GFT
Global Forex Trading | Division of Global Futures & Forex, Ltd.

4760 E. Fulton, Suite 201 Ada, Michigan USA
phone | 616.956.9273
toll free | 800.465.4373
fax | 616.974.3682

DealBook 360 CTL Change Guide 2.11.4.12.9

The guide describes the changes in CTL and Chart Studio from the previous version of Dealbook 360 (2.10.927.36) to the newest version (2.11.4.12.9).

Trailing Stops

It is now possible to issue trailing stop orders via automated strategies. A trailing stop is a stop order that follows the market bid price by *x* number of pips/points. For example if the market is at 10, and you place a trailing stop Sell at 7 with a pip trail of 3, when the market moves up to 11 your stop will automatically adjust to 8. When the market moves to 12 your stop will adjust to 9. The trailing stop can only adjust automatically in one direction, so if the market moves back down to 11, your trailing stop will be locked in at 9. If the market eventually moves down to 9, the stop would turn into a market order and fill (in this example it would sell). Note that due to the dynamic nature of the trailing stop, backtesting using a trailing stop does not provide any degree of accuracy.

The inputs for the new trailing stop functions are:
trailing_stop_sell(lots, piptrail);
trailing_stop_buy(lots, piptrail);

Example usage:

```
strategy TStop_SMA_Example;
input lots = 1,
     piptrail = 20,
     smaperiod = 30;
vars sma1(series);
begin
sma1 := SMA(close, smaperiod);

if positionvolume() = 0 and crossup(close, sma1) then begin
  buy(lots);
  trailing_stop_sell(lots,piptrail);
end;
if positionvolume() = 0 and crossdown(close, sma1) then begin
  sell(lots);
  trailing_stop_buy(lots,piptrail);
end;

end.
```

Canceling Stops and Limits

In this version of CTL you can programmatically cancel stops and limits. Please note that you cannot issue an OCO stop and limit in this version, and manual order placement (e.g., orders not placed from the strategy) will interfere with the strategy logic. For this reason it is suggested to use either only a Stop or Limit as one loss or profit target and use other logic to exit the trade and cancel the order.

The cancellation functions are:

Cancel_all_limits()

Cancel_all_stops()

Cancel_buy_limits()

Cancel_buy_stops()

Cancel_sell_limits()

Cancel_sell_stops()

Example usage:

```
strategy EURUSD_SMA_Example;

input lots = 1,

     account_lotsize = 100000,

     profitFPL = 200,

     smaperiod = 30;

vars sma1(series), i(number);

begin

sma1 := SMA(close, smaperiod);
```

```
i := back(close);
if i <= front(close) + 1 then return;
if (positionvolume() = 0) and crossup(close, sma1) then begin
  buy(lots);
  //placing a stop sell at the current close price minus 20 pips
  stop_sell(lots * account_lotsize, close[i] - 0.0020);
end;
if (positionvolume() = 0) and crossdown(close, sma1) then begin
  sell(lots);
  //placing a stop buy at the current close price plus 20 pips
  stop_buy(lots * account_lotsize, close[i] + 0.0020);
end;
//exiting based on fpl reaching the "profitFPL" input
if fpl() >= profitFPL then begin
if long() then exitlong();
if short() then exitshort();
cancel_all_stops();
end;
end.
```

Dynamic Line Styles and Colors

It is now possible to dynamically adjust the thickness, style, and color of indicator lines displayed in Dealbook 360 using the “setlineproperties()” function.

setlineproperties(lineName, color, style, thickness)

Example usage:

```
indicator changecolor_example;
draw line("Line");
vars i(number);
begin

line := SMA(close, 10);
i := back(close);
if i <= front(close) + 11 then return;
  if line[i] > line[i - 1] then setlineproperties(line, green, default_style, 1);
  if line[i] < line[i - 1] then setlineproperties(line, red, default_style, 1);
end.
```

Account Controls

If you would like your module to only be tradable for specific login names or have an expiration date, you can do so with the “accounts” and “expiration” declarations. Example usage:

```
indicator account_control;

accounts "MyAccountLoginName", "MyOtherAccountLoginName";

expiration "07302008"; //mmddyyyy

input period = 10;

draw line("Line");

begin

Line := sma(close, 10);

end.
```