order with or without executions
4	Expired	Order has been canceled in broker's system due to TimeInForce(59) instructions. The only exceptions are Fill or Kill and Immediate or Cancel orders that have Canceled as terminal order state.
3	Partially Filled	Outstanding order with executions and remaining quantity
2	New	Outstanding order with no executions
2	Rejected	Order has been rejected by sell-side (broker, exchange, ECN). NOTE: An order can be rejected subsequent to order acknowledgment, i.e. an order can pass from New to Rejected status.
2	Pending New	Order has been received by sell-side's (broker, exchange, ECN) system but not yet accepted for execution. An execution message with this status will only be sent in response to a OrderStatusRequest(35=H) message.
1	Accepted for bidding	Order has been received and is being evaluated for pricing. It is anticipated that this status will only be used with the BidType(394) = 2 (Disclosed style) List Order Trading model.

3 General Order State Change Matrices

The following matrices are included to clarify the sequence of messages and the status of orders involved in the submission and processing of new orders, executions, cancel requests, cancel/replace requests and order status requests. The matrices have been arranged in groups as follows.

3.1 Overview of Scenarios

A Vanilla

Ref	Description
A.1.a	Filled order
A.1.b	Part-filled day order, done for day

B Cancel

Ref	Description
B.1.a	Cancel request issued for a zero-filled order
B.1.b	Cancel request issued for a part-filled order – executions occur whilst cancel request is active
B.1.c	Cancel request issued for an order that becomes filled before cancel request can be accepted
B.1.d	Cancel request issued for an order that has not yet been acknowledged
B.1.e	Cancel request issued for an order that has not yet been acknowledged – the acknowledgment and the cancel request 'cross'
B.1.f	Cancel request issued for an unknown order

C Cancel/Replace quantity changes

C.1 Replace to increase quantity

Ref	Description
C.1.a	Zero-filled order, cancel/replace request issued to increase order qty
C.1.b	Part-filled order, followed by cancel/replace request to increase order qty, execution occurs whilst order is pending replace
C.1.c	Filled order, followed by cancel/replace request to increase order quantity

C.2 Replace not for quantity change

Ref	Description
C.2.a	Cancel/replace request (not for quantity change) is rejected as a fill has occurred

C.3 Replace to decrease quantity

Ref	Description
C.3.a	Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty. Order is replaced then filled
C.3.b	Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty
C.3.c	Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty

D Cancel/Replace sequencing and chaining

D.1 Sequencing

Ref	Description
D.1.a	One cancel/replace request is issued which is accepted – another one is issued which is also accepted
D.1.b	One cancel/replace request is issued which is rejected before order becomes pending replace – then another one is issued which is accepted
D.1.c	One cancel/replace request is issued which is rejected after it is in pending replace – then another one is issued which is accepted

D.2 Chaining

Ref	Description
D.2.a	One cancel/replace request is issued followed immediately by another – broker processes sequentially
D.2.b	One cancel/replace request is issued followed immediately by another – broker processes pending replaces before replaces
D.2.c	One cancel/replace request is issued followed immediately by another – both are rejected
D.2.d	One cancel/replace request is issued followed immediately by another – broker rejects the second as order is pending replace

E Unsolicited/Reinstatement

Ref	Description
E.1.a	Telephoned order
E.1.b	Unsolicited cancel of a part-filled order

E.1.c	Unsolicited replacement of a part-filled order
E.1.d	Unsolicited reduction of order quantity by sell side (e.g. for US ECNs to communicate Nasdaq SelectNet declines)
E.1.e	Unsolicited cancel of ‘cancel if not best’ order
E.1.f	Order is sent to exchange, held waiting for activation and then activated

F Order Reject

Ref	Description
F.1.a	Order rejected due to duplicate ClOrdID
F.1.b	PossResend and duplicate ClOrdID
F.1.c	Order rejected because the order has already been verbally submitted

G Status

Ref	Description
G.1.a	Order status request rejected for unknown order
G.1.b	Transmitting a CMS-style “Nothing Done” in response to a status request
G.1.c	Order sent, immediately followed by a status request. Subsequent status requests sent during life of order

H GT

Ref	Description
H.1.a	GTC order partially filled, restated (renewed) and partially filled the following day
H.1.b	GTC order with partial fill, a 2:1 stock split then a partial fill and fill the following day
H.1.c	GTC order partially filled, restated(renewed) and canceled the following day
H.1.d	GTC order partially filled, restated(renewed) followed by replace request to increase quantity

I TimeInForce

Ref	Description
I.1.a	Fill or Kill order cannot be filled
I.1.b	Immediate or Cancel order that cannot be immediately hit

J Execution Cancels/Corrects

Ref	Description
J.1.a	Filled order, followed by correction and cancellation of executions
J.1.b	A canceled order followed by a busted execution and a new execution
J.1.c	GTC order partially filled, restated (renewed) and partially filled the following day, with corrections of quantity on both executions
J.1.d	Part-filled order Done for day followed by trade correction and bust

K Trading Halt

Ref	Description
K.1.a	Trading Halt – Reinstate
K.1.b	Trading Halt – Cancel

L Miscellaneous

Ref	Description
L.1.a	Transmitting a guarantee of execution prior to execution (Stopped/Guarantee)
L.1.b	Use of CashOrderQty

3.2 Scenarios for State Transitions

The table below shows which state transitions have been illustrated by the matrices in this section (marked with an asterisk). The row represents the current value of OrdStatus(39) and the column represents the next value as reported back to the buy-side via an execution report or order cancel reject message. Next to each OrdStatus(39) value is its precedence – this is used when the order exists in a number of states simultaneously to determine the value that should be reported back. Note that absence of a scenario should not necessarily be interpreted as meaning that the state transition is not allowed:

OrdStatus (precedence value)	New (2)	Partially Filled (4)	Filled (8)	Done For Day (10)	Pending Cancel (12)	Pending Replace (11)		Canceled (5)	Rejected (2)	Stopped (7)
Pending New (2)	*								*	
New (2)	*	*	*	*	*	*			*	*

OrdStatus (precedence value)	New (2)	Partially Filled (4)	Filled (8)	Done For Day (10)	Pending Cancel (12)	Pending Replace (11)		Canceled (5)	Rejected (2)	Stopped (7)
Partially Filled (4)		✱	✱	✱	✱	✱		✱		
Filled (8)		✱	✱			✱				
Done for Day (10)		✱								
Pending Cancel (12)	✱	✱	✱		✱			✱		
Pending Replace (11)	✱	✱	✱			✱		✱		
Canceled (5)										
Rejected (2)										
Stopped (7)		✱								

A Vanilla

A.1.a - Filled order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by salesperson*
2		Execution(X)	New	New	10000	0	10000	0	
3		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by trader/exchange*
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Trade	Filled	10000	10000	0	7000	Execution of 7000

A.1.b – Part-filled day order, done for day

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Done for Day	Done for Day	10000	3000	0	0	Assuming day order. See other examples which cover GT orders

B Cancel

B.1.a – Cancel request issued for a zero-filled order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected*
2		Execution(X)	New	New	10000	0	10000	0	
3	Cancel Request(Y,X)				10000				
4		*Cancel Reject (Y,X)*		*New*					*If rejected by salesperson*
4		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	0	10000	0	Acknowledge the cancel request
5		*Cancel Reject (Y,X)*		*New*					*If rejected by trader/exchange*
5		Execution (Y,X)	Canceled	Canceled	10000	0	0	0	Confirm that order has been canceled

B.1.b – Cancel request issued for a part-filled order – executions occur whilst cancel request is active

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
4	**Cancel Request(Y,X)**				**10000**				
4		**Execution(X)**	**Trade**	**Partially Filled**	**10000**	**5000**	**5000**	**3000**	**Execution for 3000. This execution passes the cancel request on the connection**
5		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected*
5		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	5000	5000	0	‘Pending cancel’ order status takes precedence over ‘partially filled’ order status
6		Execution(X)	Trade	Pending Cancel	10000	6000	4000	1000	Execution for 1000 whilst order is pending cancel – ‘pending cancel’ order status takes precedence over ‘partially filled’ order status.
7		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected*
7		Execution (Y,X)	Canceled	Canceled	10000	6000	0	0	‘Canceled’ order status takes precedence over ‘partially filled’ order status

B.1.c – Cancel request issued for an order that becomes filled before cancel request can be accepted

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
4	**Cancel Request(Y,X)**				**10000**				
4		**Execution(X)**	**Trade**	**Partially Filled**	**10000**	**5000**	**5000**	**3000**	**Execution for 3000. This execution passes the cancel request on the connection**
5		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected*
5		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	5000	5000	0	'Pending cancel' order status takes precedence over 'partially filled' order status
6		Execution(X)	Trade	Pending Cancel	10000	10000	0	5000	Execution for 5000 whilst order is pending cancel. 'Pending cancel' order status takes precedence over 'filled' order status
7		Cancel Reject (Y,X)		Filled					Cancel request rejected – CxlRejectReason = 0 (too late to cancel)

B.1.d – Cancel request issued for an order that has not yet been acknowledged

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2	Cancel Request(Y,X)				10000				Order sender immediately wishes to cancel the order
3		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	0	10000	0	OrigClOrd set to X even though X has not yet been ‘accepted’.
4		Execution(X)	New	New	10000	0	10000	0	Order accepted before cancel request is processed.
5		Execution (Y,X)	Canceled	Canceled	10000	0	0	0	Order canceled.
6	New Order(A)				5000				
7	Cancel Request(B,A)				5000				Order sender immediately wishes to cancel the order
8		Execution (B,A)	Pending Cancel	Pending Cancel	5000	0	5000	0	OrigClOrd set to A even though A has not yet been ‘accepted’.
9		Execution (B,A)	Canceled	Canceled	5000	0	0	0	Order canceled before it is accepted. Note OrigClOrdID set to A even though A has not yet been accepted

B.1.e – Cancel request issued for an order that has not yet been acknowledged – the acknowledgment and the cancel request 'cross'

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2	**Cancel Request(Y,X)**				**10000**				**Order sender immediately wishes to cancel the order**
2		**Execution(X)**	**New**	**New**	**10000**	**0**	**10000**	**0**	**This message crosses the cancel request**
3		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	0	10000	0	
4		Execution (Y,X)	Canceled	Canceled	10000	0	0	0	Order canceled.

B.1.f – Cancel request issued for an unknown order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	Cancel Request(Y,X)				10000				
2		Cancel Reject (Y,X)		Rejected					Cancel request rejected with reject reason of "Unknown Order", OrdStatus is "Rejected" and OrderID is "NONE"

NOTE: It is important to note that rejecting a cancel request for an unknown OrigClOrdID(41) does not cause the sell-side to consume the OrigClOrdID(41) used in the OrderCancelReplaceRequest(35=G).

C Cancel/Replace quantity changes

C.1 Replace to increase quantity

C.1.a – Zero-filled order, cancel/replace request issued to increase order qty

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3	Replace Request(Y,X)				11000				Request to increase order qty to 11000
4		*Cancel Reject (Y,X)*		*New*					*If request is rejected by salesperson*
4		Execution (Y,X)	Pending Replace	Pending Replace	10000	0	10000	0	Acknowledge the Replace request
5		*Cancel Reject (Y,X)*		*New*					*If rejected by trader/exchange*
5		Execution (Y,X)	Replace	New	11000	0	11000	0	Confirm order has been replaced
6		Execution(Y)	Trade	Partially Filled	11000	1000	10000	1000	Execution for 1000. Use Y as the new ClOrdID.
7		Execution(Y)	Trade	Partially Filled	11000	3000	8000	2000	Execution for 2000

C.1.b – Part-filled order, followed by cancel/replace request to increase order qty, execution occurs whilst order is pending replace

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				12000				Request increase in order quantity to 12000
5		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected*
5		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	'Pending replace' order status takes precedence over 'partially filled' order status
6		Execution(X)	Trade	Pending Replace	10000	1100	8900	100	Execution for 100 before cancel/replace request is dealt with
7		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected*
7		Execution (Y,X)	Replace	Partially Filled	12000	1100	10900	0	Confirm replace has been accepted
8		Execution(Y)	Trade	Filled	12000	12000	0	10900	Execution for 10900

C.1.c – Filled order, followed by cancel/replace request to increase order quantity

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Filled	10000	10000	0	10000	Execution for 10000
4	Replace Request(Y,X)				12000				Request increase in order quantity to 12000
5		*Cancel Reject (Y,X)*		*Filled*					*If request is rejected*
5		Execution (Y,X)	Pending Replace	Pending Replace	10000	10000	0	0	‘Pending replace’ order status takes precedence over ‘partially filled’ order status
6		*Cancel Reject (Y,X)*		*Filled*					*If request is rejected*
6		Execution (Y,X)	Replace	Partially Filled	12000	10000	2000	0	. Confirm order has been replaced
7		Execution(Y)	Trade	Filled	12000	12000	0	2000	Execution for 2000

C.2 Replace not for quantity change

C.2.a – Cancel/replace request (not for quantity change) is rejected as a fill has occurred

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	**Replace Request(Y,X)**				**10000**				**Assume in this scenario that client does not wish to increase qty (e.g. client wants to amend limit price)**
4		**Execution (X)**	**Trade**	**Filled**	**10000**	**10000**	**0**	**9000**	**Execution for 9000 – the replace request message and this execution report pass each other on the connection**
5		Cancel Reject (Y,X)		Filled					CxlRejectReason = 0 (too late to cancel)

C.3 Replace to decrease quantity

C.3.a – Cancel/replace request sent whilst execution is being reported – the requested order qty exceeds the cum qty. Order is replaced then filled

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	**Replace Request(Y,X)**				**8000**				**Request a decrease order quantity to 8000 (leaving 7000 open)**
4		**Execution(X)**	**Trade**	**Partially Filled**	**10000**	**1500**	**8500**	**500**	**Execution for 500 sent. Replace request and this execution report pass each**
5		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected by salesperson*
5		Execution (Y,X)	Pending Replace	Pending Replace	10000	1500	8500	0	‘Pending replace’ order status takes precedence over ‘partially filled’ order status
6		Execution(X)	Trade	Pending	10000	1600	8400	100	Execution for 100 occurs before cancel/replace request is accepted
7		*Cancel Reject (Y,X)*		*Partially Filled*					*If request is rejected by trader/exchange*
7		Execution (Y,X)	Replace	Partially Filled	8000	1600	6400	0	Replace is accepted as requested order qty exceeds cum qty
8		Execution (Y)	Trade	Filled	8000	8000	0	6400	Execution for 6400.

C.3.b – Cancel/replace request sent whilst execution is being reported – the requested order qty equals the cum qty – order qty is amended to cum qty

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3	**Replace Request(Y,X)**				**7000**				**Client wishes to amend order qty to 7000**
3		**Execution(X)**	**Trade**	**Partially Filled**	**10000**	**7000**	**3000**	**7000**	**Execution for 7000 - the replace message and this execution report pass each other on the connection**
4		Execution (Y,X)	Replace	Filled	7000	7000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the 'filled' order status takes precedence over 'canceled'.

C.3.c – Cancel/replace request sent whilst execution is being reported – the requested order qty is below cum qty – order qty is amended to cum qty

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3	**Replace Request(Y,X)**				**7000**				**Client wishes to amend order qty to 7000**
3		**Execution(X)**	**Trade**	**Partially Filled**	**10000**	**8000**	**2000**	**8000**	**Execution for 8000 - the replace message and this execution report pass each other on the connection**
4		Execution (Y,X)	Replace	Filled	8000	8000	0	0	The replace request is interpreted as requiring the balance of the order to be canceled – the 'filled' order status takes precedence over 'canceled'.

D Cancel/Replace sequencing and chaining

D.1 Sequencing

D.1.a – One cancel/replace request is issued which is accepted – another one is issued which is also accepted

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	'Pending replace' order status takes precedence over 'partially filled' order status
6		Execution(X)	Trade	Pending Replace	10000	1500	8500	500	Execution for 500
7		Execution (Y,X)	Replace	Partially Filled	8000	1500	6500	0	
8		Execution (Y)	Trade	Partially Filled	8000	3500	4500	2000	Execution for 2000
9	Replace Request(Z,Y)				6000				Request decrease in order quantity to 6000, leaving 2500 open
10		Execution (Z,Y)	Pending Replace	Pending Replace	8000	3500	4500	0	
11		Execution(Y)	Trade	Pending Replace	8000	4000	4000	500	Execution for 500
12		Execution (Z,Y)	Replace	Partially Filled	6000	4000	2000	0	
13		Execution(Z)	Trade	Filled	6000	6000	0	2000	Execution for 2000

D.1.b – One cancel/replace request is issued which is rejected before order becomes pending replace – then another one is issued which is accepted

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Cancel Reject (Y,X)		Partially Filled					Request is rejected
6		Execution(X)	Trade	Partially Filled	10000	1500	8500	500	Execution for 500
7		Execution(X)	Trade	Partially Filled	10000	3500	6500	2000	Execution for 2000
8	Replace Request(Z,X)				6000				Request decrease in order quantity to 6000, leaving 2500 open. Note that OrigClOrdID = X , as this is the last non rejected ClOrdID
9		Execution (Z,X)	Pending Replace	Pending Replace	10000	3500	6500	0	Note that OrigClOrdID = X
10		Execution (Z,X)	Replace	Partially Filled	6000	3500	2500	0	Note that OrigClOrdID = X
11		Execution(Z)	Trade	Partially Filled	6000	5000	1000	1500	Execution for 1500

D.1.c - One cancel/replace request is issued which is rejected after it is in pending replace – then another one is issued which is accepted

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	
6		Execution(X)	Trade	Pending Replace	10000	1500	8500	500	Execution for 500. 'Pending replace' order status takes precedence over 'partially filled' order status
7		Cancel Reject (Y,X)		Partially Filled					Request is rejected (e.g. by trader/exchange)
8		Execution(X)	Trade	Partially Filled	10000	3500	6500	2000	Execution for 2000
9	Replace Request(Z,X)				6000				Request decrease in order quantity to 6000, leaving 2500 open. Note that OrigClOrdID = X as this is the last non rejected ClOrdID
10		Execution (Z,X)	Pending Replace	Pending Replace	10000	3500	6500	0	
11		*Cancel Reject (Z,X)*		*Partially Filled*					*If request is rejected (e.g. by trader/exchange)*
11		Execution (Z,X)	Replace	Partially Filled	6000	3500	2500	0	
12		Execution(Z)	Trade	Partially Filled	6000	5000	1000	1500	Execution for 1500

D.2 Chaining

D.2.a – One cancel/replace request is issued followed immediately by another – broker processes sequentially

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request(Z,Y)				7000				Request decrease in order quantity to 7000, leaving 6000 open. Note OrigClOrdID set to last non rejected ClOrdID i.e. Y (on an ‘optimistic’ basis)
6		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	Broker processes Replace (Y,X) first
7		Execution (Y,X)	Replace	Partially Filled	8000	1000	7000	0	Broker processes Replace (Y,X) first
8		Execution (Z,Y)	Pending Replace	Pending Replace	8000	1000	7000	0	Broker then processes Replace (Z,Y). Note OrigClOrdID set to last accepted ClOrdID i.e. Y
9		Execution (Z,Y)	Replace	Partially Filled	7000	1000	6000	0	Broker then processes Replace (Z,Y)
10		Execution(Z)	Trade	Filled	7000	7000	0	6000	Execution for 6000

D.2.b – One cancel/replace request is issued followed immediately by another – broker processes pending replaces before replaces

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request(Z,Y)				7000				Request decrease in order quantity to 7000, leaving 6000 open. Note OrigClOrdID set to last non rejected ClOrdID i.e. Y
6		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	Broker processes Replace (Y,X) first
7		Execution (Z,X)	Pending Replace	Pending Replace	8000	1000	7000	0	Broker then processes Replace (Z,Y). Note OrigClOrdID set to last accepted ClOrdID i.e. X
8		Execution (Y,X)	Replace	Pending Replace	8000	1000	7000	0	Broker processes Replace (Y,X) first Note OrigClOrdID set to last accepted ClOrdID i.e. X. OrdStatus of Pending Replace takes precedence over Partially Filled
9		Execution (Z,Y)	Replace	Partially Filled	7000	1000	6000	0	Broker then processes Replace (Z,Y) Note OrigClOrdID set to last accepted ClOrdID i.e. Y
10		Execution(Z)	Trade	Filled	7000	7000	0	6000	Execution for 6000

D.2.c – One cancel/replace request is issued followed immediately by another – both are rejected

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request(Z,Y)				7000				Request decrease in order quantity to 7000, leaving 6000 open. Note OrigCOrdID set to last non rejected ClOrdID i.e. Y
6		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	Broker processes Replace (Y,X) first
7		Cancel Reject (Y,X)		Partially Filled					Broker rejects first replace request Note OrigClOrdID set to last accepted ClOrdID i.e. X
8		Execution (Z,X)	Pending Replace	Pending Replace	10000	1000	9000	0	Broker then processes Replace (Z,Y). Note OrigClOrdID set to last accepted ClOrdID i.e. X
9		Cancel Reject (Z,X)		Partially Filled					Broker then rejects second replace request Note OrigClOrdID set to last accepted ClOrdID i.e. X
10		Execution(X)	Trade	Partially Filled	10000	7000	3000	6000	Execution for 6000

D.2.d – One cancel/replace request is issued followed immediately by another – broker rejects the second as order is pending replace

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Replace Request(Y,X)				8000				Request decrease in order quantity to 8000, leaving 7000 open
5	Replace Request(Z,Y)				7000				Request decrease in order quantity to 7000, leaving 6000 open Note OrigCOrdID set to last non rejected ClOrdID i.e. Y
6		Execution (Y,X)	Pending Replace	Pending Replace	10000	1000	9000	0	
7		Cancel Reject		Pending Replace					Rejected because broker does not support processing of order cancel replace request whilst order is pending cancel. CxlRejReason = ‘Order already in
8		Execution (Y,X)	Replace	Partially Filled	8000	1000	7000	0	
9		Execution (Y)	Trade	Partially Filled	8000	3000	5000	2000	Execution for 2000

This matrix illustrates the case where the broker/order receiver does not support multiple outstanding order cancel or order cancel/replace requests

E Unsolicited/Reinstatement

E.1.a – Telephoned order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1									Order for 10000 phoned to broker
2		Execution	New	New	10000	0	0	0	Confirm that the broker has accepted the order – note that broker does not need to capture a ClOrdID
3		Execution	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution	Trade	Filled	10000	10000	0	7000	Execution of 7000

E.1.b – Unsolicited cancel of a part-filled order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4									Broker verbally agrees to cancel order
5		Execution(X)	Canceled	Canceled	10000	1000	0	0	Broker signifies that order has been canceled - ExecRestatementReason = Verbal change

This scenario might occur if the buy-side has not implemented order cancel requests or alternatively there is an electronic communication problem at the point that the buy-side wishes to send a cancel request.

E.1.c – Unsolicited replacement of a part-filled order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3									Broker verbally agrees to increase order quantity to 11000
4		Execution(X)	Restated	New	11000	0	11000	0	Broker signifies that order has been replaced ExecRestatementReason = Verbal
5		Execution(X)	Trade	Partially Filled	11000	1000	10000	1000	Execution for 1000
6									Broker verbally agrees to increase order quantity to 12000
7		Execution(X)	Restated	Partially Filled	12000	1000	11000	0	Broker signifies that order has been replaced ExecRestatementReason = Verbal change

This scenario might occur if the buy-side has not implemented order cancel/replace requests or alternatively there is an electronic communication problem at the point that the buy-side wishes to send a cancel replace request

E.1.d - Unsolicited reduction of order quantity by sell side (e.g. for US ECNs to communicate Nasdaq SelectNet declines)

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Restated	New	9000	0	9000	0	ExecRestatementReason=”Partial Decline of OrderQty”
4		Execution(X)	Trade	Filled	9000	9000	0	9000	

E.1.e - Unsolicited cancel of a ‘cancel if not best’ order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**Price**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000	56				ExecInst = Z (Cancel if Not Best)
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*56*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN) (e.g. if the order book is at 56.1-57.1 prior to this order)*
2		Execution(X)	New	New	10000	56	0	10000	0	Order accepted as order book was 55.9-56.9 prior to this order. Order book is now 56.0-56.9
3		Execution(X)	Canceled	Canceled	10000	56	0	0	0	Order book moves to 56.1-57.0. Order is no longer best bid/offer so is canceled with ExecRestatementReason =”Canceled, Not Best”

E.1.f - Order is sent to exchange, held waiting for activation and then activated

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				Entry of a stop (OrdType = 3), stop limit (OrdType = 4), At the Close (TimeInForce = 7), etc, order. E.g. an order that is held off the book waiting for activation subject to specified conditions.
2		*Execution(X)*	*Rejected*	*Rejected*	*10 000*	*0*	*0*	*0*	*If order is rejected by trader / exchange*
2		Execution(X)	New	New	10 000	0	10 000	0	Trader / exchange acknowledge receipt of order but does not enter it into the book at activation conditions are not met (**WorkingIndicator = N**).
3		Execution(X)	Triggered by (Trading System)	New	10 000	0	10 000	0	Activation conditions are reached. Trader / exchange acknowledge that order is put on book (**WorkingIndicator = Y**). Note that this message can be implicit in situations where there is an immediate fill or partial fill (as any state other than New / Pending New means the order has been added to the book and is working).
4		Execution(X)	Trade	Partially Filled	10 000	2 000	8 000	2 000	Execution of 2000
5		Execution(X)	Trade	Filled	10 000	10 000	0	8 000	Execution of 8000

F Order Reject

F.1.a– Order rejected due to duplicate ClOrdID

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	New Order(X)				15000				Order submitted with the same order id.
5		Execution(X)	Rejected	Partially Filled	10000	1000	9000	0	OrdRejReason = duplicate order. Note combining a reject of the order for 15000 with a status on the first order for 10000 (which is partially filled)

F.1.b - PossResend and duplicate ClOrdID

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3	New Order(X)				10000				PossResend=Y
4		Execution(X)	Order Status	New	10000	0	10000		Because order X has already been received, confirm back the current state of the order. Last Qty not required when ExecType = Order Status
5	New Order(X)				20000				PossResend=N or not set
6		Execution(X)	Rejected	New	10000	0	10000		OrdRejReason = duplicate order. Note combining a reject of the second order for 20000 with a status on the first order for 10000.
7	New Order(Y)				15000				PossResend=Y
8		Execution(Y)	New	New	15000	0	15000	0	Because order Y has not been received before, confirm back as a new order.

F.1.c - Order rejected because the order has already been verbally submitted

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				Order for 10000 sent electronically
2									Order passed verbally as there is communication problem and order does not arrive. The verbally passed order starts getting executed
3		Execution(X)	Rejected	Rejected	10000	0	0	0	Order finally arrives and is detected as a duplicate of a verbal order and is therefore rejected. OrdRejReason = duplicate of a verbal order

Note that the sell-side may employ a number of mechanisms to detect that the electronic order is potentially a duplicate of a verbally passed order, e.g. :

- Checking the possdup flag on the order message header
- Checking the incoming order details against other orders from the same client (e.g. side, quantity)
- Looking at the transact time on the order as a guide to ‘staleness’

G Status

G.1.a - Order status request rejected for unknown order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4	Status Request (Y)								
5		Execution(Y)	Order Status	Rejected	0	0	0		OrdRejReason = unknown order LastQty not required when ExecType=Order Status

G.1.b – Transmitting a CMS-style "Nothing Done" in response to a status request

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3	Status Request(X)								
4		Execution(X)	Order Status	New	10000	0	10000	0	Text="Nothing Done"

G.1.c - Order sent, immediately followed by a status request. Subsequent status requests sent during life of order

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2	Status Request (X)								
3		Execution(X)	Order Status	Pending New	10000	0	10000		Sent in response to status request. LastQty not required when ExecType=Order Status
4		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected*
4		Execution(X)	New	New	10000	0	10000	0	
5	Status Request (X)								
6		Execution(X)	Order Status	New	10000	0	10000		Sent in response to status request.
7		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution for 2000
8	Status Request (X)								
9		Execution(X)	Order Status	Partially Filled	10000	2000	8000		Sent in response to status request
10		Execution(X)	Trade	Filled	10000	10000	0	8000	Execution for 8000
11	Status Request (X)								
12		Execution(X)	Order Status	Filled	10000	10000	0		Sent in response to status request
13	Replace Request(Y,X)				12000				Request to increase order qty
14		Execution (Y,X)	Pending Replace	Pending Replace	10000	10000	0	0	
15		Execution (Y,X)	Replace	Partially Filled	12000	10000	2000	0	

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
16	Status Request (X)								
17		Execution (Y,X)	Order Status	Partially Filled	12000	10000	2000		Sent in response to status request. Note reference to X to allow tie back of execution report to status request
18	Status Request (Y)								
19		Execution(Y)	Order Status	Partially Filled	12000	10000	2000		Sent in response to status request

H GT

H.1.a - GTC order partially filled, restated (renewed) and partially filled the following day

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Day OrderQty**	**Day CumQty**	**Comment**
Day 1,1	New Order(X)				10000						
Day 1,2		Execution(X)	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000			Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	10000	2000	8000	0	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	8000	1000	Execution for 1000

H.1.b - GTC order with partial fill, a 2:1 stock split then a partial fill and fill the following day

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Day OrderQty**	**Day CumQty**	**Comment**
Day 1,1	New Order(X)				10000						
Day 1,2		Execution(X)	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000			Execution for 2000 @ 50
Day 1,4		Execution(X)	Done for Day	Done for Day	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	20000	4000	16000	0	16000	0	Sent the following morning after the split ExecRestatementReason = GTC corporate action. AvgPx=25, DayAvgPx=0
Day 2,2		Execution(X)	Trade	Partially Filled	20000	9000	11000	5000	16000	5000	Execution for 5000
Day 2,3		Execution(X)	Trade	Filled	20000	20000	0	11000	16000	16000	Execution for 11000

H.1.c - GTC order partially filled, restated(renewed) and canceled the following day

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Day** **OrderQty**	**Day** **CumQty**	**Comment**
Day 1,1	New Order(X)				10000						
Day 1,2		Execution(X)	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000			Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	10000	2000	8000	0	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2	Cancel Request (Y,X)				10000						
Day 2,3		*Cancel Reject (Y,X)*		*Partially Filled*							*If rejected by salesperson*
Day 2,3		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	2000	8000	0	8000	0	
Day 2,4		*Cancel Reject (Y,X)*		*Partially Filled*							*If rejected by trader/exchange*
Day 2,4		Execution (Y,X)	Canceled	Canceled	10000	2000	0	0	8000	0	

H.1.d - GTC order partially filled, restated(renewed) followed by replace request to increase quantity

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Day OrderQty**	**Day CumQty**	**Comment**
Day 1,1	New Order(X)				10000						
Day 1,2		Execution(X)	New	New	10000	0	10000	0			
Day 1,3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000			Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	10000	2000	8000	0			Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	10000	2000	8000	0	8000	0	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2	Replace Request(Y,X)				15000						Increasing qty
Day 2,3		*Cancel Reject (Y,X)*		*Partially Filled*							*If rejected by salesperson*
Day 2,3		Execution (Y,X)	Pending Replace	Pending Replace	10000	2000	8000	0	8000	0	
Day 2,4		Execution (X)	Trade	Pending Replace	10000	3000	7000	1000	8000	1000	Execution for 1000
Day 2,5		*Cancel Reject (Y,X)*		*Partially Filled*							*If rejected by trader/exchange*
Day 2,5		Execution (Y,X)	Replace	Partially Filled	15000	3000	12000	0	13000	1000	

I TimeInForce

I.1.a – Fill or Kill order cannot be filled

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				Order is FOK
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Canceled	Canceled	10000	0	0	0	If order cannot be immediately filled

I.1.b – Immediate or Cancel order that cannot be immediately hit

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				Order is IOC
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4		Execution(X)	Canceled	Canceled	10000	1000	0	0	If order cannot be immediately hit

J Execution Cancels/Corrects

J.1.a – Filled order, followed by correction and cancellation of executions

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**AvgPx**	**LastQty**	**LastPx**	**ExecId** **(Exec RefID)**	**Comment**
1	New Order(X)				10000							
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*		*0*		*A*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	0		B	
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	100	1000	100	C	Execution for 1000 @ 100
4		Execution(X)	Trade	Filled	10000	10000	0	109	9000	110	D	Execution for 9000 @ 110
5		Execution(X)	Trade Cancel	Partially Filled	10000	9000	1000	110	0	0	E (C)	Cancel execution for 1000.
6		Execution(X)	Trade Correct	Partially Filled	10000	9000	1000	100	9000	100	F (D)	Correct price on execution for 9000 to 100.
7		Execution(X)	Trade	Filled	10000	10000	0	102	1000	120	G	Execution for 1000 @ 120
8		Execution(X)	Trade Correct	Filled	10000	10000	0	120	9000	120	H (F)	Correct price on execution for 9000 to 120
9	Replace Request (Y,X)				12000							Request to increase order qty
10		Execution (Y,X)	Pending Replace	Pending Replace	10000	10000	0	120	0	0	I	
11		Execution (Y,X)	Replace	Partially Filled	12000	10000	2000	120	0	0	J	
12		Execution(Y)	Trade Correct	Partially Filled	12000	10500	1500	120	9500	120	K (H)	Correct execution of 9000 @ 120 to 9500 @ 120

J.1.b - A canceled order followed by a busted execution and a new execution

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**ExecID** **(Exec RefID)**	**Comment**
1	New Order(X)				10000					
2		Execution(X)	New	New	10000	0	10000	0	A	
3		Execution(X)	Trade	Partially Filled	10000	5000	5000	5000	B	LastPx=50
4	Cancel Request(Y,X)				10000					
5		Execution (Y,X)	Pending Cancel	Pending Cancel	10000	5000	5000	0	C	
6		Execution (Y,X)	Canceled	Canceled	10000	5000	0	0	D	
7		Execution(X)	Trade Cancel	Canceled	10000	0	0	0	E(B)	Cancel of the execution. ‘Canceled’ order status takes precedence over ‘New’
8		Execution(X)	Trade	Canceled	10000	4000	0	4000	F	Fill for 4000. LastPx=51

J.1.c - GTC order partially filled, restated (renewed) and partially filled the following day, with corrections of quantity on both executions

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Day OrderQty**	**Day CumQty**	**ExecID (Exec RefID)**	**Comment**
Day 1,1	New Order(X)				10000							
Day 1,2		Execution(X)	New	New	10000	0	10000	0			A	
Day 1,3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000			B	Execution for 2000
Day 1,4		Execution(X)	Done for Day	Done for Day	10000	2000	8000	0			C	Optional at end of trading day
Day 2,1		Execution(X)	Restated	Partially Filled	10000	2000	8000	0	8000	0	D	ExecRestatementReason = GTC renewal/restatement (no change) – optionally sent the following morning
Day 2,2		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	8000	1000	E	Execution for 1000
Day 2,3		Execution(X)	Trade Correct	Partially Filled	10000	2500	7500	1500	8500	1000	F (B)	Correct quantity on previous day's execution from 2000 to 1500
Day 2,4		Execution(X)	Trade Correct	Partially Filled	10000	2000	8000	500	8500	500	G (E)	Correct quantity on today's execution from 1000 to 500

J.1.d – Part-filled order Done for day followed by trade correction and bust

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**ExecID** **(Exec RefID)**	**Comment**
1	New Order(X)				10000					
2		Execution(X)	New	New	10000	0	10000	0	A	
3		Execution(X)	Trade	Partially Filled	10000	5000	5000	5000	B	LastPx=50
4		Execution(X)	Done for Day	Done for day	10000	5000	0	0	C	Done for day message sent
5		Execution(X)	Trade Correct	Done for day	10000	4000	0	4000	D (B)	Correct quantity on execution to 4000. LastPx = 50
6		Execution(X)	Trade Cancel	Done for day	10000	0	0	0	E (D)	Done for Day OrdStatus takes precedence

K Trading Halt

K.1.a – Trading Halt – Reinstate

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				ExecInst set to reinstate on trading halt
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3									Trading halt established
4									Trading halt lifted
5		Execution(X)	Trade	Filled	10000	10000	0	10000	

K.1.b – Trading Halt – Cancel

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				ExecInst set to cancel on trading halt
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	
3									Trading halt established
4		Execution	Canceled	Canceled	10000	0	0	0	Order canceled due to trading halt. ExecRestatementReason = Canceled due to trading halt

L Miscellaneous

L.1.a– Transmitting a guarantee of execution prior to execution

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**LastPx**	**Comment**
1	New Order(X)				10000					
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*		*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0		
3		Execution(X)	Stopped	Stopped	10000	0	10000	1000	50.10	Text=”You are guaranteed to buy 1000 at 50.10”; This is similar to the concept of a ‘protected’ trade. Not actually reporting a trade, so ExecType = Stopped
4		Execution(X)	Trade	Stopped	10000	1000	9000	1000	50.00	Executed price is better than guaranteed

L.1.b- Use of CashOrderQty

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**Cash** **OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**LastPx**	**Comment**
1	New Order(X)					10000					Currency=EUR. A buy order to invest 10,000 EUR.
2		*Execution(X)*	*Rejected*	*Rejected*		10000	*0*	*0*	*0*		*If order is rejected*
2		Execution(X)	New	New	500	10000	0	500	0		Assuming product has a unit price of 20 EUR at time of order receipt
3		Execution(X)	Trade	Partially Filled	500	10000	200	300	200	20.1	Execution of 200 @20.1 (i.e. does not have to be at the ‘conversion price’ of 20)
4		Execution(X)	Trade	Filled	500	10000	500	0	300	20.2	Execution of 300 @20.2 (i.e. does not have to be at the ‘conversion price’ of 20)

4 Order State Change Matrices for Exchanges

This section addresses issues with order state changes in an exchanges or marketplace environment. These supplement the general Order State Change Matrices above and are documented as specific to exchanges and marketplaces. The titles and references have been chosen in accordance with the general matrices. These specific cases supersede the general ones when implementing the FIX Protocol for exchanges and centralized marketplaces. General Order State Changes matrices that are not mentioned in this section apply to the exchange and centralized marketplace environment. Please also refer to the general Order State Change Matrices defined in Chapter *3 General Order State Change* Matrices.

4.1 Overview of Scenarios

A Vanilla

Ref	Description
A.1.a	Filled order after order rests on book
A.1.b	Part-filled day order after order rests on book, done for day
A.1.c	Order filled upon hitting the book
A.1.d	Order partially filled upon hitting the book

I TimeInForce

Ref	Description
I.1.a	Fill or Kill order cannot be filled
I.1.b	Immediate or Cancel order that cannot be immediately hit

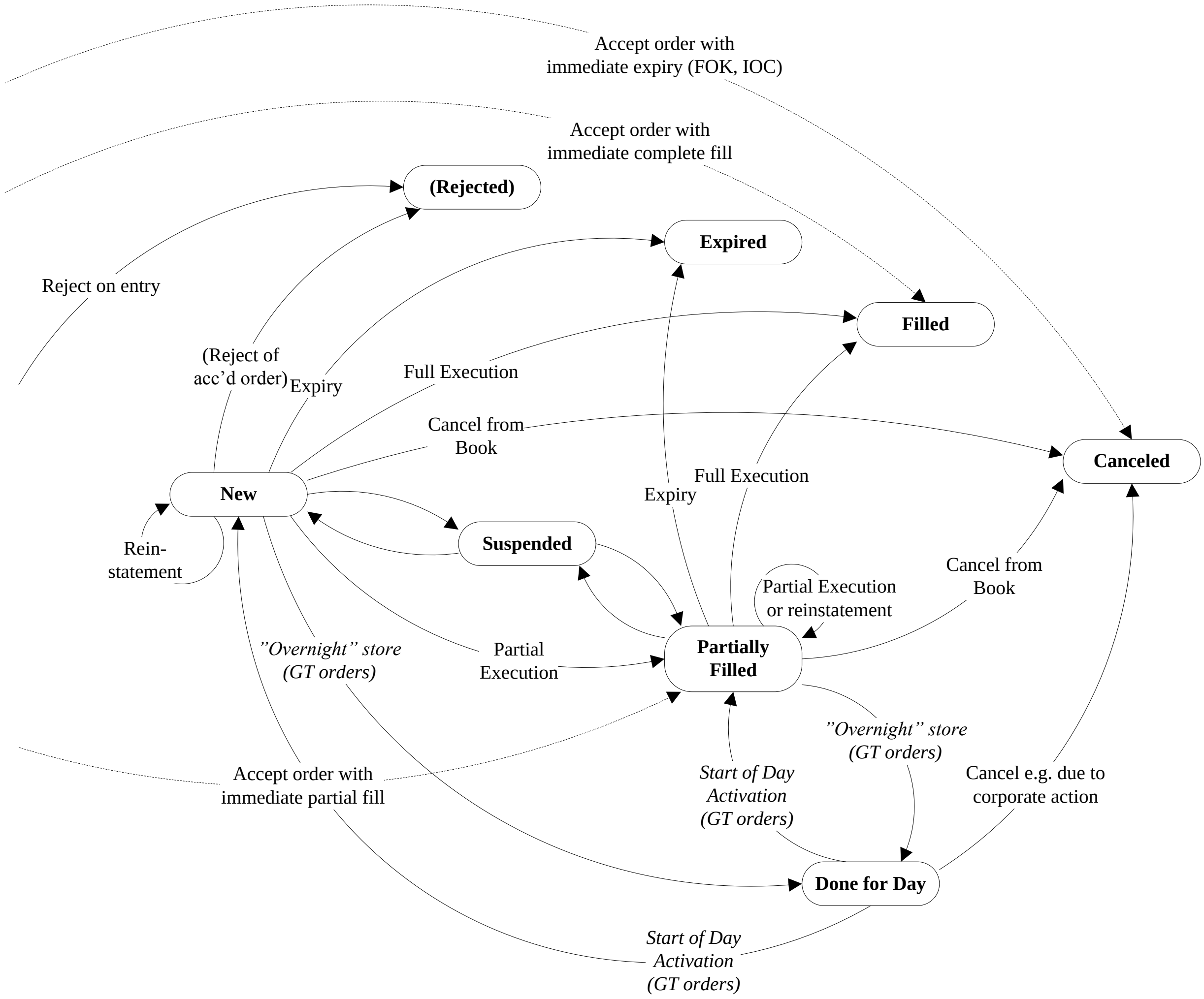
Applicability of general scenarios depicted in Chapter *3 General Order State Change Matrices* for electronic exchange/ECN environments:

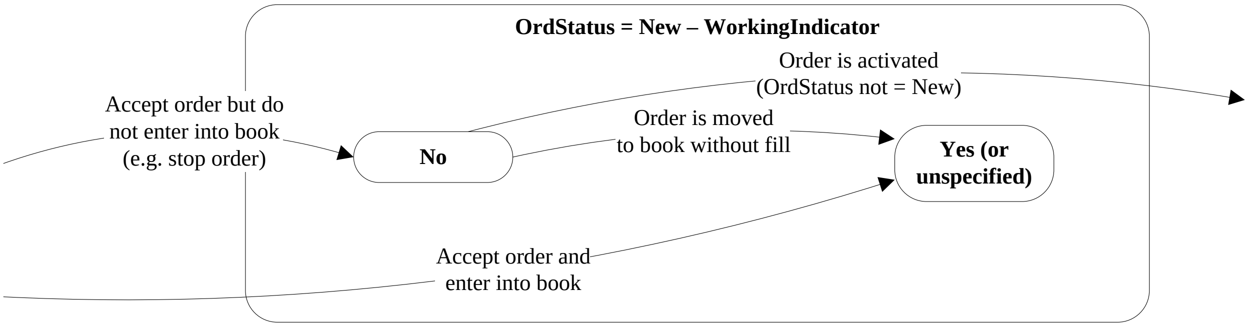
- Filled and Canceled are considered to be terminal states of an order, i.e. a state transition from Filled or Canceled to Partially Filled or Pending Replace should be avoided.
- Pending order states requiring additional messages should be avoided in the interest of performance.

The ExecType is used to identify the purpose of the execution report message. The value of ExecType will typically be New to convey the fact that a new order has been received and processed. However, the value of OrdStatus in this initial response may not necessarily be New as the order might have been executed immediately. The initial value of OrdStatus can therefore also be Partially Filled or Filled. It can even be Canceled if the order has time in force values such as Fill or Kill and Immediate or Cancel and the order could not be executed immediately (and in its entirety in case of Fill or Kill).

The following diagram illustrates the complete set of state transitions recommended for electronic exchange/ECN environments. The dotted lines lead to initial order states other than New and apply to cases where an order does not simply rest on the order book after having been accepted by the exchange/ECN. It is a possibility aimed at increasing performance by reducing the overall number of “Execution Report” messages that need to be provided and processed. Message flows with explicit messages to convey the order state New are equally possible.

4.2 Order State Change Diagrams for Exchanges


Accept order with immediate expiry (FOK, IOC)
Accept order with immediate complete fill
(Rejected)
Expired
Filled
Canceled
Reject on entry
(Reject of acc'd order)
Expiry
Full Execution
Cancel from Book
New
Rein-statement
Suspended
Full Execution
Expiry
Partial Execution or reinstatement
Cancel from Book
Partially Filled
Partial Execution
"Overnight" store (GT orders)
Accept order with immediate partial fill
"Overnight" store (GT orders)
Start of Day Activation (GT orders)
Cancel e.g. due to corporate action
Done for Day
Start of Day Activation (GT orders)


OrdStatus = New – WorkingIndicator
Order is activated (OrdStatus not = New)
Accept order but do not enter into book (e.g. stop order)
Order is moved to book without fill
No
Yes (or unspecified)
Accept order and enter into book

4.3 Scenarios for State Transitions

A Vanilla

A.1.a - Filled order after order rests on book

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by salesperson*
2		Execution(X)	New	New	10000	0	10000	0	
3		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by trader/exchange*
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Trade	Filled	10000	10000	0	7000	Execution of 7000

A.1.b – Part-filled day order after order rests on book, done for day

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by the exchange*
2		Execution(X)	New	New	10000	0	10000	0	
3		Execution(X)	Trade	Partially Filled	10000	2000	8000	2000	Execution of 2000
4		Execution(X)	Trade	Partially Filled	10000	3000	7000	1000	Execution of 1000
5		Execution(X)	Done for Day	Done for Day	10000	3000	0	0	Assuming day order. See other examples which cover GT orders

A.1.c – Order filled upon hitting the book

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by the exchange*
2		Execution(X)	Trade	Filled	10000	10000	0	10000	Immediate execution of 10000

A.1.d – Order partially filled upon hitting the book

Time	**Message Received** (ClOrdID, OrigClOrdID)	**Message Sent** (ClOrdID, OrigClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by the exchange*
2		Execution(X)	Trade	Partially Filled	10000	7000	3000	7000	Immediate execution of 7000

I TimeInForce

I.1.a – Fill or Kill order that cannot be filled

Time	**Message Received** (ClOrdID)	**Message Sent** (ClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				Order is FOK
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker,the exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	If messages are not bundled
3		Execution(X)	Canceled	Canceled	10000	0	0	0	If order cannot be immediately filled
4	New Order(Y)				10000				Order is FOK
5		*Execution(Y)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
6		Execution(Y)	Canceled	Canceled	10000	0	0	0	If message bundling is being used and order cannot be immediately filled

I.1.b – Immediate or Cancel order that cannot immediately be hit completely

Time	**Message Received** (ClOrdID)	**Message Sent** (ClOrdID)	**ExecType**	**OrdStatus**	**OrderQty**	**CumQty**	**LeavesQty**	**LastQty**	**Comment**
1	New Order(X)				10000				Order is IOC
2		*Execution(X)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
2		Execution(X)	New	New	10000	0	10000	0	If messages are not bundled
3		Execution(X)	Trade	Partially Filled	10000	1000	9000	1000	Execution for 1000
4		Execution(X)	Canceled	Canceled	10000	1000	0	0	If order cannot be immediately hit completely
5	New Order(Y)				10000				Order is IOC
6		*Execution(Y)*	*Rejected*	*Rejected*	*10000*	*0*	*0*	*0*	*If order is rejected by sell-side (broker, exchange, ECN)*
6		Execution(Y)	Trade	Canceled	10000	1000	9000	1000	If message bundling is being used and execution of 1000 occurs immediately upon hitting the book