Version as of 15 July 2021

Nordea Markets Foreign Exchange (FX) Disclosure Notice; Algo Due Diligence Template

	GENERAL
	This general section outlines the core features of the algorithm. Providers may consolidate answers 1–5 into a table or grid if they wish to cover multiple algorithms with the same template.
Q1	Algo Provider (also referred to as "you" or "your" below as required):
A1	**Nordea Bank Abp**
Q2	Algo name(s):
A2	**TWAP** **VWAP** **POHV** **Make** **Take** **Sweep**
Q3	Liquidity type (internal, external, hybrid):
A3	**Hybrid, at customers discretion**
Q4	Products covered (spot, NDF):
A4	**Spot**
Q5	Description[1] of algo(s):
A5	**TWAP:** An algorithm that targets the time-weighted average price (TWAP) benchmark and strives to minimize slippage against it. **VWAP:** An algorithm that targets the volume-weighted average price (VWAP) benchmark.

[1] You may find it helpful to refer to the 'algo archetypes' delineated in section 2.1 of FX execution algorithms and market functioning

 Nordea Bank Abp, Satamaradankatu 5, FI-00020 NORDEA, Finland, domicile Helsinki, Business ID 2858394-9 SATD013DL 10.18

	POHV: This algorithm participates in the market passively, based on the percentage of historical volume (POHV) in the traded currency pair. **Make:** A passive liquidity providing algorithm that will never cross the mid-price. **Take:** An algorithm meant for urgent but smart execution. **Sweep:** The Sweep algorithm uses the smart order routing logic to take liquidity aggressively. Sweep gathers all the available liquidity from first and last-look ECNs. The algorithm evaluates these sources dynamically to find an optimal allocation between the available liquidity sources.
Q6	Please describe any parameters or controls the user may adjust:
A6	**TWAP:** Include Liquidity: Decides whether the algorithm has access to aggress against Nordea's own liquidity pool. This pool allows the algorithm to execute without showing the interest to any other market players than Nordea. The algorithm will use liquidity from this pool only if it offers a price that is better than any other pool available to the algorithm at the time of execution. Limit Price: Price that the algo will never breach. That is, for buy orders the algo will never place orders above this limit, for sell orders it will never place orders below this limit. Price Sensitivity Feature: Price sensitivity determines how much ahead/behind compared to the schedule the algo can move when a favourable/unfavourable price emerges. A high price sensitivity means that the algo can deviate further away from the schedule than it would be able to do with a lower price sensitivity. Allowing more flexibility for the algo to deviate from benchmark may save spread for the user in situations where market exhibits a lot of volatility. For example, if a user uses high price sensitivity and price moves favourably for the order, then the algo will speed up the rate of trading. This may mean that the order completes much earlier than expected unless an unfavourable price movement is observed. Minimum Quantity: Minimum quantity the algo should complete with Price Sensitivity enabled. By using minimum quantity, the user can still make sure that the order will strive to complete when price sensitivity feature is used. Would Feature: Allows user to specify a price, which will trigger the completion of the order up to a size specified as a parameter. Would Price:

	Trigger price to invoke the feature. Would Size: The percentage of the order that should be completed by sweeping the market if would price is triggered. **VWAP:** Include Liquidity: Decides whether the algorithm has access to aggress against Nordea's own liquidity pool. This pool allows the algorithm to execute without showing the interest to any other market players than Nordea. The algorithm will use liquidity from this pool only if it offers a price that is better than any other pool available to the algorithm at the time of execution. Limit Price: Price that the algo will never breach. That is, for buy orders the algo will never place orders above this limit, for sell orders it will never place orders below this limit. Price Sensitivity Feature: Price sensitivity determines how much ahead/behind compared to the schedule the algo can move when a favourable/unfavourable price emerges. A high price sensitivity means that the algo can deviate further away from the schedule than it would be able to do with a lower price sensitivity. Allowing more flexibility for the algo to deviate from benchmark may save spread for the user in situations where market exhibits a lot of volatility. For example, if a user uses high price sensitivity and price moves favourably for the order, then the algo will speed up the rate of trading. This may mean that the order completes much earlier than expected unless an unfavourable price movement is observed. Minimum Quantity: Minimum quantity the algo should complete with Price Sensitivity enabled. By using minimum quantity, the user can still make sure that the order will strive to complete when price sensitivity feature is used. Would Feature: Allows user to specify a price, which will trigger the completion of the order up to a size specified as a parameter. Would Price: Trigger price to invoke the feature. Would Size: The percentage of the order that should be completed by sweeping the market if would price is triggered. **POHV:** Include Liquidity: Decides whether the algorithm has access to aggress against Nordea's own liquidity pool. This pool allows the algorithm to execute without showing the interest to any other market players than Nordea. The algorithm will use liquidity from this pool only if it offers a price that is better than any other pool available to the algorithm at the time

	of execution. Target Volume: Participation rate compared to the market volume. Limit Price: Price that the algorithm will never breach. That is, for buy orders the algorithm will never place orders above this limit, for sell orders it will never place orders below this limit. Would Feature: Allows user to specify a price, which will trigger the completion of the order up to a size specified as a parameter. Would Price: Trigger price to invoke the feature Would Size: The percentage of the order that should be completed by sweeping the market if would price is triggered. **Make:** Include Liquidity: Decides whether the algorithm has access to aggress against Nordea's own liquidity pool. This pool allows the algorithm to execute without showing the interest to any other market players than Nordea. The algorithm will use liquidity from this pool only if it offers a price that is better than any other pool available to the algorithm at the time of execution. Limit Price: Price that the algorithm will never breach. That is, for buy orders the algorithm will never place orders above this limit, for sell orders it will never place orders below this limit. Aggression (Default = Medium): Decides which fraction of the market trades the strategy strives to be. Possible values are: • Low • Medium • High = keep active order in the market all the time by replenishing after each fill. Price Aggression (Default = Medium): Decides how the algorithm will post limit orders relative to bid and ask. Possible values are: • Low = top of book • Medium = between top of book and mid-price • High = mid-price

	Price aggression represents the maximum aggression. Make algorithm with High price aggression will be able to place limit orders up to the mid-point of the spread, but will not do so unless it has fallen behind the schedule. Max Display Size: Maximum size that can be posted across all venues available to the algorithm at a time. Would Feature: Allows user to specify a price, which will trigger the completion of the order up to a size specified as a parameter. Would Price: Trigger price to invoke the feature. Would Size: The percentage of the order that should be completed by sweeping the market if would price is triggered. **Take:** Include Liquidity: Decides whether the algorithm has access to aggress against Nordea's own liquidity pool. This pool allows the algorithm to execute without showing the interest to any other market players than Nordea. The algorithm will use liquidity from this pool only if it offers a price that is better than any other pool available to the algorithm at the time of execution. Limit Price: Price that the algorithm will never breach. That is, for buy orders the algorithm will never place orders above this limit, for sell orders it will never place orders below this limit. Aggression (Default = Medium): Decides the minimum pace that the algorithm needs to keep up with. Possible values are Low, Medium and High. Stop-Loss Threshold: When the algorithm starts, it calculates sweep-to-fill price that is an estimate of the cost that would be accrued if the whole amount was traded at once. Afterwards, the algorithm will continue to update the sweep-to-fill price to reflect order book changes. If this price moves more than the stop-loss threshold, the algorithm will automatically sweep the remaining from the market. **Sweep:** Limit Price: Price that the algorithm will never breach. That is, for buy orders the algorithm will never place orders above this limit, for sell orders it will never place orders below this limit.
Q7	Please specify if the product is built internally or externally:

A7	**The algo suite is a mix of in-house software and externally procured software and services.**
	CONFLICTS OF INTEREST
	Some conflicts of interest may be expected but it is important to know what they are and what steps have been taken to manage them. This way the Algo User can make an informed decision.
Q8	If principal liquidity interacts with the Algo User's order, how does this happen and what steps are taken to ensure the fill is a fair one from the order's point of view?
A8	**If internal Nordea liquidity is enabled by the customer, the smart orders router (SOR), will elect to trade on the Nordea provided price, should it have the best price for the child order in that particular point in time. The algo may clean up (trade the last remaining part- usually a very small amount) internally, in order to achieve the full amount of the parent order – regardless of there being a better price in the market. This is necessary as some market places does not allow to trade small amounts.**
Q9	If another part of your business needs to hedge or trade in the same direction as the Algo User's order, how are fills allocated between the two?
A9	**The algos execute in the market independently of other liquidity consumers, internally. There is no allocation or prioritization, It is strictly first come, first serve.**
Q10	Are there any particular commercial interests in trading venues or other relevant service providers that interact with the algorithm provided by you? If so, how are such conflicts addressed?
A10	**No**
Q11	Please elaborate on your role as regards market risk, counterparty risk, and settlement risk.
A11	**The market risk is assumed by the customer, the counterparty and settlement risk for the executed child orders in the market is assumed by Nordea.**
Q12	Is there anything else of which you feel the Algo User should be aware?
A12	**No**
	ALLOCATION POLICY
	There are many different approaches to allocations. It is important to understand what happens in circumstances where multiple clients wish to trade or, indeed, when one order would be used to fill the order of another client.
Q13	If you have more than one client order wishing to trade in the same pair and on the same side, how are fills allocated amongst these orders?

A13	**Each algo runs on its own independent trajectory. There is no cross communication between the algos, and they will be allocated the fills achieved as a result of their independent operation.**
Q14	If two client orders are eligible for execution netting, how does this process work?
A14	**Netting takes place by trading on the Nordea internal exchange, on a first come first serve basis – per child order.**
	ROUTING POLICY
	Routing policy is an important topic. There are several components such as how execution venues are evaluated, curated, and prioritised. Also covered is the question of what fair-value mid the algo uses to make routing decisions and how information leakage is avoided when placing lit orders. Finally, internalisation is defined: some providers have a strict definition such as 'two algo orders netting' whereas others will include midbooks and trades where they have shown a skew through mid externally to incentivise another counterparty to fill them.
Q15	How are hedging execution venues evaluated, including both observable (spread, impact) and implicit costs (information leakage)?
A15	**Venues are curated on a high level and to a large extend shared with the principal business. Spreads are evaluated at run time by the SOR. Impact, fill rate and information leakage is done on a statistical basis.**
Q16	How do you prioritise between different execution venues (both external and internal sources) when routing orders?
A16	**Prioritization for consuming liquidity is done based on price and size available by the SOR. Posting liquidity is done in a semi random fashion, taking into account where the SOR deems the highest possibility of a fill will be.**
Q17	If multiple clients enter orders in the same pair, will you aggregate these orders before placing orders externally or treat each client order individually and place multiple similar orders, which may compete with one another for fills?
A17	**Orders are independent, and there is no coordination between two competing orders.**
Q18	What – if any – ongoing work do you do in order to curate execution venues, where curation is possible? Approximately how often is this conducted?
A18	**Curation takes place on an ad-hoc basis**
Q19	Do you have any logic to avoid orders on venues where the order book is visible to all participants (lit execution venues) causing information leakage? If so, please describe it.
A19	**It is possible for the customer to request that no lit markets be used.**

Q20	Does the mid/fair-value used by the algorithm differ from the one used by your own market making system for pricing and risk management? If yes, please specify.
A20	**Yes. The systems are independent and does not share codebase.**
Q21	Please define your understanding of 'internalisation' and, using an example, describe how this works in practice, demonstrating if/how your Algo Clients benefit from this process. If you wish to do so you may provide an indication of how much volume is internalised on average.
A21	**Internalization, as defined by Nordea, can only pertain to two scenarios:** **1) The algo will post a firm order on the Nordea internal exchange, and some other part of the Nordea trading franchise aggresses on that order, resulting in an external fill.** **2) The algo compares Nordeas risk transfer price to all other prices in the market, and finds the Nordea price best- and thus executes on it.** **We do not define mid book matches, trades on skew etc. as internalized.** **Internalization ratio is reported in the TCA report.**
	SEGREGATION POLICY
	Segregation policy is all about keeping order information private and reducing the risk of signalling.
Q22	Please describe if and how the algo orders are segregated within your institution.
A22	**Access to the algo control center is only granted to a specific support team within the business, as well as named staff who has the responsibility of maintaining and supporting the product.** **Front office access to the underlying trading books are only granted to personel with a need to know.**
Q23	Can sales and trading personnel who provide intraday 'market colour' view algo orders at any stage? If so, what steps have been taken to minimise the risk of information leakage?
A23	**Steps have been taken to make sure that as few people in trading and sales as possible has access to the trading and sales book. No trader with a risk mandate has access to the book or the algo control center.**
Q24	Can discretionary traders who may enter or exit risk for your institution view algo orders at any stage? If so, what steps have been taken to minimise the risk of information leakage?
A24	**No**

Q25	Can an electronic market making system view algo orders at any stage? If so, what steps have been taken to minimise the risk of information leakage or misuse of information?
A25	**The electronic market making system cannot view algo orders. The electronic market making system can only see child orders, if the algo trades with the internal liquidity. In that case the electronic market making system does not see the whole parent order, but just the particular fill.**
Q26	Are algo order flows included in any market positioning tools or analyses that other clients may use?
A26	**No**
	SAFETY FEATURES
	Safety features might include fat-finger limits, kill switches or protections that automatically suspend the order when it trades too fast or in certain market conditions.
Q27	Please describe any in-built safety features you have that may cause an order to be suspended or rejected.
A27	**There is a size cap on auto acceptance for algos. Placing algos above this limit requires manual intervention by business support team, in order to provide the customer security in the fact that there is an additional layer of control.**
Q28	Please explain what you have done, and will continue to do, to ensure the integrity of the electronic trading system you provide for clients to use (including the system's reliability, security, capacity and contingency measures).
A28	**The support and monoitoring of the algo suite follows the same principles as the rest of the eFX business, where we strive to have a secure and reliable setup.**
	TCA
	TCA is an increasingly important part of the service. Where the TCA is not third party it is important to understand internal metrics. For example, if you have 'beaten risk transfer price' by 3bp how is that risk transfer price calculated?
Q29	Do you support any TCA or analytics? If so, please specify which providers.
A29	**Yes. We provide our own TCA as well as 3rd party TCA, by BestX.**
Q30	If you provide proprietary analytics, please describe how relevant metrics are calculated (mid-price, risk-transfer benchmarks, etc.).
A30	**Arrival price is determined by looking at the TOB spread on the primary market. TWAP benchmarks are deduced from snapshots of primary markets during the time of the execution.**

Q31	If you provide proprietary analytics, is there a difference in data provided to different users? If so, please elaborate.
A31	**Data in the Nordea TCA is deduced from the same data sources for all customers. Data in the BestX TCA is provided by 3rd party, and not under Nordea control.**
	SWAPS
	Algo Users may have a need to roll an algo execution entirely/partially to one or more forward value date/s. If roll forwards are executed with the Algo Provider, it is crucial to understand if the respective swap prices are competitive and whether potentially sensitive order information is exposed. For example, does the swaps trader know which side of the quote the algo execution is on or do they receive a two-sided RFQ? Also, does the swap trader know they are quoting a captive spot fill or does it appear the same as RFQs that are priced in competition with other banks?
Q32	What information is provided to the STIRT desk when there is a request for swap pricing from an algo order?
A32	**Not applicable.**