



# Trade Event Notification Service

Rules of engagement

# Content

Content .....	1
1 Document properties .....	4
1.1 Purpose of this document.....	4
1.2 Document location.....	4
1.3 Revision history.....	4
2 Preface .....	5
2.1 Overview .....	5
2.2 Pre-requisites .....	6
3 Functionality .....	7
3.1 Delivery methods.....	7
3.1.1 Delivery aspects .....	7
3.1.2 E-mail.....	7
3.1.3 File Transfer .....	7
3.1.4 SWIFT .....	7
3.1.5 Fix .....	7
3.2 Notification formats.....	8
3.2.1 XML.....	8
3.2.2 SWIFT .....	8
3.2.3 Fix Session.....	8
3.3 Notification triggers.....	8
3.3.1 Products.....	8
3.3.2 Actions .....	10
3.3.3 Actions and products.....	10
3.4 XML Description.....	12
3.4.1 Positions .....	12
3.4.2 Orders .....	14
3.4.3 Margin Calls .....	17
3.4.4 Funding .....	18
3.5 SWIFT Description .....	19
3.5.1 Positions .....	19
3.6 FIX Description.....	21
3.6.1 Positions .....	21
3.6.2 Orders .....	23
3.6.3 Margin Calls .....	26
3.6.4 Funding .....	27
4 Error Handling .....	28
4.1 E-mail Troubleshooting.....	28
4.1.1 Missing/Invalid content .....	28
4.2 SWIFT Troubleshooting.....	29
4.2.1 Broken communication .....	29
4.2.2 Invalid Message Formatting.....	29
4.3 File Transfer Troubleshooting.....	29
4.3.1 Broken communication .....	29
4.3.2 Invalid Credentials.....	29
4.4 FIX Troubleshooting.....	30
4.4.1 Broken Communication .....	30
4.4.2 Invalid Credentials.....	30
4.4.3 Sequence Numbers out of sync.....	30
5 Acceptance Test.....	31
5.1 Functional Testing.....	31

5.2	Connectivity Testing .....	31
6	Maintenance .....	32
6.1	Schedule .....	32
6.2	Notifications.....	32
6.3	Support .....	32
6.3.1	Onboarding Support .....	32
6.3.2	Maintenance Support .....	32
	Appendix 1 - XML schema .....	33
	Appendix 2 - Option Exercise .....	38
	Create position .....	38
	Exercise Option .....	39
	Appendix 3 - Option Expire.....	43
	Create Position .....	43
	Expire Option.....	44
	Appendix 4 – Allocation .....	45
	Appendix 5 - Position netting.....	46
	Create two opposing positions .....	46
	Leave positions open.....	48
	Appendix 6 - Cash Settlement .....	49
	Appendix 7 - Corporate Action .....	50
	Appendix 8 - Saxo Initiated trade .....	51
	Create position .....	51
	Appendix 9 - Customer trade.....	52
	Create position .....	52
	Place Stop order.....	53
	Appendix 10 - Margin Call .....	55
	Margin utilization exceeds 100%.....	55
	Margin utilization exceeds 115%.....	56
	Margin utilization exceeds 125% - stop out .....	56
	Appendix 11 - Customer Order .....	57
	Order Placed and modified to market order.....	58
	Place order.....	58
	Modify to market order .....	59
	Order placed, modified and cancelled .....	61
	Place order.....	61
	Change price.....	62
	Cancel Order.....	62
	Cancel all orders.....	63
	Place related order (If Done).....	64
	Place order.....	64
	Place two related orders.....	65
	Slave order changed to master order .....	68
	Old main order executed .....	69
	Placing Related orders (OCO) .....	71
	Main order executed.....	73
	Appendix 12 - Trade correction .....	75
	Create position .....	75
	Saxo Group performs trade correction .....	76
	Appendix 13 – Bonds.....	77

Appendix 15 - Partial Fill.....	79
Place order .....	79
Partial fill .....	80
Order executed fully.....	81
Place sell order .....	81
Sell order executed fully.....	82
Appendix 16 - Future Contract – New Order .....	84
Appendix 17 - Future Contract – New Position .....	85
Appendix 18 - Contract Option – New Order.....	86
Appendix 19 - Contract Option – New Position .....	87
Appendix 20 – Tool ID .....	88

# 1 Document properties

## 1.1 Purpose of this document

The purpose of this document is to provide a detailed description of the functionality of the Trade Event Notification Service and enable the Institutional Client to integrate the TENS solution with own systems.

## 1.2 Document location

Commercial Resources

## 1.3 Revision history

DATE	Version	Change
10-04-2012	0.1	First draft
21-06-2012	1.0	First version
14-11-2013	1.1	Section 3.4.3.1, 3.6.3.1 and Appendix 1, have been updated with margin level dropped action call.
14-11-2013	1.2	Section 3.4.3.1, 3.6.3.1, Appendix 1 and Appendix 10, have been updated, 5 fields have been added to the margin call notification.
12-04-2014	1.3	Added field Expiry Date to contract Options and Contract Furtures on the positions and the orders notification events.
31-03-2015	2.0	New look&feel
19-10-2015	2.1	Actions vs Products Table updated
31-05-2016	2.2	Added Bond Sample to Annex

## 2 Preface

### 2.1 Overview

The Trade Event Notification Service is an optional integration functionality built to support Institutional Clients, who has a need for a real-time supplement to the End of Day files.

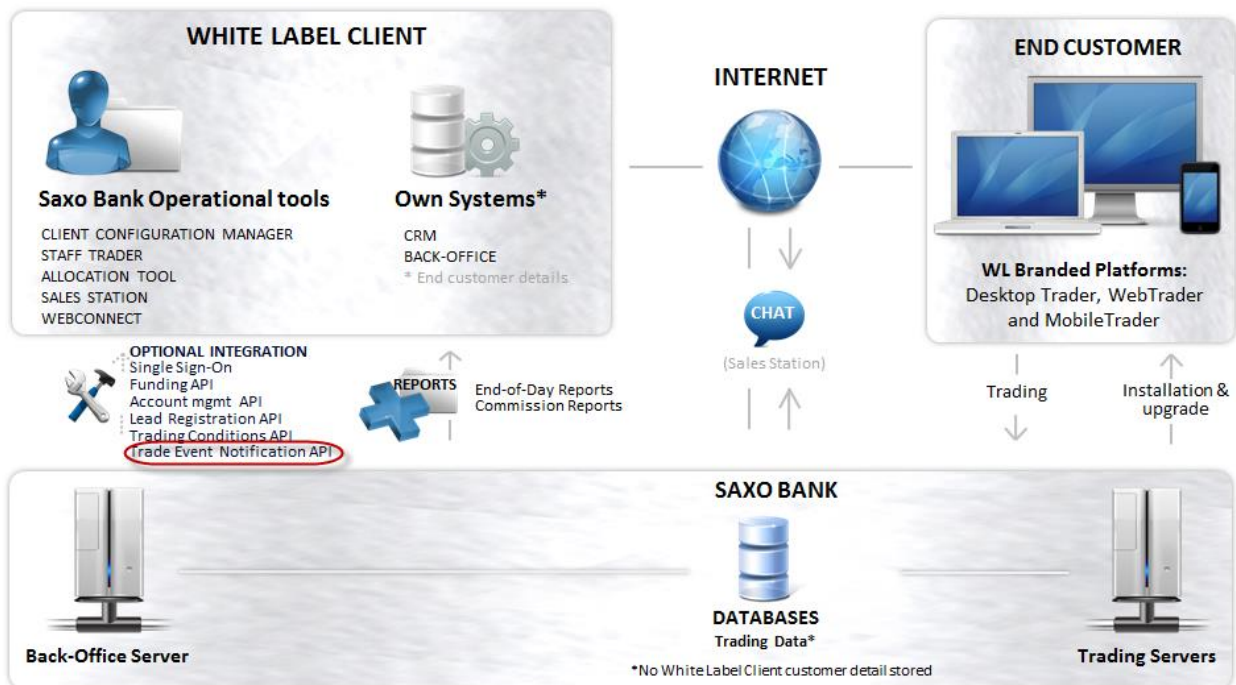


FIGURE 1 OVERVIEW OF SAXO SOLUTION WITH LINK TO TENS

TENS monitors the trade and process flows at Saxo Group. It publishes trades, position events, and system notifications. These notifications are available for both live and simulated systems.

TENS can be used for a number of purposes: As a risk management tool to record margin calls, as a service alert passed on to underlying customers (for a White Label Client) and for resolving specific reporting and reconciliation tasks at client side.

The TENSservice is able to provide real-time notifications in the following areas:

- Positions
- Orders
- Margin Calls
- Funding

Please see below figure for an overview of the notifications and the various delivery methods available.

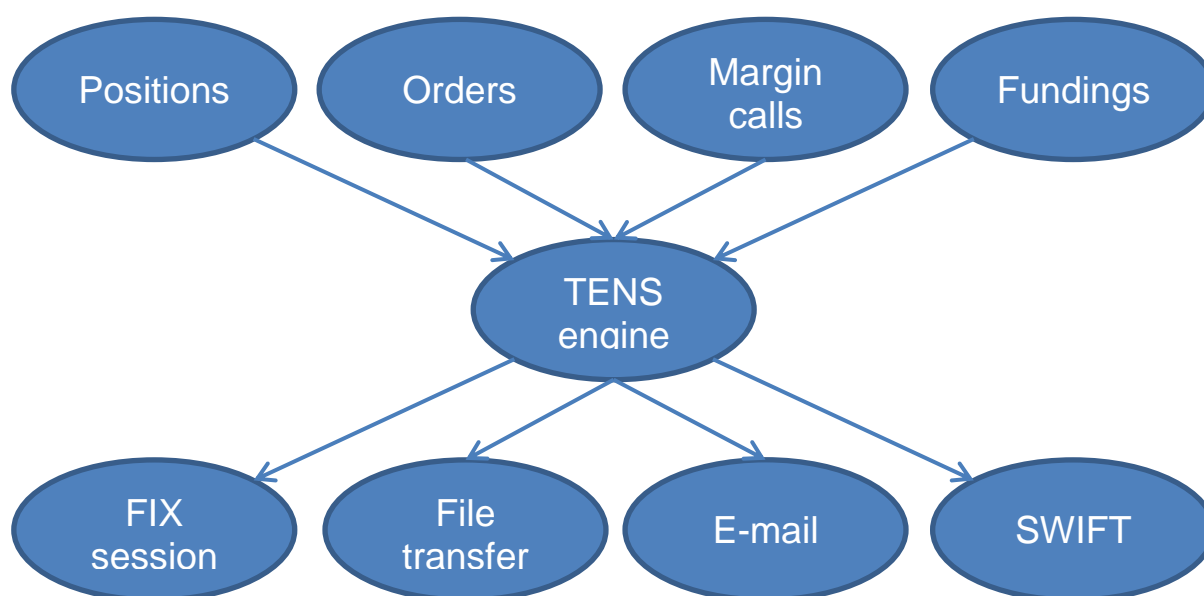


FIGURE 2 OVERVIEW OF TRADE EVENT NOTIFICATION SERVICE

## 2.2 Pre-requisites

In order to implement TENS it is important that the Institutional Client's organisation has the necessary knowledge and technical skills to implement the solution.

Task	Suggested skillset
Work with XML files (file transfer and e-mail)	Software development: XML parsing and XML Schema Definition (XSD).
Work with FIX messages	Software development: FIX protocol and engine.
Work with SWIFT messages	Software development: SWIFT parsing.
Establish file transfer	IT operations: File transfer protocols (FTPS/SFTP) and configuration of either a server or a client.
Extract e-mail	Software development: E-mail client API allowing access to client inbox.
Connect to FIX server	IT operations: Establishing connectivity using VPN.
Receive SWIFT messages	Banking: Understanding the SWIFT ecosystem.

## 3 Functionality

### 3.1 Delivery methods

TENS can deliver notifications using a variety of delivery methods. The notifications delivered using a particular delivery method will use a suitable format. E.g. if the deliver method is FIX the FIX format will be used.

#### 3.1.1 Delivery aspects

There are four different delivery methods. According to the estimated number of notifications per day (volume), each of these mechanisms may be more or less suitable. It is also worth to consider whether security, latency, format and reliability have importance.

- Volume: calculated in notifications pr. day
- Latency: time from trade event to the notification has arrived at the client
- Reliability: the general conception of reliability, or resistance to failure
- Security: is notification data encrypted between sender and receiver
- Format: notification format used with the corresponding transport type

Delivery	Volume	Latency	Reliability	Secure	Format
E-mail	<10K	<10s	Low	No	XML
File Transfer	<100K	<5s	Medium	Yes	XML
SWIFT	<100K	<5s	High	Yes	SWIFT
FIX	>100k	<1s	High	Yes	FIX

#### 3.1.2 E-mail

Each notification is delivered as a single mail message, which makes this delivery method most suitable for low volumes. When delivered by e-mail the content may either be attached as a file, or in the body text. This is not recommended if guaranteed delivery is required. Each configuration can specify several distinct e-mail addresses. E-mail delivery is through common SMTP mail servers at both the ends of the party, which may be burdened by other business tasks. This may increase the latency in some periods.

#### 3.1.3 File Transfer

The file based transports requires that the client has a server that fully supports the selected transfer protocol. Saxo Group will either initiate the connection, and login with a client supplied user account and password. Notifications are hereafter delivered as individual files in the root directory of the supplied user account. Or the client can login to the Saxo Group EFT server and download new notifications at scheduled intervals. Each notification is delivered as a separate file, named after the type of event, including a unique ID.

Messages stored on Saxo Group's EFT server will be purged after 30 days. If any additional clean-up is required by the client, the client will be given access to remove messages according to their need.

#### 3.1.4 SWIFT

SWIFT may be suitable where the infrastructure is already in place. The SWIFT delivery method is however restricted to position events and a subset of contract types. There may also apply a charge pr. notification sent in the SWIFT net, which may make it less attractive for medium and high volume.

#### 3.1.5 Fix

The FIX protocol is the best choice for high volume and low latency transfers. The clients are permanently connected to Saxo Group, and receive the events with a minimum of latency. The FIX protocol is based on version 4.4, and appended with a custom set of Saxo Group fields and messages. The communication is not encrypted by the FIX standards, which therefore requires a VPN connection to be secure.

##### 1.1.1.1 Connection hours:

Log on can start 00:00:00 UTC Sunday



Log off can be done at 23:50 UTC Friday

**NOTE:** Saxo Group is using two VPN solutions, based on either SSL or IPSEC. Currently only the SSL based solution is working with FIX delivery.

## 3.2 Notification formats

Each delivery method uses a specific format:

Delivery Method	Format
E-mail	XML
File Transfer	XML
SWIFT	SWIFT
FIX Session	FIX 4.4

The following sections describe each format in detail.

### 3.2.1 XML

The XML format used for notifications is defined by an XSD (see section 0 on page 33).

### 3.2.2 SWIFT

Only position notifications can be delivered using SWIFT. Furthermore only notifications about a subset of contracts can be delivered. Different contract types use different SWIFT message types as outlined below.

Contract type	SWIFT message type
FX Spot	MT 300
CFD, Future	MT 515
Bond, Stock	MT 541 / MT 542

**Note:** The amount of information contained in SWIFT notifications is considerable less compared to the other transports.

### 3.2.3 Fix Session

The FIX protocol used by TENS is FIX 4.4. Four different user defined messages have defined for each notification type and where appropriate user defined fields are being used in these messages. All user defined fields and messages are prefixed with "Saxo".

## 3.3 Notification triggers

When a notification is triggered, it is triggered by an action in the Saxo Group systems. Notifications are triggered whenever a change is detected. Changes happen when for instance positions are created and netted and when orders are created and filled or cancelled.

However, other changes may lead to updates. E.g. when the order price of a trailing stop order is changed because the market has moved a notification is generated. And when positions are rolled over each day a notification is generated with the updated value of the position.

These updates may happen frequently and it is not possible to suppress them based on a criteria. Subscribing to notifications is an all or nothing proposition where you either get all updates or none at all.

### 3.3.1 Products

The Trade Event Notification Service generates notifications for all available products. Below table lists all available products, maps them to the TENS contract type and lists the corresponding Saxo system abbreviation.

Product	TENS Contract type	System Abbreviation
Forex Spot & Forwards	FXSpot	VT
Forex Options	FxVanillaOption FxBinaryOption FxKnockInOption FxKnockOutOption FxNoTouchOption FxOneTouchOption	VO
Commodities	CfdOnFuture	FF
CFDs	Cfd	SF
Contract Options	ContractOption	CO
Share Options	ShareOption	SO
Contract Futures	ContractFuture	CF
Shares	Share	SH
Managed Funds	ManagedFund	MF
Bonds	Bond	BO

### 3.3.2 Actions

A notification can be triggered by several different actions. Below table describes the actions, which can be performed within the Saxo Group systems and the actions where the Saxo Group systems are affected.

Actions	Description
Option Exercise	When an Option is exercised
Option Expire	When an Option expires
Allocation	When a trade is allocated to a customer account via an allocation key
Position netting	When a position is netted by Saxo Group's Back Office
Cash Settlement	When a Cash Settlement is performed
Corporate Action	When a corporate action occurs
Saxo Initiates trade	When an employee of Saxo Group initiates a trade on a customer account
Customer trade	When a customer or a White Label Client Employee initiates a trade on a customer account
Margin Calls	When a margin call is triggered by the Margin Service
Customer Order	When a customer or White Label Client Employee places orders on a customer account
Trade Correction	The correction of a trade parameter that has not been settled or netted
Position Correction	The correction of a trade that is settled or has become a part of netting
Change ISIN code	ISIN code changed
Partial fill	When an order is partially filled

### 3.3.3 Actions and products

Not all actions are applicable to all products and some actions are applicable to more than one product. The table below outlines the actions that can be performed for the various products.

Action type	VT	VO	FF	SF	CO	SO	CF	SH	MF	BO
Option Exercise	-	X	-	-	-	-	-	-	-	-
Option Expire	-	X	-	-	X	X	-	-	-	-
Allocation	X									
Position netting	X									
Cash Settlement	X	-	-	-	-	-	-	-	-	-
Corporate Action	-	-	-	X			-	-	-	-
Saxo Initiated trade	X									
Customer trade	X					-				
Margin Calls	X									
Customer Order* (Order actions)	X				-					
Trade Correction	X									
Position Correction	X									
Change ISIN code	X									
Partial fill	X									

The following key is used:

-	This type of action cannot be performed for this instrument
x                      x	This type of action must be regarded individually for all instruments selected
x	This type of action can be shared across all instruments selected

Please see Appendix 2-15 for examples of the actions listed above.

### 3.3.3.1 Order behaviour

When subscribing to order notifications the notifications sent may not accurately reflect all changes made to the order. For instance if an order is partially filled three times in quick succession before it is completely filled and deleted, only some or even none of the partial fills may lead to a notification being generated. This only happens when an order changes very quickly within tenths of a second.

When an order notification is being generated it still accurately reflects that state of the order at the point in time it was generated and a notification for the final deletion of the order when it is completely filled is always generated. Furthermore, when an order is either filled or partially filled a notification for the resulting position is always generated.

Orders can be deleted for three reasons:

- The order is cancelled
- The order is filled
- The order expires

The exact information why an order is deleted is not included in the notification. However by applying a few rules the reason can be deducted:

- If a position is created having a SourceOrderId that is the same as the OrderId of the order then the order has been filled. Note that the position is also created or updated for a partial fill.
- If the order has a duration (i.e. has a Duration different from GoodTilCancel) and the order is deleted when the point in time defined by the duration of the order is reached then the order has expired.
- Otherwise the order was cancelled.

## 3.4 XML Description

### 3.4.1 Positions

The position notification is created when an event related to a position occurs. Cash deposit and withdrawal events are not included in the position notifications but are instead reported using the funding notification.

Below is a sample message for a new position.

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>77820</AccountId>
  <Amount>10</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>3179470</ClientId>
  <ContractType>Cfd</ContractType>
  <Created>2011-09-26T08:28:51.477</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>NASDAQ</ExchangeId>
  <ExecutionTime>2011-09-26T08:28:51.333</ExecutionTime>
  <Instrument>NAS100.I</Instrument>
  <IsinCode>US6311011026</IsinCode>
  <OpenPrice>2211.91</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>59846081</PositionId>
  <RegistrationTime>2011-09-26T08:28:51.437</RegistrationTime>
  <SourceOrderId>44085966</SourceOrderId>
  <Symbol>NAS100.I</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2011-09-26</ValueDate>
</Position>
```

#### 3.4.1.1 Position fields

Name	Required	Type	Description
AccountId	Yes	string	Identifies the account involved in the transaction.
Amount	No	decimal	Amount (e.g. number of securities) ordered.
BuySell	No	BuySell	Identifies the side of the transaction. <i>Buy:</i> Buy the security. <i>Sell:</i> Sell the security.
CallPut	No	CallPut	Identifies if the option is a call option or a put option. <i>Call:</i> The option is a call option. <i>Put:</i> The option is a put option.
ClientId	Yes	int	Identifies the client involved in the transaction.
Commission	No	decimal	The commission on the position (if there is any).
ContractType	No	ContractType	Identifies the type of the security. <i>Bond:</i> Bond. <i>Cfd:</i> Contract for difference. <i>CfdOnFuture:</i> Contract for difference on future. <i>ContractOption:</i> Option. <i>FutureContract:</i> Future. <i>FxBinaryOption:</i> Forex binary option. <i>FxKnockInOption:</i> Forex knock in option. <i>FxKnockOutOption:</i> Forex knock out option. <i>FxNoTouchOption:</i> Forex no touch option. <i>FxOneTouchOption:</i> Forex one touch option.

*FxSpot*: Forex spot.

*FxVanillaOption*: Forex vanilla option.

*ManagedFund*: Managed fund.

*Share*: Stock.

*ShareOption*: Stock option.

ConversionRate	No	decimal	The conversion rate used at the time of the transaction.
Created	Yes	dateTime	The time when this notification was created.
CurrencyCode	No	string	The currency of the transaction.
Delta	No	decimal	The delta value of the option.
ExchangeFee	No	decimal	Exchange fee charged for the trade. The currency is the currency of the contract.
Exchangeld	No	string	Identifies the exchange where the security is traded.
ExecutionTime	No	dateTime	Timestamp when the business transaction represented by the notification occurred.
ExpiryCut	No	string	The scheme used to determine the time of day the option expires. NY (New York) is 10 AM New York time. TY (Tokyo) is 3 PM Tokyo time.
ExpiryDate	No	date	Expiry date of the contract.
Instrument	No	string	The Saxo Group instrument.
IsinCode	No	string	The ISIN code for the instrument traded.
LowerBarrier	No	decimal	Exotic option lower barrier limit price.
OpenPrice	No	decimal	Price per security.
OpenSpot	No	decimal	Forex spot rate.
OpenSwap	No	decimal	Forex forward points added to OpenSpot. May be a negative value.
OptionExpiryDate	No	Date	Expiry date associated to a Contract Option or a Contract Future
OriginalAccountld	No	string	Account that the position originally belonged to (if it has been moved).
OriginatingPositionld	No	int	Identifies the option this position is associated with.
PositionEvent	Yes	PositionEvent	Describes the position event. <i>Deleted</i> : The position is deleted. <i>MarginStopOut</i> : The position is created as a result of a margin stop out. <i>New</i> : The position is created. <i>OptionExercised</i> : The position has been deleted when an option was exercised. <i>OptionExpired</i> : The position has been deleted when an option expired. <i>Updated</i> : The position has been updated.
Positionld	Yes	int	Identifies the position.
PremiumAmount	No	decimal	The premium of an FX option.
PremiumAmountCcy2	No	decimal	TBD
PriceType	No	PriceType	<i>Amount</i> : The quoted price is an amount <i>Ccy1Percentage</i> : The quoted price is a percentage of the first currency of the pair <i>Ccy1Pips</i> : The quoted price is in pips of the first currency of the currency pair <i>Ccy2Percentage</i> : The quoted price is a percentage of the second currency of the pair <i>Ccy2Pips</i> : The quoted price is in pips of the second currency of the currency pair <i>ThirdCurrency</i> : The quoted price is in the third currency
RegistrationTime	No	dateTime	The time when the transaction was registered by the bank.
RelatedOrderCount	No	int	The number of related orders (maximum 2).
RelatedPositionld	No	int	Identifies the position that this order/position is related to.
SourceOrderld	No	int	Identifies the order that was the source for this position.
SpotDate	No	date	The date when the initial transfer of the product of the transaction takes place.
StampDuty	No	decimal	Tax applied by an exchange.
StrikePrice	No	decimal	Option strike price.

Symbol	No	string	Symbol identifying the security using a common "human understood" representation.
SystemOrigin	No	SystemOrigin	Identifies the system where the notification originates from. <i>B2B</i> : The notification is created as a result of an action in the B2B service. <i>ClientStation</i> : The notification is created as a result of an action in DesktopTrader. <i>MobileTrader</i> : The notification is created as a result of an action in MobileTrader. <i>Other</i> : The notification is created as a result of an action in another system, e.g. internally in Saxo Group. <i>WebTrader</i> : The notification is created as a result of an action in WebTrader.
ToOpenClose	No	ToOpenClose	Specifies whether the trade is closing an existing position or opening a new position. <i>ToClose</i> : The trade is closing an existing position. <i>ToOpen</i> : The trade is opening a new position.
UpperBarrier	No	decimal	Exotic option upper barrier limit price.
ValueDate	No	date	The date when the final transfer of the product of the transaction takes place.
Volatility	No	decimal	The volatility value of the option.

### 3.4.2 Orders

The order notification is created when an event related to an order occurs.

Below is a sample message for a changed order.

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>77820</AccountId>
  <Amount>30000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>3179470</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2011-09-19T17:48:24.747</Created>
  <CurrencyCode>JPY</CurrencyCode>
  <Duration>DayOrder</Duration>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <FilledAmount>20</FilledAmount>
  <Instrument>USDJPY</Instrument>
  <OrderId>130790572</OrderId>
  <OrderRelation>Oco</OrderRelation>
  <OrderType>StopIfOffered</OrderType>
  <Price>75.78</Price>
  <RegistrationTime>2011-09-19T17:48:24.747</RegistrationTime>
  <Symbol>USD/JPY</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <UserId>4662393</UserId>
</Order>
```

#### 3.4.2.1 Order fields

Name	Required	Type	Description
AccountId	Yes	string	Identifies the account involved in the transaction.
Amount	No	decimal	Amount (e.g. number of securities) ordered.
BuySell	No	BuySell	Identifies the side of the transaction. <i>Buy</i> : Buy the security. <i>Sell</i> : Sell the security.
CallPut	No	CallPut	Identifies if the option is a call option or a put option. <i>Call</i> : The option is a call option.

			<i>Put</i> : The option is a put option.
ClientId	Yes	int	Identifies the client involved in the transaction.
ClientOrderId	No	string	Client assigned order identification.
ContractType	No	ContractType	Identifies the type of the security. <i>Bond</i> : Bond. <i>Cfd</i> : Contract for difference. <i>CfdOnFuture</i> : Contract for difference on future. <i>ContractOption</i> : Option. <i>FutureContract</i> : Future. <i>FxBinaryOption</i> : Forex binary option. <i>FxKnockInOption</i> : Forex knock in option. <i>FxKnockOutOption</i> : Forex knock out option. <i>FxNoTouchOption</i> : Forex no touch option. <i>FxOneTouchOption</i> : Forex one touch option. <i>FxSpot</i> : Forex spot. <i>FxVanillaOption</i> : Forex vanilla option. <i>ManagedFund</i> : Managed fund. <i>Share</i> : Stock. <i>ShareOption</i> : Stock option.
Created	Yes	dateTime	The time when this notification was created.
CurrencyCode	No	string	The currency of the transaction.
Duration	No	Duration	Specifies how long the order remains in effect. <i>AtTheClose</i> : The order is either filled at the close of the market or, if that is not possible, canceled. <i>AtTheOpening</i> : The order is either filled at the opening of the market or, if that is not possible, canceled. <i>DayOrder</i> : The order is automatically canceled at the end of the trading day. <i>FillOrKill</i> : The order is either immediately filled for the entire amount or, if that is not possible, canceled. <i>GoodTillCancel</i> : The order is in effect until canceled by the client. <i>GoodTillDate</i> : The order is in effect until the specified expiry date. <i>ImmediateOrCancel</i> : The order is either immediately filled or, if that is not possible, canceled. If only a partial fill is possible the order is still filled.
ExchangedId	No	string	Identifies the exchange where the security is traded.
ExecutionType	Yes	ExecutionType	Describes the order event. <i>Changed</i> : Order has been changed. <i>Deleted</i> : Order has been deleted either as a result of being filled or canceled or it has expired. <i>New</i> : New order.
ExpiryDate	No	date	Expiry date of the contract.
FilledAmount	No	decimal	Amount (e.g. number of securities) filled.
Instrument	Yes	string	The Saxo Group instrument.
IsinCode	No	string	The ISIN code for the instrument traded.
OptionExpiryDate	No	Date	Expiry date associated to a Contract Option or a Contract Future.
OrderId	Yes	int	Identifies the order.
OrderRelation	No	OrderRelation	<i>IfDoneMaster</i> : The order is a stop order with a related order that comes into effect if this order is executed. <i>IfDoneSlave</i> : The order is a related order that only comes into effect when the related order is executed. <i>IfDoneSlaveOco</i> : The order is a related order that only comes into effect when the related order is executed. Furthermore, another related order exists that if executed will cancel this order. <i>Oco</i> : One-Cancels-the-Other. The order is canceled if the related order is executed. <i>StandAlone</i> : The order is a stand-alone order and execution does not depend on other



orders.

OrderType	No	OrderType	<p>The type of the order.</p> <p><i>Algorithmic</i>: Is an order to optimize execution by minimizing the combination of price impact and the risk of potential price movements. Supported Algorithmic Order Types: Reload, VWAP, IS, Iceberg, With Volume, Smart Dark, Pre-Market.</p> <p><i>CallLimit</i>: TBD</p> <p><i>CallStop</i>: TBD</p> <p><i>GuaranteedStop</i>: TBD</p> <p><i>Limit</i>: The order is a limit order.</p> <p><i>Market</i>: The order is a market order.</p> <p><i>MarketExpiry</i>: The order is a market order. Gets triggered on expiry for instruments that have an expiration date. (CFDs on Commodities + Futures)</p> <p><i>MarketStopOut</i>: The order is a market order. The order gets triggered automatically if margin requirement is exceeded.</p> <p><i>StopIfBids</i>: The order is a stop order. The order execution criterion is evaluated by comparing the bid price to the limit price.</p> <p><i>StopIfOffered</i>: The order is a stop order. The order execution criterion is evaluated by comparing the offer price to the limit price.</p> <p><i>StopIfTraded</i>: The order is a stop order. The order execution criterion is evaluated by comparing the trade price to the limit price.</p> <p><i>StopLimit</i>: An order placed with a broker that combines the features of stop order with those of a limit order. A stop-limit order will be executed at a specified price (or better) after a given stop price has been reached. Once the stop price is reached, the stop-limit order becomes a limit order to buy (or sell) at the limit price or better.</p>
OriginalOrderId	No	int	Identifies the original order that was used to create this order.
OriginatingPositionId	No	int	Identifies the option this position is associated with.
Price	No	decimal	Price per unit.
RegistrationTime	No	dateTime	The time when the transaction was registered by the bank.
RelatedOrderId	No	int	Identifies the related order for complex order types.
RelatedPositionId	No	int	Identifies the position that this order/position is related to.
RelatedSecondOrderId	No	int	Identifies the second related order for complex order types.
StrikePrice	No	decimal	Option strike price.
Symbol	No	string	Symbol identifying the security using a common "human understood" representation.
SystemOrigin	No	SystemOrigin	<p>Identifies the system where the notification originates from.</p> <p><i>B2B</i>: The notification is created as a result of an action in the B2B service.</p> <p><i>ClientStation</i>: The notification is created as a result of an action in SaxoTrader.</p> <p><i>MobileTrader</i>: The notification is created as a result of an action in MobileTrader.</p> <p><i>Other</i>: The notification is created as a result of an action in another system, e.g. internally in Saxo Group.</p> <p><i>WebTrader</i>: The notification is created as a result of an action in WebTrader.</p>
ToOpenClose	No	ToOpenClose	<p>Specifies whether the trade is closing an existing position or opening a new position.</p> <p><i>ToClose</i>: The trade is closing an existing position.</p> <p><i>ToOpen</i>: The trade is opening a new position.</p>
TrailingStopPriceChangeThreshold	No	decimal	The trailing step size for a trailing stop order. When the market price moves in the direction of the trade and the distance from the order price exceeds the sum of the distance to market and the trailing step size the order price is adjusted in the direction of the trade to trail the market.
TrailingStopPriceDifference	No	decimal	The distance to market for a trailing stop order. When the market price moves in the direction of the trade and the distance from the order price exceeds the sum of the distance to market and the trailing step size the order price is adjusted in the direction of the trade to trail the market.
UserId	No	int	Identifies the user that created the order.

### 3.4.3 Margin Calls

The margin call notification is created when an event related to a margin call occurs.

Below is a sample message for a margin call.

```
<?xml version="1.0" encoding="utf-8"?>
<MarginCall>
  <BaseCurrency>USD</BaseCurrency>
  <ClientId>1973785</ClientId>
  <Created>2013-11-11T09:55:46.817</Created>
  <DefaultAccountId>19373785</DefaultAccountId>
  <MarginCallAction>MarginCall</MarginCallAction>
  <MarginCallLevel>110</MarginCallLevel>
  <MarginDeficit>28295.91</MarginDeficit>
  <MarginRequired>298506.73</MarginRequired>
  <NetEquityForMargin>270210.82</NetEquityForMargin>
  <NetFreeBalance>-28295.91</NetFreeBalance>
  <UseOfEquityForMargin>1.10</UseOfEquityForMargin>
</MarginCall>
```

#### 3.4.3.1 Margin call fields

Name	Required	Type	Description
BaseCurrency	Yes	string	The default currency for the client.
ClientId	Yes	int	Identifies the client involved in the transaction.
Created	Yes	dateTime	The time when this notification was created.
DefaultAccountId	Yes	string	The default account ID for the client.
MarginCallAction	Yes	MarginCallAction	Describes the action associated with this margin call. <i>MarginCall:</i> The client is requested to restore the necessary margin required. <i>Reinstate:</i> The system has restored the necessary margin required. <i>StopOut:</i> The client has been forcefully stopped out to restore the necessary margin required. <i>LevelDropped:</i> Margin level drooped, The client has restored the necessary margin level required, according to the margin call profile defined. This margin call action will appear if the client margin level dropped under the defined margin levels percentage as set in the margin call profile.
MarginCallLevel	Yes	int	The percentage of margin used. Please Note, in a LevelDropped margin call scenario, where the margin level is 0, this field <b>will not be</b> included in the XML message. Margin call level equal to 0 can accrue in 2 different scenarios. Scenario 1: If the margin level has drooped under the last percentage defined in the margin call profile, then the margin call level will be equal to 0, in this case the <i>MarginCallLevel</i> field will not be included in the Margin Call notification. Scenario 2: In case of account funding where the account balance was negative, margin call level will be equal to 0, in this case the <i>MarginCallLevel</i> field will not be included in the Margin Call notification.
MarginDeficit	Yes	String	The value in this field represents the available amount that the client has for trading on margin, the value displayed in this field reflecting Saxo group point of view, hence the value will always be positive during a margin call.
MarginRequired	Yes	String	The value in this field represents the total margin amount required to support the various exposures, created by the client's prevailing holdings in any margin tradable product.
NetEquityForMargin	Yes	String	Total net equity available for margin trading.
NetFreeBalance	Yes	String	The value in this field represents the available amount that the client has for trading on margin, the value displayed in this field is reflecting the client point of view. Pending on the margin call level, this field could have a negative representation during a margin call.

UseOfEquityForMargin	Yes	String	The value in this field represents the percentage of the net exposure in relation to the account value.
----------------------	-----	--------	---

### 3.4.4 Funding

The funding notification is created when money is added or deducted from an account.

Below is a sample message for a funding deposit.

```
<?xml version="1.0" encoding="utf-8"?>
<Funding>
  <AccountId>77820</AccountId>
  <Amount>14000</Amount>
  <ClientId>3179470</ClientId>
  <Created>2010-08-18T08:14:14.58</Created>
  <CurrencyCode>NOK</CurrencyCode>
  <FundingEvent>New</FundingEvent>
  <FundingType>Deposit</FundingType>
  <PositionId>276302329</PositionId>
  <RegistrationTime>2010-08-18T08:14:14.573</RegistrationTime>
  <ValueDate>2010-08-19</ValueDate>
</Funding>
```

#### 3.4.4.1 Funding fields

Name	Required	Type	Description
AccountId	Yes	string	Identifies the account involved in the transaction.
Amount	Yes	decimal	Amount (e.g. number of securities) ordered.
ClientId	Yes	int	Identifies the client involved in the transaction.
ConversionRate	No	decimal	The conversion rate used at the time of the transaction.
Created	Yes	dateTime	The time when this notification was created.
CurrencyCode	Yes	string	The currency of the transaction.
FundingEvent	Yes	FundingEvent	Describes the funding event. <i>Deleted:</i> The funding position is deleted. <i>New:</i> The funding position is created. <i>Updated:</i> The funding position has been updated.
FundingType	Yes	FundingType	Describes the type of the funding. <i>Deposit:</i> Money is added to the account. <i>Withdrawal:</i> Money is deducted from the account.
PositionId	Yes	int	Identifies the position.
RegistrationTime	Yes	dateTime	The time when the transaction was registered by the bank.
ValueDate	Yes	date	The date when the final transfer of the product of the transaction takes place.

## 3.5 SWIFT Description

### 3.5.1 Positions

The position notification is created when an event related to a position occurs.

#### 3.5.1.1 FX positions

Notifications about FX positions are sent using the SWIFT MT 300 foreign exchange confirmation message type.

Name	Required	Type	Description
New Sequence	15A	Yes	(Empty field.)
Sender's Reference	20	Yes	Identifies the position.
Related Reference	21	No	When the type of operation is AMND or CANC the same value as sender's reference (the position ID).
Type of Operation	22A	Yes	AMND: The position is updated. CANC: The position is deleted. NEWT: The position is created.
Common Reference	22C	Yes	A common reference created from sender BIC, counterparty BIC and the price.
Party A	82A	Yes	Sender BIC (Saxo Group).
Party B	87A	Yes	Counterparty BIC (the receiver of the SWIFT message).
New Sequence	15B	Yes	(Empty field.)
Trade Date	30T	Yes	Timestamp when the business transaction represented by the notification occurred.
Value Date	30V	Yes	The date when the final transfer of the product of the transaction takes place.
Exchange Rate	36	Yes	Price per unit.
Amount Bought: Currency, Amount	32B	Yes	When buying an FX pair this is the counter currency amount sold. When selling an FX pair this is the base currency amount bought.
Amount Bought: Receiving Agent	57A	Yes	BIC of the financial institution buying the counter currency when an FX pair is bought and buying the base currency when an FX pair is sold (Saxo Group).
Amount Sold: Currency, Amount	33B	Yes	When buying an FX pair this is the base currency amount bought. When selling an FX pair this is the counter currency amount sold.
Amount Sold: Receiving Agent	57A	Yes	BIC of the financial institution selling the counter currency when an FX pair is bought and buying the counter currency when an FX pair is sold (the receiver of the SWIFT message).

#### 3.5.1.2 CFD and future positions

Notifications about CFD and future positions are sent using the SWIFT MT 515 client confirmation of purchase or sale message type.

Name	Required	Type	Description
Start of Block	16R	Yes	GENL
Sender's Message Reference	20C	Yes	Identifies the position.
Function of the Message	23G	Yes	CANC/DUPL: The position is deleted. NEWM/DUPL: The position is created or updated.
Trade Transaction Type	22F	Yes	TRTR/TRAD
Start of Block	16R	No	LINK
Previous Message Reference	20C	No	When the function of the message is CANC the same value as sender's message reference (the position ID).
End of Block	16S	No	LINK
Start of Block	16R	Yes	LINK
Deal Reference	20C	Yes	TRRF//ABCDEFGHIJKLMNPO

End of Block	16S	Yes	LINK
End of Block	16S	Yes	GENL
Start of Block	16R	Yes	CONFDET
Settlement Date/Time	98A	Yes	The date when the final transfer of the product of the transaction takes place.
Trade Date/Time	98C	Yes	Timestamp when the business transaction represented by the notification occurred.
Deal Price	90B	Yes	The total price (amount × price per unit).
Buy/Sell Indicator	22H	Yes	Identifies the side of the transaction. <i>BUY</i> : Buy the security. <i>SELL</i> : Sell the security.
Payment Indicator	22H	Yes	APMT
Start of Block	16R	Yes	CONFPTY
ETC Service Provider 1	95P	Yes	Sender BIC (Saxo Group)
Cash Account	97A	Yes	Identifies the account involved in the transaction.
Declaration Details Narrative	70E	Yes	Identifies the type of the security. <i>CFD</i> : Contract for difference. <i>FUT</i> : Future.
End of Block	16S	Yes	CONFPTY
Quantity of Financial Instrument Confirmed	36B	Yes	Amount (e.g. number of securities) ordered.
Identification of the Financial Instrument	35B	Yes	ISIN identifier if it is defined for the instrument. Otherwise the ticker symbol.

### 3.5.1.3 Bond and stock positions

Notifications about bond and stock positions are sent using the SWIFT MT 541 receive against payment message type when securities are bought and the SWIFT MT 543 deliver against payment message type when securities are sold.

Name	Required	Type	Description
Start of Block	16R	Yes	GENL
Sender's Message Reference	20C	Yes	Identifies the account involved in the transaction.
Function of the Message	23G	Yes	<i>CANC</i> : The position is deleted. <i>NEWM</i> : The position is created or updated.
Start of Block	16R	No	LINK
Previous Message Reference	20C	No	When the function of the message is <i>CANC</i> the same value as sender's message reference (the account ID).
End of Block	16S	No	LINK
End of Block	16S	Yes	GENL
Start of Block	16R	Yes	TRADET
Settlement Date/Time	98A	Yes	Timestamp when the business transaction represented by the notification occurred.
Identification of the Financial Instrument	35B	Yes	The ISIN code for the instrument traded.
End of Block	16S	Yes	TRADET
Start of Block	16R	Yes	FIAC
Quantity of Financial Instrument	36B	Yes	Amount (e.g. number of securities) ordered.
Safekeeping Account	97A	Yes	Identifies the account involved in the transaction.
End of Block	16S	Yes	FIAC

Start of Block	16R	Yes	SETDET
Type of Settlement Transaction Indicator	22F	Yes	SETR/TRAD
Start of Block	16R	Yes	SETPRTY
Delivering Agent or Receiving Agent	95P	Yes	When buying (MT 541) <i>DEAG</i> or selling (MT 543) <i>REAG</i> and the sender BIC (Saxo Group)
End of Block	16S	Yes	SETPRTY
Start of Block	16R	Yes	SETPRTY
Seller or Buyer	95P	Yes	When buying (MT 541) <i>SELL</i> or selling (MT 543) <i>BUYR</i> and the sender BIC (Saxo Group).
End of Block	16S	Yes	SETPRTY
Start of Block	16R	Yes	AMT
Settlement Amount	19A	Yes	The total price (amount x price per unit).
End of Block	16S	Yes	AMT
End of Block	16S	Yes	SETDET

## 3.6 FIX Description

### 3.6.1 Positions

FIX MsgType for a position message is U4.

The position notification is created when an event related to a position occurs. Cash deposit and withdrawal events are not included in the position notifications but are instead reported using the funding notification.

Below is a sample message for a new position. The SOH field delimiter character is displayed as '\*'.

```
8=FIX.4.4*9=250*35=U4*1=77820*14=10*22=4*37=44085966*44=2211.91*48=US6311011026*54=1*55=
NAS100.I*60=20110926-08:28:51*100=NASDAQ*109=3179470*769=20110926-
08:28:51*20003=C*20005=20110926-
08:28:51*20006=USD*20014=NAS100.I*20023=59846081*20024=0*20034=0*20039=20110926*10=190*
```

#### 3.6.1.1 Position fields

Name	Tag	Required	Type	Description
Account	1	Yes	String	Identifies the account involved in the transaction. The default account ID for the client.
ClientID	109	Yes	String	Identifies the client involved in the transaction. The client identifier is a non-negative integer (31 bits).
CumQty	14	No	Qty	Amount (e.g. number of securities) filled.
ExDestination	100	No	Exchange	Identifies the exchange where the security is traded.
LastForwardPoints	195	No	PriceOffset	Forex forward points added to LastSpotRate (194). May be a negative value.
LastSpotRate	194	No	Price	Forex spot rate.
OrderID	37	No	String	Identifies the order. The order identifier is a non-negative integer (31 bits).
Price	44	No	Price	Price per unit.
SaxoCallPut	20001	No	Char	Identifies if the option is a call option or a put option. <i>Call (C)</i> : The option is a call option. <i>Put (P)</i> : The option is a put option.

SaxoCommission	20002	No	Amt	The commission on the position (if there is any).
SaxoContractType	20003	No	Char	Identifies the type of the security. FxSpot (0): Forex. FxVanillaOption (1): Forex vanilla option. FxKnockInOption (2): Forex knock in option. FxKnockOutOption (3): Forex knock out option. FxBinaryOption (4): Forex binary option. FxOneTouchOption (5): Forex one touch option. FxNoTouchOption (6): Forex no touch option. FutureContract (7): Future. ContractOption (8): Option. Share (9): Share. ShareOption (A): Share option. Bond (B): Bond. Cfd (C): Contract for difference. ManagedFund (D): Managed fund. CfdOnFuture (G): Contract for difference on future.
SaxoConversionRate	20004	No	Float	The conversion rate used at the time of the transaction.
SaxoCreated	20005	Yes	UTCTimeStamp	The time when this notification was created.
SaxoCurrencyCode	20006	No	Currency	The currency of the transaction.
SaxoDelta	20007	No	Float	The delta value of the option.
SaxoExchangeFee	20008	No	Amt	Exchange fee charged for the trade. The currency is the currency of the contract.
SaxoExpiryCut	20010	No	String	The scheme used to determine the time of day the option expires. NY (New York) is 10 AM New York time. TY (Tokyo) is 3 PM Tokyo time.
SaxoExpiryDate	20011	No	UTCDateOnly	Expiry date of the contract.
SaxoInstrument	20014	No	String	The Saxo Group instrument.
SaxoLowerBarrier	20015	No	Price	Exotic option lower barrier limit price.
SaxoOptionExpiryDate	20077	No	UTCDateOnly	Expiry Date associated to Contract Options or Contract Futures.
SaxoOriginalAccountID	20020	No	String	Account that the position originally belonged to (if it has been moved).
SaxoOriginatingPositionID	20022	No	Int	Identifies the option this position is associated with.
SaxoPositionID	20023	Yes	Int	Identifies the position. The position identifier is a non-negative integer (31 bits).
SaxoPositionEvent	20024	Yes	Char	Describes the position event. New (0): The position is created. Updated (1): The position has been updated. Deleted (2): The position is deleted. MarginStopOut (3): The position is created as a result of a margin stop out. OptionExercised (4): The position has been deleted when an option was exercised. OptionExpired (5): The position has been deleted when an option expired.
SaxoPremiumAmount	20025	No	Price	The premium of an FX option.
SaxoPremiumAmountCcy2	20026	No	Price	TBD
SaxoPriceType	20027	No	Char	Amount: The quoted price is an amount Ccy1Percentage: The quoted price is a percentage of the first currency of the pair Ccy1Pips: The quoted price is in pips of the first currency of the currency pair Ccy2Percentage: The quoted price is a percentage of the second

				currency of the pair <i>Ccy2Pips</i> : The quoted price is in pips of the second currency of the currency pair <i>ThirdCurrency</i> : The quoted price is in the third currency
SaxoRelatedOrderCount	20028	No	Int	The number of related orders (maximum 2).
SaxoRelatedPositionID	20030	No	Int	Identifies the position that this order/position is related to.
SaxoSpotDate	20032	No	UTCDateOnly	The date when the initial transfer of the product of the transaction takes place.
SaxoStampDuty	20033	No	Amt	Tax applied by an exchange.
SaxoSystemOrigin	20034	No	int	Identifies the system where the notification originates from. Please refer to Appendix 20 for a list of all Origins
SaxoToOpenClose	20037	No	Char	Specifies whether the trade is closing an existing position or opening a new position. <i>ToClose (C)</i> : The trade is closing an existing position. <i>ToOpen (O)</i> : The trade is opening a new position.
SaxoUpperBarrier	20038	No	Price	Exotic option upper barrier limit price.
SaxoValueDate	20039	No	UTCDateOnly	The date when the final transfer of the product of the transaction takes place.
SaxoVolatility	20040	No	Float	The volatility value of the option.
SecurityID	48	No	String	The ISIN code for the instrument traded.
SecurityIDSource	22	No	String	The type of the SecurityID field. <i>ISIN (4)</i> : The SecurityID is an ISIN code.
Side	54	No	Char	Identifies the side of the transaction. <i>Buy (1)</i> : Buy the security. <i>Sell (2)</i> : Sell the security.
StrikePrice	202	No	Price	Option strike price.
Symbol	55	No	String	Symbol identifying the security using a common "human understood" representation.
TransactTime	60	No	UTCTimeStamp	Timestamp when the business transaction represented by the notification occurred.
TrdRegTimestamp	769	No	UTCTimeStamp	The time when the transaction was registered by the bank.

### 3.6.2 Orders

FIXMsgType for an order message is U3.

The order notification is created when an event related to an order occurs.

Below is a sample message for a changed order. The SOH field delimiter character is displayed as '•'.

```
8=FIX.4.4•9=214•35=U3•115=4662393•1=77820•14=20•37=130790572•38=30000•44=75.78•54=1•55=U
SD/JPY•59=0•100=SBFX•109=3179470•769=20110919-17:48:24•20003=0•20005=20110919-
17:48:24•20006=JPY•20009=1•20014=USDJPY•20018=1•20019=4•20034=0•10=221•
```

#### 3.6.2.1 Order fields

Name	Tag	Required	Type	Description
Account	1	Yes	String	Identifies the account involved in the transaction. The default account ID for the client.
ClientID	109	Yes	String	Identifies the client involved in the transaction. The client identifier is a non-negative integer (31 bits).
ClOrdID	11	No	String	Client assigned order identification.



CumQty	14	No	Qty	Amount (e.g. number of securities) filled.
ExDestination	100	No	Exchange	Identifies the exchange where the security is traded.
OrderID	37	Yes	String	Identifies the order. The order identifier is a non-negative integer (31 bits).
OrderQty	38	No	Qty	Amount (e.g. number of securities) ordered.
Price	44	No	Price	Price per unit.
SaxoCallPut	20001	No	Char	Identifies if the option is a call option or a put option. <i>Call (C)</i> : The option is a call option. <i>Put (P)</i> : The option is a put option.
SaxoContractType	20003	No	Char	Identifies the type of the security. <i>FxSpot (0)</i> : Forex. <i>FxVanillaOption (1)</i> : Forex vanilla option. <i>FxKnockInOption (2)</i> : Forex knock in option. <i>FxKnockOutOption (3)</i> : Forex knock out option. <i>FxBinaryOption (4)</i> : Forex binary option. <i>FxOneTouchOption (5)</i> : Forex one touch option. <i>FxNoTouchOption (6)</i> : Forex no touch option. <i>FutureContract (7)</i> : Future. <i>ContractOption (8)</i> : Option. <i>Share (9)</i> : Stock. <i>ShareOption (A)</i> : Stock option. <i>Bond (B)</i> : Bond. <i>Cfd (C)</i> : Contract for difference. <i>ManagedFund (D)</i> : Managed fund. <i>CfdOnFuture (G)</i> : Contract for difference on future.
SaxoCreated	20005	Yes	UTCTimeStamp	The time when this notification was created.
SaxoCurrencyCode	20006	No	Currency	The currency of the transaction.
SaxoExecType	20009	Yes	Char	Describes the order event. <i>New (0)</i> : New order. <i>Changed (1)</i> : Order has been changed. <i>Deleted (2)</i> : Order has been deleted either as a result of being filled or canceled or it has expired.
SaxoExpiryDate	20011	No	UTCDateOnly	Expiry date of the contract.
SaxoInstrument	20014	No	String	The Saxo Group instrument.
SaxoOptionExpiryDate	20077	No	UTCDateOnly	Expiry Date associated to Contract Options or Contract Futures.
SaxoOrderRelation	20018	No	Char	Describes the relationship between this order and any related orders identified by SaxoRelatedOrderID and SaxoRelatedSecondOrderID. <i>Oco (1)</i> : One-Cancels-the-Other. The order is canceled if the related order is executed. <i>IfDoneMaster (2)</i> : The order is a stop order with a related order that comes into effect if this order is executed. <i>IfDoneSlave (3)</i> : The order is a related order that only comes into effect when the related order is executed. <i>IfDoneSlaveOco (4)</i> : The order is a related order that only comes into effect when the related order is executed. Furthermore, another related order exists that if executed will cancel this order.
SaxoOrderType	20019	No	Char	The type of the order. <i>Market (1)</i> : The order is a market order. <i>Limit (2)</i> : The order is a limit order. <i>StopIfBid (3)</i> : The order is a stop order. The order execution criterion is evaluated by comparing the bid price to the limit price. <i>StopIfOffered (4)</i> : The order is a stop order. The order execution criterion is evaluated by comparing the offer price to the limit price.

				<p><i>StopIfTraded (5):</i> The order is a stop order. The order execution criterion is evaluated by comparing the trade price to the limit price.</p> <p><i>GuaranteedStop (6):</i> TBD</p> <p><i>StopLimit (7):</i> An order placed with a broker that combines the features of stop order with those of a limit order. A stop-limit order will be executed at a specified price (or better) after a given stop price has been reached. Once the stop price is reached, the stop-limit order becomes a limit order to buy (or sell) at the limit price or better.</p> <p><i>MarketStopOut (8):</i> The order is a market order. The order gets triggered automatically if margin requirement is exceeded.</p> <p><i>StopIfBid:</i> The order is a stop order. The order execution criterion is evaluated by comparing the bid price to the limit price.</p> <p><i>CallLimit (9):</i> TBD</p> <p><i>CallStop (A):</i> TBD</p> <p><i>MarketExpiry (B):</i> The order is a market order. Gets triggered on expiry for instruments that have an expiration date. (CFDs on Commodities + Futures)</p> <p><i>Algorithmic (C):</i> Is an order to optimize execution by minimizing the combination of price impact and the risk of potential price movements. Supported Algorithmic Order Types: Reload, VWAP, IS, Iceberg, With Volume, Smart Dark, Pre-Market.</p>
SaxoOriginalOrderID	20021	No	Int	Identifies the original order that was used to create this order.
SaxoOriginatingPositionID	20022	No	Int	Identifies the option this position is associated with.
SaxoRelatedOrderID	20029	No	Int	Identifies the related order for complex order types.
SaxoRelatedPositionID	20030	No	Int	Identifies the position that this order/position is related to.
SaxoRelatedSecondOrderID	20031	No	Int	Identifies the second related order for complex order types.
SaxoSystemOrigin	20034	No	int	Identifies the system where the notification originates from. Please refer to Appendix 20 for a list of all Origins
SaxoToOpenClose	20037	No	Char	Specifies whether the trade is closing an existing position or opening a new position. <i>ToClose (C):</i> The trade is closing an existing position. <i>ToOpen (O):</i> The trade is opening a new position.
SaxoTrailingStopPriceChangeThreshold	20035	No	PriceOffset	The trailing step size for a trailing stop order. When the market price moves in the direction of the trade and the distance from the order price exceeds the sum of the distance to market and the trailing step size the order price is adjusted in the direction of the trade to trail the market.
SaxoTrailingStopPriceDifference	20036	No	PriceOffset	The distance to market for a trailing stop order. When the market price moves in the direction of the trade and the distance from the order price exceeds the sum of the distance to market and the trailing step size the order price is adjusted in the direction of the trade to trail the market.
SecurityID	48	No	String	The ISIN code for the instrument traded.
SecurityIDSource	22	No	String	The type of the SecurityID field. <i>ISIN (4):</i> The SecurityID is an ISIN code.
Side	54	No	Char	Identifies the side of the transaction. <i>Buy (1):</i> Buy the security. <i>Sell (2):</i> Sell the security.
StrikePrice	202	No	Price	Option strike price.
Symbol	55	No	String	Symbol identifying the security using a common "human understood" representation.
TimeInForce	59	No	Char	Specifies how long the order remains in effect. <i>Day (0):</i> The order is automatically canceled at the end of the trading day. <i>GoodTillCancel (1):</i> The order is in effect until canceled by the client. <i>AtTheOpening (2):</i> The order is either filled at the opening of the

market or, if that is not possible, canceled.

*ImmediateOrCancel (3):* The order is either immediately filled or, if that is not possible, canceled. If only a partial fill is possible the order is still filled.

*FillOrKill (4):* The order is either immediately filled for the entire amount or, if that is not possible, canceled.

*GoodTillDate (6):* The order is in effect until the specified expiry date.

*AtTheClose (7):* The order is either filled at the close of the market or, if that is not possible, canceled.

TrdRegTimestamp	769	No	UTCTimeStamp	The time when the transaction was registered by the bank.
-----------------	-----	----	--------------	---

### 3.6.3 Margin Calls

FIX MsgType for a margin call message is U2.

The margin call notification is created when an event related to a margin call occurs.

Below is a sample message for a margin call. The SOH field delimiter character is displayed as '.'.

```
8=FIX.4.4 •9=214 •35=U2 •34=580 •49=SAXOBANK •52=20131021-11:29:20.944 •56=XXXX
•1=S1429USD •109=258142 •20000=USD •20005=20131021-11:29:20 •20016=4 •20017=0 •20071=-
3071311.50 •20072=1770658.50 •20073=4841970.00 •20074=3071311.50 •20075=0.37 •10=143
```

#### 3.6.3.1 Margin call fields

Name	Tag	Required	Type	Description
Account	1	Yes	String	Identifies the account involved in the transaction. The default account ID for the client.
ClientID	109	Yes	String	Identifies the client involved in the transaction. The client identifier is a non-negative integer (31 bits).
SaxoBaseCurrency	20000	Yes	Currency	The default currency for the client.
SaxoCreated	20005	Yes	UTCTimeStamp	The time when this notification was created.
SaxoMarginCallAction	20016	Yes	Char	Describes the action associated with this margin call. <i>MarginCall (1):</i> The client is requested to restore the necessary margin required. <i>StopOut (2):</i> The client has been forcefully stopped out to restore the necessary margin required. <i>Reinstate (3):</i> The system has restored the necessary margin required. <i>LevelDrop (4):</i> Margin level drooped, The client has restored the necessary margin level required, according to the margin call profile defined. This margin call action will appear if the client margin level dropped under the defined margin levels percentage as set in the margin call profile.
SaxoMarginCallLevel	20017	Yes	Int	The percentage of margin used. Please Note, in a LevelDropped margin call scenario, where the margin level is 0, this field <b>will be</b> included in the FIX message. Margin call level equal to 0 can accrue in 2 different scenarios. Scenario 1: If the margin level has dropped under the last percentage defined in the margin call profile, then the margin call level will be equal to 0, in this case the <i>MarginCallLevel</i> field will not be included in the Margin Call notification. Scenario 2: In case of account funding where the account balance was negative, margin call level will be equal to 0, in this case the <i>MarginCallLevel</i> field will not be included in the Margin Call notification.
MarginDeficit	20071	Yes	Float	The value in this field represents the available amount that the client has for trading on margin, the value displayed in this field reflecting Saxo group point of view, hence the value will always be positive during a margin call.

MarginRequired	20072	Yes	Float	The value in this field represents the total margin amount required to support the various exposures, created by the client's prevailing holdings in any margin tradable product.
NetEquityForMargin	20073	Yes	Float	Total net equity available for margin trading.
NetFreeBalance	20074	Yes	Float	The value in this field represents the available amount that the client has for trading on margin, the value displayed in this field is reflecting the client point of view. Pending on the margin call level, this field could have a
UseOfEquityForMargin	20075	Yes	Float	The value in this field represents the percentage of the net exposure in relation to the account value.

### 3.6.4 Funding

FIX MsgType for a funding message is U1.

The funding notification is created when money is added or deducted from an account.

Below is a sample message for a funding deposit. The SOH field delimiter character is displayed as '\*'.

```
8=FIX.4.4*9=138*35=U1*1=77820*14=14000*109=3179470*769=20100818-08:14:14*20005=20100818-08:14:14*20006=NOK*20012=0*20013=D*20023=276302329*20039=20100819*10=110*
```

#### 3.6.4.1 Funding fields

Name	Tag	Required	Type	Description
Account	1	Yes	String	Identifies the account involved in the transaction. The default account ID for the client.
ClientID	109	Yes	String	Identifies the client involved in the transaction. The client identifier is a non-negative integer (31 bits).
CumQty	14	No	Qty	Amount (e.g. number of securities) filled.
SaxoConversionRate	20004	No	Float	The conversion rate used at the time of the transaction.
SaxoCreated	20005	Yes	UTCTimeStamp	The time when this notification was created.
SaxoCurrencyCode	20006	No	Currency	The currency of the transaction.
SaxoFundingEvent	20012	Yes	Char	Describes the funding event. <i>New (0)</i> : The funding position is created. <i>Updated (1)</i> : The funding position has been updated. <i>Deleted (2)</i> : The funding position is deleted.
SaxoFundingType	20013	No	Char	Describes the type of the funding. <i>Deposit (D)</i> : Money is added to the account. <i>Withdrawal (W)</i> : Money is deducted from the account.
SaxoPositionID	20023	Yes	Int	Identifies the position. The position identifier is a non-negative integer (31 bits).
SaxoValueDate	20039	No	UTCDateOnly	The date when the final transfer of the product of the transaction takes place.
TrdRegTimestamp	769	No	UTCTimeStamp	The time when the transaction was registered by the bank.

## 4 Error Handling

Saxo Group monitors all operations, and has failover procedures and backup for all systems. Internal failures are therefore handled immediately, and normally not noticeable to the outside. We therefore urge you to troubleshoot locally before contacting us, as we are already monitoring all operations to keep downtime and outages to a minimum.

### 4.1 E-mail Troubleshooting

Getting notifications via email is a one way communication. The immediate sources of error will therefore typically be missing/delayed emails or content related problems. Missing/Delayed emails

If the flow of emails is stopping or there is reason to be suspicious about a sudden lack/delay of incoming messages, the problem can be either related to the sending or receiving mail server. The mail servers at Saxo Group are designed for extremely high volumes, and are therefore not to be considered as the primary source. The receiving mail server (@client) is usually more exposed to varying loads, as well as DOS attacks from the outside. The following procedure is suggested to troubleshoot missing/delayed emails:

1. Check that the receiving mail server is generally able to receive mail from outside domains.
2. Check that other external parties can send mails to the address in use.
3. Check that the receiving mail server is not rejecting or blacklisting the Saxo Group servers.
4. Check that the receiving mail server is not filtering or relocating Saxo Group mails to spam or anti-virus mail boxes
5. Contact Saxo Group for further investigation

#### 4.1.1 Missing/Invalid content

Mails are sent in plain text, which is highly compatible with any mail reader. However, sometimes certain characters may look foreign, and sometimes content may be missing. This is usually caused by using or defaulting to invalid character encodings in the mail reading client. Saxo Group messages are encoded with UTF8, which is a widely used international standard. It's also possible to receive notifications as attachments, which also requires the mail server and mail clients not to discard or filter out attachments. Saxo Group sends the notifications as files with either .xml or .csv extensions, which should be enabled on the mail server and clients. The following procedure is suggested to troubleshoot missing/invalid content:

1. Check that your encoding is set to UTF8
2. Try forward the incoming mail to another mail client (different brand), to see if it renders correctly.
3. Check that you can send and receive attachments of type .xml and .csv, by composing a mail with one of these, and send it to an internal address for verification.
4. Contact Saxo Group for further investigation

## 4.2 SWIFT Troubleshooting

SWIFT messages are transferred by message queuing servers, which handle all the aspects of message delivery, consistency and reliability between parties. The usual source of failures in SWIFT communication is broken communication and invalid message formatting.

### 4.2.1 Broken communication

The local message queuing server servicing SWIFT messages may be unable to communicate with the outside servers. Communication should be verified by a SWIFT technician.

### 4.2.2 Invalid Message Formatting

All SWIFT messages are encoded in UTF8, and formatted to comply with the SWIFT message standards. However, in some circumstances the SWIFT messages might be unreadable or not formatted exactly as the standards specifies. If the message format is suspect to be invalid, please make a copy of the message and send it to Saxo Group for further analysis.

## 4.3 File Transfer Troubleshooting

File transfers are normally only troubled by broken communications, or misconfiguration of the account login/password. Under normal operation file transfers are very reliable due to the built-in error correction in the underlying TCP protocol. In Saxo Group, file transfers are handled by dedicated servers designed for high volume.

### 4.3.1 Broken communication

If the communication is suspect to be broken, please verify the following:

1. Check that your file transfer client/server is able to communicate with other parties.
2. Check that you can connect (or ping) the Saxo Group server
3. Contact Saxo Group for further investigation

### 4.3.2 Invalid Credentials

Depending on how the file transferring is setup, the account login details have to be authenticated either by the Saxo Group or the customer server. One of the ends will supply the login/password and the other end will verify (authenticate) these.

1. If the customer server is authenticating the login request, make sure that the credentials supplied to Saxo Group are still valid on the customer server, by trying login in locally.
2. If the Saxo Group server is authenticating the login, make sure that you are using the same credentials as originally provided by Saxo Group.
3. Contact Saxo Group for further investigation

## 4.4 FIX Troubleshooting

The source of error for FIX communication, are typically invalid credentials, network outages or sequence numbers becoming out of sync. The FIX protocol is resistant to short network outages, as it will continuously keep trying to connect to the other party. As this is a built in feature by design, it may normally go unnoticed for a short while without actually interrupting and manifest itself as an error. However, a substantial network outage will of course affect the flow of transmission and need attention. Saxo Group is running the FIX server, and FIX clients (@customers) are connecting to this via VPN tunnels.

### 4.4.1 Broken Communication

If the FIX communication is broken, the message stream will stop. The following procedure is suggested to troubleshoot network problems:

1. Check that the FIX client can access the local network
2. Check that the VPN connection is up and that the FIX client can ping the Saxo Group FIX server
3. Contact Saxo Group for further investigation

### 4.4.2 Invalid Credentials

The Saxo Group FIX server requires four pieces of information in order to establish a connection. These are the SenderCompId, TargetCompId, Login and Password. If these are not matching the values supplied by Saxo Group, the login process will fail.

If the client is able to connect, but unable to login, please contact Saxo Group in order to verify that the credentials are correct.

### 4.4.3 Sequence Numbers out of sync

If sequence numbers goes out of sync, it is usually a sign of either the FIX server or client has lost its state. Both ends keep track of the incoming and outgoing sequence numbers, and are storing these on the local machine to be available whenever they reconnect again. If one of the ends loses the numbers, they will mismatch upon connection and the sequence numbers will become out of sync.

In order to re-sync, Saxo Group is able to either manually adjust the sequence numbers, or reset the sequence numbers so they will start over from 1 again. If a reset is executed, the client should also reset their side.

When contacting Saxo Group, please state the SenderCompId and whether you will need a reset or an adjustment. If you need to make an adjustment, you will also have to inform about what values we should apply to the incoming and outgoing sequence numbers.

NOTE: Whenever sequence numbers are adjusted or reset, there is a potential of losing older FIX messages.

## 5 Acceptance Test

The goal of the acceptance test is to verify and ensure the complete functionality of a client subscribed to TENS. The test is divided into two stages – a simulation stage, and a live stage.

The purpose of the simulation stage is to ensure that all of the functionality is working. This includes any customer post processing and system integration of the TENS notifications. The simulation stage, gives the customer the ability to trade and test functionality for an extended period of time, without risk.

When the end-to-end functionality is tested complete on the simulation system, it safe to go to the live stage. As it is currently not possible to test funding on the simulation system, we suggest that the customer prepare a small funding test to be executed on the live system.

### 5.1 Functional Testing

The table below describes how to generate the four types of notifications delivered by TENS. Please note that initial testing and tuning should be performed on the simulation system, as this is free of risks and costs.

Message	Simulation	Live
Positions	TBD	TBD
Orders	TBD	TBD
Margin Calls	TBD	N/A
Funding	N/A	TBD

Will be updated.

### 5.2 Connectivity Testing

For every delivery method, it's important to ensure that the communication is failsafe and able to withstand network and system outages. If possible we suggest exercising the chosen delivery system by making trades on the platform, while the local delivery system is made unavailable. Make the delivery system available again, and confirm that the test did not result in any loss of notifications.



## 6 Maintenance

Maintenance and further development of the code residing on Saxo Group's servers will be conducted by Saxo Group.

### 6.1 Schedule

Updates are run at the same time as other Saxo systems. It is usually done on Saturdays when markets are closed.

### 6.2 Notifications

Important updates are notified monthly.

### 6.3 Support

#### 6.3.1 Onboarding Support

Saxo will provide resources during the onboarding period to support the Institutional Client with questions during the implementation and configuration of the service. It is expected that the Institutional Client has development resources.

#### 6.3.2 Maintenance Support

Saxo Group's Service Center supports the solution 24/6½ and can be reached on +39 77 40 01 and [servicecenter@saxobank.com](mailto:servicecenter@saxobank.com).

## Appendix 1 - XML schema

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="unqualified" elementFormDefault="qualified">
  <xs:simpleType name="BuySell">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Buy" />
      <xs:enumeration value="Sell" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="CallPut">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Call" />
      <xs:enumeration value="Put" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ContractType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Bond" />
      <xs:enumeration value="Cfd" />
      <xs:enumeration value="CfdOnFuture" />
      <xs:enumeration value="ContractOption" />
      <xs:enumeration value="FutureContract" />
      <xs:enumeration value="FxBinaryOption" />
      <xs:enumeration value="FxKnockInOption" />
      <xs:enumeration value="FxKnockOutOption" />
      <xs:enumeration value="FxNoTouchOption" />
      <xs:enumeration value="FxOneTouchOption" />
      <xs:enumeration value="FxSpot" />
      <xs:enumeration value="FxVanillaOption" />
      <xs:enumeration value="ManagedFund" />
      <xs:enumeration value="Share" />
      <xs:enumeration value="ShareOption" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="Duration">
    <xs:restriction base="xs:string">
      <xs:enumeration value="AtTheClose" />
      <xs:enumeration value="AtTheOpening" />
      <xs:enumeration value="DayOrder" />
      <xs:enumeration value="FillOrKill" />
      <xs:enumeration value="GoodTillCancel" />
      <xs:enumeration value="GoodTillDate" />
      <xs:enumeration value="ImmediateOrCancel" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ExecutionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Changed" />
      <xs:enumeration value="Deleted" />
      <xs:enumeration value="New" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="FundingEvent">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Deleted" />
      <xs:enumeration value="New" />
      <xs:enumeration value="Updated" />
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="FundingType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Deposit" />
    </xs:restriction>
  </xs:simpleType>

```

```

        <xs:enumeration value="Withdrawal" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MarginCallAction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="MarginCall" />
        <xs:enumeration value="Reinstate" />
        <xs:enumeration value="StopOut" />
        <xs:enumeration value="LevelDrop" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="OrderRelation">
    <xs:restriction base="xs:string">
        <xs:enumeration value="IfDoneMaster" />
        <xs:enumeration value="IfDoneSlave" />
        <xs:enumeration value="IfDoneSlaveOco" />
        <xs:enumeration value="Oco" />
        <xs:enumeration value="StandAlone" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="OrderType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Algorithmic" />
        <xs:enumeration value="CallLimit" />
        <xs:enumeration value="CallStop" />
        <xs:enumeration value="GuaranteedStop" />
        <xs:enumeration value="Limit" />
        <xs:enumeration value="Market" />
        <xs:enumeration value="MarketExpiry" />
        <xs:enumeration value="MarketStopOut" />
        <xs:enumeration value="StopIfBid" />
        <xs:enumeration value="StopIfOffered" />
        <xs:enumeration value="StopIfTraded" />
        <xs:enumeration value="StopLimit" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PositionEvent">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Deleted" />
        <xs:enumeration value="MarginStopOut" />
        <xs:enumeration value="New" />
        <xs:enumeration value="OptionExercised" />
        <xs:enumeration value="OptionExpired" />
        <xs:enumeration value="Updated" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="PriceType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Amount" />
        <xs:enumeration value="Ccy1Percentage" />
        <xs:enumeration value="Ccy1Pips" />
        <xs:enumeration value="Ccy2Percentage" />
        <xs:enumeration value="Ccy2Pips" />
        <xs:enumeration value="ThirdCurrency" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SystemOrigin">
    <xs:restriction base="xs:string">
        <xs:enumeration value="B2B" />
        <xs:enumeration value="ClientStation" />
        <xs:enumeration value="MobileTrader" />
        <xs:enumeration value="Other" />
        <xs:enumeration value="WebTrader" />
    </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType name="ToOpenClose">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ToClose" />
    <xs:enumeration value="ToOpen" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="Funding">
  <xs:complexType>
    <xs:all>
      <xs:element name="AccountId" type="xs:string" />
      <xs:element name="Amount" type="xs:decimal" />
      <xs:element name="ClientId" type="xs:int" />
      <xs:element name="ConversionRate" type="xs:decimal" minOccurs="0"/>
      <xs:element name="Created" type="xs:dateTime" />
      <xs:element name="CurrencyCode" type="xs:string" />
      <xs:element name="FundingEvent" type="FundingEvent" />
      <xs:element name="FundingType" type="FundingType" />
      <xs:element name="PositionId" type="xs:int" />
      <xs:element name="RegistrationTime" type="xs:dateTime" />
      <xs:element name="ValueDate" type="xs:date" />
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="MarginCall">
  <xs:complexType>
    <xs:all>
      <xs:element name="BaseCurrency" type="xs:string" />
      <xs:element name="ClientId" type="xs:int" />
      <xs:element name="Created" type="xs:dateTime" />
      <xs:element name="DefaultAccountId" type="xs:string" />
      <xs:element name="MarginCallAction" type="MarginCallAction" />
      <xs:element name="MarginCallLevel" type="xs:int" minOccurs="0"/>
      <xs:element minOccurs="0" maxOccurs="1" name="MarginDeficit" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="MarginRequired" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="NetEquityForMargin" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="NetFreeBalance" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="UseOfEquityForMargin" type="xs:string" />
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="Order">
  <xs:complexType>
    <xs:all>
      <xs:element name="AccountId" type="xs:string" />
      <xs:element name="Amount" type="xs:decimal" minOccurs="0" />
      <xs:element name="BuySell" type="BuySell" minOccurs="0" />
      <xs:element name="CallPut" type="CallPut" minOccurs="0" />
      <xs:element name="ClientId" type="xs:int" />
      <xs:element name="ClientOrderId" type="xs:string" minOccurs="0" />
      <xs:element name="ContractType" type="ContractType" minOccurs="0" />
      <xs:element name="Created" type="xs:dateTime" />
      <xs:element name="CurrencyCode" type="xs:string" minOccurs="0" />
      <xs:element name="Duration" type="Duration" minOccurs="0" />
      <xs:element name="ExchangeId" type="xs:string" minOccurs="0" />
      <xs:element name="ExecutionType" type="ExecutionType" />
      <xs:element name="ExpiryDate" type="xs:date" minOccurs="0" />
      <xs:element name="FilledAmount" type="xs:decimal" minOccurs="0" />
      <xs:element name="Instrument" type="xs:string" />
      <xs:element name="IsinCode" type="xs:string" minOccurs="0" />
      <xs:element name="OrderId" type="xs:int" />
    </xs:all>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="OrderRelation" type="OrderRelation" minOccurs="0" />
<xs:element name="OrderType" type="OrderType" minOccurs="0" />
<xs:element name="OriginalOrderId" type="xs:int" minOccurs="0" />
<xs:element name="OriginatingPositionId" type="xs:int" minOccurs="0" />
<xs:element name="Price" type="xs:decimal" minOccurs="0" />
<xs:element name="RegistrationTime" type="xs:dateTime" minOccurs="0" />
<xs:element name="RelatedOrderId" type="xs:int" minOccurs="0" />
<xs:element name="RelatedPositionId" type="xs:int" minOccurs="0" />
<xs:element name="RelatedSecondOrderId" type="xs:int" minOccurs="0" />
<xs:element name="Symbol" type="xs:string" minOccurs="0" />
<xs:element name="SystemOrigin" type="SystemOrigin" minOccurs="0" />
<xs:element name="ToOpenClose" type="ToOpenClose" minOccurs="0" />
<xs:element name="TrailingStopPriceChangeThreshold" type="xs:decimal"
minOccurs="0" />
<xs:element name="TrailingStopPriceDifference" type="xs:decimal" minOccurs="0"
/>
<xs:element name="UserId" type="xs:int" minOccurs="0" />
</xs:all>
</xs:complexType>
</xs:element>
<xs:element name="Position">
<xs:complexType>
<xs:all>
<xs:element name="AccountId" type="xs:string" />
<xs:element name="Amount" type="xs:decimal" minOccurs="0" />
<xs:element name="BuySell" type="BuySell" minOccurs="0" />
<xs:element name="CallPut" type="CallPut" minOccurs="0" />
<xs:element name="ClientId" type="xs:int" />
<xs:element name="Commission" type="xs:decimal" minOccurs="0" />
<xs:element name="ContractType" type="ContractType" minOccurs="0" />
<xs:element name="ConversionRate" type="xs:decimal" minOccurs="0" />
<xs:element name="Created" type="xs:dateTime" />
<xs:element name="CurrencyCode" type="xs:string" minOccurs="0" />
<xs:element name="Delta" type="xs:decimal" minOccurs="0" />
<xs:element name="ExchangeFee" type="xs:decimal" minOccurs="0" />
<xs:element name="ExchangeId" type="xs:string" minOccurs="0" />
<xs:element name="ExecutionTime" type="xs:dateTime" minOccurs="0" />
<xs:element name="ExpiryCut" type="xs:string" minOccurs="0" />
<xs:element name="ExpiryDate" type="xs:date" minOccurs="0" />
<xs:element name="Instrument" type="xs:string" minOccurs="0" />
<xs:element name="IsinCode" type="xs:string" minOccurs="0" />
<xs:element name="LowerBarrier" type="xs:decimal" minOccurs="0" />
<xs:element name="OpenPrice" type="xs:decimal" minOccurs="0" />
<xs:element name="OpenSpot" type="xs:decimal" minOccurs="0" />
<xs:element name="OpenSwap" type="xs:decimal" minOccurs="0" />
<xs:element name="OriginalAccountId" type="xs:string" minOccurs="0" />
<xs:element name="OriginatingPositionId" type="xs:int" minOccurs="0" />
<xs:element name="PositionEvent" type="PositionEvent" />
<xs:element name="PositionId" type="xs:int" />
<xs:element name="PremiumAmount" type="xs:decimal" minOccurs="0" />
<xs:element name="PremiumAmountCcy2" type="xs:decimal" minOccurs="0" />
<xs:element name="PriceType" type="PriceType" minOccurs="0" />
<xs:element name="RegistrationTime" type="xs:dateTime" minOccurs="0" />
<xs:element name="RelatedOrderCount" type="xs:int" minOccurs="0" />
<xs:element name="RelatedPositionId" type="xs:int" minOccurs="0" />
<xs:element name="SourceOrderId" type="xs:int" minOccurs="0" />
<xs:element name="SpotDate" type="xs:date" minOccurs="0" />
<xs:element name="StampDuty" type="xs:decimal" minOccurs="0" />
<xs:element name="StrikePrice" type="xs:decimal" minOccurs="0" />
<xs:element name="Symbol" type="xs:string" minOccurs="0" />
<xs:element name="SystemOrigin" type="SystemOrigin" minOccurs="0" />
<xs:element name="ToOpenClose" type="ToOpenClose" minOccurs="0" />
<xs:element name="UpperBarrier" type="xs:decimal" minOccurs="0" />
<xs:element name="ValueDate" type="xs:date" minOccurs="0" />
<xs:element name="Volatility" type="xs:decimal" minOccurs="0" />

```

```
</xs:all>  
</xs:complexType>  
</xs:element>  
</xs:schema>
```

## Appendix 2 - Option Exercise

This is an example of how the notification flow is when an option is exercised. The following scenarios are shown;

1. A position is created and
2. The option is exercised.

### Create position

A position is placed: EURGBP Long Call position at strike .8257 with expiry 17th April. The position has Position ID: 68754792. One notification received.

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Buy</BuySell>
  <CallPut>Call</CallPut>
  <ClientId>2147123712</ClientId>
  <ContractType>FxVanillaOption</ContractType>
  <ConversionRate>1.21129</ConversionRate>
  <Created>2012-04-17T04:10:06.16</Created>
  <CurrencyCode>GBP</CurrencyCode>
  <Delta>0.479085721</Delta>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:10:00.45</ExecutionTime>
  <ExpiryCut>NY</ExpiryCut>
  <ExpiryDate>2012-04-17</ExpiryDate>
  <Instrument>EURGBP</Instrument>
  <OpenPrice>0.0016</OpenPrice>
  <OpenSpot>0.0016</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68754792</PositionId>
  <PremiumAmount>160</PremiumAmount>
  <PremiumAmountCcy2>160</PremiumAmountCcy2>
  <PriceType>Ccy2Pips</PriceType>
  <RegistrationTime>2012-04-17T04:10:06.127</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <StrikePrice>0.8257</StrikePrice>
  <Symbol>EUR/GBP</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
  <Volatility>0.087177194</Volatility>
</Position>
```

## Exercise Option

The above mentioned position is then exercised. Four (4) notifications received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>1.21121</ConversionRate>
  <Created>2012-04-17T04:14:05.913</Created>
  <CurrencyCode>GBP</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:14:05.887</ExecutionTime>
  <Instrument>EURGBP</Instrument>
  <OpenPrice>0.8257</OpenPrice>
  <OpenSpot>0.8257</OpenSpot>
  <OriginatingPositionId>68754792</OriginatingPositionId>
  <PositionEvent>New</PositionEvent>
  <PositionId>68754794</PositionId>
  <RegistrationTime>2012-04-17T04:14:05.91</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/GBP</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```



## Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Buy</BuySell>
  <CallPut>Call</CallPut>
  <ClientId>2147123712</ClientId>
  <ContractType>FxVanillaOption</ContractType>
  <ConversionRate>1.21129</ConversionRate>
  <Created>2012-04-17T04:14:06.073</Created>
  <CurrencyCode>GBP</CurrencyCode>
  <Delta>0.479085721</Delta>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:10:00.45</ExecutionTime>
  <ExpiryCut>NY</ExpiryCut>
  <ExpiryDate>2012-04-17</ExpiryDate>
  <Instrument>EURGBP</Instrument>
  <OpenPrice>0.0016</OpenPrice>
  <OpenSpot>0.0016</OpenSpot>
  <PositionEvent>OptionExercised</PositionEvent>
  <PositionId>68754792</PositionId>
  <PremiumAmount>160</PremiumAmount>
  <PremiumAmountCcy2>160</PremiumAmountCcy2>
  <PriceType>Ccy2Pips</PriceType>
  <RegistrationTime>2012-04-17T04:10:06.127</RegistrationTime>
  <SourceOrderId>68754794</SourceOrderId>
  <SpotDate>2012-04-19</SpotDate>
  <StrikePrice>0.8257</StrikePrice>
  <Symbol>EUR/GBP</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
  <Volatility>0.087177194</Volatility>
</Position>
```

## Notification 3

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>1.21121</ConversionRate>
  <Created>2012-04-17T04:14:06.103</Created>
  <CurrencyCode>GBP</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:14:05.887</ExecutionTime>
  <Instrument>EURGBP</Instrument>
  <OpenPrice>0.82562</OpenPrice>
  <OpenSpot>0.82562</OpenSpot>
  <OriginatingPositionId>68754792</OriginatingPositionId>
  <PositionEvent>New</PositionEvent>
  <PositionId>68754796</PositionId>
  <RegistrationTime>2012-04-17T04:14:06.1</RegistrationTime>
  <RelatedPositionId>68754794</RelatedPositionId>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/GBP</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```



## Notification 4

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>1.21121</ConversionRate>
  <Created>2012-04-17T04:14:06.103</Created>
  <CurrencyCode>GBP</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:14:05.887</ExecutionTime>
  <Instrument>EURGBP</Instrument>
  <OpenPrice>0.8257</OpenPrice>
  <OpenSpot>0.8257</OpenSpot>
  <OriginatingPositionId>68754792</OriginatingPositionId>
  <PositionEvent>Updated</PositionEvent>
  <PositionId>68754794</PositionId>
  <RegistrationTime>2012-04-17T04:14:05.91</RegistrationTime>
  <RelatedPositionId>68754796</RelatedPositionId>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/GBP</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Appendix 3 - Option Expire

This is an example of how the notification flow is when an option is exercised. The following scenarios are shown;

1. A position is created and
2. The option is expired.

### Create Position

A position is placed: CHFJPY short Put at strike 87.77, expiry 17th April. The position has Position ID: 68754798. One notification received.

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Sell</BuySell>
  <CallPut>Put</CallPut>
  <ClientId>2147123712</ClientId>
  <ContractType>FxVanillaOption</ContractType>
  <ConversionRate>0.009477</ConversionRate>
  <Created>2012-04-17T04:24:09.637</Created>
  <CurrencyCode>JPY</CurrencyCode>
  <Delta>-0.483923093</Delta>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:24:09.283</ExecutionTime>
  <ExpiryCut>NY</ExpiryCut>
  <ExpiryDate>2012-04-17</ExpiryDate>
  <Instrument>CHFJPY</Instrument>
  <OpenPrice>0.04</OpenPrice>
  <OpenSpot>0.04</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68754798</PositionId>
  <PremiumAmount>4000</PremiumAmount>
  <PremiumAmountCcy2>4000</PremiumAmountCcy2>
  <PriceType>Ccy2Pips</PriceType>
  <RegistrationTime>2012-04-17T04:24:09.633</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <StrikePrice>87.77</StrikePrice>
  <Symbol>CHF/JPY</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
  <Volatility>0.155</Volatility>
</Position>
```

## Expire Option

Position created above is then expired. One notification received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Sell</BuySell>
  <CallPut>Put</CallPut>
  <ClientId>2147123712</ClientId>
  <ContractType>FxVanillaOption</ContractType>
  <ConversionRate>0.009477</ConversionRate>
  <Created>2012-04-17T04:29:43.363</Created>
  <CurrencyCode>JPY</CurrencyCode>
  <Delta>-0.483923093</Delta>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:24:09.283</ExecutionTime>
  <ExpiryCut>NY</ExpiryCut>
  <ExpiryDate>2012-04-17</ExpiryDate>
  <Instrument>CHFJPY</Instrument>
  <OpenPrice>0.04</OpenPrice>
  <OpenSpot>0.04</OpenSpot>
  <PositionEvent>OptionExpired</PositionEvent>
  <PositionId>68754798</PositionId>
  <PremiumAmount>4000</PremiumAmount>
  <PremiumAmountCcy2>4000</PremiumAmountCcy2>
  <PriceType>Ccy2Pips</PriceType>
  <RegistrationTime>2012-04-17T04:24:09.633</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <StrikePrice>87.77</StrikePrice>
  <Symbol>CHF/JPY</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
  <Volatility>0.155</Volatility>
</Position>
```

## Appendix 4 – Allocation

Will be updated during Q1 2017.

## Appendix 5 - Position netting

This is an example of how the notification flow is when a position is netted by Saxo Group's Back Office overnight. The following scenarios are shown;

1. Create two opposing positions
2. Leave positions open during a week-night

### Create two opposing positions

One buy and one sell position are created. Two notifications received.

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>57.4</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.1323415</ConversionRate>
  <Created>2012-04-17T09:06:10.053</Created>
  <CurrencyCode>NOK</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T09:06:10.003</ExecutionTime>
  <Instrument>EURNOK</Instrument>
  <OpenPrice>7.5578</OpenPrice>
  <OpenSpot>7.5578</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68578373</PositionId>
  <RegistrationTime>2012-04-17T09:06:10.053</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/NOK</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>10000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>57.4</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.132337</ConversionRate>
  <Created>2012-04-17T09:06:22.36</Created>
  <CurrencyCode>NOK</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T09:06:22.33</ExecutionTime>
  <Instrument>EURNOK</Instrument>
  <OpenPrice>7.5549</OpenPrice>
  <OpenSpot>7.5549</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68790802</PositionId>
  <RegistrationTime>2012-04-17T09:06:22.36</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/NOK</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```



## Leave positions open

The positions are left open overnight. The following morning two notifications are received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>57.4</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.132337</ConversionRate>
  <Created>2012-04-17T22:23:59.027</Created>
  <CurrencyCode>NOK</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T09:06:22.33</ExecutionTime>
  <Instrument>EURNOK</Instrument>
  <OpenPrice>7.55394</OpenPrice>
  <OpenSpot>7.5549</OpenSpot>
  <OpenSwap>-0.00096</OpenSwap>
  <PositionEvent>Updated</PositionEvent>
  <PositionId>68790802</PositionId>
  <RegistrationTime>2012-04-17T22:23:59.05</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/NOK</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-20</ValueDate>
</Position>
```

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <Created>2012-04-17T22:23:59.427</Created>
  <PositionEvent>Deleted</PositionEvent>
  <PositionId>68578373</PositionId>
</Position>
```

## Appendix 6 - Cash Settlement

Will be updated during Q1 2017.

## Appendix 7 - Corporate Action

Will be updated during Q1 2017.

## Appendix 8 - Saxo Initiated trade

This is an example of how the notification flow is when a Saxo Group employee initiates a trade on a customer's account. (EXAMPLE?) The following scenarios are shown:

- 1) Create position

### Create position

A position is created: EURGBP Long Call position at strike .8257 with expiry 17th April. The position has Position ID: 68754792. One notification received.

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>100000</Amount>
  <BuySell>Buy</BuySell>
  <CallPut>Call</CallPut>
  <ClientId>2147123712</ClientId>
  <ContractType>FxVanillaOption</ContractType>
  <ConversionRate>1.21129</ConversionRate>
  <Created>2012-04-17T04:10:06.16</Created>
  <CurrencyCode>GBP</CurrencyCode>
  <Delta>0.479085721</Delta>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:10:00.45</ExecutionTime>
  <ExpiryCut>NY</ExpiryCut>
  <ExpiryDate>2012-04-17</ExpiryDate>
  <Instrument>EURGBP</Instrument>
  <OpenPrice>0.0016</OpenPrice>
  <OpenSpot>0.0016</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68754792</PositionId>
  <PremiumAmount>160</PremiumAmount>
  <PremiumAmountCcy2>160</PremiumAmountCcy2>
  <PriceType>Ccy2Pips</PriceType>
  <RegistrationTime>2012-04-17T04:10:06.127</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <StrikePrice>0.8257</StrikePrice>
  <Symbol>EUR/GBP</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
  <Volatility>0.087177194</Volatility>
</Position>
```

## Appendix 9 - Customer trade

This is an example of how the notification flow is when a customer creates a position in any of the available trading platforms. (This example is based on Desktop Trader (Client Station)). The following scenarios are shown:

- 1) Create Position
- 2) Place stop Order

### Create position

Create position: Long spot EURCHF position @1.20211. One notification received.

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>9.1</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.831968</ConversionRate>
  <Created>2012-04-17T04:33:06.447</Created>
  <CurrencyCode>CHF</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:33:06.417</ExecutionTime>
  <Instrument>EURCHF</Instrument>
  <OpenPrice>1.20211</OpenPrice>
  <OpenSpot>1.20211</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68578337</PositionId>
  <RegistrationTime>2012-04-17T04:33:06.44</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/CHF</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Place Stop order

Then a stop order is placed with regards to above position. Two notifications received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>9.1</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.831968</ConversionRate>
  <Created>2012-04-17T04:43:45.447</Created>
  <CurrencyCode>CHF</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:33:06.417</ExecutionTime>
  <Instrument>EURCHF</Instrument>
  <OpenPrice>1.20211</OpenPrice>
  <OpenSpot>1.20211</OpenSpot>
  <PositionEvent>Updated</PositionEvent>
  <PositionId>68578337</PositionId>
  <RegistrationTime>2012-04-17T04:33:06.44</RegistrationTime>
  <RelatedOrderCount>1</RelatedOrderCount>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/CHF</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T04:43:45.447</Created>
  <CurrencyCode>CHF</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>EURCHF</Instrument>
  <OrderId>44309531</OrderId>
  <OrderType>StopIfOffered</OrderType>
  <Price>1.1961</Price>
  <RegistrationTime>2012-04-17T04:43:45.423</RegistrationTime>
  <RelatedPositionId>68578337</RelatedPositionId>
  <Symbol>EUR/CHF</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

## Appendix 10 - Margin Call

This is an example of how the notification flow is when a customer's account hits three predefined margin call levels. The following scenarios are shown;

- 1) Margin utilization exceeds 100%
- 2) Margin utilization exceeds 115%
- 3) Margin utilization exceeds 125% and the customer is stopped out.

### Margin utilization exceeds 100%

When the customer's account hit the margin call at 100% one e-mail and one notification were received.

#### Margin call e-mail 1

```
Margin Call for client number 197373785. Passed 100.0%%

Wednesday, April 18, 2012 8:25:44 AM GMT
Net Equity for Margin:      270210.82EUR
Margin Required:           298506.73EUR
Use of Equity for Margin:  110.1%
```

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<MarginCall>
  <BaseCurrency>USD</BaseCurrency>
  <ClientId>1973785</ClientId>
  <Created>2012-04-18T08:25:44.817</Created>
  <DefaultAccountId>19373785</DefaultAccountId>
  <MarginCallAction>MarginCall</MarginCallAction>
  <MarginCallLevel>110</MarginCallLevel>
  <MarginDeficit>28295.91</MarginDeficit>
  <MarginRequired>298506.73</MarginRequired>
  <NetEquityForMargin>270210.82</NetEquityForMargin>
  <NetFreeBalance>-28295.91</NetFreeBalance>
  <UseOfEquityForMargin>1.10</UseOfEquityForMargin>
</MarginCall>
```



## Margin utilization exceeds 115%

When the customer's account hit the margin call at 115% one e-mail and one notification were received.

### Margin call e-mail 2

Margin Call for client number 19373785. Passed 115.0%%

Wednesday, April 18, 2012 9:18:47 AM GMT

Net Equity for Margin: 945.97EUR

Margin Required: 6011.24EUR

Use of Equity for Margin: 600.35%

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<MarginCall>
  <BaseCurrency>USD</BaseCurrency>
  <ClientId>19373785</ClientId>
  <Created>2012-04-18T09:18:47.1</Created>
  <DefaultAccountId>19373785</DefaultAccountId>
  <MarginCallAction>MarginCall</MarginCallAction>
  <MarginCallLevel>115</MarginCallLevel>
  <MarginDeficit>5065.27</MarginDeficit>
  <MarginRequired>6011.24</MarginRequired>
  <NetEquityForMargin>945.97</NetEquityForMargin>
  <NetFreeBalance>-5065.27</NetFreeBalance>
  <UseOfEquityForMargin>6.35</UseOfEquityForMargin>
</MarginCall>
```

## Margin utilization exceeds 125% - stop out

When the customer's account hit margin utilization of 125%, all outstanding positions were squared. One notification received.

### Notification 3

```
<?xml version="1.0" encoding="utf-8"?>
<MarginCall>
  <BaseCurrency>USD</BaseCurrency>
  <ClientId>19373785</ClientId>
  <Created>2013-11-18T10:12:26.77</Created>
  <DefaultAccountId>19373785</DefaultAccountId>
  <MarginCallAction>StopOut</MarginCallAction>
  <MarginCallLevel>125</MarginCallLevel>
  <MarginDeficit>206.72</MarginDeficit>
  <MarginRequired>810.48</MarginRequired>
  <NetEquityForMargin>603.76</NetEquityForMargin>
  <NetFreeBalance>-206.72</NetFreeBalance>
  <UseOfEquityForMargin>1.34</UseOfEquityForMargin>
</MarginCall>
```

## Appendix 11 - Customer Order

This is an example of how the notification flow is when placing orders. The following scenarios are shown:

- 1) Order placed and modified to market order
  - a. Place order
  - b. Modify to market order
- 2) Order placed, modified and cancelled
  - a. Place order
  - b. Change price
  - c. Cancel order
- 3) Cancel all orders
  - a. Create two orders
  - b. Cancel all orders using "Cancel all" functionality
- 4) Place related order (if done)
  - a. Place order
  - b. Place two related orders
  - c. Slave order changed to Master order
  - d. Old main order executed
- 5) Place related order (OCO – One cancels other)

## Order Placed and modified to market order

- 1) Order placed and modified to market order
  - a. Place order
  - b. Modify to market order

### Place order

An order is placed: Limit order to buy GBPCHF spot @ 1.44900. The order has Order ID: 44309579. One notification received.

#### Notification a.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:19:21.657</Created>
  <CurrencyCode>CHF</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>GBPCHF</Instrument>
  <OrderId>44309579</OrderId>
  <Price>1.449</Price>
  <RegistrationTime>2012-04-17T05:19:21.643</RegistrationTime>
  <Symbol>GBP/CHF</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

Modify to market order

This order is then changed to market order. Three notifications received.

#### Notification b.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:23:03.63</Created>
  <CurrencyCode>CHF</CurrencyCode>
  <Duration>DayOrder</Duration>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <Instrument>GBPCHF</Instrument>
  <OrderId>44309579</OrderId>
  <OrderType>Market</OrderType>
  <RegistrationTime>2012-04-17T05:23:03.617</RegistrationTime>
  <Symbol>GBP/CHF</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

#### Notification b.2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:23:03.8</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>GBPCHF</Instrument>
  <OrderId>44309579</OrderId>
  <Symbol>GBP/CHF</Symbol>
</Order>
```

#### Notification b.3

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.8321575</ConversionRate>
  <Created>2012-04-17T05:23:04.093</Created>
  <CurrencyCode>CHF</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T05:23:03.73</ExecutionTime>
  <Instrument>GBPCHF</Instrument>
  <OpenPrice>1.45543</OpenPrice>
  <OpenSpot>1.45543</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68781511</PositionId>
```

```
<RegistrationTime>2012-04-17T05:23:03.777</RegistrationTime>  
<SourceOrderId>44309579</SourceOrderId>  
<SpotDate>2012-04-19</SpotDate>  
<Symbol>GBP/CHF</Symbol>  
<SystemOrigin>Other</SystemOrigin>  
<ValueDate>2012-04-19</ValueDate>  
</Position>
```

## Order placed, modified and cancelled

- 2) Order placed, modified and cancelled
  - a. Place order
  - b. Change price
  - c. Cancel order

### Place order

An order is placed: Limit order to buy EURNOK spot @ 7.5180. Order has Order ID: 44309581. One notification received.

#### Notification a.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:27:30.117</Created>
  <CurrencyCode>NOK</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>EURNOK</Instrument>
  <OrderId>44309581</OrderId>
  <Price>7.518</Price>
  <RegistrationTime>2012-04-17T05:27:30.1</RegistrationTime>
  <Symbol>EUR/NOK</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

## Change price

Then this order's price was changed to 7.5175. One notification received.

### Notification b.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:30:00.143</Created>
  <CurrencyCode>NOK</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <Instrument>EURNOK</Instrument>
  <OrderId>44309581</OrderId>
  <Price>7.5175</Price>
  <RegistrationTime>2012-04-17T05:30:00.133</RegistrationTime>
  <Symbol>EUR/NOK</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

## Cancel Order

Then this order was cancelled. One notification received.

### Notification c.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:31:46.333</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURNOK</Instrument>
  <OrderId>44309581</OrderId>
  <Symbol>EUR/NOK</Symbol>
</Order>
```

## Cancel all orders

I have a couple of orders for my account with order ID's 44309531 and 44309595. I cancelled them using "Cancel All" button from client station. Two notifications received.

### Notification b.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:41:03.117</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURCHF</Instrument>
  <OrderId>44309531</OrderId>
  <Symbol>EUR/CHF</Symbol>
</Order>
```

### Notification b.2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T05:41:03.36</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURNOK</Instrument>
  <OrderId>44309595</OrderId>
  <Symbol>EUR/NOK</Symbol>
</Order>
```



## Place related order (If Done)

- 4) Place related order (if done)
  - a. Place order
  - b. Place two related orders
  - c. Slave order changed to Master order
  - d. Old main order executed

### Place order

An order is placed: EURUSD limit order @ 1.30250. Main order ID is Order ID: 44309635. One notification received.

#### Notification a.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:37:17.77</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309635</OrderId>
  <Price>1.3025</Price>
  <RegistrationTime>2012-04-17T06:37:17.767</RegistrationTime>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

## Place two related orders

Two related orders were also placed along with the main order (44309635). Related order ID's: 44309636, 44309637. Five notifications received.

### Notification b.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:37:18.347</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309636</OrderId>
  <OrderRelation>IfDoneSlave</OrderRelation>
  <Price>1.309</Price>
  <RegistrationTime>2012-04-17T06:37:18.33</RegistrationTime>
  <RelatedOrderId>44309635</RelatedOrderId>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

### Notification b.2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:37:18.347</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309635</OrderId>
  <OrderRelation>IfDoneMaster</OrderRelation>
  <Price>1.3025</Price>
  <RegistrationTime>2012-04-17T06:37:17.767</RegistrationTime>
  <RelatedOrderId>44309636</RelatedOrderId>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

### Notification b.3

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
```

```

<Amount>5000</Amount>
<BuySell>Sell</BuySell>
<ClientId>2147123712</ClientId>
<ContractType>FxSpot</ContractType>
<Created>2012-04-17T06:37:18.77</Created>
<CurrencyCode>USD</CurrencyCode>
<ExchangeId>SBFX</ExchangeId>
<ExecutionType>New</ExecutionType>
<Instrument>EURUSD</Instrument>
<OrderId>44309637</OrderId>
<OrderRelation>IfDoneSlaveOco</OrderRelation>
<OrderType>StopIfOffered</OrderType>
<Price>1.296</Price>
<RegistrationTime>2012-04-17T06:37:18.75</RegistrationTime>
<RelatedOrderId>44309635</RelatedOrderId>
<RelatedSecondOrderId>44309636</RelatedSecondOrderId>
<Symbol>EUR/USD</Symbol>
<SystemOrigin>ClientStation</SystemOrigin>
<UserId>2147123712</UserId>
</Order>

```

#### Notification b.4

```

<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:37:18.77</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309635</OrderId>
  <OrderRelation>IfDoneMaster</OrderRelation>
  <Price>1.3025</Price>
  <RegistrationTime>2012-04-17T06:37:17.767</RegistrationTime>
  <RelatedOrderId>44309636</RelatedOrderId>
  <RelatedSecondOrderId>44309637</RelatedSecondOrderId>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>

```

#### Notification b.5

```

<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:37:18.77</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <Instrument>EURUSD</Instrument>

```

```
<OrderId>44309636</OrderId>  
<OrderRelation>IfDoneSlaveOco</OrderRelation>  
<Price>1.309</Price>  
<RegistrationTime>2012-04-17T06:37:18.33</RegistrationTime>  
<RelatedOrderId>44309635</RelatedOrderId>  
<RelatedSecondOrderId>44309637</RelatedSecondOrderId>  
<Symbol>EUR/USD</Symbol>  
<SystemOrigin>ClientStation</SystemOrigin>  
<UserId>2147123712</UserId>  
</Order>
```

Slave order changed to master order

Then one of the Slave orders (order ID: 44309637) was switched to master order. One notification received.

#### Notification c.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:46:59.443</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309636</OrderId>
  <OrderRelation>IfDoneSlaveOco</OrderRelation>
  <Price>1.309</Price>
  <RegistrationTime>2012-04-17T06:46:59.423</RegistrationTime>
  <RelatedOrderId>44309635</RelatedOrderId>
  <RelatedSecondOrderId>44309637</RelatedSecondOrderId>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>Other</SystemOrigin>
</Order>
```

Old main order executed

Then the order 44309635 was executed. Five notifications received.

#### Notification d.1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:53:29.977</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309636</OrderId>
  <Symbol>EUR/USD</Symbol>
</Order>
```

#### Notification d.2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:53:29.977</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309637</OrderId>
  <Symbol>EUR/USD</Symbol>
</Order>
```

#### Notification d.3

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>10</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.76357</ConversionRate>
  <Created>2012-04-17T06:53:29.98</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T06:53:29.92</ExecutionTime>
  <Instrument>EURUSD</Instrument>
  <OpenPrice>1.3025</OpenPrice>
  <OpenSpot>1.3025</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68781521</PositionId>
  <RegistrationTime>2012-04-17T06:53:29.95</RegistrationTime>
  <RelatedPositionId>68781523</RelatedPositionId>
  <SourceOrderId>44309635</SourceOrderId>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
```

```
</Position>
```

#### Notification d.4

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T06:53:29.977</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURUSD</Instrument>
  <OrderId>44309635</OrderId>
  <Symbol>EUR/USD</Symbol>
</Order>
```

#### Notification d.5

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>10</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.76357</ConversionRate>
  <Created>2012-04-17T06:53:30.04</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T06:53:30</ExecutionTime>
  <Instrument>EURUSD</Instrument>
  <OpenPrice>1.30963</OpenPrice>
  <OpenSpot>1.30963</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68781523</PositionId>
  <RegistrationTime>2012-04-17T06:53:30.02</RegistrationTime>
  <RelatedPositionId>68781521</RelatedPositionId>
  <SourceOrderId>44309636</SourceOrderId>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/USD</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Placing Related orders (OCO)

I placed a buy order for EURNZD @ 1.59800 and a related OCO order @ 1.61405. Main Order ID: 44309649, slave order ID: 44309650. Following emails (3) were received:

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T07:10:16.813</Created>
  <CurrencyCode>NZD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>EURNZD</Instrument>
  <OrderId>44309649</OrderId>
  <Price>1.598</Price>
  <RegistrationTime>2012-04-17T07:10:16.793</RegistrationTime>
  <Symbol>EUR/NZD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T07:10:17.3</Created>
  <CurrencyCode>NZD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>EURNZD</Instrument>
  <OrderId>44309650</OrderId>
  <OrderRelation>Oco</OrderRelation>
  <OrderType>StopIfBid</OrderType>
  <Price>1.61405</Price>
  <RegistrationTime>2012-04-17T07:10:17.233</RegistrationTime>
  <RelatedOrderId>44309649</RelatedOrderId>
  <Symbol>EUR/NZD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <UserId>2147123712</UserId>
</Order>
```

### Notification 3

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
```



```
<Created>2012-04-17T07:10:17.3</Created>  
<CurrencyCode>NZD</CurrencyCode>  
<ExchangeId>SBFX</ExchangeId>  
<ExecutionType>Changed</ExecutionType>  
<Instrument>EURNZD</Instrument>  
<OrderId>44309649</OrderId>  
<OrderRelation>Oco</OrderRelation>  
<Price>1.598</Price>  
<RegistrationTime>2012-04-17T07:10:16.793</RegistrationTime>  
<RelatedOrderId>44309650</RelatedOrderId>  
<Symbol>EUR/NZD</Symbol>  
<SystemOrigin>ClientStation</SystemOrigin>  
<UserId>2147123712</UserId>  
</Order>
```

Main order executed

Then the main order (44309649) was executed via order commander. Three notifications received.

#### Notification e.1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>12.2</Commission>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.62275</ConversionRate>
  <Created>2012-04-17T07:16:36.787</Created>
  <CurrencyCode>NZD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T07:16:36.73</ExecutionTime>
  <Instrument>EURNZD</Instrument>
  <OpenPrice>1.598</OpenPrice>
  <OpenSpot>1.598</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68781529</PositionId>
  <RegistrationTime>2012-04-17T07:16:36.77</RegistrationTime>
  <SourceOrderId>44309649</SourceOrderId>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>EUR/NZD</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

#### Notification e.2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T07:16:36.777</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURNZD</Instrument>
  <OrderId>44309649</OrderId>
  <Symbol>EUR/NZD</Symbol>
</Order>
```

#### Notification e.3

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <Created>2012-04-17T07:16:36.78</Created>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>EURNZD</Instrument>
  <OrderId>44309650</OrderId>
  <Symbol>EUR/NZD</Symbol>
```

</Order>

## Appendix 12 - Trade correction

This is an example of how the notification flow is when a trade correction is done. (EXAMPLE?) The following scenarios are shown;

- 1) Create position
- 2) Saxo Group performs trade correction

### Create position

A position is created: Long GBPUSD spot position @1.58859. The position has Position ID: 68578339. One notification received.

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.76227</ConversionRate>
  <Created>2012-04-17T04:53:04.173</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:53:04.137</ExecutionTime>
  <Instrument>GBPUSD</Instrument>
  <OpenPrice>1.58859</OpenPrice>
  <OpenSpot>1.58859</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68578339</PositionId>
  <RegistrationTime>2012-04-17T04:53:04.17</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>GBP/USD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Saxo Group performs trade correction

The position is corrected and modified by a Saxo Group Employee. Two notifications received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.76227</ConversionRate>
  <Created>2012-04-17T04:56:31.953</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:53:04.137</ExecutionTime>
  <Instrument>GBPUSD</Instrument>
  <OpenPrice>1.58859</OpenPrice>
  <OpenSpot>1.58859</OpenSpot>
  <PositionEvent>Deleted</PositionEvent>
  <PositionId>68578339</PositionId>
  <RegistrationTime>2012-04-17T04:53:04.17</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>GBP/USD</Symbol>
  <SystemOrigin>ClientStation</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>6000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>FxSpot</ContractType>
  <ConversionRate>0.76227</ConversionRate>
  <Created>2012-04-17T04:56:31.967</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>SBFX</ExchangeId>
  <ExecutionTime>2012-04-17T04:56:31.89</ExecutionTime>
  <Instrument>GBPUSD</Instrument>
  <OpenPrice>1.58859</OpenPrice>
  <OpenSpot>1.58859</OpenSpot>
  <PositionEvent>New</PositionEvent>
  <PositionId>68754800</PositionId>
  <RegistrationTime>2012-04-17T04:56:31.963</RegistrationTime>
  <SpotDate>2012-04-19</SpotDate>
  <Symbol>GBP/USD</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-04-19</ValueDate>
</Position>
```

## Appendix 13 – Bonds

### Position Buy event

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>165XXXINETUSD</AccountId>
  <Amount>50000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>547000</ClientId>
  <Commission>87.56</Commission>
  <ContractType>Bond</ContractType>
  <Created>2015-11-03T15:45:57.615</Created>
  <CurrencyCode>USD</CurrencyCode>
  <ExchangeId>BONDS</ExchangeId>
  <ExecutionTime>2015-11-03T15:45:57.608</ExecutionTime>
  <Instrument>XS0460546798</Instrument>
  <IsinCode>XS0460546798</IsinCode>
  <OpenPrice>79.099652778</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>742815636</PositionId>
  <RegistrationTime>2015-11-03T15:45:57.615</RegistrationTime>
  <Symbol>XS0460546798</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2015-11-05</ValueDate>
</Position>
```

### Position Sell Event

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>165XXXINETUSD</AccountId>
  <Amount>100000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>547000</ClientId>
  <Commission>80</Commission>
  <ContractType>Bond</ContractType>
  <Created>2015-11-03T14:18:45.323</Created>
  <CurrencyCode>EUR</CurrencyCode>
  <ExchangeId>BONDS</ExchangeId>
  <ExecutionTime>2015-11-03T14:18:45.319</ExecutionTime>
  <Instrument>XS1063244682</Instrument>
  <IsinCode>XS1063244682</IsinCode>
  <OpenPrice>79.759221311</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>742815616</PositionId>
  <RegistrationTime>2015-11-03T14:18:45.323</RegistrationTime>
  <Symbol>XS1063244682</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2015-11-05</ValueDate>
</Position>
```

## Deletion of position (position Netting)

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>165XXXINETUSD</AccountId>
  <ClientId>547000</ClientId>
  <Created>2015-11-04T01:45:26.179</Created>
  <PositionEvent>Deleted</PositionEvent>
  <PositionId>742815616</PositionId>
</Position>
```

## Appendix 15 - Partial Fill

This is an example of how the notification flow is when an order is partially filled. The following scenarios are shown;

- 1) Place buy order
- 2) Partial fill
- 3) Order executed fully
- 4) Place sell order
- 5) Sell order executed fully

### Place order

An order is placed: Order to buy contract DANSKE:xcse, quantity 20000. Order ID: 44328657. One notification received:

#### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
  <Order>
    <AccountId>xalk_test</AccountId>
    <Amount>20000</Amount>
    <BuySell>Buy</BuySell>
    <ClientId>2147123712</ClientId>
    <ContractType>Cfd</ContractType>
    <Created>2012-05-17T10:10:15.017</Created>
    <CurrencyCode>DKK</CurrencyCode>
    <ExchangeId>CSE</ExchangeId>
    <ExecutionType>New</ExecutionType>
    <Instrument>DANSKE:xcse</Instrument>
    <IsinCode>DK0010274414</IsinCode>
    <OrderId>44328657</OrderId>
    <Price>82</Price>
    <RegistrationTime>2012-05-17T10:10:15.01</RegistrationTime>
    <Symbol>DANSKE</Symbol>
    <SystemOrigin>Other</SystemOrigin>
  </Order>
```



## Partial fill

Out of 20000 quantity, 5000 was executed. Two notifications received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <Amount>20000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <ContractType>Cfd</ContractType>
  <Created>2012-05-17T10:10:31.623</Created>
  <CurrencyCode>DKK</CurrencyCode>
  <ExchangeId>CSE</ExchangeId>
  <ExecutionType>Changed</ExecutionType>
  <FilledAmount>5000</FilledAmount>
  <Instrument>DANSKE:xcse</Instrument>
  <IsinCode>DK0010274414</IsinCode>
  <OrderId>44328657</OrderId>
  <Price>82</Price>
  <RegistrationTime>2012-05-17T10:10:15.01</RegistrationTime>
  <Symbol>DANSKE</Symbol>
  <SystemOrigin>Other</SystemOrigin>
</Order>
```

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>5000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>409.1545</Commission>
  <ContractType>Cfd</ContractType>
  <ConversionRate>0.13453815</ConversionRate>
  <Created>2012-05-17T10:10:31.64</Created>
  <CurrencyCode>DKK</CurrencyCode>
  <ExchangeId>CSE</ExchangeId>
  <ExecutionTime>2012-05-17T10:10:31.587</ExecutionTime>
  <Instrument>DANSKE:xcse</Instrument>
  <IsinCode>DK0010274414</IsinCode>
  <OpenPrice>81.8309</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>69645699</PositionId>
  <RegistrationTime>2012-05-17T10:10:31.623</RegistrationTime>
  <SourceOrderId>44328657</SourceOrderId>
  <Symbol>DANSKE</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-05-21</ValueDate>
</Position>
```

## Order executed fully

Then rest of the amount (15000) was executed. Two notifications received.

The existing position is modified, cumulated and no new position is created. Also position event "<PositionEvent>Updated</PositionEvent>" was changed to "Updated" and amount and other fields gets updated as per the xml in attached document.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>Cfd</ContractType>
  <Created>2012-05-17T10:11:10.563</Created>
  <ExchangeId>CSE</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>DANSKE:xcse</Instrument>
  <IsinCode>DK0010274414</IsinCode>
  <OrderId>44328657</OrderId>
  <Symbol>DANSKE</Symbol>
</Order>
```

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>20000</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>1636.618</Commission>
  <ContractType>Cfd</ContractType>
  <ConversionRate>0.13453796</ConversionRate>
  <Created>2012-05-17T10:11:10.55</Created>
  <CurrencyCode>DKK</CurrencyCode>
  <ExchangeId>CSE</ExchangeId>
  <ExecutionTime>2012-05-17T10:11:10.55</ExecutionTime>
  <Instrument>DANSKE:xcse</Instrument>
  <IsinCode>DK0010274414</IsinCode>
  <OpenPrice>81.8309</OpenPrice>
  <OpenSpot>81.8309</OpenSpot>
  <PositionEvent>Updated</PositionEvent>
  <PositionId>69645699</PositionId>
  <RegistrationTime>2012-05-17T10:10:31.623</RegistrationTime>
  <SourceOrderId>44328657</SourceOrderId>
  <Symbol>DANSKE</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-05-21</ValueDate>
</Position>
```

## Place sell order

After executing complete 20000 buy position, a sell order of 20000 quantity is placed for the same instrument. One notification received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
```

```
<Amount>20000</Amount>
<BuySell>Sell</BuySell>
<ClientId>2147123712</ClientId>
<ContractType>Cfd</ContractType>
<Created>2012-05-17T10:15:54.58</Created>
<CurrencyCode>DKK</CurrencyCode>
<ExchangeId>CSE</ExchangeId>
<ExecutionType>New</ExecutionType>
<Instrument>DANSKE:xcse</Instrument>
<IsinCode>DK0010274414</IsinCode>
<OrderId>44328675</OrderId>
<Price>81.95</Price>
<RegistrationTime>2012-05-17T10:15:54.57</RegistrationTime>
<Symbol>DANSKE</Symbol>
<SystemOrigin>Other</SystemOrigin>
</Order>
```

## Sell order executed fully

Then this order was executed completely via order commander. Two notifications received.

### Notification 1

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>xalk_test</AccountId>
  <Amount>20000</Amount>
  <BuySell>Sell</BuySell>
  <ClientId>2147123712</ClientId>
  <Commission>1638.18</Commission>
  <ContractType>Cfd</ContractType>
  <ConversionRate>0.1345379</ConversionRate>
  <Created>2012-05-17T10:16:11.46</Created>
  <CurrencyCode>DKK</CurrencyCode>
  <ExchangeId>CSE</ExchangeId>
  <ExecutionTime>2012-05-17T10:16:11.437</ExecutionTime>
  <Instrument>DANSKE:xcse</Instrument>
  <IsinCode>DK0010274414</IsinCode>
  <OpenPrice>81.909</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>69645721</PositionId>
  <RegistrationTime>2012-05-17T10:16:11.453</RegistrationTime>
  <SourceOrderId>44328675</SourceOrderId>
  <Symbol>DANSKE</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2012-05-21</ValueDate>
</Position>
```

### Notification 2

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>xalk_test</AccountId>
  <ClientId>2147123712</ClientId>
  <ContractType>Cfd</ContractType>
  <Created>2012-05-17T10:16:11.453</Created>
  <ExchangeId>CSE</ExchangeId>
  <ExecutionType>Deleted</ExecutionType>
  <Instrument>DANSKE:xcse</Instrument>
  <IsinCode>DK0010274414</IsinCode>
  <OrderId>44328675</OrderId>
```

<Symbol>DANSKE</Symbol>  
</Order>

## Appendix 16 - Future Contract – New Order

Below you can find a XML example for a Future Contract – New Order event notification:

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>58756INET</AccountId>
  <Amount>1</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>197573</ClientId>
  <ContractType>FutureContract</ContractType>
  <Created>2014-04-04T06:10:39.85</Created>
  <CurrencyCode>EUR</CurrencyCode>
  <Duration>DayOrder</Duration>
  <ExchangeId>EUREX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>FEPPJ5</Instrument>
  <IsinCode>DE000A0Z3068</IsinCode>
  <OrderId>164690950</OrderId>
  <OrderType>Market</OrderType>
  <Price>14.6</Price>
  <RegistrationTime>2014-04-04T06:10:39.853</RegistrationTime>
  <Symbol>FEPPJ5</Symbol>
  <SystemOrigin>WebTrader</SystemOrigin>
  <UserId>6231222</UserId>
  <OptionExpiryDate>2015-04-23</OptionExpiryDate>
</Order>
```

## Appendix 17 - Future Contract – New Position

Below you can find a XML example for a Future Contract – New Position event notification:

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>58756INET</AccountId>
  <Amount>1</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>197573</ClientId>
  <Commission>10</Commission>
  <ContractType>FutureContract</ContractType>
  <ConversionRate>1.36968</ConversionRate>
  <Created>2014-04-04T06:10:50.173</Created>
  <CurrencyCode>EUR</CurrencyCode>
  <ExchangeFee>3</ExchangeFee>
  <ExchangeId>EUREX</ExchangeId>
  <ExecutionTime>2014-04-04T06:10:50.153</ExecutionTime>
  <Instrument>FEPPJ5</Instrument>
  <IsinCode>DE000A0Z3068</IsinCode>
  <OpenPrice>14.6</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>1019215708</PositionId>
  <RegistrationTime>2014-04-04T06:10:50.167</RegistrationTime>
  <SourceOrderId>164690950</SourceOrderId>
  <Symbol>FEPPJ5</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ValueDate>2014-04-04</ValueDate>
  <OptionExpiryDate>2015-04-23</OptionExpiryDate>
</Position>
```

## Appendix 18 - Contract Option – New Order

Below you can find a XML example for a Contract Option – New Order event notification:

```
<?xml version="1.0" encoding="utf-8"?>
<Order>
  <AccountId>58756INET</AccountId>
  <Amount>1</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>197573</ClientId>
  <ContractType>ContractOption</ContractType>
  <Created>2014-04-04T06:11:03.14</Created>
  <CurrencyCode>HKD</CurrencyCode>
  <Duration>DayOrder</Duration>
  <ExchangeId>HKEX</ExchangeId>
  <ExecutionType>New</ExecutionType>
  <Instrument>00005/J14C80:xhkg</Instrument>
  <OrderId>164690951</OrderId>
  <OrderType>Market</OrderType>
  <Price>0.66</Price>
  <RegistrationTime>2014-04-04T06:11:03.14</RegistrationTime>
  <Symbol>00005</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <StrikePrice>80</StrikePrice>
  <CallPut>Call</CallPut>
  <OptionExpiryDate>2014-04-29</OptionExpiryDate>
</Order>
```

## Appendix 19 - Contract Option – New Position

Below you can find a XML example for a Contract Option – New Position event notification

```
<?xml version="1.0" encoding="utf-8"?>
<Position>
  <AccountId>58756INET</AccountId>
  <Amount>1</Amount>
  <BuySell>Buy</BuySell>
  <ClientId>197573</ClientId>
  <Commission>45</Commission>
  <ContractType>ContractOption</ContractType>
  <ConversionRate>0.128909</ConversionRate>
  <Created>2014-04-04T06:11:03.823</Created>
  <CurrencyCode>HKD</CurrencyCode>
  <ExchangeFee>3</ExchangeFee>
  <ExchangeId>HKEX</ExchangeId>
  <ExecutionTime>2014-04-04T06:11:03.807</ExecutionTime>
  <Instrument>00005/J14C80:xhkg</Instrument>
  <OpenPrice>1</OpenPrice>
  <PositionEvent>New</PositionEvent>
  <PositionId>1019216206</PositionId>
  <RegistrationTime>2014-04-04T06:11:03.807</RegistrationTime>
  <SourceOrderId>164690951</SourceOrderId>
  <SpotDate>2014-04-04</SpotDate>
  <StrikePrice>80</StrikePrice>
  <Symbol>00005</Symbol>
  <SystemOrigin>Other</SystemOrigin>
  <ToOpenClose>ToOpen</ToOpenClose>
  <ValueDate>2014-04-04</ValueDate>
  <CallPut>Call</CallPut>
  <OptionExpiryDate>2014-04-29</OptionExpiryDate>
</Position>
```



## Appendix 20 – Tool ID

Below you can find a conversion table for the ToolID as broadcasted in Tag 20034

ToolID	Actor	External Description
1	Client	Saxo Trader
2	Saxo Employee	Manual Trade Input
3	Saxo Employee	Manual Trade Input
4	Client	Saxo Trader
5	Saxo Bank	Saxo Bank Office
6	Saxo Bank	FX Option Exercize
7	Saxo Bank	Cash Transfer
8	Client	Order Commander
9	Client API	B2B Server
10	Saxo Bank	Corporate Action
11	Saxo Bank	Trade Correction
12	Saxo Bank	Position Correction
17	Saxo Bank	Margin Stop Out
19	Saxo Bank	Futures Expiry
22	Saxo Bank	Security Transfer
24	Saxo Bank	FX Option Barrier Service
27	Client	Open Order Monitor
28	Saxo Bank	Corporate Action
29	Client	Saxo Platforms
30	Client	Mobile Trader
34	Saxo Employee	Manual Trade Input
35	Saxo Bank	Bond Drawing
36	Saxo Bank	Bond Maturity
37	Client	AllocationService
40	Client	OpenAPI
41	Client	Saxo Trader for ETOS
42	Client	Saxo Trader
44	Client	Saxo Trader
45	Client	Saxo Trader
46	Client	Saxo Trader
47	Client	C-Card Payment

ToolID	Actor	External Description
48	Client API	Saxo Prime
49	Saxo Bank	ETO Assignment
50	Saxo Bank	ETO Expiry
53	Saxo Employee	ETO Correction
54	Saxo Employee	ETO Correction
55	Saxo Employee	ETO Correction
59	Client	AutoTradeEngine
60	Client	Traiana GiveIN
62	Client API	Saxo Prime
63	Client API	Saxo Prime
64	Client	Saxo Trader
65	Client	Saxo Trader
66	Client	Saxo Trader
67	Client	Saxo Trader
69	Client API	Saxo Prime
72	Client API	Saxo Prime
73	Client API	Saxo Prime
79	Client API	Saxo Prime
80	Client API	Saxo Prime
81	Client API	Saxo Prime
84	Client API	Saxo Prime
85	Client API	Saxo Prime
86	Client API	Saxo Prime
87	Client API	Saxo Prime
89	Saxo Employee	Position Correction
90	Saxo Bank	Cascaded Expiry
91	Client	FX Options TradeCapture
2001	Client	SaxoTraderGO
2005	Client	Open API MiniTrader
2016	Client	ExcelTrader
2028	Client	SaxoTrader Pro