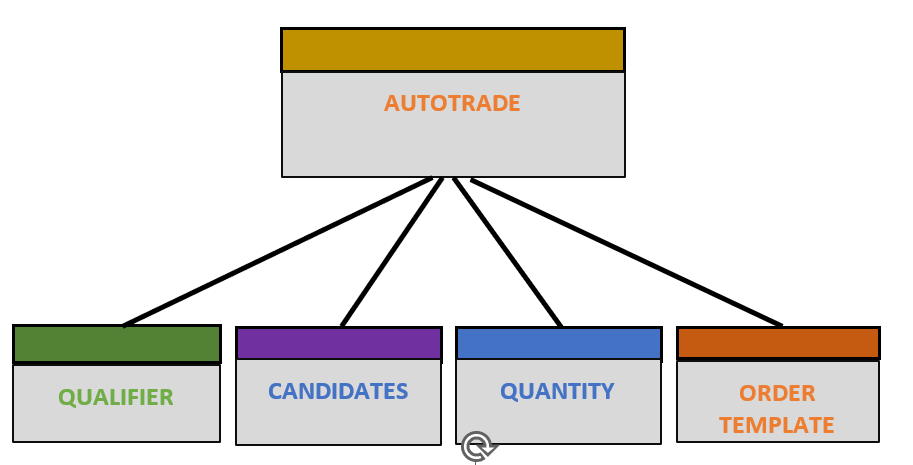
**Specification  
  
TrendTrader**Version 2.2  
2023-09-09

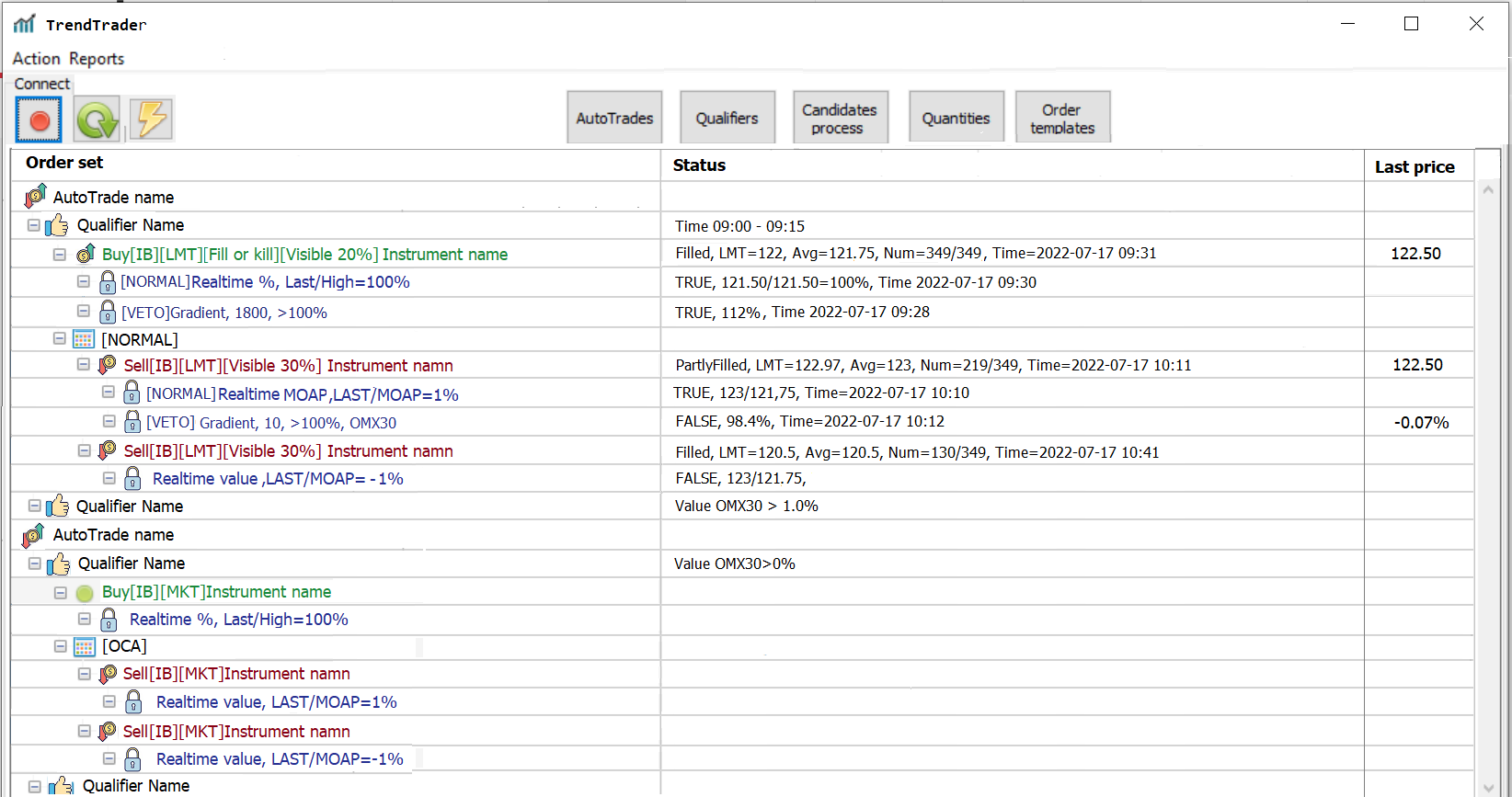
**AutoTrade**

An Autotrade is a combination of a Qualifier, Candidates, Quantity, and an Order-template.



**PROCESS**

**Main form**

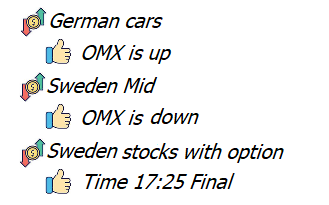


**Status kolumn/AutoTrade**  
Empty  
  
**Status column/Qualifier**1. [Compare]Instrument1>Instrument2 *and/or*  
 [Value]Instrumet>X *and/or*  
 [Time]From-To *and/or*  
 [Parse] Parse name  
  
**Status column/Order**  
1. Status: PendingSubmit, PendingCancel, PreSubmitted, Submitted, Cancelled,  
 Filled, PartlyFilled, Sleeping, Error *or* Not Considered  
2. Limit order: If LMT then LMT=X  
3. Average price: Avg=X  
4. Numbers: Num=Received/Total number in order  
5. Time last : Time

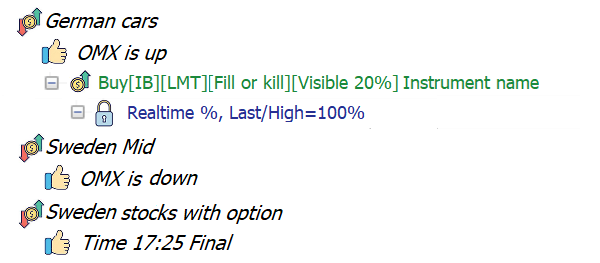
How it will work:

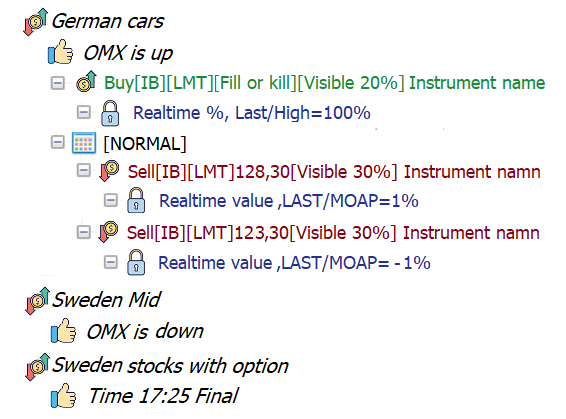
1. En bild som visar text

   Automatiskt genererad beskrivningWhen clicking on START button  on main form all active AutoTrades will start and be visible on main form. An AutoTrade is active when its active checkbox is checked. An active AutoTrade will become TRUE when its qualifier is TRUE.  
     
     
     
     
   If none of the three upper checkboxes are checked in the AutoTrade-form, is automatically TRUE otherwise all its checked conditions must be true.
2. When an AutoTrade becomes TRUE its qualifier will be visible. and it will start its Candidates process.



1. When there are instruments (candidates) in a candidates-list which reach up to the candidate’s limit, an order set will be created from the Quantity and the Order template. The mother orders in those order groups will then be visible in the monitor form. Factor will not be used in AutoOrder. The choice of instrument is included in Order and Condition.



1. When mother order is *filled*, the order group with child orders will be visible.   
   
2. If “*Hide filled orders”* is checked this Auto order will be removed.

En bild som visar bord

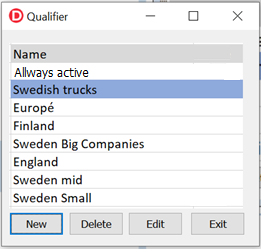
Automatiskt genererad beskrivning**AutoTrade**  
An AutoTrade has a name and must include one Qualifier, one Candidates process, one quantity and one order template. It can be active or not.   
If it’s not active, it will never start.   
If it is active, it will start when its qualifier is true.

All AutoTrades has a name and must include one Qualifier, one Candidates process, one quantity and one order template

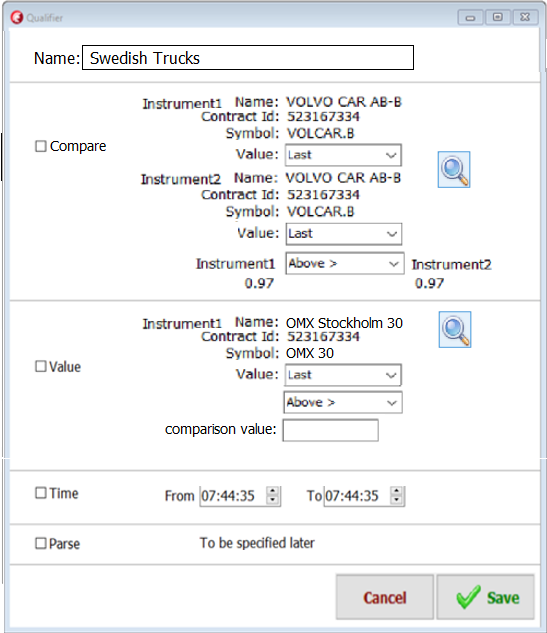
En bild som visar text

Automatiskt genererad beskrivning

**Qualifier**A Qualifier is a part which activates an AutoTrade.   
A Qualifier will be true and activate it’s AutoTrade if it’s active itself and all other checkboxes which are checked have conditions which is true.   
If no checkbox is checked the **Qualifier** is true.   
  
Parse will be specified later. It will be an external applications that send a signal to our application. (Example: An application search text from Reuters and find “big order to Volvo cars” it will then send signal to our application)   
When clicking on Qualifier button you open the Qualifier-List.

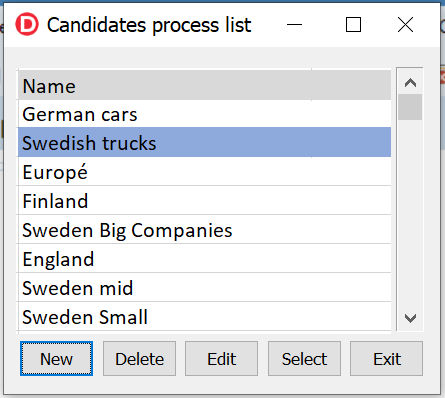


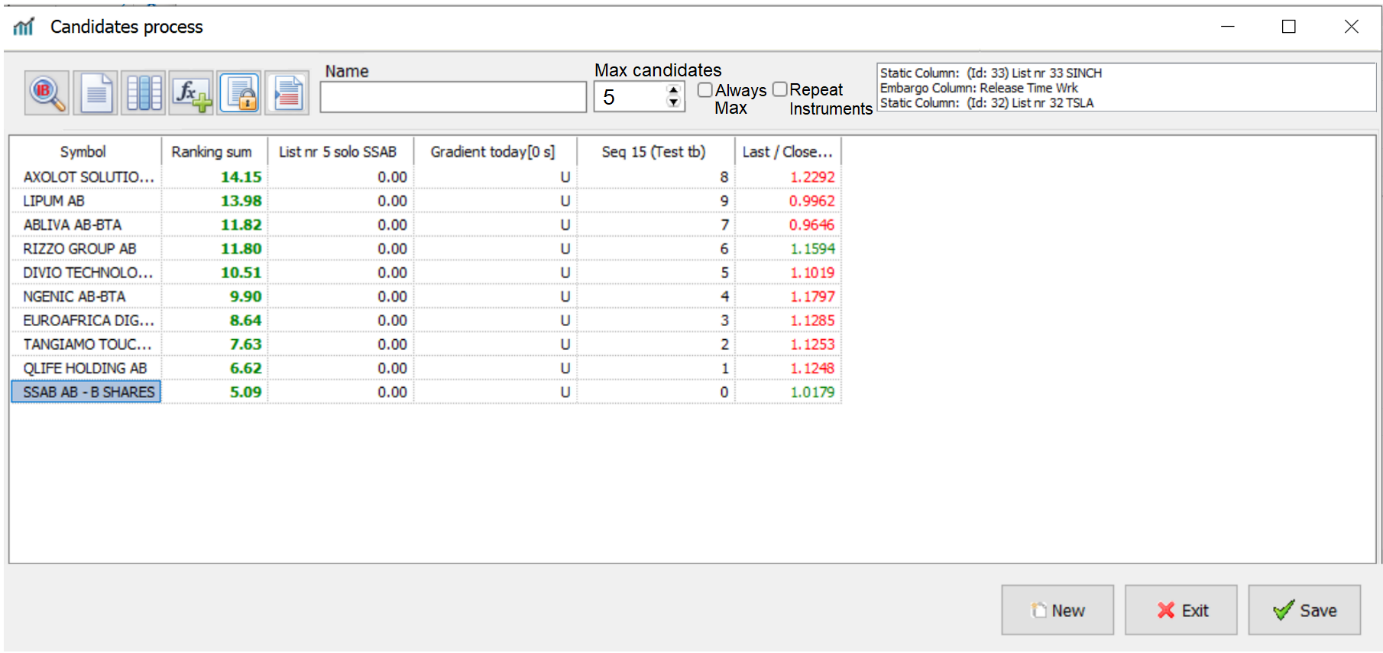
When double click on an item or selecting New or Edit you open the Qualifier-form.



**Candidates process**(change name from Scan to Candidates process)

In the candidates-form we produce a list of instruments.   
The list is produced with *IBScanner, Static lists, Added columns* and *Gradient*. All rows in the candidate list will have a ranking sum. This sum is produced by adding each columns value multiplied by its weight value. This ranking sum is compared with Candidates limit and if the ranking sum is grater or equal to this limit the instrument is ready to be processed in an *AutoTrade*.   
The most important part in the candidates process is the *IBScanner* which allow you to search and create a list of instruments which have behaved in an asked way.   
You need to add at least one *Static list* or one *IBScanner*to produce a list. There can maximum be one IBScanner and maximum one Gradient but there can be several static lists and several added columns.  
The list is continually updated in real time.





 Opens IBScanner. En bild som visar text, skärmbild, bildram

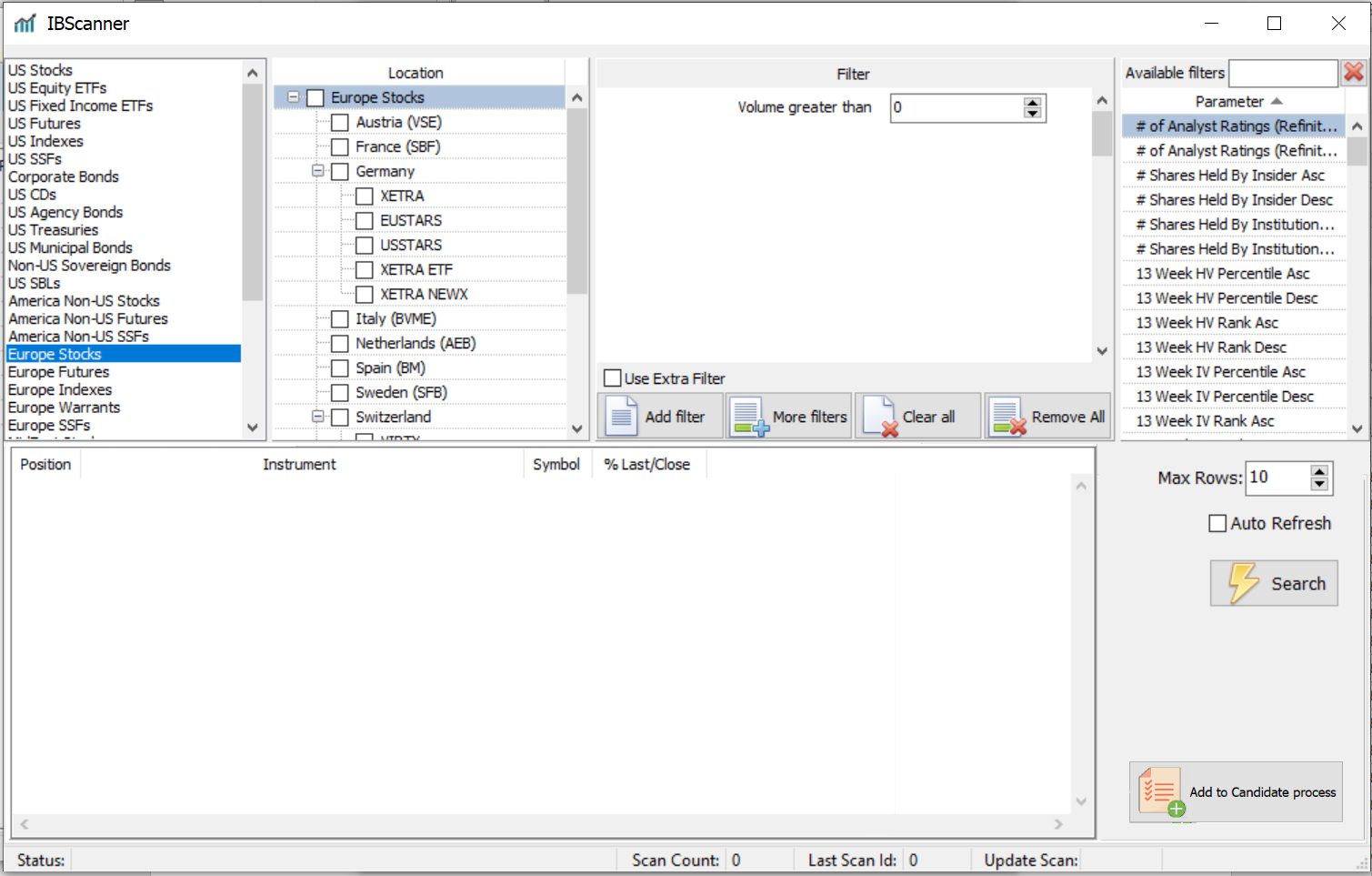
Automatiskt genererad beskrivning Opens Static lists.  Opens Add Column. En bild som visar text, bildram

Automatiskt genererad beskrivning Opens Gradient.  
 Opens Embargo.  Opens PriceChange  
  
Max candidates (changed from Candidates limit) is the maximum number of auto-orders that can be active in this Auto-Trade. If checkbox “Always Max” is checked then a new auto-order can be added if a previous auto-order is completed.  
If checkbox “Repeat instrument” is checked then a new auto-order, with an instrument that already has been used, can be used again. But two auto-orders with the same instruments can not be active at the same time.

**Candidates / IBScanner**(Change name from MarketScanner to IBScanner)

IBScanner is a functionality which produce a list of instruments. It can find instrument in almost every part of the world which have behaved in a specific way.   
To produce this, you select:

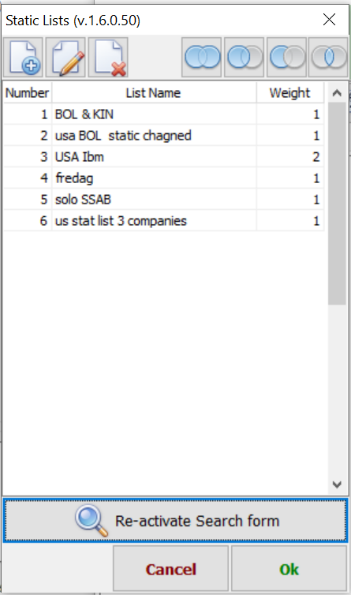
1. A region and type. (Example: “Europe Stocks”)  
2. A location. (Example: “Sweden (SFB)”)  
3. A parameter (Example: “Top % Gainers Since Open”)  
4. You may add extra filters (Example: “Revenue over 200 000 000”)

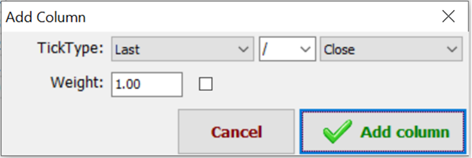
****

Erase “Exclude instruments”. It’s not necessary. It can be done with a Static list added to candidate list with  function.

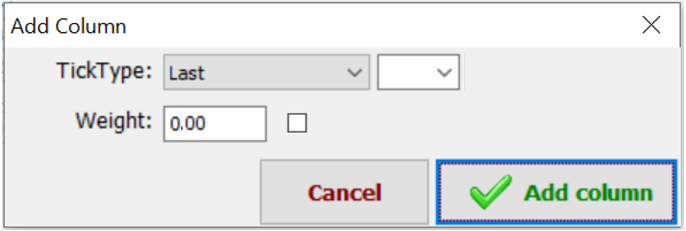
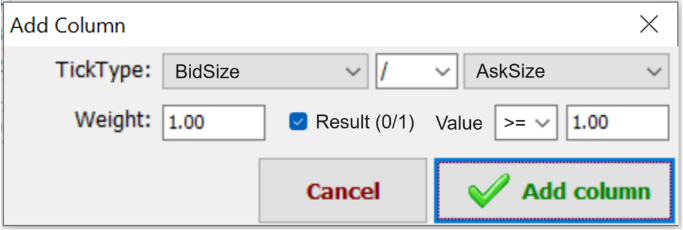
**En bild som visar text

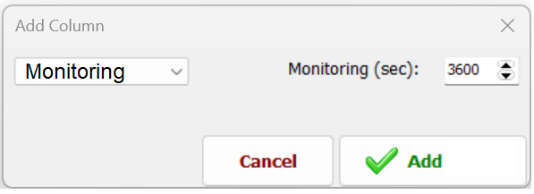
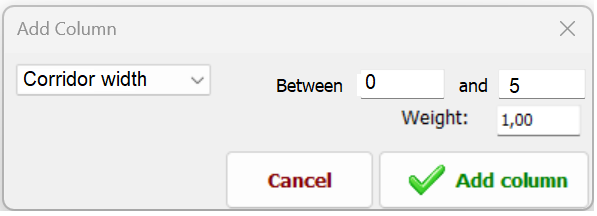
Automatiskt genererad beskrivning**En bild som visar text

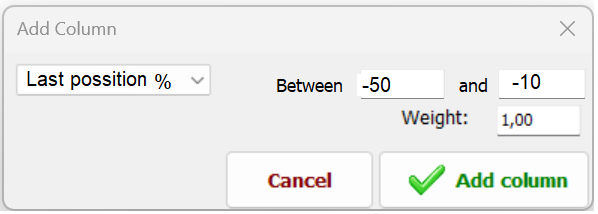
Automatiskt genererad beskrivning**Candidates / Static list**This is user produced lists. A static list includes a number of instruments which can be used when collecting candidates.  
First open all static lists: Editing or adding a new list with  :  
  
  
  
  
  
  
  
  
  
  
  
**Adding functions**When adding a list to the candidate process there is four   
types of adding function:   
(1) Add all instruments/values in the statis list to candidates,   
(2) Add only instruments/values from static list which already exists in candidates.  
(3) Erase candidates which exists in static list.  
(4) Keep only candidates which exists in both lists.

**Candidates / Add columns**You can add new columns to the Candidates process and give them weight value. If a comparing sign is chosen the two tick-types values are composed and shown in the list.   
Example: If Last is 102 and Close is 100 and comparing sign is “/” and Weight is 1.00 the value is (102/100)\*1.00=1,02 witch will be shown in the column and added to “Ranking sum”.

If the checkbox is checked then the result value, added to the “Ranking sum”, is either 0 or 1.  
If the second comparation is TRUE the value added to “Ranking sum” is 1\*weight.

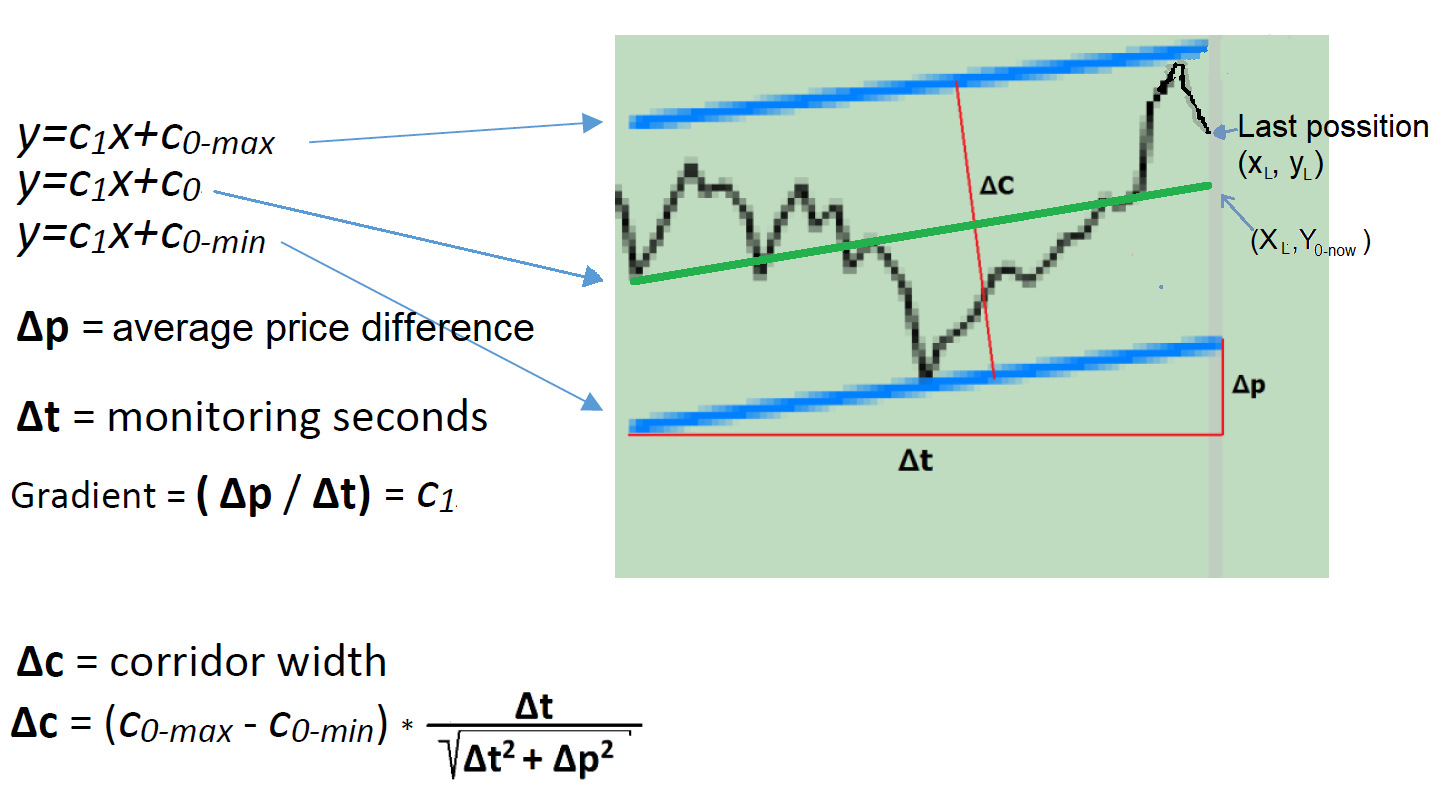
****  
  
  
  
  
  
If a comparing sign is NOT chosen only the first tick-type value is shown in the column and the weight is set to 0.00 (default)

**Candidates Functions**

****  
  
 **Gradient and corridor position and corridor width**All three of these functions (*Gradient, Corridor Width* and *Last Position*) return 1 or 0.  
1 if the condition is true otherwise 0.

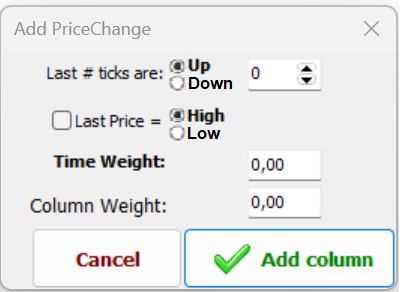
To calculate the line and gradient use the ***method of least squares*** to find a line that approximate all LAST-price value the last monitoring seconds (from condition dialog) (xn,yn) where x=*time* and y=*price.* When you have found the function for the line *y=c1x+c0* the Gradient = *c1*

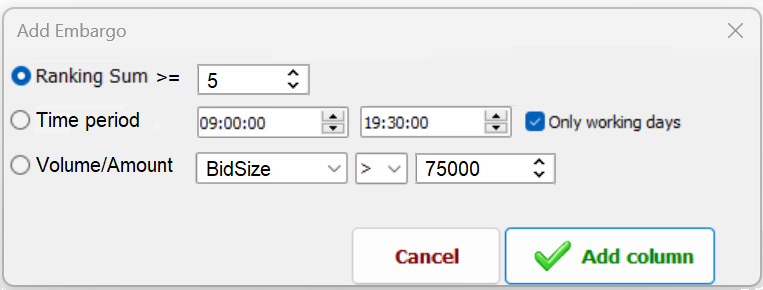
******Find ***C0-max*** by putting in the LAST-price value that gives the   
highest result of *y-c1x.* Similar with ***C0-min***



**Last position**  
Y0-now = C1XL+C0   
Last position is in percent (%) between -100 and 100. When touching the top-line it is 100% and when touching the bottom-line it is -100%.   
if YL is > Y0-now then   
  
*Last position*= 100\*(YL- Y0-now)/(C0-max - Co)  
  
else   
  
*Last position*= 100\*(YL- Y0-now)/( Co - C0-min).

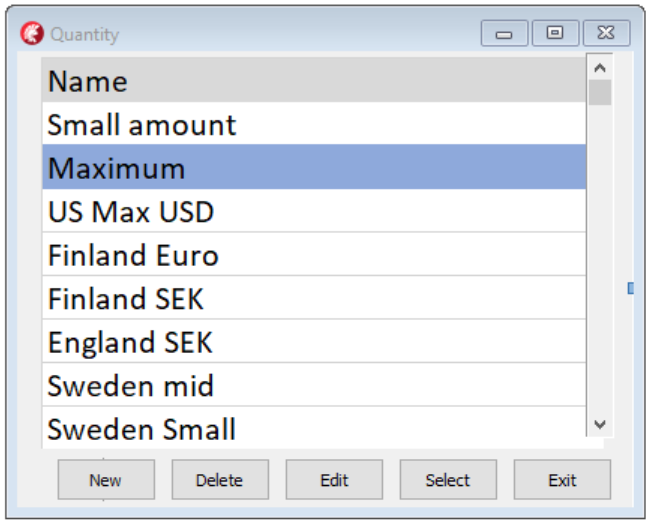
**Candidates /** **Price change**  
The “Price change” function returns a value of how fast the price is changing iof the conditions is fulfilled. If mother order is a BUY order, the radio buttons shall be “Up” and “High”. If mother order is a SELL order the radio buttons shall be “Down” and “Low”.   
The value this function return is 0 if not all conditions are true.   
The first condition is that the last # price ticks are “Up” for BUY orders or down if mother order is SELL.  
The second condition is relevant if the checkbox is checked and then the last price must be the highest of today LASTPRICE=HIGH (BUY) or lowest today LASTPRICE=LOW (SELL).  
  
The column value is set to **Weight/X** where **X** is the time in seconds between the first LAST-PRICE-tick and the last LAST-PRICE-tick of   
these #. X cant be less than 1.  
  
Example for a BUY-order where #=3 and checkbox is checked:   
If last price is 34,10 at 09:45:63:19   
 34,20 at 09:45:64:23   
 34,30 at 09:45:64:41   
and 34,30 is the highest price this day, X is 1,22 seconds and the column value will be set to 10,00/1,22 = 8,1967



**Candidates / Embargo**  
A candidate process must have at least one embargo. All embargoes must be true before the auto-order can be activated. An embargo can be:  
  
1: ***Ranking Sum***.   
 The ranking sum for an instrument in the list must be greater or equal to the   
 value set in the field.  
2: ***Time period***.   
 The current time must be in the specified period.  
3: ***Volume/Amount***.   
 Compares the value of the volume with the specified amount. The size reseived   
 with the specified tick type will be multiplied with the “last price” and then   
 compared with specified amount.  
 *Example: If you receive a “BidSize” of 810 for a instrument in the list and the   
 “last price” for this instrument is 100 you will have a* *volume value of 81000.  
 If* *the compare sign is “>” and the specified amount is 75000 then the   
 embargo is TRUE.*  
 For embargo Volume/Amount:  
 1. The volume value is shown in the column list. In this case 81000.  
 2. For embargo Volume/Amount the possible tick types is:  
 [“AskSize”, “BidSize”, “LastSize”, “Volume”, “AngVolume”]  
 3. The compare sign can be “>” or “<”.

**Quantity**

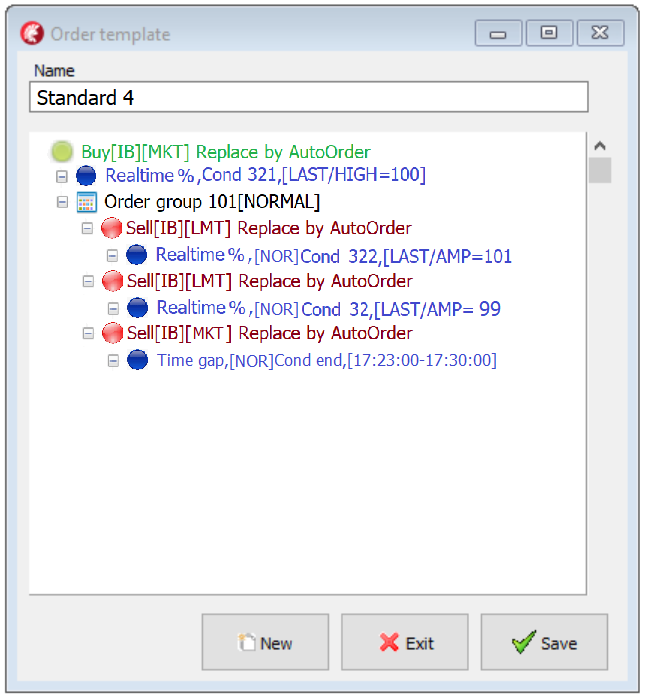
Each AutoTrade has a quantity. It decides how big the amount is to spend in total and how big the amount is for each order. It’s also saying what currency is used.  
If an instrument is ordered in another currency the amount must be converted.  
The total amount is the sum of all activates mother orders which uses this specific Quantity.



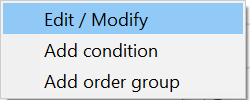
Clicking on Edit will open this dialog.

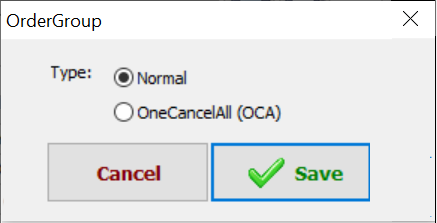
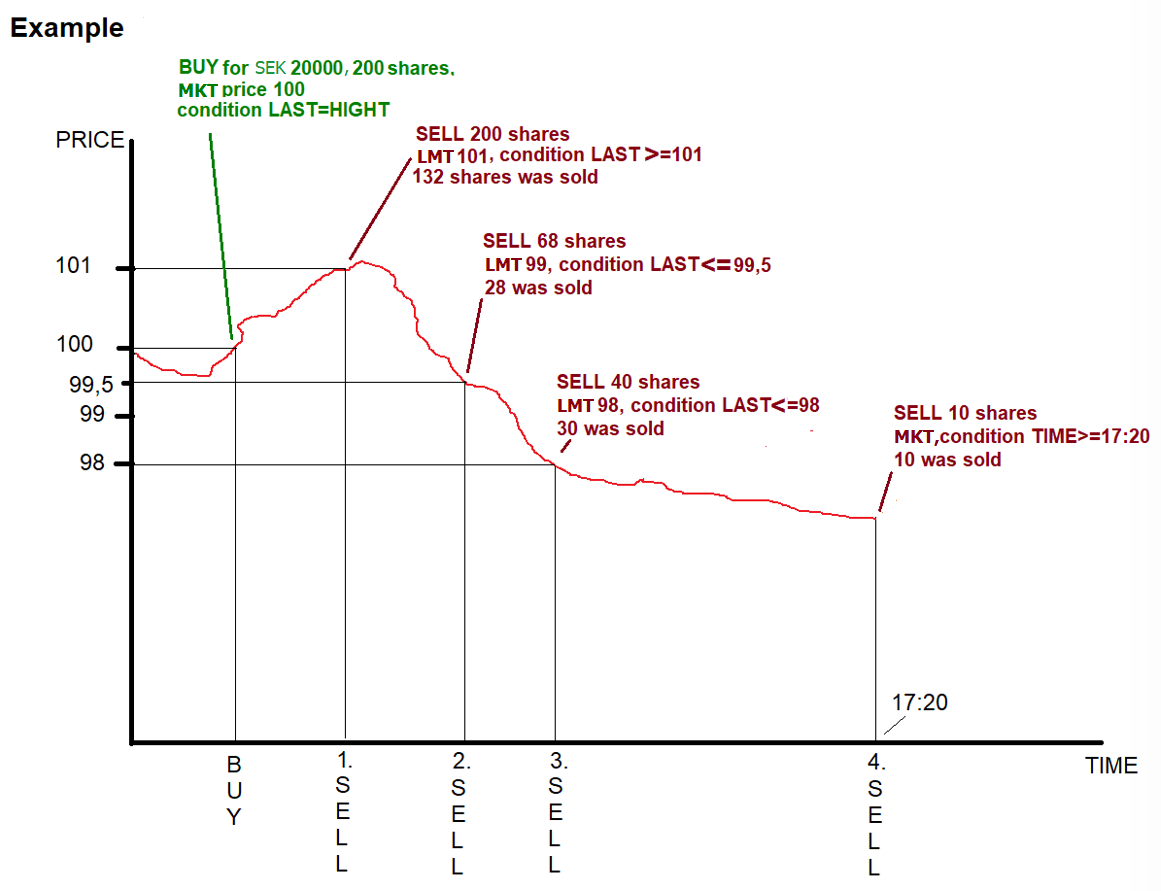
En bild som visar text

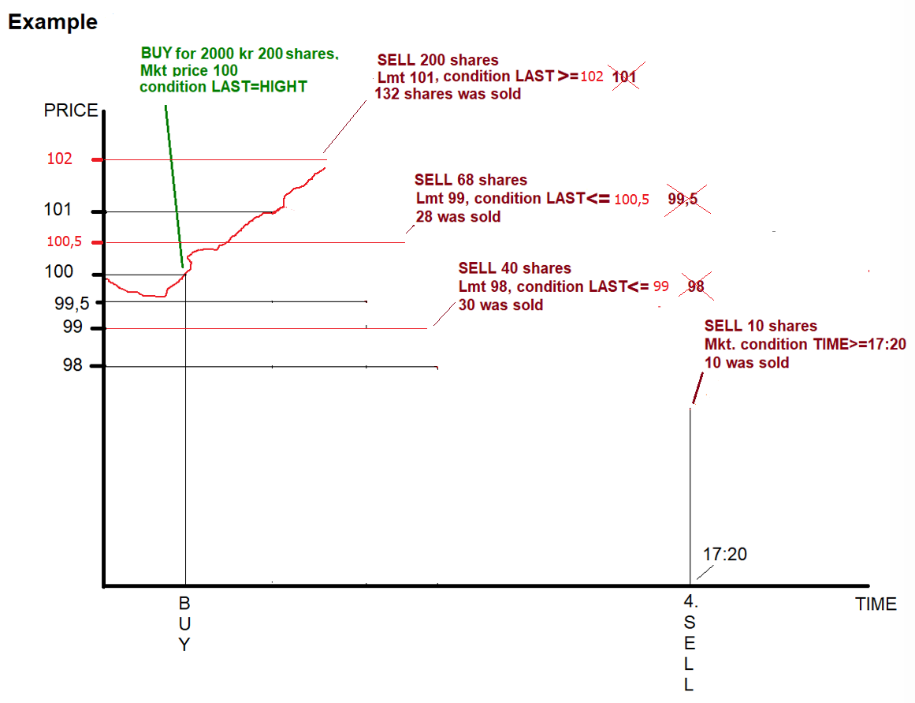
Automatiskt genererad beskrivning

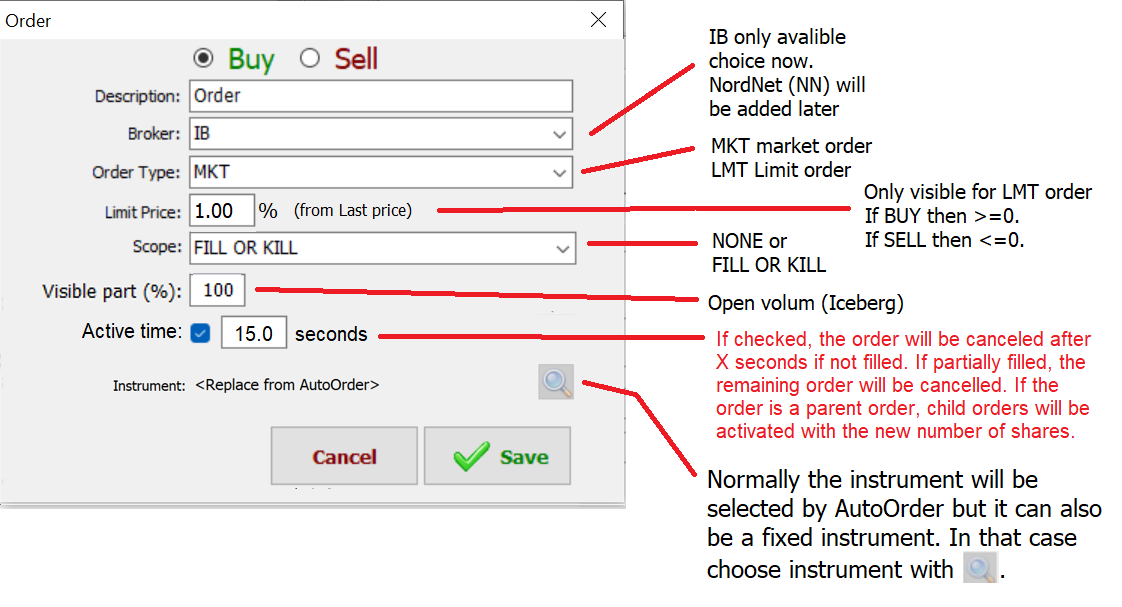
En bild som visar text

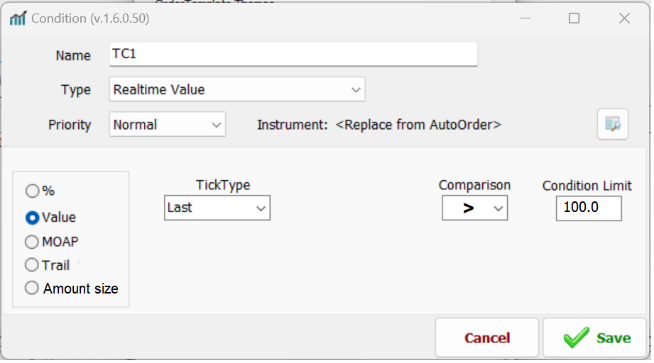
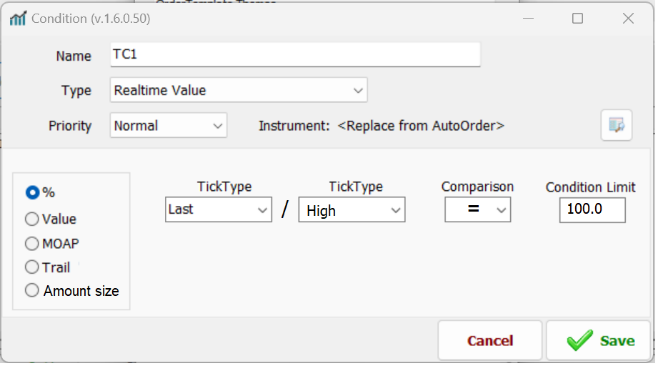
Automatiskt genererad beskrivningEn bild som visar text

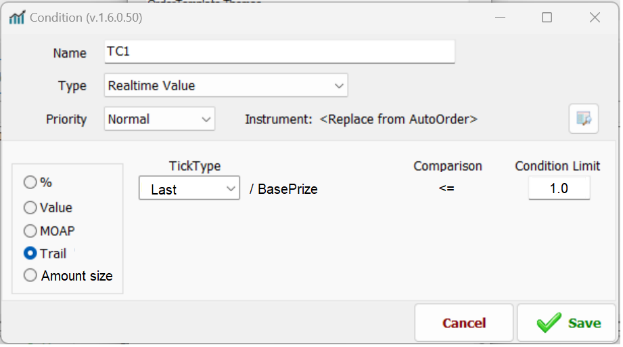
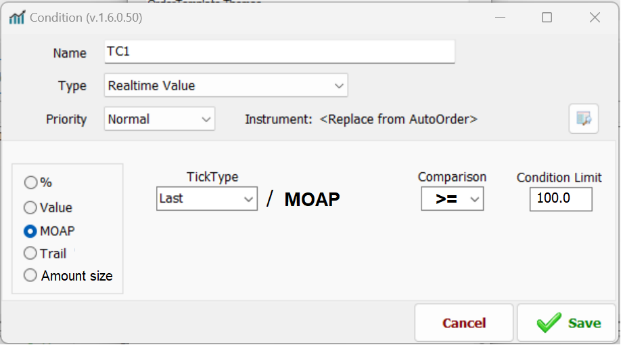
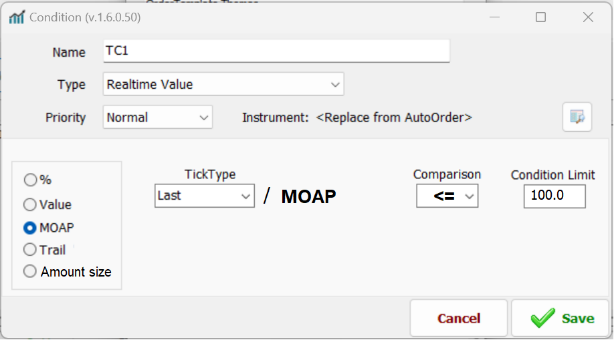
Automatiskt genererad beskrivning**Order templates**  
  
An order template consist normally of a   
mother order and an order group with   
child orders. Normally the mother order   
is a buy order and the child orders are   
normally sell orders. But it can be the   
opposite.  
  
When clicking on the order-template   
button on main form you open   
the order-template-list.  
You can edit an order template with  
New or Edit -button or with a double click.  
  
  
  
When starting a new template,   
you open a window tree structure   
including only one mother order.  
When the mother order is selected   
the pop-up menu includes look like this:  
  
  
  
For “Edit / Modify” se Order.  
For “Add condition” se Condition.  
If an order group already is added   
then “Add order group” is disabled.  
Next step is to “Add an order group” .   
Se Order group.

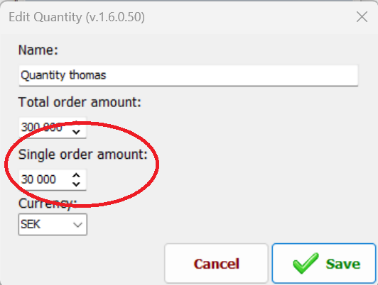
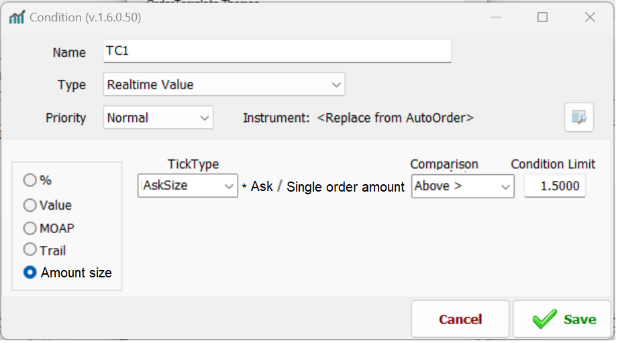
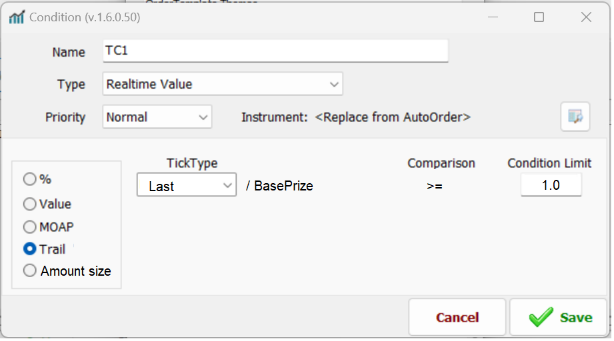
**Order Group**  
An order group decides how its child orders shall behave.  
  
  
  
*Order Group “****Normal****”*This order group type activates and deactivates different orders depending on what happens to the price of the instrument in question.  
Example:  
Mother order buys 200 shares for SEK 20000 (MOAP=100).  
The first child orders condition that becomes true will send its order to TWS.  
If another orders condition becomes true the first order will be canceled and this order will be send to TSW with the number of shares which is remaining.  
If a third orders condition becomes true the second order will be canceled and this order will be sent to TSW with the number of shares which not sold. Etc  
  
If an order is a limit order it is not certain that you will succeed in selling / buying all the shares. Example:  
   
In the same way, you do not know what price you have to pay for the mother order. Therefore, it is important to calculate the Mother Order Average Price (**MOAP**).   
Example:   
If you want to buy 100 shares and manage to buy 700 for SEK 100 and 300 for SEK 101, the average price will be   
(700 \* 100 + 300 \* 101) / 1000 = SEK 100.30

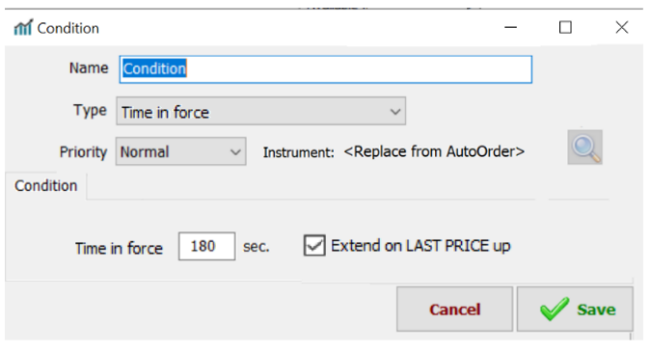
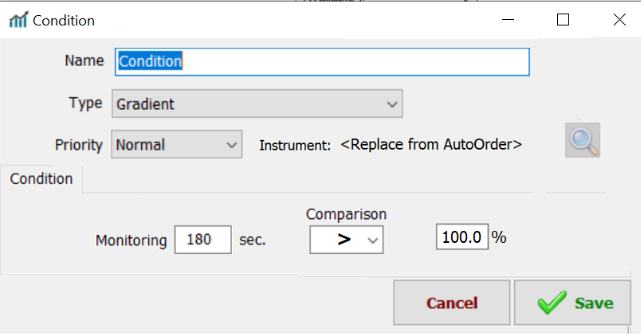
We don’t want to sell instruments and the buy the same back again because they still in candidate list with a value over the limit value.  
So, if sale order becomes active and the instrument is still in candidate list with a value over the limit value, do recalculate all limit values for all orders in the order group.  
  
*Order Group “****OneCancelAll****”*The first orders condition that becomes true will send its order to TWS.  
The other orders in the group are canceled.

**Order**  
  
An order is either a BUY or a SELL.  
The order type is either a market order (MKT) or a limit order (LMT).  
If it is a limit order (LMT) it has set a limit. The limit is set when the   
order is executed and is calculated as a % from the last price at that moment.  
  
*Example:   
If the limit order is a BUY-order and is executed when last price is 100 and  
then limit percent is 0,5 then the limit is 100,5.  
If the limit order is a SELL-order and is executed when last price is 100 and  
then limit percent is -0,5 then the limit is 99,5.*  
Scope is NONE or FILL OR KILL.   
NONE means I receive what the market can give me.  
FILL OR KILL means: I will receive everything or nothing. If the market can’t fill the order, it will, otherwise I receive nothing.   
If visible part is less the 100%, only a part of the order will be visible on the market.   
  
Active time is, if it’s checked, the maximum number of seconds the order is active. When the time has passed and the order is not filled, the order (or the remaining order) is cancelled.   
If the order is partially filled and the order is a mother order, child orders are activated with the net new number of shares

**Condition**  
  
A Conditions has a Description, a Type, a Priority, an Instrument.  
A Condition is either false or true.  
There can be several Conditions connected to an order.  
All Conditions together decides if the order shall be executed or not.  
What decides if the order shall be executed or not are two things. It is if a Condition is true or not and its priority.  
The instrument which the condition uses is normally set by the Auto Order, but it can also, as an option, be set manually.  
TickType: **Last, High, Low, Bid, Ask, Bidsize, Asksize, Volume,   
 Open, Close** and **NotSet**Comparison: **=** (Equal) **>** (Above), **>=** (Above or Equal),   
 **<** (Below), **<=** (Below or Equal)  
**Condition/Priority**  
  
If Priority is **Normal** then this Condition together with all other condition with priority **Normal** must be true.  
If Priority is **Exclusive** then it is enough if this Condition is true.  
If Priority is **Veto** then this Condition must be true.  
  
**Condition/Type**  
The types can be **Realtime %**, **Realtime value**, **Realtime MOAP**, **Realtime Trail,   
Realtime Amount,** **Time gap**, **Gradient** or **Time in force**.  
  
  
  
**Realtime value %**  
This condition divides two TickTypes with   
each other and compare with a condition   
limit.  
*Example: This condition becomes true if the   
last price is the highest price today.   
Value%=100\*(Last/High)*  
  
  
  
  
**Realtime value**   
This condition becomes true a ticktype filles   
the comparison condition  
*Example: This condition becomes true if the   
last price for Volvo Cars AB-B   
passes 100.*

****Realtime value MOAP**  
It is only useful for child orders.  
This condition compares the Last price with the *mother order average price* (MOAP).  
  
*Example 1: It becomes true if the   
last price 1% higher than MOAP.  
Condition value=100\*(Last price - MOAP)/MOAP*  
  
  
*Example 2: It becomes true if the   
last price is 1% lower than MOAP.  
Condition value=100\*(Last price - MOAP)/MOAP*  
  
  
  
  
**Realtime Trail**Realtime Trail is a dynamic stop loss function. It means that if you make an investment, and the price does not go the way you want, you shall leave the investment at a specific limit (and maybe lose some money) and this limit can change. This function is there to limit any loss.  
It can only be used for child-orders  
  
*Mother-order is a BUY-order*  
This condition compares the Last price with a BasePrice. The BasePrice is at start *mother order average price* (MOAP), but if LAST PRICE goes up and the mother order is a BUY-order, then the BasePrice goes up with the same amount. If LAST PRICE goes down, it does not affect BasePrice. If LAST PRICE reaches the limit the condition becomes TRUE.  
The comparison sign is locked to **<=**.  
  
*Example: It becomes true if the   
last price 1% lower than* BasePrice*.  
Condition value=100\*(Last price - BasePrice)/* *BasePrice  
If we buy instrument X for stock price 100. At this   
moment the condition becomes true when   
LAST PRICE=99. If LAST PRICE goes up to 100,20 then  
then BasePrice is 100,20 and condition becomes true   
when LAST PRICE<=99,20.*

*Mother-order is a SELL-order*  
This condition compares the Last price with a BasePrice. The BasePrice is at start *mother order average price* (MOAP), but if LAST PRICE goes down and the mother order is a SELL-order, then the BasePrice goes down with the same amount. If LAST PRICE goes up, it does not affect BasePrice. If LAST PRICE reaches the limit the condition becomes TRUE.  
The comparison sign is locked to **>=**.  
 *Example: It becomes true if the   
last price 1% higher than* BasePrice*.*  *Condition value=100\*(Last price - BasePrice)/* *BasePrice  
If we sell instrument X for stock price 100. At this   
moment the condition becomes true when   
LAST PRICE=101. If LAST PRICE goes down to 99,80 then  
then BasePrice is 99,80 and condition becomes true   
when LAST PRICE>=100,80.*  
  
**Realtime Amount size**Since some order types are not available on all   
markets, such as "Kill or Fill", it must be possible   
to see the quantity (amount) available for   
purchase (ASK) or for sale (BID). But also, what   
the size of the last deal was.  
Therefore, it should be possible to compare the   
amount we want to use on a purchase with the   
amounts for BID, ASK and LAST.   
  
The only TickTypes available in this condition   
type is AskSize, BidSize and LastSizeE.   
  
The comparison value is for AskSize: (AskSize\*Ask)/”Single order amount”  
for BidSize: (BidSize\*Bid)/”Single order amount” and   
for LastSize: (LastSize\*Last)/”Single order amount”.  
  
“Single order amount” is taken from AutoTrade/Quantity  
  
The quantity is always in SEK so if the actual instruments  
currency is another, it have to be recalculated.  
  
  
  
  
  
  
**Explanation for selling shares you don't own.**When you think the price of a specific stock is going to drop.   
It is possible to sell these shares even if you do not own them. It's called being short. In these cases, you borrow these shares from a bank or a broker (and pay some interest) and sell them.   
At a certain point, you buy them back and give them back to the bank or broker.  
If you were right, you make some money.

**En bild som visar text

Automatiskt genererad beskrivningCondition/Time gap**  
This condition compares the current time  
with gap time.  
*In this example the condition is true   
between 17:23 and 17:30.* **Condition/Gradient**  
This condition compares the last price   
for an instrument with the price   
X seconds ago.   
In this example the condition is true   
if the price is higher than it was   
180 seconds ago.  
X0=Price now (Last Price)  
X1=Price 180 sec ago.  
  
Value%=100\*(X0/X1)  
 **Condition/Time in force**  
This condition becomes true X seconds   
after mother order was filled.  
If the checkbox **“*Extend on LAST PRICE up*”**  
is checked, a new ***“Time in force”***-period  
will start every time the instruments   
LAST PRICE goes up.  
It is only useful for child orders.  
  
*Example 1: The Checkbox* ***“Extend on LAST PRICE up”*** *is NOT checked and time in force is 175. Mother order is filled at 09:12:41. Condition becomes true at 09:15:36.  
  
Example 2: The Checkbox* ***“Extend on LAST PRICE up”*** *is checked and time in force is 175. Mother order is filled at 09:12:41. Last price increase at 09:14:11. Condition becomes true at 09:17:06. (unless LAST PRICE goes up before that)  
  
Example 3: The Checkbox* ***“Extend on LAST PRICE up”*** *is checked and time in force is 240. Mother order is filled at 09:17:55. Condition will be true at 09:21:55. Last price increase at 09:20:11. Condition will be true at 09:24:11.  
Last price increase again at 09:23:23. Condition will be true at 09:27:23.*

**Orders set**

An order set is containing one mother-order and a group of child-orders.  
It is created from an order template.

Example:   
In the mother-order I buy 1000 shares of IBM for SEK 100 each.  
Total cost SEK 100.000. Mother order average price (MOAP) is SEK 100 /share.  
  
I then like to sell them all if they go up 1% with a limit price of 1%  
If they go down 1% I likes to sell them with no limit (market-order)  
If nothing happens I like to sell them at 17:20 (before the market close)  
So it will be one mother-order and 3 child-order.

**Instrument search**

En bild som visar text, skärmbild, inomhus

Automatiskt genererad beskrivning