\*\*LIVE ORDER PLACEMENT\*\*

OFFERS TABLE

ENTER TRADE

sOfferID = "";

iAmount = Convert.ToInt32(lotSize);

sBuySell = "";

CreateTrueMarketOrder(sOfferID, sAccountID, iAmount, sBuySell);

audcadJustSold = false;

audcadReEnter = true;

acctPos = acctPos + 1;

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": LONG: " + offerTableRow.Instrument + Environment.NewLine); }));

EXIT TRADE

sOfferID = "";

iAmount = Convert.ToInt32(audcadPosLot);

sBuySell = "";

CreateTrueMarketOrder(sOfferID, sAccountID, iAmount, sBuySell);

audcadJustSold = true;

audcadReEnter = true;

netPandL = netPandL + Convert.ToDouble(moveString);

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": CLOSED:" + offerTableRow.Instrument + " For:" + audcadPL.Text + Environment.NewLine); }));

CLOSED TRADES TABLE

audcadJustSold = true;

audcadPosPL = 0;

audcadEntryPrice = 0;

audcadSideString = "";

acctPos = acctPos - 1;

audcadBuyLongCheck = 0;

audcadPosLot = 0;

this.Invoke(new MethodInvoker(delegate { audcadPosSize.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audcadPL.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audcadSide.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audcadEntry.Text = "--"; }));

\*\*SIMULATION TRADING\*\*

FULL OFFERS TABLE CLASS

// If triggered pair is "AUD/USD"

if (offerTableRow.Instrument == "AUD/USD")

{

// Fill in Bid, Ask and Spreads

this.Invoke(new MethodInvoker(delegate { audusdBid.Text = bids; }));

this.Invoke(new MethodInvoker(delegate { audusdAsk.Text = asks; }));

this.Invoke(new MethodInvoker(delegate { audusdSpread.Text = spread; }));

// Fill in past price if ask price has a value

if (audusdPastBox.Text == "--" && audusdAsk.Text != "--")

{

try

{

this.Invoke(new MethodInvoker(delegate { audusdPastBox.Text = audusdAsk.Text; }));

this.Invoke(new MethodInvoker(delegate { audusdPast = Convert.ToDouble(audusdAsk.Text); }));

}

catch (Exception fillErr)

{

Console.WriteLine(fillErr);

}

}

// If "Past" TextBox is not null

if (audusdPastBox.Text != "--")

{

// Move calculations

double moveCalc = (ask - audusdPast);

double moveConvert = moveCalc \* 10000;

audusdMove = Convert.ToDecimal(moveCalc);

buyCheck = moveCalc;

try

{

// \*\*ENTER POSITION CHECKS\*\*

// LONG: Position entry check

if (moveCalc > 0)

{

string move = Convert.ToString(moveConvert).Substring(0, 4);

// Change the text in move TextBox for positive's

audusdMoveBox.BackColor = System.Drawing.Color.Blue;

this.Invoke(new MethodInvoker(delegate { audusdMoveBox.Text = move; }));

// Check if a Buy Long is triggered

if (buyCheck >= Convert.ToDouble(buyLong) & startSpread < maxSpread && audusdEntry.Text == "--" && audusdReEnter == false)

{

audusdSideString = "Long";

audusdEntryPrice = Convert.ToDecimal(ask);

audusdReEnter = true;

acctPos = acctPos + 1;

audusdPosLot = lotSize;

this.Invoke(new MethodInvoker(delegate { audusdEntry.Text = asks; }));

this.Invoke(new MethodInvoker(delegate { audusdSide.Text = "Long"; }));

this.Invoke(new MethodInvoker(delegate { audusdPosSize.Text = Convert.ToString(lotSize); }));

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": LONG: " + offerTableRow.Instrument + " @ " + offerTableRow.Ask + Environment.NewLine); }));

}

}

// SHORT: Position entry check

else if (moveCalc < 0)

{

string move = Convert.ToString(moveConvert).Substring(0, 5);

// Change the text in move TextBox for negative's

audusdMoveBox.BackColor = System.Drawing.Color.Red;

this.Invoke(new MethodInvoker(delegate { audusdMoveBox.Text = "(" + move + ")"; }));

// Check if a Sell Short is triggered

if (buyCheck <= Convert.ToDouble(sellShort) & startSpread < maxSpread && audusdEntry.Text == "--" && audusdReEnter == false)

{

audusdEntryPrice = Convert.ToDecimal(bid);

audusdSideString = "Short";

audusdReEnter = true;

acctPos = acctPos + 1;

audusdPosLot = lotSize;

this.Invoke(new MethodInvoker(delegate { audusdEntry.Text = bids; }));

this.Invoke(new MethodInvoker(delegate { audusdSide.Text = "Short"; }));

this.Invoke(new MethodInvoker(delegate { audusdPosSize.Text = Convert.ToString(lotSize); }));

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": SHORT: " + offerTableRow.Instrument + " @ " + offerTableRow.Bid + Environment.NewLine); }));

}

}

else

{

this.Invoke(new MethodInvoker(delegate { audusdMoveBox.Text = "0"; }));

this.Invoke(new MethodInvoker(delegate { audusdMoveBox.BackColor = Color.FromArgb(60, 60, 60); }));

}

}

catch (Exception posErr)

{

Console.WriteLine(posErr);

}

// LONG: Position Manager

if (audusdSideString == "Long")

{

// Subtract entry price from bid price

audusdBuyLongCheck = (bidd - audusdEntryPrice);

audusdPosMoveDec = (bidd - audusdEntryPrice) \* audusdPosLot;

// P/L or Position move string conversion

audusdPosPL = Convert.ToDouble(audusdPosMoveDec);

string moveString = Convert.ToString(audusdPosPL);

// Changes to P/L

if (audusdPosMoveDec >= 0)

{

audusdPL.BackColor = System.Drawing.Color.Blue;

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "$" + moveString; }));

}

else if (audusdPosMoveDec < 0)

{

audusdPL.BackColor = System.Drawing.Color.Red;

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "$" + moveString; }));

}

// LONG: Sell Trigger

if (audusdBuyLongCheck >= goalLong)

{

netPandL = netPandL + Convert.ToDouble(moveString);

audusdEntryPrice = 0;

audusdPosPL = 0;

audusdSideString = "";

acctPos = acctPos - 1;

audusdBuyLongCheck = 0;

audusdPosLot = 0;

this.Invoke(new MethodInvoker(delegate { audusdPosSize.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": CLOSED:" + offerTableRow.Instrument + " @ " + offerTableRow.Bid + " For:" + audusdPL.Text + Environment.NewLine); }));

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdSide.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdEntry.Text = "--"; }));

}

// LONG: Stop Loss

if (audusdBuyLongCheck <= (stopLoss \* -1))

{

netPandL = netPandL + Convert.ToDouble(moveString);

audusdEntryPrice = 0;

audusdSideString = "";

acctPos = acctPos - 1;

audusdBuyLongCheck = 0;

audusdPosLot = 0;

this.Invoke(new MethodInvoker(delegate { audusdPosSize.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": STOP LOSS: " + offerTableRow.Instrument + " @ " + offerTableRow.Bid + " For:" + audusdPL.Text + Environment.NewLine); }));

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdSide.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdEntry.Text = "--"; }));

}

}

// SHORT: Position Manager

else if (audusdSideString == "Short")

{

// Subtract entry price from bid price

audusdShortSaleCheck = (audusdEntryPrice - askd);

audusdPosMoveShortDec = (audusdEntryPrice - askd) \* audusdPosLot;

// P/L or Position move string conversion

audusdPosPL = Convert.ToDouble(audusdPosMoveShortDec);

string moveString = Convert.ToString(audusdPosPL);

// Changes to P/L

if (audusdShortSaleCheck >= 0)

{

audusdPL.BackColor = System.Drawing.Color.Blue;

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "$" + moveString; }));

}

else if (audusdShortSaleCheck < 0)

{

audusdPL.BackColor = System.Drawing.Color.Red;

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "$" + moveString; }));

}

// SHORT: Buy Trigger

if (audusdShortSaleCheck >= goalLong)

{

netPandL = netPandL + Convert.ToDouble(moveString);

audusdEntryPrice = 0;

audusdPosPL = 0;

audusdSideString = "";

acctPos = acctPos - 1;

audusdShortSaleCheck = 0;

audusdPosLot = 0;

this.Invoke(new MethodInvoker(delegate { audusdPosSize.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": CLOSED:" + offerTableRow.Instrument + " @ " + offerTableRow.Ask + " For:" + audusdPL.Text + Environment.NewLine); }));

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdSide.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdEntry.Text = "--"; }));

}

// SHORT: Stop Loss

if (audusdShortSaleCheck <= (stopLoss \* -1))

{

netPandL = netPandL + Convert.ToDouble(moveString);

audusdEntryPrice = 0;

audusdPosPL = 0;

audusdSideString = "";

acctPos = acctPos - 1;

audusdShortSaleCheck = 0;

audusdPosLot = 0;

this.Invoke(new MethodInvoker(delegate { audusdPosSize.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { actiBox.AppendText(DateTime.Now + ": STOP LOSS: " + offerTableRow.Instrument + " @ " + offerTableRow.Ask + " For:" + audusdPL.Text + Environment.NewLine); }));

this.Invoke(new MethodInvoker(delegate { audusdPL.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdSide.Text = "--"; }));

this.Invoke(new MethodInvoker(delegate { audusdEntry.Text = "--"; }));

}

}

else

{

audusdSideString = "";

}

}

}