

Notes for Coders

I was learning to code at the same time I was writing the original MPTM and the code is all over the place. Making additions/alterations to it can be a nightmare.

I have learned a lot in the succeeding decade. This code is tighter, should be much easier to follow and simpler to edit.

Void OnTimer

The first code block checks that EA's are enabled and aborts the function if not, along with a check to see if the user has Rene's news monitor EA warning of an impending news event.

The second code block starts with this comment:

“//In case a global order close/deletion failed” and mops up any global/shirt protection order close/deletion failures.

The third code block starts with the comment, “//Build a picture of the position” and calls my usual CountOpenTrades() function.

The forth code block starts with the comment, “//Management times” and aborts the function if MPTM should be asleep.

The fifth code block starts with the comment, “//Any trades to manage?” and calls the DoTradeManagement() function if so.

void CountOpenTrades()

This function cycles through the open trades on the platform and:

- Works out whether MPTM should be managing the trade (AreWeManagingThisTrade(ticket)).
- Calls the individual trade closure function (if (ShouldThisTradeBeClosed(ticket))
- If MPTM is managing the individual trade and it has not been closed, stores the OrderTicket() in the FifoTicket array for easy global/shirt protection closure, and for the individual trade management function DoTradeManagement().
- Makes a tally of the cash and pips UPL if the user is using global closure or shirt protection.
- If the user is allowing hedging:
 - Calls the function that sends hedge trades. Look for the comment “//Does this trade need hedging?”
 - Calls the function that closes hedge trades. Look for the comment, “//If the trade is a hedge trade, then can it be closed?”

void DoTradeManagement()

Here is where MPTM examines each of the trades it is managing and calls the functions that manipulate the stop loss and take profit fields. It is called from the bottom of the OnTimer() function, after all the global and hedging functions have been dealt with.

It will be easy to add extra functionality here. There are four steps::

1. Add your extern variables for the user.
2. Add code to DisplayUserFeedback() to remind the user of his choice.
3. Code a function that performs the task you want adding. Declare the function with 'ticket' as the parameter i.e.
 - bool DoSomething(int ticket).
 - Add the BetterOrderSelect check at the top of each function just in case the trade has been closed by another source.
4. Add a call to that function from within DoTradeManagement() i.e.
 - DoSomething(ticket);

Hedging

I have moved away from the 'panic' hedging employed by my original MPTM. Instead I have based the decision to open/close hedge trades on my favourite SHF indis, plus an extra feature for the long-term traders wanting a layer of protection over the weekend.

There are two parent functions which pass calls to functions that make the hedging decision, both called from within CountOpenTrades():

- bool DoesThisTradeNeedHedging(int ticket) calls the functions that decide whether an individual trade needs hedging. The functions are:
 - bool UsingBobMaForHedging(int ticket)
 - bool UsingHgiToOpenHedge(int ticket)
 - bool UsingSuperSlopeToOpenHedge(int ticket)
 - bool TimeToSendProtectiveHedge(int ticket)
- bool DoesTheHedgeNeedClosing(int ticket) calls the functions that decide whether an individual hedge trade needs closing. The functions are:
 - bool DoesTheHedgeNeedClosing(int ticket)
 - bool UsingBobMaForHedgeClosing(int ticket)
 - bool UsingHgiToCloseHedge(int ticket)
 - bool TimeToCloseProtectiveHedge(int ticket)

The sub-functions all return 'true' if a trade should be hedged or a hedge trade closed, else they return 'false' to the parent function. The parent function then performs the order send/close.

This makes it easy to add your own favourite indicators. There are four steps:

1. Add your extern variables for the user.
2. Add code to DisplayUserFeedback() to remind the user of his choice.
3. Code the function that takes the decision to hedge and add a call to it to DoesThisTradeNeedHedging(int ticket).
4. Code the function that takes the decision to close a hedge trade and add a call to it to DoesTheHedgeNeedClosing(int ticket) .

Indicator-based closures

I decided to add individual trade closure by my favourite SHF indicators but keep this separate from hedging. The variable names are the same apart from the last letter, which is 'H' when it is hedging and 'C' when it is for closure i.e. SsTimeFrameH and

SsTimeFrameC,

bool ShouldThisTradeBeClosed(int ticket) is the parent function and calls the sub-functions that decide whether the trade should close or not. You should have got the message by now and can work out for yourselves the usual four steps involved in adding your own indi's to the list.