Exchanges / ECN Working Group

Recommended Best Practices

Phase 1

January 25, 2007

Revision 1.0

Proposal Status: Final

For Global Technical Committee Governance Internal Use Only

Submission Date:		Control Number:	
Submission Status	Not Submitted	Ratified Date	
Primary Contact Person:	Hanno Klein, Deutsche Börse	Release Identifier:	

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.

DRAFT OR NOT RATIFIED PROPOSALS (REFER TO PROPOSAL STATUS AND/OR SUBMISSION STATUS ON COVER PAGE) ARE PROVIDED "AS-IS" TO INTERESTED PARTIES FOR DISCUSSION ONLY. PARTIES THAT CHOOSE TO IMPLEMENT THIS DRAFT PROPOSAL DO SO AT THEIR OWN RISK. IT IS A DRAFT DOCUMENT AND MAY BE UPDATED, REPLACED, OR MADE OBSOLETE BY OTHER DOCUMENTS AT ANY TIME. THE FPL GLOBAL TECHNICAL COMMITTEE WILL NOT ALLOW EARLY IMPLEMENTATION TO CONSTRAIN ITS ABILITY TO MAKE CHANGES TO THIS SPECIFICATION PRIOR TO FINAL RELEASE. IT IS INAPPROPRIATE TO USE FPL WORKING DRAFTS AS REFERENCE MATERIAL OR TO CITE THEM AS OTHER THAN “WORKS IN PROGRESS”. THE FPL GLOBAL TECHNICAL COMMITTEE WILL ISSUE, UPON COMPLETION OF REVIEW AND RATIFICATION, AN OFFICIAL STATUS ("APPROVED") TO THE PROPOSAL AND A RELEASE NUMBER.

No proprietary or ownership interest of any kind is granted with respect to the FIX Protocol (or any rights therein).

Table of Contents

Document History

Revision	Date	Author	Revision Comments
0.1	Nov. 22, 2006	Hanno Klein, Deutsche Börse	Initial
0.2	Dec. 19, 2006	Hanno Klein, Deutsche Börse	Added summary of specification changes, added detail to order identification, included various comments.
0.3	Jan. 5, 2007	Hanno Klein, Deutsche Börse	Added order state event diagram, incorporated various review comments.
0.4	Jan. 22, 2007	Hanno Klein, Deutsche Börse	Updated order state event diagram, incorporated various review comments for clarification.
1.0	Jan. 25, 2007	Hanno Klein, Deutsche Börse	Prepared final version of the document

1 Introduction

1.1 Background

The purpose of this document is to provide recommendations for best practices for the usage of FIX by exchanges and ECNs. The work was conducted under the FIX Protocol Limited (FPL) umbrella within the FPL Exchanges / ECN Working Group (EEWG). OMX and Deutsche Börse started the work prior to the EEWG being reestablished on June 1, 2006. The work between OMX and Deutsche Börse resulted in the *Proposal for a Harmonized Exchange Standard* document that served as input to the EEWG. The document is available on the FPL website of the EEWG (http://www.fixprotocol.org/documents/2654/Harmonized Exchange Standard v1_0 Draft.pdf).

In general, FIX offers a large variety of possibilities in order to be able to cover various requirements. However, existing implementations of FIX interfaces to exchanges and ECNs often differ for the same requirement, e.g. how to convey changes to the quantity of an existing order from the user to the exchange. The EEWG believes that increased awareness about such differences in conjunction with guidelines for a more consistent usage across exchanges and ECNs is important to achieve a more widespread adoption of FIX in these environments.

The FIX messages used in this document are based on FPL's FIX 5.0 released December 30th, 2006. The recommendations and guidelines in this document are not necessarily specific to certain asset types unless explicitly stated.

The protocol message flows covered by this document include:

- Orders
- Execution Reports
- Trading Session and Security Status

The remainder of the document is made up of the following chapters:

- Chapter 2 briefly describes the different areas of interest for which this document provides recommendations for best practices.
- Chapter 3 is the main section and details the various issues that have been discussed in the EEWG and presents recommendations and guidelines for best practices.
- Chapter 4 shows, where applicable, detailed message flows for the issues described in Chapter 3.
- Appendix A covers usage examples of various exchanges for a number of issues described in Chapter 3.
- Appendix B contains additional discussions and comments related to the issues described in Chapter 3.

1.2 Summary of Recommendations

The following sections list the key messages for exchanges that are explained in detail in Chapter 3.

1.2.1 Identification

- ClOrdID should be mandatory as currently designated in the FIX specification. OrigClOrdID and OrderID should be conditionally required for order identification in the case of order modifications and deletions.
- Exchanges do not need to check ClOrdID for uniqueness and can treat orders with duplicate values as separate entries.
- Exchanges offering enforcement of ClOrdID uniqueness should limit its scope to active orders within each order book.
- Exchanges should issue two order identifications (private and public) whereby the private one does not change, i.e. is persistent.
- Key fields for security identification can only be specified in the context of a specific asset class.
- The field Symbol and the fields SecurityID/SecurityIDSource should be conditionally required where a message has a mandatory Instrument component block.

- The text associated with the enum value "8" of the field SecurityIDSource should be given a more generic description, e.g. "Marketplace assigned identifier".
- Market segments should be defined by means of a separate (new) message in FIX.
- The field TradingSessionID should be used to refer to an extended period of time that can easily also be expressed informally in terms of the trading day (e.g. entire day, half-day, multi-day).
- The field TradingSessionSubID should be used to refer to trading phases that a trading session consists of (e.g. pre-open, continuous trading, closing, post-trading).
- The values for TradingSessionID and TradingSessionSubID should remain subject to bi-lateral agreement between counterparties but a number of common values are suggested.
- The values for TradingSessionID and TradingSessionSubID should be used to restrict the validity of orders or quotes to these periods (e.g. "DAY", "PRE-OPENING") whereas complex order restrictions should use the field ExecInst (e.g. i="ImbalanceOnly").

1.2.2 Order Handling

- Status requests for orders that have become inactive (cancelled, filled, expired) should be optional.
- Exchanges offering status requests for inactive orders should limit the scope to the current business day and return an execution report with quantity and status information.
- Modification requests for orders that have become inactive (cancelled, filled, expired) should be rejected in the same way as an invalid order identification.
- The ranking of an order for price/time priority should not be based on the order identification issued by the exchange since this would require the reassignment of the exchange identifier.
- Orders should be persistent during order modification, i.e. the order continues to exist, relevant fields are modified and the exchange does not issue a new order identification (OrderID).
- Key quantity fields are total investor quantity (OrderQty), total executed quantity (CumQty), last executed quantity (LastQty), remaining quantity (LeavesQty).
- The field OrderQty should be interpreted as new total investor quantity. The representation of a delta value could be offered as an additional option but should be done so with a (new) separate field, e.g. DeltaQty.
- In-flight modifications should be allowed and supported.
- Order entry and modification from different sources should not require different methods for order identification across FIX and non-FIX interfaces.
- Order entry and modification from different sources should require execution reports to be returned only via the interface issuing the request. Drop-copies can be provided on an optional basis.
- Drop-copies of fills should use Trade Capture Reports instead of Execution Reports if the receiver does not need order status updates.

1.2.3 Performance

- Orders that are (partially) filled upon hitting the book should result in a single execution report with execution type "Trade" and order status "(Partially) Filled".
- Orders that are automatically cancelled (IOC/FOK) should result in a single execution report with execution type "Trade" and order status "Cancelled", quantity field CumQty shows if (partial) fill occurred.
- Multiple fills of an order in a single match operation should result in a single execution report for all partial fills occurring in a single match. This should be conveyed with a (new) separate message, e.g. MassExecutionReport.
- The issue of response messages by an exchange should be configurable for requests related to quotes.
- Pending order states should be avoided with only a single execution report returned upon completion or rejection of a request.
- Mandatory, unsolicited transfer of order restatements should be avoided in general. If necessary, the transfer should include only active orders at the beginning of a trading session and all orders at the end of a trading session.
- Corporate actions should either lead to the deletion of all corresponding active orders or the adjustment of price and/or quantity. In both cases, a confirmation of the changes should be sent to the user.

1.2.4 Market Data

- Status changes that apply to a single security should use the SecurityStatus message.
- Status changes that apply to a group of securities should use the TradingSessionStatus message and store the (exchange specific) group name in a new field SecurityGroup as part of the Instrument component block.
- Market events including state changes and that may or may not change the trading rules should be communicated by means of a (new) separate message, e.g. MDEvent or by additional fields in the SecurityStatus message.

1.2.5 Additional Topics

- Order rejections during specific trading phases not allowing order management should result in a BusinessMessageReject message with reject reason 4 = “Application not available”
- Order rejections due to lack of authorization should result in a BusinessMessageReject message with reject reason 6 = “Not authorized”

1.3 Summary of Changes to the FIX Specification

Some of the recommendations in this document require a formal change to the FIX specification if they are to be adopted. This section briefly lists merely these changes. A separate, formal document with detailed message and field definitions is required to submit these changes to the FIX Global Technical Committee for approval. This should be the next step and conducted by the technical sub-group of the EEWG.

1.3.1 Identification

- Field OrigClOrdID and OrderID should be conditionally required for order modification and deletion
- Field ClOrdID should be optional for drop-copies of execution reports sent to other sessions.
- Field Symbol and the fields SecurityID/SecurityIDSource should be conditionally required whenever a message has a mandatory Instrument component block
- Field SecurityIDSource should have 8=“Marketplace assigned identifier” (currently “Exch Symb”)
- There should be a new message to describe market segments comprising a list of securities, trading sessions and a single schedule.

1.3.2 Performance

- There should be a new message to bundle multiple fills occurring in a single match in a single message.
- There should be new entries in the Order State Change Matrix showing the “final status” approach for IOC and FOK orders. There should also be a diagram to further illustrate the state changes for orders.

1.3.3 Market Data

- There should be a new field SecurityGroup in the Instrument component block to contain exchange specific names related to a group of securities in order to convey status information for more than one security with a single message
- The values of field SecurityTradingStatus are incomplete. The field should either be converted to a free string type or a generic integer value such as 99=”other” should be added in conjunction with an additional field describing the content, e.g. SecurityTradingStatusText.
- Fields SecurityTradingStatus and HaltReason should be added to the market data messages MarketData-SnapshotFullRefresh and MarketDataIncrementalRefresh to enable bundling of orderbook and status data.
- There should be a new message MDEvent or additional fields in the SecurityStatus message to convey market data events other than price or status changes.

2 Areas of Interest

2.1 Introduction

The EEWG discussed a wide range of issues that have been grouped into different areas as described below.

2.2 Identification

The usage and interpretation of fields in FIX that are intended to identify messages and/or entities is one of the crucial areas where harmonization can lead to benefits for users of exchanges and ECNs. The relevant fields for identification sometimes also depend on the asset class involved.

Topics in this area are related to the identification of orders, securities and trading sessions.

2.3 Performance

Performance is one of the main drivers for exchanges and ECNs. An electronic interface needs to support this target, especially in terms of the number of messages required for a specific business workflow. FIX was not designed with performance as a prime target. With the development of the FAST Protocol for the transport of FIX messages, the focus has shifted and the reduction of messages that need to be sent in the first place becomes very important.

Topics in this area are message bundling, pending order states, restatement of orders and corporate actions.

2.4 Order Handling

Order entry and subsequent changes are at the core of FIX since its initial version. FIX was originally developed for trading between the buy-side and the sell-side. An exchange and ECN environment behaves differently in some respect as the individual order of a user typically interacts with a large number of orders from other users. Some of the concepts discussed in this area are quite fundamental so that existing systems of exchanges and ECNs might not be easily adapted to fulfill the recommendations described. The understanding is rather to see the recommendations as a basis for the design of a new system or whenever major changes to an existing system are planned.

Topics in this area are order lifetime, order modification and order entry and modification from different sources.

2.5 Market Data

Market data is a separate area and also covered by its own FPL working group *Market Data and Optimization* (MDOWG). However, there are some specific aspects in the way exchanges and ECN distribute market data to users including data vendors. Recommendations in this area are provided to and discussed with MDOWG wherever useful. MDOWG recommendations have recently been approved by the FIX Global Technical Committee and included in Version 5.0 of the FIX specification.

Topics in this area are the change of trading session status and change of security status.

2.6 Additional Topics

This areas includes miscellaneous, smaller topics. Currently, only order rejections are discussed in this area.

3 Issues and Discussion Points

3.1 Identification

3.1.1 Order Identification

FIX provides a number of fields for the identification of orders, some of which are meant for the buy-side (exchange/ECN user) and others that are meant for the sell-side (exchange/ECN). Each side has more or less one key field to identify an order. FIX foresees the client order identification field ClOrdID for the exchange user to identify an order when it is sent to the exchange. FIX allows the exchange to assign an identification to this order using the field OrderID.

Currently in the FIX specification, the ClOrdID is mandatory for order entry, modification and deletion requests submitted via FIX and its values need to be unique for a specific institution and business day, possible even across multiple days if the order is valid beyond the current business day. Every change of an order by the user requires the sender to provide a new, unique ClOrdID. This allows a chaining of messages related to an order. The current order can be identified by the most recent value of ClOrdID.

The OrderID is mandatory in the response to order entry but optional for order modification and deletion requests. The OrderID does not need to change in response to order modification requests, i.e. this is left up to the exchange. If it is provided by the exchange its most recent value can be used to identify the current order.

Thus, an order can be uniquely identified either by means of the last accepted ClOrdID or the OrderID once an order has been accepted by the receiver. Cancel/replace requests that the receiver rejects result in the ClOrdID to be consumed without changing the ClOrdID of the order. Using ClOrdID as identification allows sending a request for an order before the receiver has issued an OrderID, e.g. for immediate cancellation or modification of an order. The concept of FIX goes one step further by protecting the sender from entering an order more than once. The unique nature of ClOrdID enables the receiver to reject an order if he previously received an order with the same ClOrdID (and the PossResend or PossDup flag, depending on the situation, is not set). This behavior is currently required by FIX. In case of verbally submitted orders which lack a value for ClOrdID, FIX also allows the receiver to compare order details (e.g. side, quantity) of subsequent orders to determine if an order is a duplicate.

The following sections discuss the key issues in the area of order identification, implementation options and provide recommendations for best practices.

3.1.1.1 Mandatory vs. Optional Nature of ClOrdID and OrderID

A mandatory nature of ClOrdID for all usage cases does not reflect current practice for a number of exchanges. Exchanges typically receive orders from many different participants, each of which likely has his own order identification scheme. The order identification values of the participants are stored as additional information but are usually not used to identify or access an order. Exchanges typically issue their own order identification (OrderID) and often require this information for any subsequent actions on this order. ClOrdID can be specified by the user but often needs to be mapped to the OrderID by the exchange to process the request internally.

The following three interface models exist as current practice of exchanges:

1. Exchange only maintains its own identification (OrderID)
2. Exchange maintains its own identification (OrderID) and the identification of the user (ClOrdID)
3. Exchange only maintains the user identification (ClOrdID)

The first and third model only offer a single order identification which is deemed inferior to the second model which in turn is more complex than the other two. The recommendation for the second model needs to distinguish between the entry of new orders and changes to existing orders.

Usage of ClOrdID will not deliver the best performance compared with OrderID if mapping needs to occur but is more user friendly. On the other hand, usage of OrderID does not allow the user to modify or delete orders which the exchange has not confirmed yet and provided their identification for. The mandatory nature of ClOrdID for order entry, modification and deletion as currently defined by FIX for electronically submitted orders, is therefore confirmed. Equally, the mandatory nature of OrderID in the Execution Report from the exchange for these new orders for messages to the original sender and session is confirmed. However, drop-copies to other locations of a user across separate sessions should be allowed to omit ClOrdID and only contain OrderID as uniqueness of ClOrdID is unlikely to be provided by the user across multiple locations. Otherwise, genuine responses and drop-copies might contain the same ClOrdID.

The value of ClOrdID has limited value outside of the FIX session during which it was originally provided to the exchange. ClOrdID serves primarily as a message identifier to enable chaining of requests and responses, i.e. the user is required to provide a new ClOrdID value for a modification or deletion although he refers to an existing entity. OrderID serves as a unique entity identifier issued by the exchange. It should be stable over the lifetime of the order.

Drop-copies of fills can be provided with Trade Capture Reports instead of Execution Reports in cases where the recipient does not require order status information. This might be the case for clearing firms but is not applicable to fail-over locations that will still require Execution Reports.

The modification or deletion of existing orders should be possible in two variants in terms of order identification, i.e. by means of OrigClOrdID or OrderID, whereby OrderID is preferred as soon as it has been made available by the exchange. The current FIX specification does not foresee existing orders to be solely identified with the exchange-assigned identification (OrderID). However, this is required in the following cases.

- Order was not electronically submitted, e.g. provided via telephone, followed by an electronic Execution Report
- Order was electronically submitted through a legacy exchange interface that does not accept a user-assigned order identification
- Order needs to be changed from a user location that did not originally enter the order and received a drop-copy without a value for ClOrdID

Option	Behavior	Best Practice
1	ClOrdID and OrigClOrdID are mandatory, OrderID is unused or used for first ClOrdID	
2	ClOrdID and OrigClOrdID are mandatory, OrderID is optional	
3	ClOrdID and OrigClOrdID are optional, OrderID is mandatory	
4	ClOrdID is mandatory, OrigClOrdID and OrderID are conditionally required for order modification and deletion	X

3.1.1.2 Enforcement of ClOrdID Uniqueness by the Exchange

The enforcement of ClOrdID uniqueness by the exchange could require ClOrdID to be provided for all order entries, regardless whether they are conducted through an electronic interface or verbally submitted whereby the electronic interface can be FIX as well as non-FIX (proprietary). FIX limits the scope of uniqueness to the orders submitted electronically via FIX. Unless there is only a single, electronic FIX interface, it does not seem to be realistic to recommend complete enforcement. Unless ClOrdID is also used internally by the exchange, the check for uniqueness on incoming orders is likely to reduce the achievable performance.

The check is required in case of the usage of the field PossResend. If the flag PossResend is set, the receiver should check to see if the message has been previously received using the ClOrdID which serves as the message identifier. If the receiver has seen the order, he can either ignore the duplicate or send back an Execution Report with the current status of the order. If the flag PossDup is sent, the message sequence number is used to check uniqueness. If the receiver has seen the order, then the order is to be dropped and ignored.

However, a validation of ClOrdID values on a best effort basis and within a well defined scope is likely to be beneficial to the user also in the absence of the aforementioned flags. Validation should be limited to the orders that are still in the book (active orders) and uniqueness should only be considered per sender and order book. The possibilities to offer such functionality without impacting performance depend on the exchange architecture and makes specific recommendations impractical.

It should be noted that exchanges offering to enforce ClOrdID uniqueness need to be aware of potential misuse by users that can resend order entries without having to consider duplicates, e.g. in a recovery situation.

Option	Behavior	Best Practice
1	Check ClOrdID and reject orders with duplicate values	
2	Do not check ClOrdID and treat orders with duplicate values as separate entities	X

3.1.1.3 Persistence of Exchange Issued Order Identification

The exchange issued order identification (OrderID) is not required to change upon every order modification. The FIX specification even requests it to be unique for each chain of orders. However, issuing a new OrderID for order modifications changing the price or increasing the quantity is common practice for some exchanges that use order identification fields for ranking purposes of matching algorithms supporting price/time priority. The latter is not recommended in order to avoid dependencies between the two functionalities (order identification and ranking).

OrderID is recommended as private or static identification that does not change over the lifetime of an order. FIX already provides SecondaryOrderID as an additional field for order identification by the exchange. This field should be used as a public or dynamic identifier that changes depending on the specific rules of the exchange. It should also be the identifier distributed in the market data if the exchange supports that level of granularity. Another example for its usage are reserve orders where the exchange wants to hide the nature of the order by issuing a new order identification each time it is replenished. OrderID then represents the reserve order and its parameters and is only known to the originating user and the exchange.

Option	Behavior	Best Practice
1	Issue single order identification and optionally change it during the order lifetime	
2	Issue two order identifications (private/static and public/dynamic) whereby the private one does not change	X

3.1.2 Security Identification

3.1.2.1 Introduction

Marketplaces support various types of security (or book) identifiers. Conventions differ by:

- Asset class (equity, fixed income, options, futures, commodity, FX, energy, etc)
  - Equities are e.g. sometimes traded using symbols that are more or less very short mnemonic names
  - Options often have symbols that are composed of a mnemonic for the underlying, the strike price and the expiration month
- Geographical location
  - CUSIP's are common in the US
  - SEDOL's in the UK
  - ISIN's in large parts of the world
- Market place
  - Reuters use their native RIC's
  - Native static synthetic ID's (proprietary identification)

Some identifiers are used in the investor trading interface, others in the marketplace interface, while yet others are more applicable to clearing / settlement. When a single marketplace offers trading in multiple books the book and security might need separate identifiers (e.g. a symbol for trading and an ISIN for settlement). In cases where multiple marketplaces trade the same security they could choose the same symbol, but that requires that the marketplace itself is represented in the message (could be implied as in the connection to a specific marketplace).

FIX covers different identifier types quite well. However, there is a need for more clarity in order to promote standardization:

- The <Instrument> component is mandatory in orders and the Symbol (55) tag is thereby mandatory according to the FIX 5.0 specification (not shown in FIXimate). As not all markets use Symbols, the mandatory nature of this tag is a problem for some exchanges and leads to unintended usages. Given that Symbol is not a repeating group there is no reason for it to be a required field, other than it was the most common identifier for equities.
- The <Instrument> component block provides the field SecurityID (48) and a SecurityIDSource (22) value of 8 = "Exchange Symbol". The question is what the relationship between this and the Symbol (55) tag is. Should marketplaces use Symbol or SecurityID + SecurityIDSource = 8? We have found no reason to support such an additional exchange symbol.
- The SecurityDesc (107) tag is used as an identifier in certain domains (futures and unstandardized options). The FIX specification does not explain when and how this field is relevant (see examples for security identifier in Volume 1 of the FIX Specification). It is e.g. unclear why the SecurityID (48) is not used instead (they are both String values).
- When a marketplace starts to use FIX and requires support for a native, marketplace-specific, security identifier, it seems that a new enum needs to be added to the SecurityIDSource (22). The existing enum values are quite different in nature and some correspond to specific marketplaces. As the current SecurityIDSource = 8 (Exchange Symbol) has little meaning, we propose that the description is changed to "Marketplace assigned identifier" and the enum value is used for that purpose.

The recommendation is to see the single field Symbol and the two fields SecurityID and SecurityIDSource as two equally valid options to represent a mandatory Instrument component block. They should be conditionally required, thus eliminating the need to define an artificial value for Symbol in cases where there is no applicable value. The fields SecurityID and SecurityIDSource should be required in the absence of the field Symbol.

3.1.2.2 Key Fields for Security Identification

There is a fairly large number of fields in the <Instrument> component block that can be used in the context of security identification. The following table list those that are deemed to be most relevant for exchange environments and one or more asset classes.

FIX 5.0 (<Instrument> Component Block)				
Tag	**Field Name**	**Req'd**	**Comments**	**EEWG Comment**
55	Symbol	Y	Common, "human understood" representation of the security. SecurityID value can be specified if no symbol exists (e.g. non-exchange traded Collective Investment Vehicles) Use "[N/A]" for products which do not have a symbol.	For securities where Symbol is an eligible identifier. Used for: 1. security traded on symbolic name 2. identifying an underlying symbol (requires additional fields for complete identification). When various symbology can be used, parties need to agree on what is used
65	SymbolSfx	N	Used in Fixed Income with a value of "WI" to indicate "When Issued" for a security to be reissued under an old CUSIP or ISIN or with a value of "CD" to indicate a EUCP with lump-sum interest rather than discount price.	– " – Also used for equities to identify class and preferred stocks

FIX 5.0 (<Instrument> Component Block)				
48	SecurityID	N	Takes precedence in identifying security to counterparty over SecurityAltID block. Requires SecurityIDSource if specified.	Alternative identifier as specified in SecurityIDSource
22	SecurityIDSource	N	Required if SecurityID is specified.	Modified description for value 8 = "Marketplace assigned identifier"
461	CFICode	N	Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values. It is recommended that CFICode be used instead of SecurityType for non-Fixed Income instruments.	Part of alternative identifier for derivatives (options vs. futures). Details see below.
200	MaturityMonthYear	N	Specifies the month and year of maturity. Applicable for standardized derivatives which are typically only referenced by month and year (e.g. S&P futures). Note MaturityDate (a full date) can also be specified.	Part of alternative identifier for derivatives
541	MaturityDate	N	Specifies date of maturity (a full date). Note that standardized derivatives which are typically only referenced by month and year (e.g. S&P futures). May use MaturityMonthYear and/or this field. When using MaturityMonthYear, it is recommended that markets and sell sides report the MaturityDate on all outbound messages as a means of data enrichment.	Part of alternative identifier for derivatives. Used instead of MaturityMonthYear when there are multiple strikes per month.
202	StrikePrice	N	Used for derivatives, such as options and covered warrants	Part of alternative identifier for derivatives
207	SecurityExchange	N	Market used to help identify a security. Valid values: See "Appendix 6-C"	Values are MICs. Used when the identifier provided above is not sufficient to identify the execution venue.
100	ExDestination	N	Execution destination as defined by institution when order is entered. Valid values: See "Appendix 6-C"	Values are MICs.
106	Issuer	N	Name of security issuer (e.g. International Business Machines, GNMA). see also Volume 7: "PRODUCT: FIXED INCOME - Euro Issuer Values"	Used to identify issuer of a bond
107	SecurityDesc	N	Security description	Used for complex securities (e.g. strategies) or display purposes (human readable representation) In fixed income it is used to provide a

FIX 5.0 (<Instrument> Component Block)				
				description of a bond, e.g. "GM 8.375 07/15/2033"

Char 1 *Category*	Char 2 *Group*	Char 3 *Scheme*	Char 4 *Underlying Asset*	Char 5 *Delivery*	Char 6 *Stdized/Non-Std*
O= Options	C=Call P=Put M=Other X= Unknown (n/a)	A=American E=European X=Unknown (n/a)	B=Basket S=Stock-Equities D=Interest rate/ notional debt sec T=Commodities C=Currencies I=Indices O=Options F=Futures W=Swaps M=Other X=Unknown (n/a)	P=Physical C=Cash X=Unknown (n/a)	S=Standardized terms (maturity date, strike price, contract size) N=Non- standardized terms X=Unknown (n/a)

CFI code for options

Char 1 *Category*	Char 2 *Group*	Char 3 *Underlying Asset*	Char 4 *Delivery*	Char 5 *Stdized/Non-Std*	Char 6 *Strategy*
F= Futures	F=Financial Futures C= Commodity Futures M=Others X= Unknown (n/a)	A=Agriculture, forestry, and fishing B=Basket S=Stock-Equities (for financial future) or Services (for commodities futures) D=Interest rate/ notional debt sec C=Currencies I=Indices (for financial futures) or Industrial Products (for commodities futures) O=Options F=Futures W=Swaps M=Other X=Unknown(n/a)	P=Physical C=Cash X=Unknown (n/a)	S=Standardized terms (maturity date, strike price, contract size) N=Non- standardized terms X=Unknown (n/a)	S=Strategy O=Outright X=Not applicable / undefined

CFI code for futures

3.1.2.3 Recommendations for Unique Security Identifiers

Due to the different nature of asset classes and the acceptance of different standards for identification in different parts of the world, it does not make sense to issue a single recommendation for exchanges. The following table recommends the key fields that should be used in various asset classes whereby there might be more than one alternative.

Asset Class	Identification Method	Key Fields	Comment
Equities	Symbol	Symbol, SymbolSfx Optional: SecurityExchange	Use SymbolSfx in case of multiple trading venues
	ID	SecurityID, SecurityIDSource Optional: SecurityExchange	
Futures	Complex	SecurityID, SecurityIDSource, Symbol, CFICode, SecurityDesc, MaturityMonthYear	Use Symbol to contain non-unique product ID
Options	Symbol	Symbol	
	Option characteristics (month, year)	Symbol, SymbolSfx, CFICode, MaturityMonthYear, StrikePrice	Use Symbol and SymbolSfx to contain underlying
	Option characteristics (date)	Symbol, SymbolSfx, CFICode, MaturityDate, StrikePrice	Use Symbol and SymbolSfx to contain underlying
	Complex	SecurityDesc	Exchange specific
Fixed Income	ID	SecurityID, SecurityIDSource Optional: SymbolSfx, SecurityExchange, SecurityDesc, Issuer	
Foreign Exchange	Symbol	Symbol, CFICode, SettlDate	Symbol should contain CCY1/CCY2 where CCY1 and CCY2 are ISO currency codes
Commodities	Symbol	Symbol	
	ID	SecurityID, SecurityIDSource Optional: SecurityExchange	

It is further recommended that

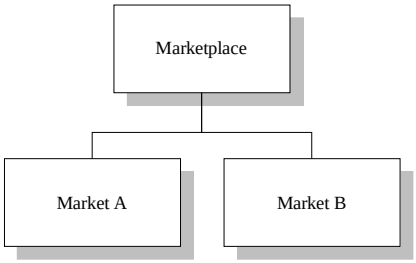
- the single field Symbol and the two fields SecurityID and SecurityIDSource should be conditionally required;
- the text associated with the enum value “8” of the field SecurityIDSource should be given a more generic description, e.g. “Marketplace assigned identifier”.

3.1.3 Trading Session Identification

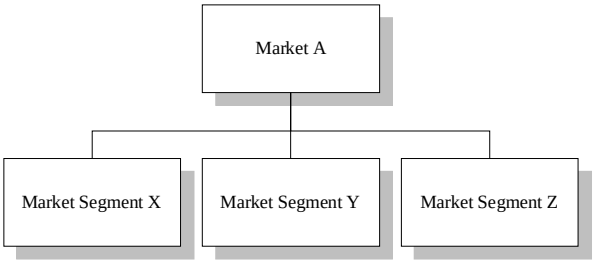
3.1.3.1 Introduction

Trading sessions are components of a number of FIX messages (e.g. NewOrderSingle). Additionally, FIX provides messages that are specific to trading sessions (e.g. TradingSessionList). Exchanges have a wide variety of models regarding trading sessions and the term is not used consistently. The following attempts to define trading sessions and their properties in a way that should be applicable to most exchange environments.

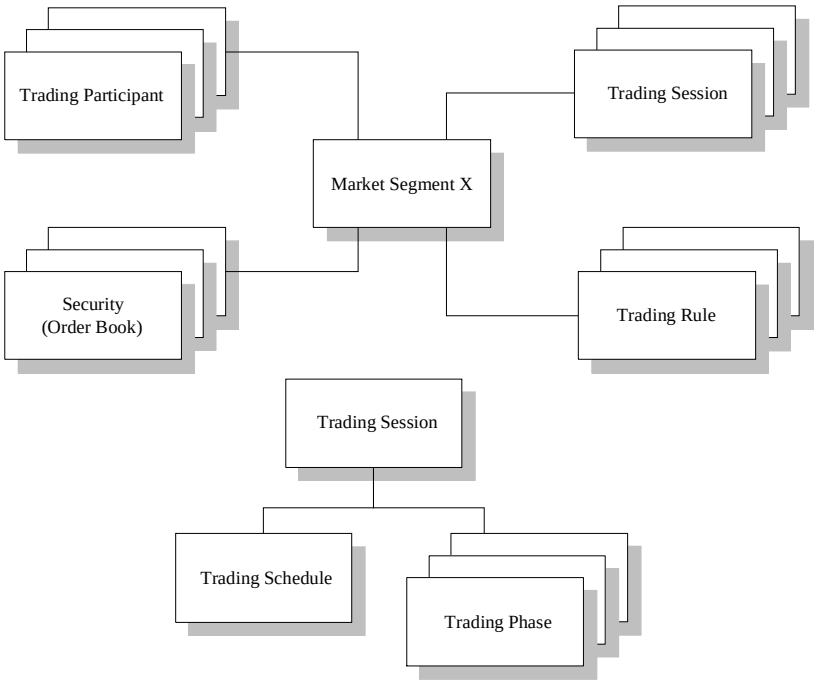
Trading sessions are typically part of an overall data model that starts with a marketplace consisting of one or more markets. A marketplace may group trading into separate markets. Such grouping is frequent when various asset classes are traded in one and the same marketplace, e.g. Stock, Fixed Income, Options and Futures. Another type of segmentation occurs when one exchange covers multiple domiciles, like OMX in Northern Europe and the Xetra system covering German, Irish and Austrian trading. The various markets may have different members (or trading participants), be served by separate trading systems etc.



Each market can have one or more market segments whereby the criteria for segmentation can be very different and are usually driven by business requirements. One market segment could e.g. be configured for wholesale trading, another for retail. Separate market segments could be devised for liquid stock using continuous auto-execution facilities, while another segment uses recurring call auctions to best fit less liquid instruments.



Each market segment includes a number of securities whereby a security can be part of more than one market segment, for example when a market is split between wholesale and retail trading.
Within a given market segment, trading is organized into one or more trading sessions, each of which has a single trading schedule. This schedule defines planned starting and ending times of trading phases that represents elements of the trading sessions. There can be additional trading phases that do not have pre-defined times.



The appendix *5.1.2 Trading Session* contains a number of examples for current exchanges.

3.1.3.2 Definition of Market Segments

FIX currently does not offer messages to define market segments. It is recommended to define an explicit representation of market segments in FIX messages. A separate (new) message should include the following information for each market segment:

- Identification and description of the segment
- Identification of the exchange to which the segment belongs to
- List of securities that are traded within this segment
- List of trading sessions that make up trading in this segment

Note that a market segment is not the same as a group of securities for which a new field SecurityGroup is recommended (see *3.1.2 Security Identification*). The latter is merely a means to convey information about more than a single security without having to use a repeating group. A group of securities is not necessarily confined to a single market segment. The detailed definition of new messages and fields goes beyond the scope of this document and requires a separate, formal document to be presented to the FIX Global Technical Committee for approval.

3.1.3.3 Definition of Trading Sessions and Trading Phases

The definition of trading sessions and trading phases is quite challenging and needs to take into account the specific properties of the market, for example the asset classes that are traded and the market model defining the trading rules. However, there are a number of principles that can be recommended as guidelines for this task. The terms used by an exchange to describe sessions and phases differ and it seems unrealistic to harmonize them. Only the underlying semantics are relevant and these are comparable.

1. A trading session spans an extended period of time that can easily also be expressed informally in terms of the trading day (e.g. entire day, half-day, multi-day)

2. A trading session should always consist of three basic phases[1] of which at least the last one is scheduled no more than once during the entire session.
    a. Preparation for trading (e.g. pre-open, opening, pre-trading)
    b. Trading (e.g. continuous trading, regular trading)
    c. Preparation for ending the trading (e.g. pre-close, closing, post-trading)

3. A trading session has a single schedule that contains at least two timestamps for starting and ending the session. Additional timestamps related to trading phases during the session can be defined.

4. Trading phases only have a scheduled starting time (if at all). The ending time of a trading phase is implicitly given by the starting time of another trading phase or the ending time of the entire session.

5. Trading sessions can overlap within a single market segment. Trading phases should not overlap within a given trading session.

The implication of the principles are best illustrated by giving a number of examples for business requirements and corresponding recommendations. It should be noted that other solutions are possible. The usage of multiple segments, sessions and phases depends on different properties and trading rules an exchange wants to attach to each of these elements. The complexity increases with the level of granularity so that the general recommendation is to distinguish as little as possible and as much as necessary.

[1] It is debatable whether the period after the end of a session (no interaction between exchange and firm possible) should be considered a phase or merely a status. It is not automatically the period between ending and starting time if there is more than one session per day.

Business Requirement	Recommendation
Trading occurs in the morning as well as in the afternoon with a pre-opening phase to start trading. The market closes for 2 hours during the day.	Define two trading sessions, e.g. “Morning”, “Afternoon”. Define three phases, e.g. “Opening”, “Trading”, “Closing” that are valid for both sessions.
Interest rate and equity index derivatives are traded with two different schedules having a large overlap. Other product groups have similar or identical schedules. Users might only be eligible to trade in a single product group. 1. Detailed schedules need to be disseminated electronically 2. Schedule information is only available on the website and in the documentation of the exchange	1. Define two (or more) trading sessions, e.g. “Session 1”, “Session 2” and assign them to two market segments, e.g. “Interest Rate”, “Equity Index” 2. Define a single trading session without sending schedule information and assign it to two market segments, e.g. “Interest Rate”, “Equity Index”

3.1.3.4 FIX Messages for Trading Sessions and Trading Phases

The electronic dissemination of reference data regarding market segments, trading sessions, schedules and phases should not be required in order to be able to trade. The user should not have to expect a change of the basic structure every day. Most changes will be related to the trading schedules.

FIX offers the message TradingSessionList to describe sessions and phases in the sense of reference data. The message TradingSessionListRequest can be used to request descriptions. It is recommended to use multiple TradingSessionList messages to convey schedules and phases separately. The first message is optional and should only contain the trading schedule of each session. That should be followed a one additional message per session containing the different phases without repeating the schedule.

The fields TradingSessionID and TradingSessionSubID can be used to identify a session and a phase within that session. Both fields have data type “string” and can thus contain any value. The following values are assumed to be most relevant within exchange environments.

TradingSessionID	Description
DAY	Single session during a calendar day
HALF-DAY	Single session during a calendar day
MORNING	To be used in conjunction with AFTERNOON and/or EVENING for multiple sessions during a calendar day
AFTERNOON	To be used in conjunction with MORNING and/or EVENING for multiple sessions during a calendar day
EVENING	To be used in conjunction with MORNING and/or AFTERNOON for multiple sessions during a calendar day
AFTER-HOURS	To be used in conjunction with DAY for secondary session in the evening
SESSION <n>	Multiple sessions with different schedules running in parallel

TradingSessionSubID	Description
TRADING	Phase for normal trading (also called continuous or regular trading)
PRE-OPEN	Phase prior to opening with limited interaction and visibility, no matching (same meaning as PRE-TRADING)
PRE-CLOSE	Phase prior to closing with limited interaction and visibility, no matching
PRE-TRADING	Phase prior to opening with limited interaction and visibility, no matching (same meaning as PRE-OPEN)
POST-TRADING	Phase after closing with limited interaction and visibility, no matching
OPENING	Transition phase to commence regular trading, possibly with limited interaction and visibility, matching occurs
CLOSING	Transition phase to end regular trading, possibly with limited interaction and visibility, matching occurs
OPENING AUCTION	Transition phase to commence regular trading by using an auction
INTRADAY AUCTION	Temporary interruption of regular trading to conduct an auction
CLOSING AUCTION	Transition phase to end regular trading by using an auction

3.1.3.5 Usage of Trading Session and Trading Phase Identifiers

The identification of trading sessions and their phases is static and is likely to be disseminated as reference data prior to actual trading. The identifiers are used during normal trading by the exchange as well as by the user for various purposes.

The main usage by the exchange is the dissemination of status information related to an entire session, its current phase or related to an individual security. This is discussed in detail in section *3.4.1 Trading Session and Security Status.*

The user might also need to use trading session and phase identifiers. The order and quote entry messages (NewOrderSingle, NewOrderMultiLeg, Quote, MassQuote) contain a repeating group of TradingSessionID and TradingSessionSubID. This allows to specify trading restrictions for an order or quote by conveying the sessions and /or phases during which an order or (tradable) quote is allowed to match. One could also specify just a single session or phase representing the last session or phase of the current business day (TimeInForce=0=”Day (or Session)”) that an order or quote is to be active. This shows that the design granularity of trading sessions and phases also depends on the trading rules of the given market segment. If the trading of an order or quote can be restricted to specific periods during a trading session, the trading phases could be chosen to coincide with these periods. The main advantage of this approach is that the field TradingSessionSubID is a string value and can be chosen at will.

An alternative for the entry of trading restrictions is to use fields such as ExecInst and TimeInForce, both of which are enum values, i.e. have a restricted list of permissible values. It is recommended to chose TimeInForce if the restriction is closely related to a trading phase. Currently, TimeInForce includes the values “AtTheOpening” and “AtTheClose” for this purpose.

There are other trading restrictions that are less related to a trading phase and more to a security. It is recommended to use ExecInst for this purpose. For example, there could be a market imbalance for a security that is currently going through an auction. Users can enter orders that are only valid if this case occurs. This part of the auction could be defined as a phase, e.g. order book balancing and orders could specify this phase in TradingSessionSubID. However, this case is better expressed by using ExecInst that offers a value “ImbalanceOnly” (ExecInst = i).

The disadvantage of this alternative is that the fields are enum values and additional values cannot easily be defined by an exchange coming up with new market rules but need to approved by the FIX Global Technical Committee.

3.2 Order Handling

3.2.1 Order Lifetime

The lifetime of an order needs to be defined from the perspective of the user. The lifetime ends when all or partial information about an order is no longer accessible for the user through an interface of any kind. In general, FIX foresees orders to have an "indefinite" lifetime (limited by the current business day or the date specified in the field TimeInForce), e.g. the sender of an order can increase its quantity or inquire its status from the receiver at any time. This is not common practice for exchanges where the scope is more limited and a number of events lead to the end of an order lifetime, i.e. a terminal state.

Exchanges typically only allow their users to access an order while it is active and resting in the order book. All orders are archived for auditing purposes but have a limited lifetime from the viewpoint of an external party. An indefinite lifetime would have a severe impact on the performance of an exchange, especially in the area of algorithmic trading where there could be a very large number of small orders.

In an exchange environment, there are a number of possible reasons why the lifetime of an order has ended, i.e. the order is no longer active.

- Order has been completely filled
- Order has expired
- Order has been cancelled by the trader
- Order has been cancelled by the exchange (e.g. corporate action)
- Order has been modified and the exchange has issued a new order

The last case is a special case where the lifetime of an order ends but at the same time a new order is issued by the exchange and inherits most of the characteristics of the previous one.

The following sections discuss the key issues in the area of order lifetime, implementation options and provide recommendations for best practices.

3.2.1.1 Status Request for Inactive Orders

Orders that have been cancelled, filled or that are expired are considered to be inactive orders. Exchanges ensure reliable delivery of the final status of such orders to the user. Rejected orders that do not receive an order identification from the exchange never become active and can be considered out of scope. It should be noted that this definition does not apply to stop orders that are considered to become active as soon as they have been accepted by the exchange and not only after they have been triggered. Stop orders could be cancelled or expire without ever being triggered and would only then become inactive.

Users are likely to see a benefit if they can inquire the status of inactive orders. However, most exchanges currently do not offer such a functionality, i.e. inactive orders are not accessible to the user. It might not provide the best performance but is significantly more user friendly to offer access. Rejecting such requests has the drawback that one can then not differentiate valid orders from orders that have not existed in the first place. The order identification given by the user is likely deemed to be unknown in both cases.

Making inactive orders available for status requests by the user can impact the overall performance, especially if active and inactive orders are separately stored. It can also create additional complexities or even inconsistencies if status requests need to check multiple databases that need to be synchronized as active orders are transferred from one database to another. Regardless of its implementation challenges, the recommendation is to limit the status request scope to orders that have become inactive on the current business day. It should not be considered core functionality and be offered on an optional basis, if at all.

Option	Behavior	Best Practice
1	Return ExecutionReport on an optional basis with status and quantity information for orders that have become inactive on the current business day	X
2	Reject status request in the same way as an invalid order identification	

3.2.1.2 Modification of Inactive Orders

This issue is similar in nature to the status request for inactive orders although the focus is on orders that have been filled. The recommendation is to reject modifications of inactive orders even if the increase of the quantity for a filled order might be useful in an environment with a high (manual) effort to create a new order. The feeling is that this does not apply to electronic trading venues and can therefore be neglected.

Option	Behavior	Best Practice
1	Accept increase of order quantity and set order status back to “partially filled”	
2	Reject modification request of an inactive order in the same way as an invalid order identification	X

3.2.2 Order Modification

A large number of the attributes of an order can be modified. The vast majority of attributes relate to simple information about the order without having any further impact upon modification. This is typically not the case for the price and the quantity of an order. The increase or decrease of the executable quantity of an order is by far the most complex of all order modifications.

FIX provides a large number of order quantity fields whereby OrderQty is the central field representing the total investor quantity. However, some exchanges interpret the order quantity received in a modification request from a user in a different way, either as remaining quantity to be executed or also as a delta quantity (relative increase or decrease of the remaining quantity).

The differences in the representation and interpretation of order quantities across exchanges leads to significantly different message flows. A special case are so-called in-flight modifications (IFMs) where order executions occur at the exchange in parallel to a user sending modification requests for these very same orders. FIX supports IFMs in general but the corresponding FIX message flows are not common practice for all exchanges. Some exchanges do not support IFMs or even standard order modifications, i.e. users are required to cancel an existing order and enter a new order with the desired characteristics.

The following sections discuss the key issues in the area of order modification, implementation options and provide recommendations for best practices.

3.2.2.1 Order Persistence

Order persistence is different from the persistence of an exchange issued order identification as the latter only applies to a single attribute of an order. An entire order can only be considered as new if its status (OrdStatus = “New”) and executed quantity (CumQty = 0) reflect this.

Some exchanges issue a new order during order modification and behave differently than currently designated in the FIX specification. This is typical for exchanges where the order identification is used for ranking purposes. The modification of an order that changes its price or increases its quantity typically requires the order to be re-ranked in the context of a price/time priority matching algorithm. The re-ranking is achieved by issuing a new order with a new order identification (OrderID) and the previous remaining executable quantity (LeavesQty) as new total quantity (OrderQty).

The recommendation is to separate the fields used by the exchange for order identification and ranking. This allows to modify the relevant (quantity) fields without issuing a new order. It is recognized that the aforementioned separation might not be achievable with a reasonable effort for existing exchange systems. However, the design of new systems should always foresee the separation.

Option	Behavior	Best Practice
1	Order continues to exist, relevant fields are modified, exchange does not issue new order identification (OrderID)	X
2	Order is replaced by a new order with a new order identification (OrderID), old order contains previously executed quantity, executed quantity of the new order is zero	

3.2.2.2 Key Quantity Fields

The representation and interpretation of quantities is of major importance in the area of order modification. FIX offers a large number of fields some of which only capture quantities of the current trading day. It is recommended that quantity fields related to the current trading day (DayOrderQty, DayCumQty) do not need to be maintained and communicated by an electronic exchange due to their relatively low significance in such environments. Exchanges allow orders to be valid for more than the current business day but it is the exchange users who typically keep track of the overall quantities as well as the quantities traded on each of the individual days.

The total investor quantity is a key quantity field in FIX. Although it is not common practice for all exchanges, the recommendation is to store and maintain this information. It allows the exchange to implement the standard FIX behaviour for IFMs where the user specifies a new total investor quantity. The exchange can calculate whether the user intended an increase or decrease of the order quantity regardless of the remaining quantity.

Option	Behavior	Best Practice
1	Key quantity fields are total investor quantity (OrderQty), total executed quantity (CumQty), last executed quantity (LastQty), remaining quantity (LeavesQty), total quantity of current day (DayOrderQty), total executed quantity of current day (DayCumQty)	
2	Key quantity fields are total investor quantity (OrderQty), total executed quantity (CumQty), last executed quantity (LastQty), remaining quantity (LeavesQty)	X
3	Key quantity fields are total executed quantity (CumQty), last executed quantity (LastQty), remaining quantity (LeavesQty)	

3.2.2.3 Interpretation of Field OrderQty

The field OrderQty contains a numeric value and is part of the request to modify an existing order. FIX defines the content to represent the new total investor quantity, assuming that the receiver maintains the value of the original total investor quantity. The existing practice for exchanges shows two more, fundamentally different interpretations of this field which lead to different behaviors. Some exchanges interpret the value to be the remaining executable quantity after the modification whereas other exchanges ask for a delta value to increase or decrease the remaining quantity.

The differences are not relevant for new order entries where an absolute value is required and the total investor quantity equals the remaining quantity by definition due to lack of execution. However, in-flight modifications (IFMs) cannot be handled in the same way for all three variants.

The exchange cannot determine whether the user wants to increase or decrease the quantity if only the remaining quantity is maintained. The user might want to reduce the remaining quantity based on outdated information if an execution occurring in parallel causes the actual remaining quantity to fall below the value specified in his modification request. This would then cause an increase of the quantity. These exchanges typically resolve this issue by rejecting IFMs and ensuring that the user has processed all updates from the exchange.

IFMs can easily be supported if the field OrderQty is interpreted as new total investor quantity or as delta value. In the former case, the delta value can be calculated as difference between the original and the new total investor

quantity whereas the delta value is explicitly given in the latter case. The delta value is to be applied to the actual remaining quantity whereby a decrease cannot reduce the quantity below zero. In this case, the order modification implicitly leads to a cancellation of the order.

Although functionally equivalent, the recommendation is to interpret OrderQty as new total investor quantity. The variant with the delta value could be offered as an additional option. However, the delta value should be stored in a separate field, e.g. DeltaQty, to avoid confusion with the existing field OrderQty. Such a field currently does not exist.

Option	Behavior	Best Practice
1	OrderQty is interpreted as new total investor quantity	X
2	OrderQty is interpreted as remaining quantity	
3	OrderQty is interpreted as positive/negative delta value to increase/decrease quantity	

3.2.2.4 Handling of In-Flight Modifications

The issue at hand is whether exchanges should support in-flight modifications (IFMs) with FIX or not. The FIX specification already provides support for IFMs. Although the issue itself is straightforward, it contains dependencies to the previously described issues for order modification. The recommendation is to allow IFMs based on the recommendations for the other issues. This means that, upon modification, orders should continue to exist. The field OrderQty is seen as one of the key quantity fields and should be interpreted as the new total investor quantity for a modification.

The alternative is to prevent IFMs from an exchange point of view. This behavior is deemed to be rather restrictive although it protects the user from unintended modifications. It should also be noted that FIX already offers the two fields TransactTime and OrigOrdModTime that can be used to enable the user to prevent acting upon outdated information.

An additional concern with the prevention of IFMs by the exchange is the possibility of a race condition for a user trying to modify an order. The situation can arise in very liquid markets where the order cannot be modified as long as a partial fill occurs after the user sends the modification request but before the exchange is able to process it.

Option	Behavior	Best Practice
1	Allow in-flight modifications	X
2	Prevent in-flight modifications	

3.2.3 Order Entry and Modification from Different Sources

There are a number of scenarios in the FIX specification (see Volume 4, *Order State Change Matrices, Scenarios E Unsolicited/Reinstatement*) where verbal communication is used for order entry, modification and cancellation. Primary reason is a disruption in the electronic communications link between the user and the exchange. The co-existence of multiple, electronic (FIX and non-FIX) and non-electronic (voice, fax) interfaces is a likely scenario for exchanges. Non-FIX, proprietary interfaces still dominate the exchange environment.

Not all exchanges permit verbal communication as it also depends on the specific regulatory environment. If it is supported, it is typically conducted by Market Operations personnel that have access to the order book. Execution reports are to be generated for these transactions and are to be submitted back to the participant as soon as the electronic link becomes available again. This should be part of the reliable transport mechanism that exchanges typically offer.

Issues arise from scenarios where interfaces can be used arbitrarily to enter, modify or cancel one and the same order. This is likely to be a common practice for existing exchanges starting to offer access through a FIX interface. This is typically done in addition to and not as a replacement of the existing proprietary interface. The at least temporary co-existence of both with a potential difference in scope is a requirement for these exchanges. It is not

desirable to have to restrict interactions with the exchange to one or the other interface for a single order. Users are interested in a smooth transition where they can use any interface to access their orders.

The following sections discuss the key issues in the area of order entry and modification from different sources, implementation options and provide recommendations for best practices.

3.2.3.1 Order Identification across FIX and Non-FIX Interfaces

Section *3.1.1.1 Mandatory vs. Optional Nature of ClOrdID and OrderID* discusses various models for order identification and recommends to make OrigClOrdID and OrderID conditionally required for order modification and deletion. The issue here is whether models should be the same across different interfaces or not. The recommendation is to maximize interoperability by offering the same model across all interfaces of a single exchange. It is recognized that this could be a challenge for existing exchange systems.

Option	Behavior	Best Practice
1	Same model for order identification for every interface	X
2	One model per interface with the option of common identification (possibly limited interoperability)	

3.2.3.2 Execution Reports across FIX and Non-FIX Interfaces

Standard behavior in FIX is to return execution reports for order entries, modifications and cancellations as well as for fills. In the case of multiple different interfaces, the question is whether interoperability should automatically be supported by sending the report multiple times. The recommendation is to foresee duplicates (drop-copies) of execution reports only on an optional basis to increase interoperability and fault-tolerance between FIX and non-FIX interfaces. Drop-copies of execution reports sent across other sessions should avoid to include the client order identification as it cannot be guaranteed that both locations use distinct numbering schemes.

Standard behavior is to only send the execution back over the interface that issued the request. This means in practice, for example, that an order entry or modification via a proprietary interface does not necessarily lead to a FIX ExecutionReport message via the FIX interface and vice versa. Drop-copies for fills could also be relayed by means of the TradeCaptureReport message. This is the preferred behavior for fills unless the receiver of the drop-copies requires an updated order status.

Option	Behavior	Best Practice
1	Multiple execution reports (all interfaces)	
2	Multiple execution reports (all interfaces) for selected business transactions	
3	Execution report only via interface issuing request	X

3.3 Performance

3.3.1 Message Bundling

FIX returns separate ExecutionReport messages for new order entries, partial fills and order cancellations whereas exchanges often bundle responses for orders that do not rest on the book and for multiple fills occurring in a single match operation. The main reason for message bundling is to optimize performance by reducing latency and the number of technical messages that need to be transmitted for a specific business transaction. Some exchanges offer the possibility to switch off response messages to certain request altogether, e.g. MassQuotes.

The following sections discuss the key issues in the area of message bundling, implementation options and provide recommendations for best practices.

3.3.1.1 Order is (Partially) Filled upon Hitting the Book

In an electronic trading environment, especially for very liquid securities, the probability of a partial fill upon entry or modification of an order is fairly high. FIX currently foresees at least two execution reports for these cases. The first confirms the request to enter or modify an order and echoes back a number of input fields, whereas the second reports the (partial) fill. The transaction boundary of many exchanges will actually be such that the matching engine never assigns the status "New" to an order as it can be matched immediately against another order in the book. The first, externally actionable status is "(Partially) Filled".

The (artificial) separation of this single transaction across two ExecutionReport messages might trigger an obsolete action by the user's trading software after the first execution report. This could be the case if the second execution report (with the same transaction time) contains a complete fill. The user cannot know that the second report is coming and might try to cancel the order resulting in an error message from the exchange.

The message flows should be consistent with the transaction model of an exchange, i.e. the recommendations here relate to additional message flow models and do not intend to replace the existing model in FIX. For example, if an exchange assigns the status "partially filled" to a new order due to immediate matching without ever explicitly communicating a previous status, it should be allowed to issue only a single execution report covering this information. If the new order cannot be immediately matched and rests on the book, the recommendation is to follow the existing model in FIX with separate messages to confirm order entry and report the execution taking place at a later point in time.

A single execution report with execution type "New" in conjunction with an order status other than "New" would require an adaptation of current FIX rules regarding conditionally required fields (e.g. LastQty is only required for ExecType="Trade"). The recommendation is therefore to use the execution type "Trade" instead for orders being (partially) filled upon hitting the book. The consequence for the user is that he has to prepare for two different ExecutionReport messages as response to an order entry, depending on whether an order was immediately executed or rests on the book.

Option	Behavior	Best Practice
1	Separate execution reports for order status "New" and "(Partially) Filled"	
2	Single execution report with execution type "Trade" and order status "(Partially) Filled"	X

3.3.1.2 Automatic Order Cancellations (IOC/FOK)

Orders can be given specific attributes to prevent them from resting on the book. An Immediate-or-Cancel (IOC) order by definition can only be executed if it is (partially) filled upon hitting the book. Regardless of the amount that was filled (0-100%), the remainder of the order is to be cancelled. A Fill-or-Kill (FOK) order is a special case where the fill percentage needs to be 100%. In this case, the order only needs to be cancelled if it was not filled as a partial fill cannot occur.

Again, FIX currently prescribes that every order status should be relayed as a separate message. However, most exchanges offering IOC and/or FOK orders will provide "Cancelled", "Expired" or "Filled" as the first external status that can be acted upon. The entry, matching and cancellation represents a single, atomic transaction for these exchanges.

The recommendation is to allow a corresponding message flow where a single ExecutionReport message is returned for an IOC or FOK order. The execution type should be "Trade" and the order status is either "Cancelled"[2] (IOC with partial execution or FOK with no execution) or "Filled" (IOC or FOK with complete execution). The quantity field CumQty in the execution report shows which amount has been executed and LeavesQty is set to zero.

[2] FIX currently defines "Expired" as order status when an "Order has been cancelled in broker's system due to time in force instructions" which applies to IOC and FOK orders. Order status "Cancelled" is better suited for other events causing an order to be deleted by the exchange or the user.

There should be new entries in the Order State Change Matrix showing the "final status" approach for IOC and FOK orders. The description of the order status "Expired" needs to exclude IOC and FOK orders to avoid ambiguities.

Option	Behavior	Best Practice
1	Separate execution reports for order status "New" and "(Partially) Filled" and "Cancelled"	
2	Up to two execution reports, one for execution type "Trade" and order status "(Partially) Filled" (IOC or FOK) or "Cancelled" (only FOK) and another for order status "Cancelled" (only IOC)	
3	Single execution report with execution type "Trade" and order status "Cancelled", quantity field CumQty shows if (partial) fill occurred	X

3.3.1.3 Multiple Fills of an Order in a Single Match Operation

As is the case with the order status, FIX currently prescribes that every (partial) fill[3] should be relayed as a separate message. Although this may be a reasonable model where an order needs to be worked before executed, it does not fit well with electronic matching systems. Similar to orders being (partially) filled upon hitting the book, users risk acting on status information that is not externally correct. The order might already be completely filled but the user is unaware of this until he has processed the last execution report related to the order.

The recommendation is to allow a corresponding message flow where a single ExecutionReport message is returned for all partial fills that occur in a single transaction of the matching engine and are related to a single order. This ensures that the most recent order status is immediately available to the user. The user can choose to react upon this status prior to processing the information about the partial fills, thereby reducing latency in his response.

Bundling partial fills at multiple price levels in a single message either requires a repeating group within the ExecutionReport message or a new message type, e.g. MassExecutionReport. The existing ExecutionReport message is a central component of FIX response messages so that the impact of a structural change to this message would be very significant and thus not desirable.

Option	Behavior	Best Practice
1	Separate execution reports for every partial fill	
2	Single execution report for all partial fills occurring in a single match	X

3.3.1.4 Configurability of Response Messages

Performance considerations target the overall number of messages that are required for electronic trading. In some areas such as quoting, it is common practice for some exchanges to offer configurability of response messages in certain areas. The general recommendation is that response messages should be configurable for selected requests. FIX currently only offers this configurability for mass quote messages where the user can specify whether he wants QuoteStatusReport messages for all, erroneous or none of the individual quotes. The MassQuote message itself is always acknowledged through a MassQuoteAcknowledgement message. Individual quotes sent with the Quote message are acknowledged by the QuoteResponse message. This should both be configurable.

A complete list of FIX messages where the response level should be configurable still needs to be defined. Configurability may be something which is established at the beginning of a session or for a specific group of products depending on the preferences of the trade rather than specified on every submission.

Option	Behavior	Best Practice
1	Response messages are always delivered	
2	Configurability of response messages for selected requests	X

[3] In fact, FIX does not require that every partial fill is relayed itself as it allows partial fills to be bundled per price level (LastPx). While marketplaces using counterparty disclosure may need to relay (partial) fills per counterparty, others may aggregate them. The latter is assumingly the best practice for ExecutionReport messages.

3.3.2 Pending Order States

FIX provides a number of order states prefixed with "Pending" that allow the sender to be informed that the receiver is working on a request without having completed it. Electronic exchanges typically act on requests immediately and only return the order status upon completion. This also reduces the overall number of messages required and is beneficial to the performance. However, there are special situations for new orders where a delayed entry into the order book is desirable, e.g. for the purpose of risk management.

Intermediate execution reports might be applicable to floor trading environments but are not considered best practice for electronic exchanges. The field WorkingIndicator (636) can be used for new orders to express that an order has been received but is not (yet) executable. An example is a stop order where the trigger condition has not been reached. In the case of stop orders, it is appropriate to return an ExecutionReport message as an acknowledgement with an order status "New" and WorkingIndicator="No". When the stop order is placed on the orderbook, another ExecutionReport message would be sent indicating that the order is now in working status.

In general, the recommendation is to return only a single execution report upon completion or rejection of a request. This equally applies to order entries and modifications as well as to order cancellations. Orders thus are not recommended to have a pending status except in situations of delayed book entry such as risk management.

Option	Behavior	Best Practice
1	Single execution report upon completion or rejection of request	X
2	Intermediate execution report upon request acceptance and final execution report upon completion or rejection of request	

3.3.3 Restatement of Orders

FIX allows information about the status of active orders to be sent at the beginning or end of a business day on an optional basis. Some electronic exchanges offer restatement services in case of multiple trading sessions or to enable consistency checks by the user. It is not used on a general basis by exchanges as it also increases the message volume for the end-of-day and start-of-day processing. In any case, the restatement should only apply to orders that are still in the book or have been modified or removed by the exchange, i.e. that have not been previously filled or cancelled. However, mandatory, unsolicited transfer of mass report messages is inefficient and not considered best practice.

The following sections discuss the key issues in the area of restatement of orders, implementation options and provide recommendations for best practices.

3.3.3.1 Restatement of Orders at the Beginning of a Trading Session

The restatement of orders could occur at the beginning of the first trading session of a business day or also at the beginning of a subsequent session, e.g. after-hours trading. The issue is about the unsolicited restatement of orders as requests by the user for information about specific orders should always be possible. The exchange is likely to discourage users from requesting information about their orders as a standard practice. In terms of unsolicited restatements, the recommendation is to offer this only on an optional basis (if at all) and to restrict them to active orders. The user can then reconcile the exchange information with his own to confirm his exposure through orders in the market.

Option	Behavior	Best Practice
1	Unsolicited restatement of active orders (if at all)	X
2	Unsolicited restatement of all orders of previous trading session	

3.3.3.2 Restatement of Orders at the End of a Trading Session

FIX provides the execution type and order status "Done for Day" to communicate the final properties (especially quantities, price) of an order. The recommendation is similar to the case of the beginning of a trading session, namely that this should only be provided on an optional basis. The recommendation differs in terms of scope, namely that (if offered) it should include all orders of the current trading session. This service can be of use for batch runs of exchanges after trading but the information could also be provided through a file transfer instead of individual messages.

Option	Behavior	Best Practice
1	Unsolicited restatement of active orders	
2	Unsolicited restatement of all orders of current trading session (if at all)	X

3.3.3.3 Re-request of Orders During a Trading Session

Mass restatement of orders during a trading session are special cases and are not considered best practice in general. Such services might be useful for users without order repositories that would otherwise not be able to continue trading after a severe problem on their side. In general, it is recommended to only offer a solicited service and to restrict the re-requests to individual orders or at least to a small subset of orders. The performance impact might be considerable and affect other users, for example if a service were to be offered that provided a complete history of all order transactions leading to multiple messages per order. This recommendation only pertains to the request of all orders of a user and not to the standard usage of the OrderMassStatusRequest message together with specific criteria to return a subset of all orders.

Option	Behavior	Best Practice
1	Restatement request for active orders	N/A
2	Restatement request for all orders of current trading session	N/A
3	Restatement request for complete history of all orders of current trading session	N/A

3.3.4 Corporate Actions

Corporate actions can be seen as s special case for the restatement of orders. In this case, the exchange might automatically modify or even cancel orders of a user for a security for which a corporate action, e.g. stock split took place. This would typically only apply to a subset of all orders a user has in the market. FIX allows to send updated information about the status of active orders after corporate actions at the beginning of a business day on an optional basis.

Some electronic exchanges adjust the price and/or quantity of active orders whereas other automatically delete any orders for the security. Practices also differ whether the exchange subsequently sends corresponding messages to the user or not.

The recommendation is that the deletion of orders as well as the adjustment of order fields are both valid approaches. The reason for the deletion or modification of orders by the exchange needs to be communicated with the electronic restatement. The ability of an exchange to offer one of these approaches possibly depends on given rules and regulations. The lack of electronic confirmation is seen as a problem for users and can lead to data inconsistencies. Harmonization of practices to confirm corporate actions is beneficial to the broker trading on multiple exchanges.

Option	Behavior	Best Practice
1	Deletion of all active orders	
2	Deletion of all active orders and confirmation to user	X
3	Adjustment of order price and/or quantity	
4	Adjustment of order price and/or quantity and confirmation to user	X

3.4 Market Data

3.4.1 Trading Session and Security Status

The identification of trading sessions in the context of an overall model for a marketplace is discussed in section *3.1.3 Trading Session Identification*. This covers the requirements for static or reference data in this area and defines the terms trading session and trading phase. A different aspect is the usage of reference data in the context of status information being disseminated by the exchange. FIX also allows a current status to be assigned to trading sessions and securities. This section discusses recommendations for exchange environments in this area.

State changes are typically disseminated as part of market data. The complexity stems from the varying scope of a status, i.e. the applicability of a status can range from a single security all the way to an entire trading session. This also depends on the asset class and the market model, i.e. trading rules of the exchange. For example, all securities within a market segment might become available for trading at a single point in time or it might be the case that groups of securities are activated separately by means of auctions that are a few minutes apart.

Furthermore, the status can be the specific trading session or trading phase itself or the status could be more generic (e.g. halted) or more specific (e.g. order book balancing part of a trading phase such as an opening auction). States are typically associated with rules, for example rules that govern permissible order management activities (order entry, modification, cancellation), orderbook transparency or reporting of privately negotiated trades. Users often use state information to trigger behavior and define rules for their local applications.

All these examples illustrate the breadth of requirements that need to be covered in an exchange environment. FIX provides the two messages TradingSessionStatus and SecurityStatus to convey status information. Within these messages, there are a number of fields that can be used. The following table lists the messages and most relevant fields for status information.

Message	Field / Component Block	Req'd	Description	Data Type
TradingSessionStatus	TradingSessionID	Yes	Name of a trading session	String
TradingSessionStatus	TradingSessionSubID	No	Name of a trading phase	String
TradingSessionStatus	TradSesStatus	Yes	Status	Int
TradingSessionStatus	Instrument	No	Security	N/A
SecurityStatus	Instrument	Yes	Security	N/A
SecurityStatus	TradingSessionID	No	Name of a trading session	String
SecurityStatus	TradingSessionSubID	No	Name of a trading phase	String
SecurityStatus	SecurityTradingStatus	No	Security specific status	Int
SecurityStatus	HaltReason	No	Explanation for halted status	Char

Both messages allow trading sessions, trading phases and securities to be specified. The scope of the status applies to different levels of granularity in the TradingSessionStatus message, depending on the existence of fields specifying a trading phase and/or security. The scope of the SecurityStatus message is restricted to securities and cannot apply to an entire trading session.

The actual status values (TradSesStatus and SecurityTradingStatus) are of data type "int", i.e. they are pre-defined as opposed to the names of trading sessions and phases. There are only very few, rather generic values for the field TradSesStatus whereas there is a long list of values for SecurityTradingStatus, some of which are equally generic.

It is recommended to use the SecurityStatus and not the TradingSession message to convey a security specific status, also because of the larger granularity of status values available in the former message. However, there is a large

number of values being used by exchanges and most of them are currently not covered in FIX. The recommendation is therefore either to change the data type of SecurityTradingStatus from "int" to "String" or to add a specific value 99=Other as well as a new field SecurityTradingStatusText that can contain the exchange specific value.

The FIX messages MarketDataSnapshotFullRefresh and MarketDataIncrementalRefresh currently do not convey the status of a security. However, some exchanges bundle a state change with an orderbook situation, for example a trading halt in a single security due to a trade price exceeding a pre-defined range. It is recommended to add appropriate fields to the market data messages to cover this usage case. If used, the market data messages should not replace the SecurityStatus message to convey status information but be used in addition to it.

The following table shows which message and fields to use for which intended scope of a status.

Scope of Status	**Message**	**Fields / ComponentBlock**
Single security	SecurityStatus, MarketDataSnapshot-FullRefresh, MarketData-IncrementalRefresh	Instrument, Trading SessionID, TradingSessionSubID, SecurityTradingStatus*, HaltReason*
Group of securities	TradingSessionStatus	Trading SessionID, TradingSessionSubID, TradSesStatus, Instrument
Trading phase	TradingSessionStatus	Trading SessionID, TradingSessionSubID, TradSesStatus
Trading session	TradingSessionStatus	TradingSessionID, TradSesStatus

*Field is currently only part of SecurityStatus message.

The status of a group of securities is a special case that is not explicitly covered in FIX today. For reasons of efficiency, it is very important for an exchange to be able to convey status information for a group of securities in a single message instead of having to send the same status for the individual securities that make up the group. Another potential usage is the identification of a group of securities in the context of market data requests. The existing fields such as Product (460), CFICode (461) or SecurityType (167) have pre-defined values and are not suited to cover exchange-specific groupings. It is recommended to introduce a new field SecurityGroup to the Instrument component block to cover this functionality.

The status of a market segment is identical to the status of the currently active trading session of this segment.

3.4.2 Events

The dissemination of status information for trading sessions and securities does not cover all requirements of a marketplace. There are also events (or warnings) that need to be published for different scopes, ranging from an entire market segment to a single security.

A marketplace may need to relay suspensions of participants, users, messages and other entities. A marketplace might offer 24x7 trading without actual session changes but still requiring to communicate changes such as the beginning of a new business day. There is no FIX support for such mechanisms.

An event could be scheduled, event driven or manually imposed to relay messages such as:

- "The market is re-opening (after trading halt) in five minutes".
- "The opening auction will be extended three minutes due to market order imbalance"
- "The market is opening, please remove any indicative quotes"

The existing FIX message "News" is far too generic and not suited for this purpose. It is therefore recommended to add a new MDEvent message or new fields to the SecurityStatus message which contain the following information.

- Type of book, class of service and trade date to which events apply
- Repeating group of events
  - Market segment, trading session and phase to which the event applies
  - List of securities (or single security) to which the event applies
  - Event type and description
  - Timestamp of event communication
- Information related to application queuing behavior for outstanding events

The detailed definition of new messages and fields goes beyond the scope of this document and requires a separate, formal document to be presented to the FIX Global Technical Committee for approval.

3.5 Additional Topics

3.5.1 Order Rejections

FIX returns the state of an order together with the current quantity values in case of an order modification being rejected. Exchanges possibly reject order modifications prior to accessing the actual order and cannot or must not return information about an existing order. Examples for this behavior include specific trading phases and authorization profiles of users restricting order visibility and access.

The recommendations for best practices cover new orders, order modifications and order cancellations. Order specific rejections should always use the ExecutionReport message (for New Order Single) or the OrderCancel-Reject message (for Order Cancel/Replace and Order Cancel). This includes the case of a non-existent order.

Issues arise due to the mandatory nature of certain fields in the ExecutionReport message which FIX requires to reject new order entries. Any rejection with this message requires to return the current order status and executed quantity (OrdStatus and CumQty are mandatory fields). The alternative would be to introduce a new message to cover the case of rejecting a new order entry. However, a new message does not seem desirable for this purpose.

Order modifications have the same issue due to the order status being mandatory in the OrderCancelReject message that is the required response for a rejection.

The following sections discuss the key issues in the area of order rejections, implementation options and provide recommendations for best practices.

3.5.1.1 Order Rejections During Specific Trading Phases

Exchanges typically distinguish different phases during a single trading session. Trading phases can have different rules in terms of permissible interactions between the user and the exchange as well as different levels of order visibility. The rules are not order specific and can thus be enforced without having to access the order itself. The main point of the response is the rejection itself and its reason. It adds little value to also return current information about the order. From an exchange perspective, this would require additional access to the order and have an impact on performance. This is especially relevant for transitions from a phase disabling order modifications to a phase where they are allowed again. Users might attempt to be the first to modify their orders and start sending modification requests around the time of the transition.

The recommendation is to use the BusinessMessageReject message for this purpose. The appropriate field value is 4, i.e. “Application not available” to indicate that the reason is not related to the content of the order.

Option	Behavior	Best Practice
1	Return order status and quantity fields upon rejection	
2	Return "Rejected" as execution type and/or order status and echo quantities of request if applicable	
3	Return (new) message "NewOrderReject" with "Rejected" as execution type	
4	Return a BusinessMessageReject message with reject reason 4 = "Application not available"	X

3.5.1.2 Order Rejections due to Lack of Authorization

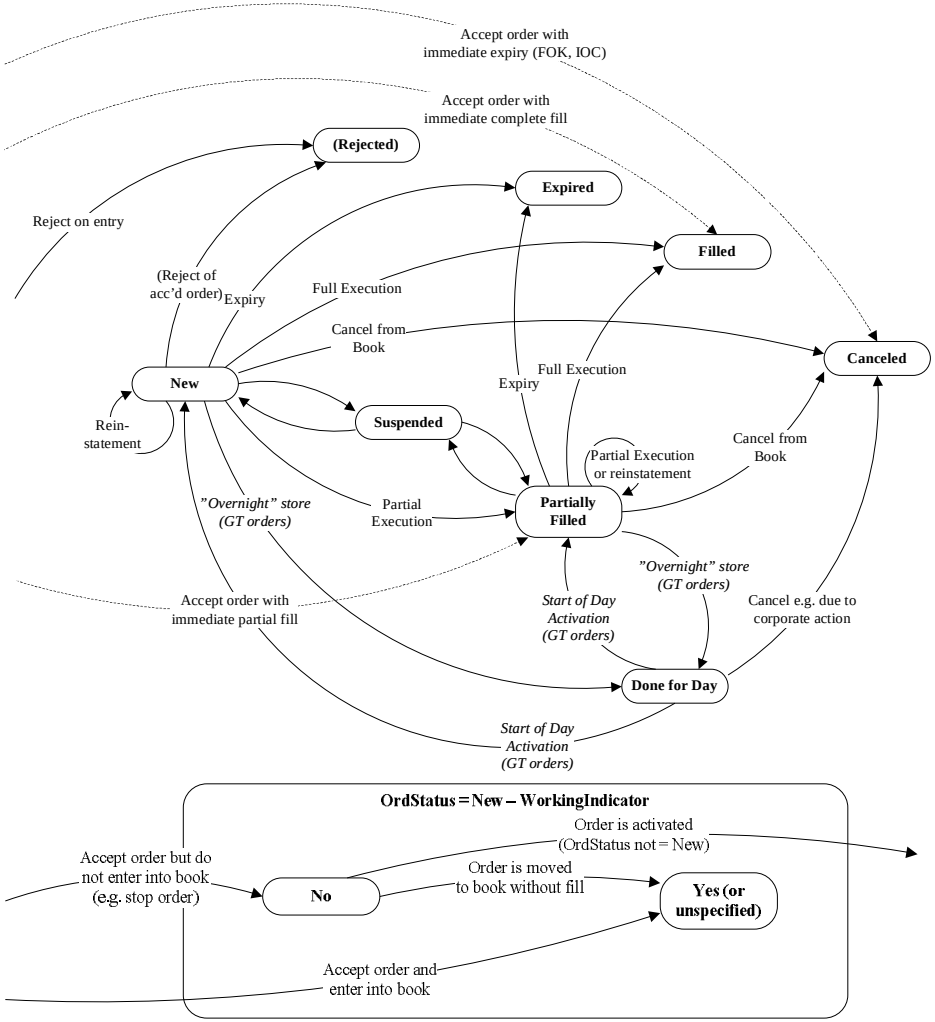
Exchanges typically maintain a complex set of authorizations that determine whether the user is entitled to view or modify an order, among other actions. Lack of authorization means that a request will be rejected. Again, the corresponding rules are specific to a user but not to the order. It is therefore likely that the exchange will check the authorization prior to accessing the order itself. In this case, the exchange must not return the status and/or quantities of the order as required by the mandatory nature of the FIX fields in the corresponding rejection messages.

The recommendation is to use the BusinessMessageReject message for this purpose. The appropriate field value is 6, i.e. "Not authorized" to indicate that the reason is not related to the content of the order. This value has been introduced with FIX 5.0.

Option	Behavior	Best Practice
1	Return order status and quantity fields upon rejection	
2	Return "Rejected" as execution type and/or order status and echo quantities of request if applicable	
3	Return (new) message "NewOrderReject" with "Rejected" as execution type	
4	Return a BusinessMessageReject message with reject reason 6 = "Not authorized"	X

4 Message Flows

4.1 Order State Change Event Diagram



4.2 Identification

4.2.1 Order Identification

4.2.1.1 Enforcement of ClOrdID Uniqueness by the Exchange

Option 1: Check ClOrdID and reject orders with duplicate values

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3	New Order(X)		A			12,000			Second order carries the same ClOrdID
4		Execution (X)	A	Rejected	New	10,000	1,000	9,000	OrdRejReason = duplicate order, current status and quantities of existing order are returned

Option 2: Do not check ClOrdID and treat orders with duplicate values as separate entities

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID of first new order
3	New Order(X)		A			12,000			Second order carries the same ClOrdID
4		Execution (X)	B	New	New	12,000	0	12,000	Exchange issues B as OrderID of second new order

4.2.1.2 Persistence of exchange issued order identification

Option 1: Issue single order identification and optionally change it during the order lifetime

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y,X)	B, A	Replace	Partially Filled	12,000	1,000	11,000	Order continues to exist with an executed quantity of 1,000, client order ID changes from X to Y, exchange order ID changes from A to B
6		Execution (Y)	B	Trade	Partially Filled	12,000	2,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 2: Issue two order identifications (private/static and public/dynamic) whereby the private one does not change

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, Secondary-OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A, N	New	New	10,000	0	10,000	Exchange issues A as private/ static order ID and N as public/ dynamic order ID
3		Execution (X)	A, N	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A/N			12,000			Request to increase total order qty from 10,000 to 12,000. Sender uses either static value A or current public value N to identify order.
5		Execution (Y,X)	A, M	Replace	Partially Filled	12,000	1,000	11,000	Order continues to exist with an executed quantity of 1,000, client order ID changes from X to Y, public/dynamic exchange order ID changes from N to M, private exchange order ID A does not change
6		Execution (Y)	A, M	Trade	Partially Filled	12,000	2,000	10,000	Execution for 1,000

4.2.2 Trading Session Identification

4.2.2.1 Dissemination of Trading Sessions and Trading Phases

The following example shows a market segment with two separate sessions with the same four trading phases.

Time	Message Received (TradSesReqID)	Message Sent (TradSesReqID)	Trading SessionID	Trading SessionSubID	TradSesStartTime	TradSesEndTime	Comment
1	TradingSessionListRequest(X)						Request reference data regarding trading sessions and related trading phases
2		TradingSessionList(X)	MORNING		20061116-09:00:00	20061116-12:00:00	First session from 9am – 12pm
2			AFTERNOON		20061116-14:00:00	20061116-17:00:00	Second session from 2pm – 5pm
3		TradingSessionList(X)	MORNING	PRE-OPEN			Phase 1 of first session
3			MORNING	OPENING			Phase 2 of first session
3			MORNING	TRADING			Phase 3 of first session
3			MORNING	CLOSING			Phase 4 of first session
3			AFTERNOON	PRE-OPEN			Phase 1 of second session
3			AFTERNOON	OPENING			Phase 2 of first session
3			AFTERNOON	TRADING			Phase 3 of first session
3			AFTERNOON	CLOSING			Phase 4 of first session

The following example shows a market segment with three concurrent sessions with the same three trading phases.

Time	Message Received (TradSesReqID)	Message Sent (TradSesReqID)	Trading SessionID	Trading SessionSubID	TradSesStartTime	TradSesEndTime	Comment
1	TradingSessionListRequest(X)						Request reference data regarding trading sessions and related trading phases
2		TradingSessionList(X)	SESSION 1		20061116-09:00:00	20061116-17:00:00	First session from 9am – 5pm
2			SESSION 2		20061116-09:05:00	20061116-17:05:00	Second session from 9:05am – 5:05pm
2			SESSION 3		20061116-09:10:00	20061116-17:10:00	Third session from 9:10am – 5:10pm
3		TradingSessionList(X)	SESSION 1	PRE-OPEN			Phase 1 of first session
3			SESSION 1	TRADING			Phase 2 of first session
3			SESSION 1	CLOSING			Phase 3 of first session
3			SESSION 2	PRE-OPEN			Phase 1 of second session
3			SESSION 2	TRADING			Phase 2 of second session
3			SESSION 2	CLOSING			Phase 3 of second session
3			SESSION 3	PRE-OPEN			Phase 1 of third session
3			SESSION 3	TRADING			Phase 2 of third session
3			SESSION 3	CLOSING			Phase 3 of third session

4.3 Order Handling

4.3.1 Order Lifetime

4.3.1.1 Status Requests for Inactive Orders

Option 1: Return ExecutionReport with status and quantity information for orders that have become inactive on the current business day

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Filled	10,000	10,000	0	Execution for 10,000
4	Status Request(X)		A						Request for information
5		Execution (X)	A	Order Status	Filled	10,000	10,000	0	Order has been completely filled with a quantity of 10,000

Option 2: Reject status request in the same way as an invalid order identification

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Filled	10,000	10,000	0	Execution for 10,000
4	Status Request(X)		A						Request for information
5		Execution (X)	A	Order Status	Rejected	0	0	0	OrdRejReason = unknown order

4.3.1.2 Modification of Inactive Orders

Option 1: Accept increase of order quantity and set order status back to "partially filled"

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Filled	10,000	10,000	0	Execution for 10,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y,X)	A	Replace	Partially Filled	12,000	10,000	2,000	Order has been completely filled with a quantity of 10,000

Option 2: Reject modification request in the same way as an invalid order identification

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Filled	10,000	10,000	0	Execution for 10,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y,X)	A	Replace	Rejected	12,000	0	0	OrdRejReason = unknown order, order quantity is echoed back

4.3.2 Order Modification

4.3.2.1 Order Persistence

Option 1: Order continues to exist, relevant fields are modified, exchange does not issue new order identification (OrderID)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y,X)	A, A	Replace	Partially Filled	12,000	1,000	11,000	Order continues to exist with an executed quantity of 1,000, client order ID changes from X to Y, exchange order ID remains A
6		Execution (Y)	A	Trade	Partially Filled	12,000	2,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 2: Order is replaced by a new order with a new order identification (OrderID), old order contains previously executed quantity, executed quantity of the new order is zero

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000 (not to set the remaining quantity to 12,000)
5		Execution (Y,X)	B, A	Replace	New	11,000	0	11,000	Order is replaced with a new one from the exchange with an executed quantity of 0, client order ID changes from X to Y, exchange issues order ID B for new order
6		Execution (Y)	B	Trade	Partially Filled	11,000	1,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

4.3.2.2 Interpretation of Field OrderQty

Option 1a: OrderQty is interpreted as new total investor quantity (persistent order)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y,X)	A,A	Replace	Partially Filled	12,000	1,000	11,000	Previous order execution quantity of 1,000 is subtracted from total, remaining quantity is calculated, exchange order ID unchanged
6		Execution (Y)	A	Trade	Partially Filled	12,000	2,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 1b: OrderQty is interpreted as new total investor quantity (non-persistent order)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y,X)	B,A	Replace	New	11,000	0	11,000	Previous order execution quantity of 1,000 is subtracted from new total order quantity and stays with order A which is removed from the book, exchange issues new order with order ID B, executed quantity is set to zero
6		Execution (Y)	B	Trade	Partially Filled	11,000	1,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 2a: OrderQty is interpreted as remaining quantity (persistent order)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to set remaining order qty to 12,000
5		Execution (Y,X)	A, A	Replace	Partially Filled	13,000	1,000	12,000	Previous order execution quantity of 1,000 is added to total, remaining quantity is set to requested value, exchange order ID unchanged
6		Execution (Y)	A	Trade	Partially Filled	13,000	2,000	11,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 2a: OrderQty is interpreted as remaining quantity (non-persistent order)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to set remaining order qty to 12,000
5		Execution (Y,X)	B, A	Replace	New	12,000	0	12,000	Previous order execution quantity of 1,000 stays with order A which is removed from the book, exchange issues new order with order ID B, remaining quantity is set to requested value, executed quantity is set to zero
6		Execution (Y)	B	Trade	Partially Filled	12,000	1,000	11,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 3a: OrderQty is interpreted as positive/negative delta value to increase/decrease quantity (persistent order)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			2,000			Request to increase order qty by 2,000
5		Execution (Y,X)	A, A	Replace	Partially Filled	12,000	1,000	11,000	Total and remaining quantity are increased by requested quantity, exchange order ID unchanged
6		Execution (Y)	A	Trade	Partially Filled	12,000	2,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 3b: OrderQty is interpreted as positive/negative delta value to increase/decrease quantity (non-persistent order)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			2,000			Request to increase order qty by 2,000
5		Execution (Y,X)	B, A	Replace	New	11,000	0	11,000	Total and remaining quantity are increased by requested quantity, exchange issues new order with order ID B
6		Execution (Y)	B	Trade	Partially Filled	11,000	1,000	10,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

4.3.2.3 Handling of In-Flight Modifications

The following table illustrates the different options for in-flight modifications (IFMs) under the two and three different models for order persistence and order quantity interpretation respectively.

Option	Allow IFM	Order Persistence	Order Quantity Interpretation
1a	YES	Retain existing order	New total investor quantity
1b	YES	Retain existing order	Remaining quantity
1c	YES	Retain existing order	Positive/negative delta value
1d	YES	Issue new order	New total investor quantity
1e	YES	Issue new order	Remaining quantity
1f	YES	Issue new order	Positive/negative delta value
2	NO	N/A	N/A

Option 1a: Allow in-flight modifications (persistent order, new total investor quantity)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Execution (Y,X)	A, A	Replace	Partially Filled	12,000	5,000	7,000	Use most recent order execution quantity of 4,000, set total order quantity to requested value, calculate remaining quantity
7		Execution (Y)	A	Trade	Partially Filled	12,000	6,000	6,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 1b: Allow in-flight modifications (persistent order, remaining quantity)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to set remaining order qty to 12,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Execution (Y,X)	A, A	Replace	Partially Filled	13,000	5,000	8,000	Concurrent order execution quantity of 4,000 is subtracted from requested remaining quantity, previous order execution quantity of 1,000 is added to total quantity
7		Execution (Y)	A	Trade	Partially Filled	13,000	6,000	7,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 1c: Allow in-flight modifications (persistent order, delta quantity)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			2,000			Request to increase order qty by 2,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Execution (Y,X)	A, A	Replace	Partially Filled	12,000	5,000	7,000	Total and most recent remaining quantity are increased by requested quantity
7		Execution (Y)	A	Trade	Partially Filled	12,000	6,000	6,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 1d: Allow in-flight modifications (non-persistent order, new total investor quantity)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Execution (Y,X)	B,A	Replace	New	7,000	0	7,000	Previous order execution quantity of 5,000 is subtracted from new total order quantity and stays with order A which is removed from the book, exchange issues new order with order ID B, executed quantity is set to zero
7		Execution (Y)	B	Trade	Partially Filled	7,000	1,000	6,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 1e: Allow in-flight modifications (non-persistent order, remaining quantity)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to set remaining order qty to 12,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Execution (Y,X)	B,A	Replace	New	8,000	0	8,000	Complete previous order execution quantity of 5,000 stays with order A which is removed from the book, exchange issues new order with order ID B, remaining quantity is set to requested value after subtraction of concurrent order execution quantity of 4,000, executed quantity is set to zero
7		Execution (Y)	B	Trade	Partially Filled	8,000	1,000	7,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 1f: Allow in-flight modifications (non-persistent order, delta quantity)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID*)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			2,000			Request to increase order qty by 2,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Execution (Y,X)	B,A	Replace	New	7,000	0	7,000	Complete previous order execution quantity of 5,000 stays with order A which is removed from the book, exchange issues new order with order ID B, remaining quantity is increased by requested value executed quantity is set to zero
7		Execution (Y)	B	Trade	Partially Filled	7,000	1,000	6,000	Execution for 1,000

* OrigOrderID is not a standard FIX field but used here for illustration purposes

Option 2: Prevent in-flight modifications

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (Y)	A	Trade	Partially Filled	10,000	5,000	5,000	Execution for 4,000
6		Cancel Reject (Y,X)	A		Rejected				Replace cannot be executed due to partial match that occurred after replace request was issued
7		Execution (Y)	A	Trade	Partially Filled	10,000	6,000	4,000	Execution for 1,000

4.4 Performance

4.4.1 Message Bundling

4.4.1.1 Order is (Partially) Filled upon Hitting the Book

Option 1: Separate execution reports for order status "New" and "(Partially) Filled"

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000

Option 2: Single execution report with execution type "Trade" and order status "(Partially) Filled"

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	Partially Filled	10,000	1,000	9,000	Execution for 1,000

4.4.1.2 Automatic Order Cancellations (IOC/FOK)

Option 1a: Separate execution reports for order status "New" and "(Partially) Filled" and "Cancelled" - IOC, partially filled

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			Order is IOC
2		Execution(X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Immediate execution for 1,000
4		Execution (X)	A	Canceled	Canceled	10,000	1,000	0	

Option 1b: Separate execution reports for order status "New" and "(Partially) Filled" and "Cancelled" - FOK, no fill

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			Order is FOK
2		Execution(X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Canceled	Canceled	10,000	0	0	No execution

Option 2a: Up to two execution reports, one for execution type "Trade" and order status "(Partially) Filled" (IOC or FOK) or "Cancelled" (only FOK) and another for order status "Cancelled" (only IOC) - IOC, partially filled

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			Order is IOC
2		Execution (X)	A	New	Partially Filled	10,000	1,000	9,000	Execution for 1,000
3		Execution (X)	A	Canceled	Canceled	10,000	1,000	0	Cancellation of remainder

Option 2b: Up to two execution reports, one for execution type "Trade" and order status "(Partially) Filled" (IOC or FOK) or "Cancelled" (only FOK) and another for order status "Cancelled" (only IOC) - FOK, no fill

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			Order is FOK
2		Execution(X)	A	New	Canceled	10,000	0	0	No execution

Option 3a: Single execution report with execution type "Trade" and order status "Cancelled", quantity field CumQty shows if (partial) fill occurred - IOC, partially filled

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			Order is IOC
2		Execution (X)	A	New	Canceled	10,000	1,000	9,000	Execution for 1,000 and implicit cancellation of remainder

Option 3b: Single execution report with execution type "Trade" and order status "Cancelled", quantity field CumQty shows if (partial) fill occurred - FOK, no fill

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID, OrigOrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			Order is FOK
2		Execution(X)	A	New	Canceled	10,000	0	0	No execution

4.4.1.3 Multiple Fills of an Order in a Single Match Operation

Option 1: Separate execution reports for every partial fill

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000 at price P1
4		Execution (X)	A	Trade	Partially Filled	10,000	3,000	7,000	Execution for 2,000 at price P2
5		Execution (X	A	Trade	Filled	10,000	10,000	0	Execution for 7,000 at price P3

Option 2: Single execution report for partial fills occurring in a single match

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		MassExecution(X)*	A	New	Filled	10,000	10,000	0	Multiple executions for 1,000, 2,000 and 7,000 at price levels P1, P2, P3

* MassExecution is not a standard FIX message but used here for illustration purposes

4.4.2 Pending Order States

Option 1a: Single execution report upon completion or rejection of request (order entry and modification)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	ExecType	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Order has been entered into the book
3	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
4		Execution (X)	A	Replace	New	12,000	0	12,000	Order has been modified

Option 1b: Single execution report upon completion or rejection of request (order cancellation)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	ExecType	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Order has been entered into the book
3	Cancel Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
4		Execution (X)	A	Replace	New	12,000	0	12,000	Order has been modified

Option 2a: Intermediate execution report upon request acceptance and final execution report upon completion or rejection of request (order entry and modification)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	ExecType	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	Pending New	Pending New	10,000	0	10,000	Order is sent to external risk mgmt system for approval
3		Execution(X)	A	New	New	10,000	0	10,000	Order has been entered into the book
4	Replace Request(Y,X)		A			12,000			Request to increase total order qty from 10,000 to 12,000
5		Execution (X)	A	Pending Replace	Pending Replace	10,000	0	10,000	Quantity increase is sent to external risk mgmt system for approval, current quantities are provided
6		Execution (X)	A	Replace	New	12,000	0	12,000	Order has been modified

Option 2b: Intermediate execution report upon request acceptance and final execution report upon completion or rejection of request (order cancellation)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	ExecType	OrdStatus	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	Pending New	Pending New	10,000	0	10,000	Order is sent to external risk mgmt system for approval
3		Execution(X)	A	New	New	10,000	0	10,000	Order has been entered into the book
4	Cancel Request(Y,X)		A			10,000			
5		Execution (X)	A	Pending Cancel	Pending Cancel	10,000	0	10,000	Request to cancel order has been accepted
6		Execution (X)	A	Canceled	Canceled	12,000	0	0	Order has been canceled

Orders held off the book until activated (e.g. stop orders)

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Last Qty	Comment
1	New Order(X)					10 000				Entry of a stop (OrdType = 3), stop limit (OrdType = 4), At the Close (TimeInForce = 7), etc, order. I.e. 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)	A	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**). Note that if the conditions are met on entry, this WorkingIndicator is not sent.
3		Execution(X)	A	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)	A	Trade	Partially Filled	10 000	2 000	8 000	2 000	Execution of 2000
5		Execution(X)	A	Trade	Filled	10 000	10 000	0	8 000	Execution of 8000

Note that:

- A Stop order will change OrdType to Market
- A Stop Limit order will change OrdType to Limit

4.4.3 Restatement of Orders

4.4.3.1 Restatement of Orders at the Beginning of a Trading Session

Option 1: Unsolicited restatement of active orders

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
Day 1,1	New Order(X)					10,000			
Day 1,2		Execution (X)	A	New	New	10,000	0	10,000	
Day 1,3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
Day 1,4	New Order(Y)					5,000			
Day 1,5		Execution (Y)	A	New	New	5,000	0	5,000	
Day 1,6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
Day 2,1		Execution (X)	A	Restated	Partially Filled	10,000	1,000	9,000	Order X is still active from previous business day
Day 2,2		Execution (X)	A	Trade	Partially Filled	10,000	3,000	7,000	Execution for 2,000

Option 2: Unsolicited restatement of all orders of previous trading session

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
Day 1,1	New Order(X)					10,000			
Day 1,2		Execution (X)	A	New	New	10,000	0	10,000	
Day 1,3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
Day 1,4	New Order(Y)					5,000			
Day 1,5		Execution (Y)	A	New	New	5,000	0	5,000	
Day 1,6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
Day 2,1		Execution (X)	A	Restated	Partially Filled	10,000	1,000	9,000	Order X is still active from previous business day
Day 2,2		Execution (Y)	A	Restated	Filled	5,000	5,000	0	Order Y was filled on previous business day
Day 2,3		Execution (X)	A	Trade	Partially Filled	10,000	3,000	7,000	Execution for 2,000

4.4.3.2 Restatement of Orders at the End of a Trading Session

Option 1: Unsolicited restatement of active orders

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution (X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	New Order(Y)					5,000			
5		Execution (Y)	A	New	New	5,000	0	5,000	
6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
7		Execution (X)	A	Done for Day	Done for Day	10,000	1,000	9,000	Order X is still active and can trade on next business day

Option 2: Unsolicited restatement of all orders of current trading session

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution (X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	New Order(Y)					5,000			
5		Execution (Y)	A	New	New	5,000	0	5,000	
6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
7		Execution (X)	A	Done for Day	Done for Day	10,000	1,000	9,000	Order X is still active and can trade on next business day
8		Execution (Y)	A	Done for Day	Done for Day	5,000	5,000	0	Order Y has been completely filled

4.4.3.3 Re-request of Orders During a Trading Session

Option 1: Restatement request for active orders

Time	Message Received (ClOrdID, OrigClOrdID, MassStatusReqID*)	Message Sent (ClOrdID, OrigClOrdID, MassStatusReqID*)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution (X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	New Order(Y)					5,000			
5		Execution (Y)	A	New	New	5,000	0	5,000	
6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
7	MassStatusRequest(N)*								MassStatusReqType=6, only active orders with LeavesQty>0
8		Execution (X, N)	A	Restated	Partially Filled	10,000	1,000	9,000	Order X is still active
9		Execution (X)	A	Trade	Partially Filled	10,000	3,000	7,000	Execution for 2,000

* MassStatusRequestID/MassStatusRequest are not a standard FIX field/message but used here for illustration purposes

Option 2: Restatement request for all orders of current trading session

Time	Message Received (ClOrdID, OrigClOrdID, MassStatusReqID*)	Message Sent (ClOrdID, OrigClOrdID, MassStatusReqID*)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution (X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	New Order(Y)					5,000			
5		Execution (Y)	A	New	New	5,000	0	5,000	
6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
7	MassStatusRequest(N)*								MassStatusReqType=6, only active orders with LeavesQty>0
8		Execution (X,N)	A	Restated	Partially Filled	10,000	1,000	9,000	Order X is still active
9		Execution (Y,N)	A	Restated	Filled	5,000	5,000	0	Order Y has been completely filled
10		Execution (X)	A	Trade	Partially Filled	10,000	3,000	7,000	Execution for 2,000

* MassStatusRequestID/MassStatusRequest are not a standard FIX field/message but used here for illustration purposes

Option 3: Restatement request for complete history of all orders of current trading session

Time	Message Received (ClOrdID, OrigClOrdID, MassStatusReqID*)	Message Sent (ClOrdID, OrigClOrdID, MassStatusReqID*)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution (X)	A	New	New	10,000	0	10,000	
3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
4	Replace Request(Y,X)					12,000			
5		Execution (Y,X)	A	Replace	Partially Filled	12,000	1,000	11,000	
6	MassStatusRequest(N)*								MassStatusReqType=6, **OrderHistory*=Y**
7		Execution (Y,N)	A	New	New	10,000	0	10,000	Order X was added to the book, TotNumReports=3
8		Execution (Y,N)	A	Trade	Partially Filled	10,000	1,000	9,000	Order X was partially filled, TotNumReports=3
9		Execution (Y,N)	A	Replace	Partially Filled	12,000	1,000	11,000	Order X was modified, TotNumReports=3
10		Execution (Y)	A	Trade	Partially Filled	12,000	3,000	9,000	Execution for 2,000

* OrderHistoy/MassStatusRequestID/MassStatusRequest are not standard FIX fields/messages but used here for illustration purposes

4.4.4 Corporate Actions

Option 1: Deletion of all active orders

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
Day 1,1	New Order(X)					10,000			
Day 1,2		Execution (X)	A	New	New	10,000	0	10,000	
Day 1,3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
Day 1,4	New Order(Y)					5,000			
Day 1,5		Execution (Y)	A	New	New	5,000	0	5,000	
Day 1,6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
Day 2,1	Status Request(X)								Orders X was for a security that had a stock 1:2 split after Day 1
Day 2,2		Execution (X)	A	Rejected	Rejected				OrdRejReason=5 (unknown order)

Option 2: Deletion of all active orders and confirmation to user

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
Day 1,1	New Order(X)					10,000			
Day 1,2		Execution (X)	A	New	New	10,000	0	10,000	
Day 1,3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
Day 1,4	New Order(Y)					5,000			
Day 1,5		Execution (Y)	A	New	New	5,000	0	5,000	
Day 1,6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
Day 2,1		Execution (X)	A	Restated	Canceled	10,000	0	0	Remainder of order has been deleted

Option 3: Adjustment of order price and/or quantity

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
Day 1,1	New Order(X)					10,000			
Day 1,2		Execution (X)	A	New	New	10,000	0	10,000	
Day 1,3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
Day 1,4	New Order(Y)					5,000			
Day 1,5		Execution (Y)	A	New	New	5,000	0	5,000	
Day 1,6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
Day 2,1	Status Request(X)								
Day 2,2		Execution (X)	A	Order Status	Partially Filled	20,000	2,000	18,000	Order X was for a security that had a 1:2 stock split after day 1

Option 4: Adjustment of order price and/or quantity and confirmation to user

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
Day 1,1	New Order(X)					10,000			
Day 1,2		Execution (X)	A	New	New	10,000	0	10,000	
Day 1,3		Execution (X)	A	Trade	Partially Filled	10,000	1,000	9,000	Execution for 1,000
Day 1,4	New Order(Y)					5,000			
Day 1,5		Execution (Y)	A	New	New	5,000	0	5,000	
Day 1,6		Execution (Y)	A	Trade	Filled	5,000	5,000	0	Execution for 5,000
Day 2,1		Execution (X)	A	Restated	Partially Filled	20,000	2,000	18,000	Order X was for a security that had a 1:2 stock split after Day 1
Day 2,2	Status Request(X)								
Day 2,3		Execution (X)	A	Order Status	Partially Filled	20,000	2,000	18,000	

4.5 Market Data

4.5.1 Trading Session and Security Status

The following example for message flows for status information related to trading sessions and securities are based on the following assumptions.

- The exchange has a single market segment trading two groups of securities, i.e. equity options "EQOPT" and treasury futures "TRFUT"
- Trading is conducted during two sessions called "MORNING" and "AFTERNOON"
- Each trading session has three trading phases called "PRE-TRADING", "TRADING" and "POST-TRADING"
- Users can connect to the system prior to "PRE-TRADING" but requests are not processed until the first trading phase starts
- Users can only subscribe to status information for a complete session, i.e. not for individual trading phases
- Users can subscribe to status information for groups of securities but not to individual securities
- All securities within a specific group are available for trading at the same time
- Trading can be interrupted for individual securities, groups of securities or the entire trading session

Time	Message Received (TradSesReqID[1] or SecurityStatusReqID[1], SubscriptionRequestType)	Message Sent (TradSesReqID or SecurityStatusReqID)	Trading-SessionID	Trading-SessionSubID	SecurityID, SecurityIDSource	TradSesStatus[2] or SecurityTradingStatus[2]	Comment
1	TradingSessionStatusRequest (X, "Snapshot Update")		MORNING				Subscribe to status messages of morning trading session and its trading phases
2	SecurityStatusRequest (Y, "Snapshot Update")		MORNING		EQOPT, 8		Subscribe to status messages of equity options for morning trading session
3	SecurityStatusRequest (Z, "Snapshot Update")		MORNING		TRFUT, 8		Subscribe to status messages of treasury futures for morning trading session
4		TradingSessionStatus(X)	MORNING	PRE-TRADING	EQOPT, 8	2 = OPEN	Pre-trading starts for equity options
5		TradingSessionStatus(X)	MORNING	PRE-TRADING	TRFUT, 8	2 = OPEN	Pre-trading starts for treasury futures
6		TradingSessionStatus(X)	MORNING	PRE-TRADING	TRFUT, 8	1 = HALTED	Trading has been temporarily halted for treasury futures during pre-trading
7		TradingSessionStatus(X)	MORNING	TRADING	EQOPT, 8	2 = OPEN	Normal trading starts for equity options
8		TradingSessionStatus(X)	MORNING	PRE-TRADING	TRFUT, 8	2 = OPEN	Pre-trading resumes for treasury futures
9		TradingSessionStatus(X)	MORNING	TRADING	TRFUT, 8	2 = OPEN	Normal trading starts for treasury futures
10		SecurityStatus(Y)	MORNING	TRADING	SEC1, 8	2 = TrdHalt	Trading interrupted for equity option SEC1 during normal trading
11		SecurityStatus(Y)	MORNING	TRADING	SEC1, 8	3 = Resume	Trading resumes for equity option SEC1 during normal trading
12		TradingSessionStatus(X)	MORNING	POST-TRADING	EQOPT, 8	2 = OPEN	Post-trading starts for equity options
13		TradingSessionStatus(X)	MORNING	POST-TRADING	TRFUT, 8	2 = OPEN	Post-trading starts for treasury futures
14		TradingSessionStatus(X)	MORNING	POST-TRADING	EQOPT, 8	3 = CLOSED	Last trading phase (post-trading) of the morning session ends for equity options
15		TradingSessionStatus(X)	MORNING	POST-TRADING	TRFUT, 8	3 = CLOSED	Last trading phase (post-trading) of the morning session ends for treasury futures

[1] TradingSessionStatusRequest uses TradSesReqID and SecurityStatusRequest uses SecurityStatusReqID as message identifier.
[2] TradingSessionStatus provides TradSesStatus and SecurityStatus provides SecurityTradingStatus to convey status information.

4.6 Additional Topics

4.6.1 Order Rejections

The message flows are almost identical for order rejections during specific trading phases and order rejections due to lack of authorization so that they are only shown once in the following tables.

Option 1: Return order status and quantity fields upon rejection

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3	New Order(X)					15,000			
4		Execution (X)	A	Rejected	New	10,000	0	10,000	OrdRejReason=0 (Exchange option), tag 58=Text with further information

Option 2: Return "Rejected" as execution type and/or order status and echo quantities of request if applicable

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3	New Order(X)					15,000			
4		Execution (X)	A	Rejected	Rejected	15,000	0	15,000	OrdRejReason=0 (Exchange option), tag 58=Text with further information

Option 3: Return (new) message "NewOrderReject" with "Rejected" as execution type

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3	New Order(X)					15,000			
4		NewOrderReject (X)	A	Rejected					OrdRejReason=0 (Exchange option), tag 58=Text with further information

Option 4: Return a BusinessMessageReject message with reject reason 4 = "Application not available" or 6 = "Not authorized"

Time	Message Received (ClOrdID, OrigClOrdID)	Message Sent (ClOrdID, OrigClOrdID)	Exchange (OrderID)	Exec Type	Ord Status	Order Qty	Cum Qty	Leaves Qty	Comment
1	New Order(X)					10,000			
2		Execution(X)	A	New	New	10,000	0	10,000	Exchange issues A as OrderID
3	New Order(X)					15,000			
4		BusinessMessage Reject (X)							BusinessRejectReason=4 (AppNA) or 6 (NotAuth), tag 58=Text with further information

5 Appendix A - Usage Examples

This chapter contains examples from various exchanges for some of the issues described in Chapter *3 Issues and Discussion Points*.

5.1 Identification

5.1.1 Security Identification

Marketplace	Security Identifier(s)	Comment
Arca Options	1. Symbol (55) + MaturityMonthYear (200) + PutOrCall (201) + StrikePrice (202) 2. Symbol (55) + StrikePrice (202) 3. Symbol (55) + MaturityMonthYear (200) + PutOrCall (201) + StrikePrice (202) 4. Symbol (55) + SymbolSfx (65)	1 = OCC style (using the OCC symbol) 2 = OPRA code style (using the OPRA symbol) 3 = Explicit style (the symbol of the underlying, not the OCC symbol) 4 = Non-option security MaturityDay (205) used for series with more than one expiration per month. (Note 205 is replaced with MaturityDate, 541, since 4.3.)
BM&F	SecurityID (55) + SecurityIDSource (48) (+SecurityExchange [207])	SecurityID is internal id, not ticker id SecurityIDSource = 8 SecurityExchange is option (defaults to XBMF)
BM&F Fixed Income	SecurityID (55) + SecurityIDSource (55) + SecurityExcchange (207) + ExchangeTradeType (5681) + StartDate (916) + EndDate (917) + SettlDate (64) + MaturityDate (541)	SecurityID = bond name; SecurityIDSource = 8; SecurityExchange = “XBMF”; ExchangeTradeType (tag 5681 – used defined), with the following values: • R – Specific collateral repo for same day settlement; • S – Specific collateral repo for forward settlement; • Z – Outright purchase and sale for same day settlement • U – Outright purchase and sale for forward settlement • E – Outright purchase and sale for same day settlement on a when issued transaction • J – Securities driven repo transaction (net settlement) • B – Securities driven repo transaction (gross settlement) StartDate (tag 916): when applicable, according to tag ExchangeTradeType; EndDate (tag 917): when applicable, according to tag ExchangeTradeType; SettlDate (64): when applicable, according to tag ExchangeTradeType; MaturityDate (tag 541): with the maturity date of the

Marketplace	Security Identifier(s)	Comment
		bond;
Brut	Symbol (55)	
CHX	Symbol (55)	
CME Exchange	Symbol (55) + SecurityDesc (107) + SecurityType (167) + SecurityID (48) Sample 55=GE 48=CME008065005 107=GEZ6 C94275 167=OPT	A single option for expressing the instrument Symbol contains product group SecurityDesc is human readable SecutityType = FUT or OPT SecurityID = unique identifier assigned by CME Strike price is built into SecurityDesc
CME Clearing	SecurityID (48) + SecurityExchange (207) + CFI Code (461) + StrikePrice (202) + MaturityMonthYear (200) + UnderlyingSecurityID (309) + UnderlyingCFICode (462) + UnderlyingMaturityMonthYear (313) Sample 48=ED 207=CME 461=OPADCS 202=94.725 200=200612 309=ED 462=FFDCSO 313=200612	A single option for expressing the instrument SecurityID contains product group. CFICode expresses security type, put/call, standard/non-standard. Underlying expresses the future instrument on which option is based. Underlying is only provided with option since it may change according to the option definition
Deutsche Börse (cash)	Symbol (55) + SecurityID (48) + SecurityIDSource (22)	All fields mandatory. Symbol = empty field (not used). SecurityID = ISIN
Eurex (derivatives)	Symbol (55) + SecurityType (167) + MaturityMonthYear (200) PutOrCall (201) StrikePrice (202) OptAttribute (206)	Symbol = Eurex product, e.g. FBND (bund future) OptAttribute = contract version number (non-capital adjusted contracts have version number 0) Future: fields Symbol, SecurityType, MaturityMonth-Year mandatory Option: all fields except OptAttribute mandatory
Euronext (cash)	Symbol (55) + SecurityID (48) + SecurityIDSource (22)	All fields mandatory. Symbol = empty field. SecurityID = ISIN
Inet	Symbol (55)	
ISE Stock	1. Symbol (55) 2. SecurityID (48) +	SecurityID = Cusip, alternative to symbol

Marketplace	Security Identifier(s)	Comment
	SecurityIDSource (22)	
LSE	1. Symbol (55) + 2. SecurityID (48) + SecurityIDSource (22) + CountryOfIssue (470)	All three fields mandatory. Symbol contains ISIN if no symbol exist. SecurityID must be ISIN.
MEFF	Symbol (55)	
NASD/ADF	Symbol (55)	
Nasdaq	Symbol(55) + SymbolSfx (65)	SecurityDesc (107) has the following values: N = Global Select or Global P = Non-NASDAQ OTC R = NASDAQ Capital Market C= CQS
NSX	Symbol (55)	SymbolSfx in 5th position of symbol for Nasdaq-listed and 4-7 for NYSE/AMEX
NYSE	Symbol (55) + SymbolSfx (65)	
OCC	Symbol (55) + SecurityID (48) + SecurityIDSource (22) + CFICode (461) + StrikePrice (202) + StrikeCurrency (947) Sample 55=IBM 48=PB 22=J 461=OPASPS 202=110 947=USD	Market symbol expressed in Symbol. OPRA code expressed in SecurityID and SecurityIDSource
Pink Sheets	1. Symbol (55) + SymbolSfx (65) 2. SecurityID (48) + SecurityIDSource (22) + CountryOfIssue (470)	Symbol contains marketplace issued id (and SecurityExchange = NQB) if no symbol exist. If Symbol = 0, the Pink Sheets SecurityID is used. SecurityID = Pink Sheets Security ID + SecurityIDSource = 151. (CUSIP also supported)

5.1.2 Trading Session Identification

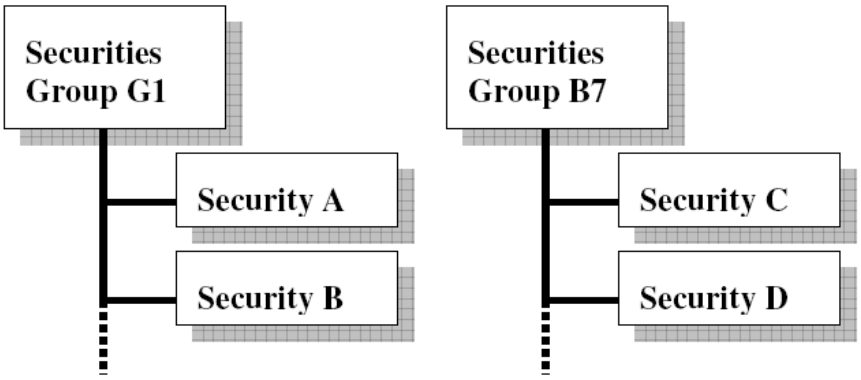
The following sections describe trading session concepts at various exchanges. Thereby, the terminology of these exchanges is used. Some of the descriptions were provided by the exchange to the EEWG whereas others represent excerpts from the publicly available documentation.

5.1.2.1 Brazilian Mercantile & Futures Exchange (BM&F)

Securities Logic Division

BM&F divides electronic traded securities in terms of groups: each group of securities has identification (G1, B7 for instance). Each one of these groups one or more securities together. The criteria used vary from Market Segment (Finance, Spot, Agriculture etc) to most traded or not so traded securities. Each group of securities has its own

trading schedule. Beyond that, groups of securities may have different status during the trading day (every instrument inside one group could be set up to an auction during the day at the same time). The most important aspect of this division is to establish a logic way to deal with securities, arranging them in groups to facilitate the managing of their securities common attributes. (Figure 1.2.1)



Trading Schedule and Time Division

A Trading Day is the basis for defining a division of time for trading a security in BM&F. During a single Trading Day, you might have the time divided in Trading Phases. A single Trading Phase acts as a Time Label for a slice of the trading day. In other words, a set of business trading rules (if an order can be entered, modified or deleted, the limits for trading the security etc) may the set to a specific trading phase. Examples of trading phases include:

- Pre-opening (Auction)
- Non-cancel order
- Opening (Auction Matching)
- Continuous Trading
- Surveillance Intervention
- Market Closing

A Trading Phase has a Start Time, so the start of one phase marks the end of the previous.

Trading Session Definition

A Trading Session is a group of Trading Phases. In other words, we might have 2 Trading Sessions during a single Trading Day.

Trading Phase Status, Security Status and Security Group Status

Both Trading Phase and Security might have different status during a trading day: for instance, in a given trading day, the trading phase could be in continuous trading mode and an order is then received by the trading engine. At this moment the trading engine determines that the order exceeds the high limit price threshold. The engine starts an auction upon this security: the security status changes to “on auction”, but the trading phase remains the continuous trading mode with no status change (nor halted or paused).

One very important point to be noted is that internally, we could change one security group status, what could affect all the securities within the group: in other words, the individual security status is changed either when a specific security change event occurs or when a "parent" security group status is changed.

Status Change Notification
At BM&F, when a trading phase changes or when a security status changes, this piece of information is sent across the hole market. The trading screens are notified about these changes in a "Market Data Delivery" fashion. This is very important for traders to know that a security is being auctioned, or to be notified about the fact that the actual trading phase does not allow him to cancel orders sent to the exchange (no business rules are sent to the traders, but the label of the trading phase recalls him about what rules are implied in that phase, as in the case of "Non cancel orders" phase).

5.1.2.2 Chicago Mercantile Exchange (CME)

The following excerpts were taken from the CME website (http://www.cme.com).

Trading Sessions and Schedules
Electronic trading on the CME Globex platform is available virtually around the clock, from Sunday evening through late Friday afternoon. Trading takes place during five daily CME Globex sessions [per week], the period of time in which a customer may enter, view, modify or cancel electronic orders.

During each CME Globex session, all contracts transition through a variety of pre-defined states, although the timing of these states and the transition between them vary by product, according to contract specifications. During each of these states certain trading functionality is allowed or disallowed, as detailed in the chart below.

For CME Globex products, the start of the CME Globex session, which usually occurs in the afternoon or evening, generally marks the beginning of the next Trading Day. (For example, orders entered during Sunday's evening session are dated for and cleared on Monday). If the CME Globex and open outcry sessions overlap for a given product, the Trading Day includes both the CME Globex session and the trading floor open outcry session (i.e., "Regular Trading Hours").

Products that trade on CME Globex are classified into three categories based on their hours of availability:

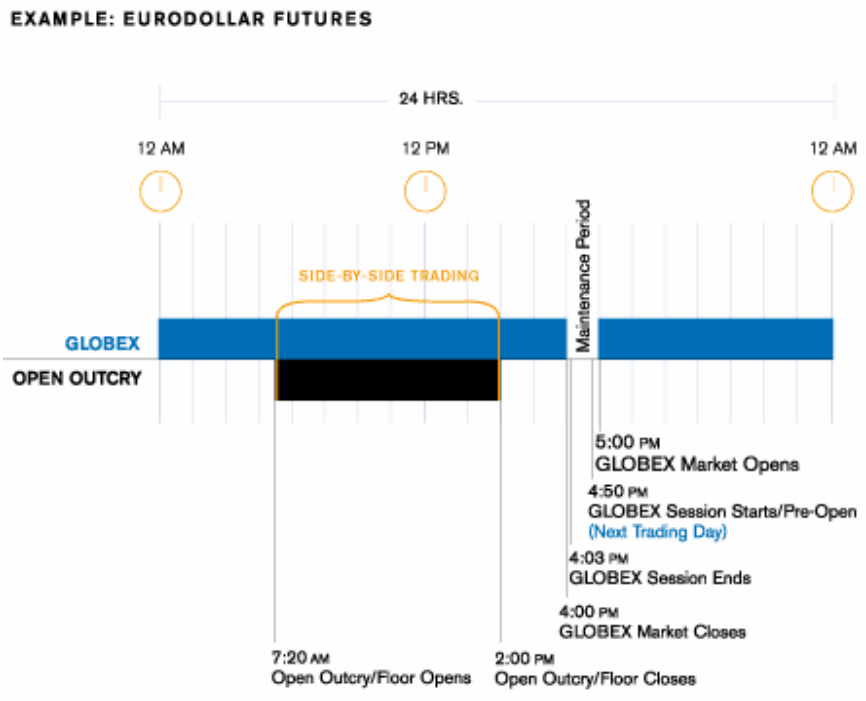
- "Side-by-Side" contracts trade on CME Globex and, for a portion of the day, simultaneously via open outcry on the trading floor.
- "Electronic-Only" contracts trade only on CME Globex.
- "After-Hours Electronic" contracts trade electronically on CME Globex only after the product stops trading via open outcry on the trading floor.

The CME Globex session for products in each of the above three categories is outlined below. Trading hours vary by product.

Side-by-Side

Some contracts trade around the clock on CME Globex while also trading side-by-side via open outcry during the day on the CME floor.

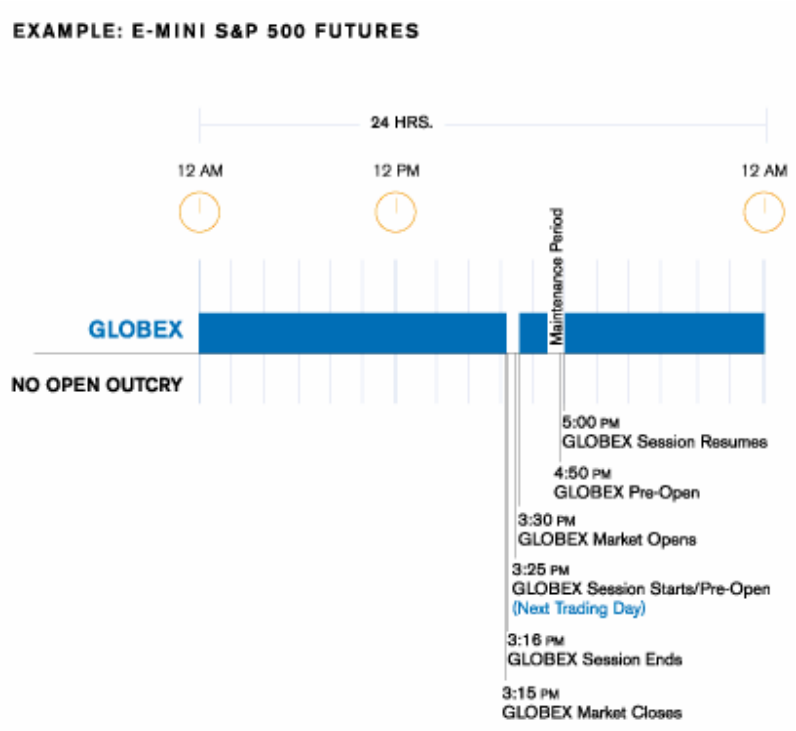
For these side-by-side products, the CME Globex session and the open outcry trading hours together compose the Trading Day. For most contracts, the first CME Globex session of the week starts on Sunday evening, with the opening rotation starting at 17:00 CT. This session continues through the open outcry trading period on the CME floor on Monday, ending on Monday afternoon. Similarly, each subsequent Trading Day begins with the start of the next CME Globex session, at varying times in the afternoon on Monday through Thursday.



Electronic-Only

Some contracts trade only on CME Globex, either "continuously" or only during U.S. day-time hours. These contracts do not trade via open outcry on the CME trading floors. For these products, the CME Globex session and the Trading Day are synonymous and concurrent.

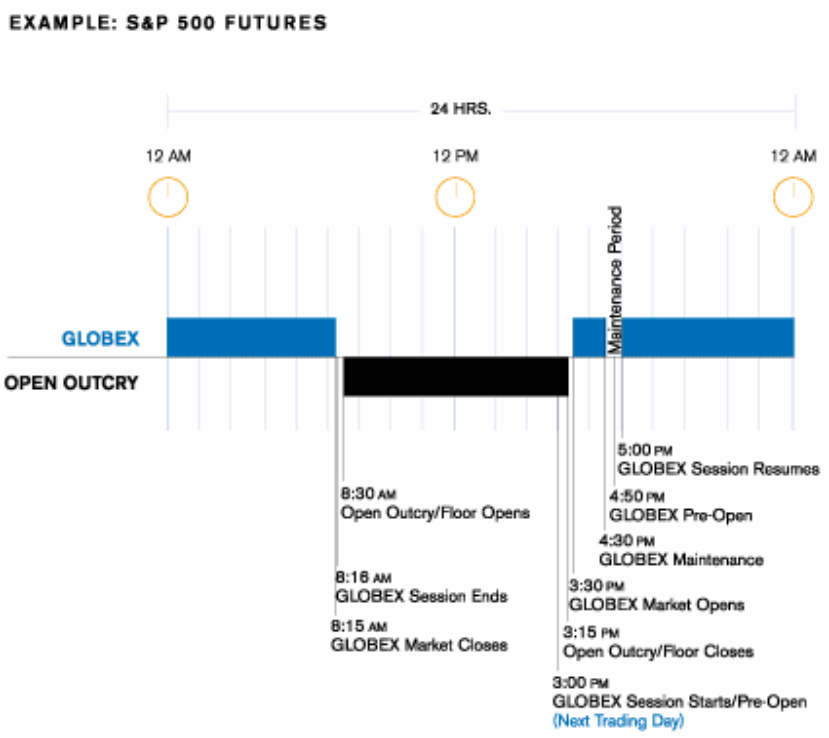
For most contracts, the first CME Globex session of the week starts on Sunday, with the opening rotation starting at 17:00 CT and trading continuing until Monday afternoon. Similarly, each subsequent Trading Day begins with the start of the next CME Globex session, at varying times in the afternoon on Monday through Thursday.



After-Hours Electronic

Some contracts trade both electronically and via open outcry on the CME trading floor, but trade electronically only when there is no trading on the floor. For these products, the Trading Day begins with the CME Globex session and ends with the close of open outcry trading.

The first Trading Day begins on Sunday, with the opening rotation starting at 17:00 CT and trading continuing on CME Globex until the next morning. Closing times vary by product, with CME Globex trading always ceasing prior to the start of trading on the floor. On Monday through Thursday, the pre-opening cycle takes place during the open outcry session to enable customers to begin entering orders for the next Trading Day.



5.1.2.3 Chicago Stock Exchange (CHX)

New Order Single:

- TradingSessionID - Indicates the trading session(s) the order is eligible to be traded in.
  - 1 = Primary session
  - 2 = Post-primary session
  - 3 = Primary and post-primary sessions

Execution Report:

- TradingSessionID - Indicates the trading session(s) the order is eligible to be traded in. If ExecType = 1 (Partially fill) or 2 (Filled), this will indicated the session the trade occurred.

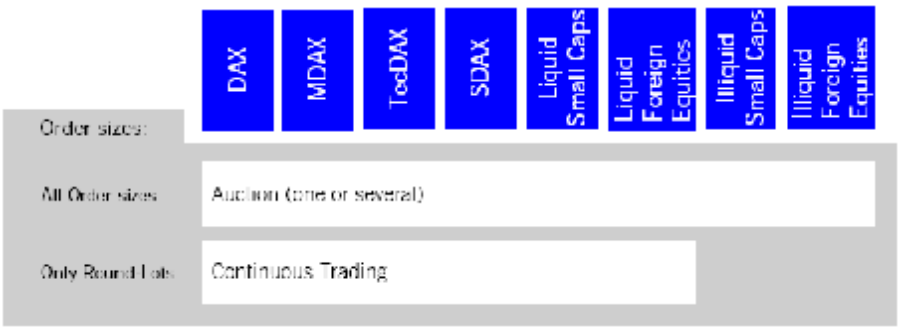
5.1.2.4 Deutsche Börse (DBAG)

The following excerpts are taken from the DBAG document *Xetra Release 7.1, Market Model Equities.*

Equities are segmented into different groups. Possible criteria for segmentation are, for example, liquidity, index affiliation and country of origin. The trading segments valid in Xetra are not dependent on the existing legally stipulated admission segments (market segments) at the Frankfurter Wertpapierbörse.

A trading segment consists of a specific number of instruments for which trading is organized in the same way. Certain parameters of the Xetra market model concerning trading model, order book transparency, trading times etc. can be configured for one trading segment. A combination of parameters is selected for each trading segment, which specifies the trading process in the respective segment.

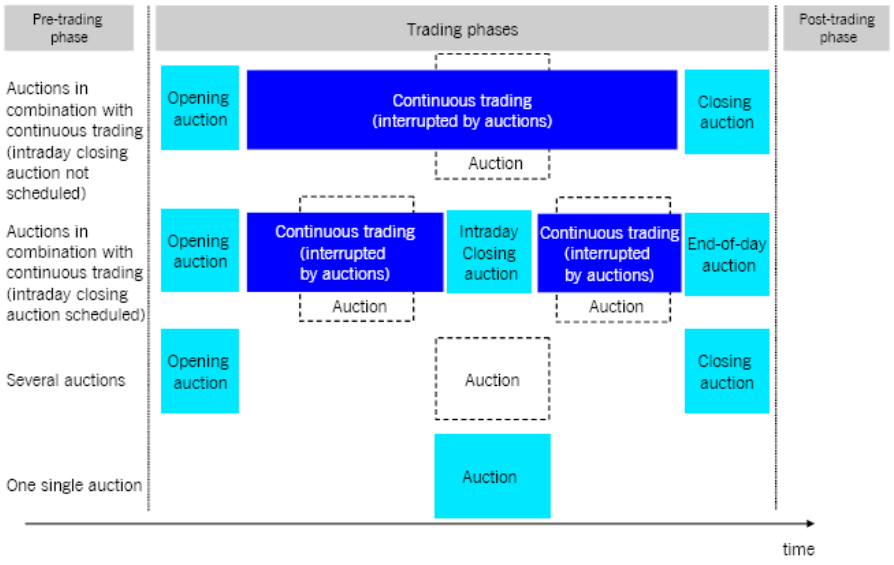
The following figure shows the trading segments and trading forms for DBAG.



The market model for equities has a number of underlying principles related to trading sessions and the corresponding status, order visibility and permissible actions as follows.

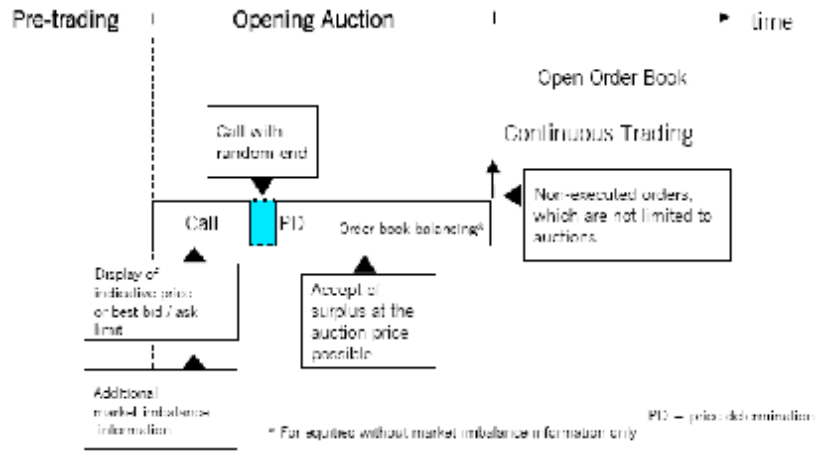
- An equity can be traded continuously or only in auctions.
- Continuous trading starts with an opening auction, can be interrupted by one or several intraday auction(s) and ends with either a closing auction or an end-of-day auction. If continuous trading ends with an end-of-day auction, an intraday closing auction is scheduled which provides an intraday valuation price. Continuous trading starts again after completion of the intraday closing auction.
- During the auction's call phase, the order book remains partially closed. The indicative auction price or the best bid and/or ask limit is displayed. Depending on the individual equity, additional market imbalance information may be disseminated. In case of an uncrossed order book, the accumulated volumes at the best bid and best ask are displayed in addition to the best bid and ask limits. In case of a crossed order book the executable volume for the indicative auction price, the side of the surplus and the volume of the surplus are displayed. Currently, the market imbalance information is disseminated for all equities.
- Trading will be interrupted if the potential price lies outside a pre-defined price range around the reference price.

Trading takes place all day and begins with the pre-trading phase followed by the trading phase and the post-trading phase. The system is not available for trading between the post-trading and pre-trading phase. The pre-trading phase and the post-trading phase are the same for all equities whereas the course of the trading phase may vary from equity to equity. According to their segmentation, individual equities are traded in different trading models and at different trading hours.



An opening auction, comprising a call phase, price determination phase and - for all equities without market imbalance information - order book balancing phase, is carried out prior to continuous trading. An intraday and intraday closing auction (if scheduled) interrupts continuous trading. When scheduled, the end-of-day auction ends continuous trading.

Like opening auctions, intraday auctions, closing auctions, end-of-day auctions consist of three phases: call phase, price determination and, for equities without market imbalance information, order book balancing phase.



To ensure price continuity, continuous trading is interrupted by a volatility interruption whenever the potential execution price of an order lies outside the dynamic and/or static price range around a reference price. A volatility interruption triggers a change of trading form: continuous trading is interrupted and an auction is initiated.

5.1.2.5 Eurex

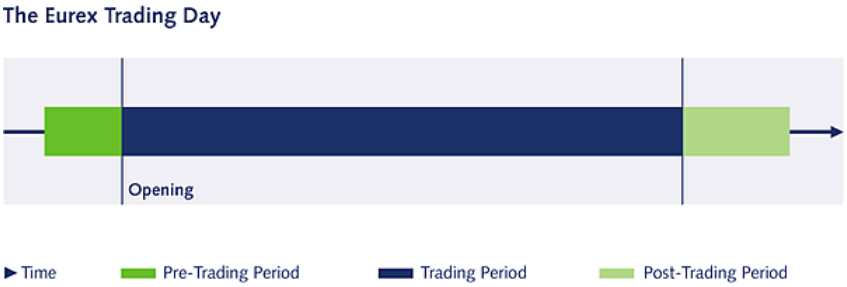
The Eurex market offers a range of products (options and futures contracts based on a specific underlying security or other value) that are grouped into a number of segments:

- Equity Derivatives
- Equity Index Derivatives
- Volatility Index Derivatives
- Exchange Traded Funds
- Interest Rate Derivatives

The trading day at Eurex typically runs from 07:30 to 22:00 CET. It consists of three phases:

- Pre-Trading
- Trading
- Post-Trading.

All products are subject to these periods, although product-specific schedules apply for each period - for instance, due to different conventions in the underlying market.



The Pre-Trading Period begins at 07:30 CET, and allows participants to prepare for the opening of trading. Quotes and orders can be entered, changed or deleted, and participants can make data inquiries, but no "inside market" (the best bid and ask prices, as well as their respective aggregated bid and offer sizes) information is available. Entering futures combination quotes, as well as quotes and orders for option strategies and option volatility strategies is not permitted during this time.

The Opening Period consists of several steps taken to uncross the order books and to start the continuous trading phase. Eurex Market Supervision has the option of implementing a "Fast Market" on a per-product basis in unusual circumstances, such as times of high volatility. This status temporarily changes the rules for quotes and mistrades. In designated futures contracts, in order to establish a closing price, the Trading Period ends with a closing auction.

At the end of the Trading Period, a product enters the Post-Trading Period which is further split in several periods (e.g. Post-Trading Full, Post-Trading Restricted) where different functions are available.

If the last effected price of a futures contract is outside one of the price ranges with respect to specific time frames, a volatility interruption of the trading period in this product occurs. Immediately thereafter, trading is resumed with a Pre-Trading Period and an Opening Period (unless a closing auction follows).

5.1.2.6 London Stock Exchange (LSE)

The definitions for the market structure of the LSE were taken from the LSE document *Guide to the new trading system, Guidance note: April 2006*. The following are excerpts from the document that abstract from specific messages.

Component	Description	Examples
Market	Used to describe the geographical elements of a trading environment - its business calendar and time zone the Market is operating in.	LSE, DTS, JSE
Segment	The Segment tier is used to define a set of Instruments that follow the same trading model, e.g. an electronic anonymous order-driven market.	SET1
Sector	The Sector provides a more granular level which defines the group of Instruments within the Segment which follow the same trading schedule.	FT10
Instrument	The lowest tier is used to describe the individual tradable instrument itself.	VOD

A Period defines the rules applicable to trading in the underlying securities. Periods usually apply to all securities in a Sector but it may be necessary for separate rules to be temporarily defined for a specific Tradable Instrument/ Currency (TI/C).

The default schedule is disseminated in the morning. As the day commences, the actual start of every period is communicated for each Sector. At this point the instrument falls out of the default schedule, into a temporary schedule. The temporary schedule is applicable only to the current day.

Similarly, messages will be disseminated throughout the day as each new period commences. In the situation where further price monitoring extensions are required, new messages describing the revised schedule are disseminated.

The security will remain in temporary periods for the entire day's schedule. At the end of the day, the instrument will return back to the default schedule for the next day's trading.

5.1.2.7 NYSE Arca

Trading Session ID

FIX 4.2 introduces specifying a list of TradingSessionID values in an order. Because TradingSessionId(336) is a repeating field, it is necessary to also send NoTradingSessions(386) to indicate the number of TradingSessionId tags that will be sent in the message. The ARCA FIX Host will support these fields in FIX 4.0 and 4.1. Firms have several options in identifying the Time in Force parameters, it is highly recommended that firms using a combination of TradingSessionID values to specify the sessions for which participation is desired.

Orders may be entered at any time after 3:30 AM.

- Session 1 (TradingSessionID value "P1") orders participate in:
  - Opening Auction
  - Session 1 (Opening Session)
  - Market Orders Auction
  - Expire at 9:30 AM
- Session 2 (TradingSessionID value "P2") orders participate in:
  - Market Order Auction
  - Session 2 (Core Session)
  - Expire at 4:00 PM
- Session 3 (TradingSessionID value "P3") orders participate in:
  - Session 3 (Late Session) OTC and Listed only
  - Expire at 8:00 PM

TradingSessionID may be combined with DAY or GTD to provide accurate control of order execution. Combinations of TradingSessionID values may be used. For example, a DAY order might be specified with TradingSessionID values of "P1" "P2" and "P3".

On the Bulletin Board Exchange you must specify P1 and P2 in order to trade in the morning trading session. These orders will always stay open in the core session unless cancelled by the client.

DAY orders with no TradingSessionID qualifiers or other factors such as EffectiveTime default to being valid for Sessions 1 and 2 if placed between 3:30 AM and 4:00 PM, or will be rejected in the after hours session. For GTD orders expiring on the same day as they were placed, TradingSessionID is ignored. However, trading session IDs

will prevail if a GTD order has an expire time that exceeds the duration of the trading session ID specified. (e.g., if you set an order to only be active during P2 session, and your GTD expire time is 6pm EST, the order will expire at the end of the core session because of the trading session ID specified. However, if you specify P2 and P3 in this order, your order will expire at 6pm as instructed.) The order is valid from the time it is placed (or EffectiveTime, if specified) until the ExpireTime.

GTD orders with an ExpireTime containing a date in the future with no TradingSessionID qualifiers will default to being valid for Session 2 only. Anyone wishing to specify that such orders are valid for other sessions must list the sessions explicitly using TradingSessionID.

Extended Hours

Extended hours trading was not well defined in the FIX specification prior to FIX 4.2. As a temporary solution, ARCA introduced the concept of a "Day+" order, which spans all trading sessions. Day+ orders must be represented in one of two ways:

- Specify a TimeInForce of DAY with NoTradingSessions(386)=3 and TradingSessionID(336) values of "P1" "P2" and "P3"

- Specify a TimeInForce of GTD and set an Expiretime. This should only be used when an order is not set to expire at the end of a trading session. When an order is set to be expired at the end of a trading session the TradingSessionId fields should always be used. Note that ExpireTime is formatted in UTC (formerly GMT). As a result, 8:00 PM Eastern Time will be represented as midnight of the following calendar day (20030306-00:00:00). This link will help you determine which time zone you should send: <http://www.dxing.com/utcgmt.htm>.

NYSE ARCA Exchange supports EffectiveTime, which enables customers to stage orders for execution at a later time. This feature is particularly useful for submitting orders between 7:30 and 9:30 AM, which will be eligible for execution at the 9:30 AM open. Alternately, firms may wish to submit DAY orders between 3:30 and 9:30 AM with a NoTradingSessions(386)=1 and a TradingSessionID(336) value of "P2".

5.1.2.8 Stockholmsbörsen

Stockholmsbörsen is a secondary market where, at present, equities, premium bonds, convertible loans, subscription rights, exchange-traded options/warrants and retail bonds are traded. All trading on the Exchange is carried out by traders. They are employed by stockbroking firms and banks which are members of Stockholmsbörsen.

Trading on Stockholmsbörsen is conducted in:

- SAXESS: equities, premium bonds (lottery bonds), convertible loans, subscription rights, exchange-traded options/warrants and other instruments
- SOX: retail bonds (bond loans, debenture loans and index loans)
- Click: derivatives (stock options, futures)

The trading day is dived into the following trading sessions:

- Pre-Trading Session
- Trading Session
    - Opening Call Auction
        - Pre-Call
        - Call Interaction
        - Uncrossing

  - Continuous Trading Session
  - Closing Call Auction
- Post-Trading Session

During the pre-trading session manual trades, so called after market II -trades in this phase, and pre-opening trades can be recorded. In addition, GTD order management together with order entry for opening call takes place.

Opening, closing and intraday calls can be formed with three sub phases; pre-call, call interaction and uncross. Pre-call phase enables order management without any market transparency. Call interaction phase allows order management with increased market transparency. Final price determination and allocation takes place in uncross. Thus, order management in opening call continues when the orderbook shifts from pre-trading to either call interaction in equities (open call) or pre-call in related instruments (hidden call). When the first orderbooks are shifted to continuous trading order entry with other orderbooks continues still in call. The time for sequential start in equities takes approximately 6 minutes after 10 am (EET) after the final equity order book is in continuous trading.

Continuous trading session is divided into the round lot and odd lot trading in automatic order matching. Manual trades, when criteria of contract transactions are met, can be concluded as well.

During post-trading session manual trades, so called after market I -trades in this phase, can be concluded. In addition, order management is possible allowing good-till-date order (GTD) cancellation as well as those order changes for GTD orders that have no effect on order priority

Ordinary trading day:

Product Group	Trading Hours	Notes
Equities	09.00 - 17:20	Closing call 17:20 - 17:30
ETF	09:00 - 17:20	
Derivatives	09.00 - 17:20	
New Market	09.15 - 17:20	Closing call 17:20 - 17:30
Warrants	09.05 - 17:20	
Equity rights	09:30 - 17:20	
Convertibles	10.00 - 17:30	
Premium (lottery) bonds	10.00 - 15.30	
SOX bonds	09.00 - 16.15 (09.00 - 12.00*)	
Fixed Income	09.00 - 16.15 (09.00 - 12.00*)	

- Market (SOX bonds and Fixed Income) is closing 12:00 day before: Epiphany, Good Friday, Labor Day, Ascension Day, National Day, Midsummer's Eve, All Saints Day Christmas Eve and New Year's Eve.

5.2 Market Data

5.2.1 Trading Session and Security Status

5.2.1.1 Chicago Mercantile Exchange (CME)

CME provides a number of messages to convey the status of individual instruments or group of instruments.

- MG Message – Instrument Status
- MJ Message – Group State Change Marker

The CME market data Instrument Status (MG) message communicates information pertinent to the trading state of an instrument. Within the Instrument Status (MG) message, Trading Status, Instrument State, and the Type of Action on an Instrument fields may contain different values for Eurodollar options instruments and Equity and FX options instruments.

The following tables describe possible values for Trading Status, Instrument State and Type of Action.

Trading Status	Description
P	Simple Reserved
H	Reserved Upwards (Stop spike)
B	Reserved Downward
S	Suspended (Stop spike)
R	Resume (Stop spike)
< >	Unchanged or Open (Stop spike)

Instrument State	Description
A (1st character)	Authorized - Order and mass quote submission, modification, and cancellation are allowed for the instrument.
I (1st character)	Forbidden - Order and mass quote submission, modification, and cancellation are not allowed for the instrument.
R (2nd character)	Reserved - If the price exceeds the upper or lower price threshold, it is said to be reserved. An instrument can be reserved automatically by Globex or manually by GCC.
S (2nd character)	Suspended - An instrument is suspended before the market opens to prevent the instrument from opening with the rest of its group. Instruments can only be suspended manually by GCC.
G (2nd character)	Frozen - Order entry temporarily denied because an order is set to be matched that would make the instrument exceed one of its price thresholds.
< > (2nd character)	Open - Order and mass quote submission, modification, cancellation, and order execution are allowed for the instrument, assuming the group state permits.

Type of Action	Description
P	Programming of deferred opening
M	Reservation or suspend by surveillance
C	Trading
O	Change to normal state (following R or S)
D	Cancellation deferred opening
A	Instrument Authorization
I	Instrument Forbidding
E	Cancellation of orders in the book for an instrument
N	State of initialization

Type of Action	Description
F	Fast Market
S	Normal Market
G	Freeze/Unfreeze the instrument

The CME market data Group State Change Marker (MJ) message communicates information pertinent to the group state of an instrument. Within the Group State Change Marker (MJ) message, the Instrument Group Status filed may contain different values for Eurodollar instrument groups and Equity and FX instrument groups. The message further contains a field Closing Indicator to show whether the Instrument Group is in a closing phase or not.

The following tables describe possible values for Instrument Group Status.

Instrument Group Status	Description
C	Start of Consultation
P	Pre-opening
O	Opening
T	Transient
E	Intervention before Opening (no cancel period for the CME)
S	Continuous Trading
M	Minibatch (resetting of statistical data for a trading session)
R	Intervention after opening (not used by the CME)
N	Surveillance intervention
F	End of Consultation
I	“Forbidden”
Z	“Interrupted”

CME provides the MH Message – Trading Day Timetables to inform subscribers about which groups of instruments will trade during the day, as well as their timetable’s changeover status. It has fields to describe numerous trading transitions, program events as well as the group state after opening (only for program event O = Opening).

5.2.1.2 Deutsche Börse (DBAG)

DBAG provides status information on an instrument level by means of two independent messages.

- Inside Market
- State Change Information

The following table describes fields with status and event information within the Inside Market and State Change Information message. Both messages specify an instrument for which the status or event information applies.

Field	Description
prcStsValCod	Code/identifier of a trading phase or fast market status, for example: • PRETR – Pre-trading • POSTR – Post-trading • TRADE – Continuous trading • VOLA – Volatility interruption of continuous trading • HALT – Halt • SUSP – Suspend • EQUON – Equity Fast Market On • EQUOF – Equity Fast Market Off • …
fmInd	Fast Market
moiInd	Market Order Interruption
volInd	Volatility Interruption during an auction
srpBidAskInd	Surplus on the bid or ask side during an auction
excpStateInd	Generic indicator, e.g. to extend call phase of an auction

5.2.1.3 Eurex

Eurex provides status information on an instrument level by means of two independent messages.

- Inside Market
- State Change Information

The following table describes possible values of the field cntrStsValCod representing status and event information within the Inside Market message which refers to a contract, i.e. a single instrument.

Contract Status	Description
P	Pre-trading
O	Opening
< >	Normal trading
A	Closing auction
C	Closed
H	Hold
V	Start Fast Market
R	Return to trading
X	Expired

The following table describes possible values of the field prcStsValCod representing status and event information within the State Change Information message which refers to a product, i.e. a group of instruments.

Product Status	Description
PRETR	Pre-trading
PREOP	Pre-opening
POSTF	Post-trading full
POSTR	Post-trading restricted
TRAD	Trading
CLAUC	Closing auction
HALT	Halt
FAST	Fast Market

6 Appendix B – Discussions and Comments

6.1 Introduction

This chapter presents discussions that have been captured during the process of discussing the topics in the workgroup. This chapter is not intended to be a complete capture of the discussions.

6.2 Identification

6.2.1 Security Identification

6.2.1.1 Exchange Symbol vs. Symbol

One possibility that has been debated is to use the Symbol (55) tag for commonly known symbols, e.g. “IBM”. This may e.g. be relevant when an investor enters an order to his broker and the broker chooses the execution venue among the ones offering trading in IBM. As a consequence the intermediary may need to specify the SecurityID = “IBM EUR” (as SecurityIDSource [22] = 8 [Exchange Symbol]) when submitting the order to the marketplace. Sometimes adding the SecurityExchange (207) can be sufficient.

Another problem is that a marketplace may choose to trade the same instrument in multiple books, e.g. in different market segments. This may be relevant e.g. when a security is traded both in block (wholesale) and retail markets but also when the same security (ISIN) is traded in multiple marketplaces covered by a single trading engine. While the investor may not care what market segment the order ends up in, the broker may have to choose what book to enter the order in. So while the investor uses a “commonly known” identifier such as “IBM” when buying the shares, the broker may have to assign “IBM xxx” to identify the particular book. Proposedly this could be done using Symbol (55) for “IBM” and SecurityID (48) + SecurityIDSource (22) = 8 (Exchange Symbol) for “IBM xxx”.

6.2.1.2 Marketplace-assigned Identifiers

Some marketplaces use unique system-generated (synthetic) identifiers for unique identification. Such id’s can be static over the lifetime of the instrument (or book) meaning other identifiers, which are outside the control of the marketplace (as ISINs, Symbols, etc) can be modified while retaining all transaction history under the same (synthetic).identifier.

6.2.1.3 SecurityDesc

The SecurityDesc (107) tag seems to come into more frequent use as a unique identifier. It is not mentioned in the examples of Volume 1 and we think it would be a good idea to expand a bit about its usage there. We understand e.g. the CME uses it and that it may be recommended for use for more complex (and unstandardized) products where you normally do not issue symbols or other standard identifiers. We however do not understand why you couldn't use SecurityID (with some SecurityIDSource) for the same purpose, as the formats of the fields are the same (String). Well, you could also deprecate Symbol and add it as an enum to SecurityIDSource (thus containing all identifiers in that tag) - but some things are tough to change, we know.

6.2.1.4 SecurityExchange vs. ExDestination

SecurityExchange (207) is part of the Instrument component block whereas ExDestination (100) is part of a number of messages including NewOrderSingle. Both require a market identifier code (MIC) as value and describe the execution venue. SecurityExchange seems to be a static attribute of a given instrument whereas ExDestination is more dynamic and seems to only apply to a specific order. It is probably confusing if both tags were to appear with different values in a single message. A security can be described without naming an execution venue, an order requires this information, either implicitly through the destination of the session or explicitly through a field with an appropriate value.

The specification of an execution venue is more complex than a simple MIC value. There could be execution venues such as systematic internalizer that do not have a MIC value or there could be multiple books.

6.2.1.5 Security Identifiers in Fixed Income

FI populate:

- Symbol = "[N/A]" (without the quotes)
- SecurityID and SecurityIDSource (mainly ISIN in Europe or CUSIP/Blmbrg in the US).
- SecurityDesc is optionally used to provide a description, e.g. "GE CAPITAL 4.500% 08/15/2014", that corresponds with the security.

Additional Issuer (106) field can also be used to identify the bond's issuer, usually an entity name is put here, e.g. "LOS ANGELES CA DEPARTMENT OF WATER AND POWER" or "FORD MOTOR CREDIT". GFIC had made an attempt to make Symbol field not required, with the rule being that EITHER Symbol or the SecurityID / SecurityIDSource be present. But there was major push back thus the "[N/A]" was a compromise. In the end we still prefer to see it not required because it causes confusion.

6.2.1.6 Security Identifiers in FX

Very simply is the standard convention of EBS symbology of "CCY1/CCY2", where CCY1 and CCY2 are ISO currency codes. This is read as "currency 2 per currency 1", e.g. USD/GBP. From the GFXC perspective this is sufficient.

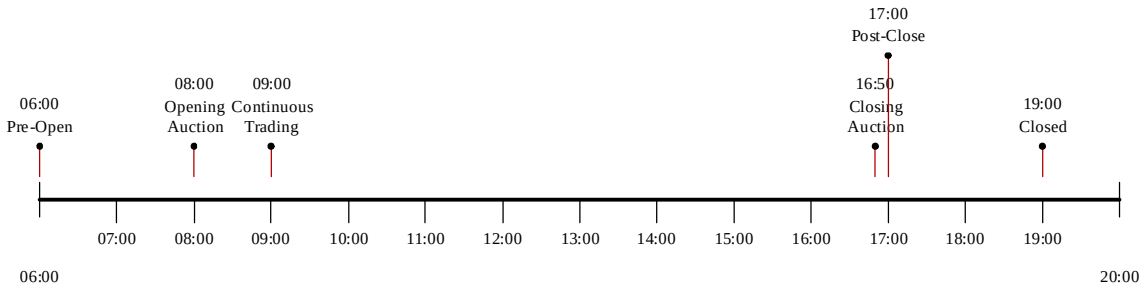
The value is stored in the Symbol field.

6.2.2 Trading Session Identification

6.2.2.1 Static / Reference Data

An exchange needs to relay trading session related static data for use by connected actors, e.g.:

- What market segments are available (or product group or any other name you choose to group instruments)
- What instruments are traded per market segment, either as a list per market segment, or by defining the market segment in the security ref. data message
- What trading schedule is associated with the market segment (or in extreme cases the individual security)
- The trading schedules including the trading periods within the schedule, e.g.:
    - Schedule = Blue chips normal day
    - Periods:
        - Pre-Open, start 06:00
        - Opening Auction, start 08:00
        - Continuous Trading, start 09:00
        - Closing Auction, start 16:50
        - Post-Close, start 17:00
        - Closed, start 19:00

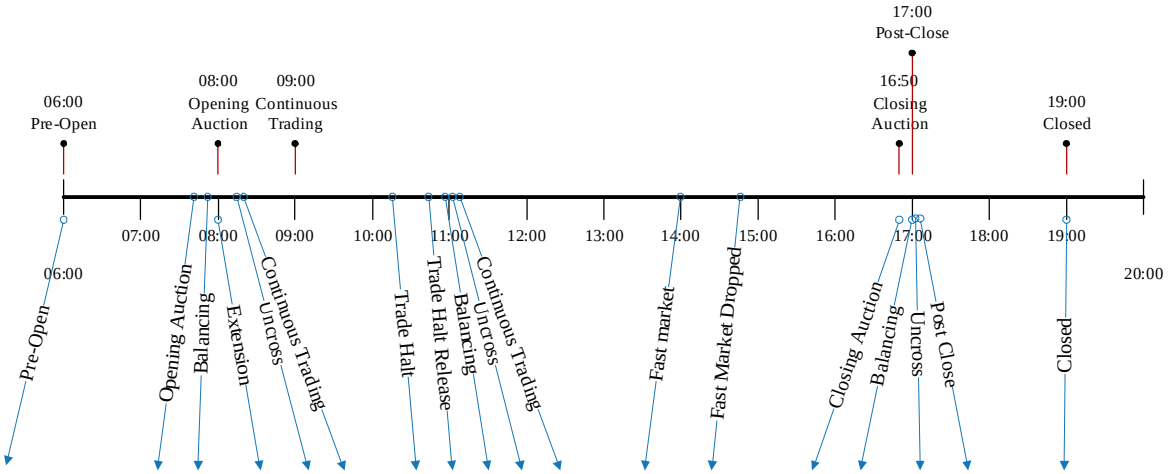


- What sub-periods are relevant for the respective periods, e.g.:
  - Auctions are divided into:
    - Pre-Auction (where the order book is published)
    - Balancing (where only imbalance information is published)
    - Auction extension (if there still is imbalance at the scheduled end)
    - Uncross (where the matching engine is busy)
- A list of events that can be published

6.2.2.2 Events Publication

As trading moves along the schedule, the exchange may want to publish events including state changes. Parties use the events not only for information but also for automation. In many cases those events are related to individual instruments, but in some cases they are published per market segment. Publication per market segment is relevant when you want to limit the number of messages sent out and the when the market change state for a whole market segment instantaneously.

Example:



6.2.2.3 Comments

The main events are defined as trading periods that are associated with rules like:

- Order maintenance is allowed (can be limited to certain actions, order types, etc)

- Privately negotiated trades are allowed
- Batch queries are allowed
- Etc

However, there may also be events that do not change the rules, but still have to be produced in a scheduled manner (sometimes associated with more logic too), e.g.:

- "The trading halt will be lifted at 11:15. Order entry allowed from 11:05"
- "Opening in 10 minutes, please remove indicative interest"
- Etc

