A quick guide to Zorro

What is Zorro?

Zorro is a free trading automation plattform. Forum

Variables

Type	Size (bytes)	Range	Step	Digits	Used for
double, var	8	-1.810308to 1.810308	2.210-308	14	prices
float	4	-3.41038to 3.41038	1.210-38	6	storing prices
fixed	4	-1048577.999 to 1048576.999	0.001	9	not for trading
int, long	4	-2147483648 to 2147483647	1	10	counting
short	2	0 to 65536	1	4	not for trading
char	1	0 to 256	1	2	text characters
bool	4	true, false	-	-	decisions
char*, string	characters+1	"..."	-	-	text
var*, vars	elements*8	-1.810308to 1.810308	2.210-308	14	arrays, series
mat	rows*cols*8+12	-1.810308to 1.810308	2.210-308	14	vectors, matrices

Skript Control

if .. else

```
if (((x+3)<9) or (y==0))   // set z to 10 if
    x+3 is below 9, or if y is equal to 0
z = 10;
else
z = 5;// set z to 5 in all other cases
```

while

```
x = 0;
while(x < 100) // repeat while x is lower
    than 100
x += 1;
```

for loop

```
int i;
for(i=0; i<5; i++) // repeat 5 times
x += i;
```

switch

```
int choice;
...
switch (choice)
{
  case 0:
  printf("Zero! ");
  break;
  case 1:
  printf("One! ");
  break;
  case 2:
  printf("Two! ");
  break;
  default:
  printf("None of them! ");
}
```

Input/Output

printf (string format, ...)

print (int to, string format,.)

TO_WINDOW print in [Test] and [Trade] mode to the message window and the log file (default for printf).

TO_LOG print in [Test] mode only to the log file, in [Trade] and in STEPWISE mode also to the message window.

TO_FILE

TO_ANY

TO_DIAG

TO_CSV

msg (string format, ...): int

0 when [No] was clicked,

1 when [Yes] was clicked.

File functions

file_copy (string dest, string src)

file_delete (string name)

file_length

file_date

file_select Opens a file dialog box at a given directory that lets the user select a file to open. Returns the selected file name including path.

file_content

file_read Reads the content of a file into a string or array. Returns the number of read bytes.

file_write Stores the content of a string, series, or other data array in a file.

file_append Opens a file and appends text or other data to it; can be used to export data to Excel or other programs. If the file does not exist, it is created.

Testing

Testing

Debugging

if(date() == 20150401) set(STEPWISE); watch ("!...", ...)

Biasis

Curve fitting bias, Market fitting bias, Peeking bias, Data mining bias, Trend bias, Granularity bias, Sample size bias

Asset Parameters

Spread, eg Spread = 3*PIP Slippage, default = 10
Commission, taken from AssetsFix.csv RollShort RollLong

Code

From Manual

Trade Management

Listing 1: Chan1ge stops and profit targets of all open long trades with the current algo and asset

```
exitLong(0,NewStop);
exitLong(0,-NewTakeProfit);
```

Listing 2: Limit the number of open positions // max. 3 open long positions per asset/algo

```
#define H24 (1440/BarPeriod)
#define H4 (240/BarPeriod)
#define H1 (60/BarPeriod)

if(my_long_condition == true) {
  exitShort(); // no hedging - close all
      short positions
  if(NumOpenLong < 3) enterlong();
}
```

Listing 3: Exit all open trades Friday afternoon GMT

```
if(dow() == FRIDAY && hour() >= 18) {
  exitLong("*");
  exitShort("*");
}
```

Listing 4: Lock 80% profit of all winning trades

```
for(open_trades)
if(TradeIsOpen && !TradeIsPool && TradeProfit
    > 0) {
  TradeTrailLock = 0.80; // 80% profit (minus
      trade costs)
  if(TradeIsShort)
  TradeTrailLimit = max(TradeTrailLimit,
      TradePriceClose);
  else
  TradeTrailLimit = min(TradeTrailLimit,
      TradePriceClose);
}
```

Listing 5: Calculate the value of all open trades with the current asset

```
var valOpen()
{
  string CurrentAsset = Asset; // Asset is
      changed in the for loop
  var val = 0;
```

```
  for(open_trades)
  if(strstr(Asset,CurrentAsset) &&
      TradeIsOpen)
  val += TradeProfit;
  return val;
}
```

Listing 6: Monitoring and modifying a certain trade

```

TRADE* MyTrade = enterlong();
...
ThisTrade = MyTrade; // connect trade
    variables to MyTrade
var MyResult = TradeProfit; // evaluate trade
     variables
...
exitTrade(MyTrade,0,TradeLots/2); // exit
    half the trade
}
```

Listing 7: Correlation / heatmap

```
#define DAYS  252 // 1 year
#define NN    30  // max number of assets

#include <profile.c>

// plot a heatmap of asset correlations
function run()
{
  BarPeriod = 1440;
  StartDate = 20150101;
  LookBack = DAYS;

  vars Returns[NN];
  var Correlations[NN][NN]; // NN*NN matrix
  int N = 0;
  while(asset(loop(.../*some assets*/ ..)))
  {
    Returns[N] = series((price(0)-price(1))/
        price(0));
    N++; // count number of assets
  }

  int i,j;
  if(!is(LOOKBACK)) {
    for(i=0; i<N; i++)
    for(j=0; j<N; j++)
    Correlations[N*i+j] = Correlation(Returns
        [i],Returns[j],DAYS);

    plotHeatmap(Correlations,N);
    quit("");
  }
}
```

Indicators Signals

Listing 8: Generate an indicator with a different asset time frame and shift

```
//extended ATR function with individual asset
     and timeframe (in minutes)
var extATR(string symbol,int period,int
    length,int shift)
{
ASSET* previous = g->asset; // store previous
     asset
if(symbol) asset(symbol);   // set new asset
if(period) TimeFrame = period/BarPeriod;
// create price series with the new asset /
    timeframe
vars H = series(priceHigh()),
L = series(priceLow()),
O = series(priceOpen()),
C = series(priceClose());
TimeFrame = 1; // set timeframe back
g->asset = previous; // set asset back
return ATR(O+shift,H+shift,L+shift,C+shift,
    length);
}
```

Listing 9: Calculate the weekend price change for gap trading

```
// use 1-hour bars, wait until Sunday Sunday
    5pm ET,
// then get the price change from Friday 5pm
    ET
if(dow() == SUNDAY && lhour(ET) == 5) {
  int FridayBar = timeOffset(ET,SUNDAY-FRIDAY
      ,5,0);
  var PriceChange = priceClose(0) -
      priceClose(FridayBar);
}
```

Listing 10: Use a series to check if something happened within the last n bars

```
// buy if Signal1 crossed over Signal2 within
     the last 7 bars
...
vars crosses = series(0); // generate a
    series and set it to 0
if(crossOver(Signal1,Signal2)
crosses[0] = 1; // store the crossover in the
     series
if(Sum(crosses,7) > 0) // any crossover
    within last 7 bars?
enterLong();
...
```

Listing 11: Use a loop to check if something happened within the last n bars

```
// buy if Signal1 crossed over Signal2 within
     the last 7 bars
...
int i;
for(i = 0; i < 7; i++)
if(crossOver(Signal1+i,Signal2+i)) { //
    crossover, i bars ago?
  enterLong();
  break; // abort the loop
}
...
```

Listing 12: Align a time frame to a certain event

```
// Let time frame start when Event == true
// f.i. frameAlign(hour() == 0); aligns to
    midnight
function frameAlign(BOOL Event)
{
  TimeFrame = 1;
  vars Num = series(0); // use a series for
      storing Num between calls
  Num[0] = Num[1]+1;     // count Num up once
      per bar
  if(!Event)
  TimeFrame = 0;       // continue current
      time frame
  else {
    TimeFrame = -Num[0]; // start a new time
        frame
    Num[0] = 0;          // reset the counter
  }
}
```

Listing 13: Shift a series into the future

```
// the future is unknown, therefore fill
// all unknown elements with the current
    value
vars seriesShift(vars Data,int shift)
{
  if(shift >= 0) // shift series into the
      past
  return Data+shift;
  else { // shift series into the future
    int i;
    for(i = 1; i <= shift; i++)
    Data[i] = Data[0];
    return Data;
  }
}
```

Listing 14: Use a function from an external DLL

```
// Use the function "foo" from the DLL "bar.dll"
// Copy bar.dll into the Zorro folder
int __stdcall foo(double v1,double v2); // foo prototype
API(foo,bar) // use foo from bar.dll

function run()
{
  ...
  int result = foo(1,2);
  ...
}
```

Listing 15: Equity curve trading (skipping trades dependent on strategy success)

```
// don't trade when the equity curve goes down
// and is below its own lowpass filtered value
function checkEquity()
{
  // generate equity curve including phantom trades
  vars EquityCurve = series(EquityLong+EquityShort);
  vars EquityLP = series(LowPass(EquityCurve,10));
  if(EquityLP[0] < LowPass(EquityLP,100) && falling(EquityLP))
  Lots = -1; // drawdown -> phantom trades
  else
  Lots = 1; // profitable -> normal trades
```

Auxiliary

Listing 16: Debugging a script

```
// Display a message box with the variables to be observed
// Click [Yes] for a single step, [No] for closing the box
static int debug = 1;
if(debug && Bar > LookBack)
debug = msg(
"High = %.5f, Low = %.5f",
priceHigh(), priceLow());
```

Listing 17: Find out if you have a standard mini or micro lot account

```
// Click [Trade] with the script below
function run()
{
  asset("EUR/USD");
  if(Bar > 0) {
    if(LotAmount > 99999)
    printf("\nI have a standard lot account!");
    else if(LotAmount > 9999)
    printf("\nI have a mini lot account!");
    else
    printf("\nI have a micro lot account!");
    quit();
  }
}
```

Listing 18: Download historic price data

```
// Click [Test] for downloading/updating the latest "NZD/USD" price data
// This extends the length of the historical price data file, therefore
// WFO strategies using that asset should be trained again afterwards
function run()
{
  NumYears = 6;      // download up to 6 years data
  loadHistory("NZD/USD",1); // update the price history
}
```

Listing 19: Export historic price data to a .csv file

```
// Click [Test] for exporting price data to a .csv file in the Data folder
// The records are stored in the format: time, open, high, low, close
// f.i. "31/12/12 00:00, 1.32205, 1.32341, 1.32157, 1.32278"
// Dependent on the locale, date and numbers might require a different format
function run()
{
  BarPeriod = 1440;
  StartDate = 2008;
  EndDate = 2012;
  LookBack = 0;

  string line = strf(
  "%02i/%02i/%02i %02i:%02i, %.5f, %.5f, %.5f, %.5f\n",
  day(),month(),year()%100,hour(),minute(),
  priceOpen(),priceHigh(),priceLow(),priceClose());
  if(is(INITRUN))
  file_delete("Data\\export.csv");
  else
  file_append("Data\\export.csv",line);
}
```

Listing 20: Print the description of a trade (like "[AUD/USD:CY:S1234

```
void printTradeID()
{
  string ls = "L", bo = "[", bc = "]";
  if(TradeIsShort) ls = "S";
  if(TradeIsPhantom) { bo = "{"; bc = "}"; }
  printf("#\n%s%s:%s:%s%04i%s ",
  bo,TradeAsset,TradeAlgo,ls,TradeID%10000,bc);
}
```

Listing 21: Plot equity curves of single assets in a multi-asset strategy

```
char name[40]; // string of maximal 39 characters
strcpy(name,Asset);
strcat(name,":");
strcat(name,Algo);
var equity = EquityShort+EquityLong;
if(equity != 0) plot(name,equity,NEW|AVG,BLUE);
```

Listing 22: Set up strategy parameters from a .ini file at the start

```
function run()
{
  static var Parameter1 = 0, Parameter2 = 0;
  if(is(INITRUN)) { // read the parameters only in the first run
    string setup = file_content("Strategy\\mysetup.ini");
    Parameter1 = strvar(setup,"Parameter1");
    Parameter2 = strvar(setup,"Parameter2");
  }
}

// mysetup.ini is a plain text file that contains
// the parameter values in a format like this:
Parameter1 = 123
Parameter2 = 456
```

Listing 23: Check every minute in Trade mode if an .ini file was modified

```
var Parameter1 = 0, Parameter2 = 0;

function tock() // run once per minute
{
  static int LastDate = 0;
  if(LastDate && !Trade) return; // already updated
```

```
  int NewDate = file_date("Strategy\\mysetup.ini");
  if(LastDate < NewDate) {
    LastDate = NewDate; // file was modified: update new parameters
    string setup = file_content("Strategy\\mysetup.ini");
    Parameter1 = strvar(setup,"Parameter1");
    Parameter2 = strvar(setup,"Parameter2");
  }
}
```

Listing 24: Trade multiple strategies and assets in a single script

```
function run()
{
  BarPeriod = 240;
  StartDate = 2010;
  set(TICKS); // set relevant variables and flags before calling asset()

  // call different strategy functions with different assets
  asset("EUR/USD");
  tradeLowpass();
  tradeFisher();

  asset("GBP/USD");
  tradeAverage();

  asset("SPX500");
  tradeBollinger();
}
```

Listing 25: Update price history of many assets

```
function run()
{
  NumYears = 2;
  while(loop("AUD/USD","EUR/USD","GBP/USD","GER30","NAS100",
  "SPX500","UK100","US30","USD/CAD","USD/CHF","USD/JPY",
  "XAG/USD","XAU/USD"))
  {
    assetHistory(Loop1,1);
  }
}
```

Listing 26: Portfolio strategy with 3 assets and 3 trade functions

```

function tradeTrendLong()
{
  algo("TRL");
  ...
}

function tradeTrendShort()
{
  algo("TRS");
  ...
}

function tradeBollinger()
{
  algo("BOL");
  ...
}

function tradeFunc(); // empty function pointer

function run()
{
  while(loop(
  "EUR/USD",
  "USD/CHF",
  "GBP/USD")) // loop through 3 assets
  while(tradeFunc = loop(
  tradeTrendLong,
  tradeTrendShort,
  tradeBollinger)) // and 3 different trade algorithms
  {
    asset(Loop1); // select asset
    tradeFunc();  // call the trade function
  }
}
```

Plotting

Listing 27: Plot Triangle above Candle Patterns

```
function run()
{
  set(PLOTNOW);
  PlotBars = 300;
  PlotScale = 8;
  if(CDLDoji())
  plot("Doji",1.002*priceHigh(),TRIANGLE4,BLUE);
  if(CDLHikkake() > 0)
  plot("Hikkake+",0.998*priceLow(),TRIANGLE,GREEN);
  if(CDLHikkake() < 0)
  plot("Hikkake-",1.002*priceHigh(),TRIANGLE4,RED);
}
```

Systems

YenTrader System

Listing 28: YenTrader System

```

bool UseEquityFilter = false;

function TradeYenTrader()
{
vars Price = series(price());
vars SMASlow = series(SMA(Price, optimize(100, 5, 200, 5)));
vars SMAFast = series(SMA(Price, optimize(50, 5, 200, 5)));

Stop = optimize(15, 5, 50) * PIP;
Trail = optimize(1, 1, 20) * PIP;

// Equity curve filter
var LotsBackup = Lots;

if (UseEquityFilter)
{
vars EquityCurve = series(EquityLong+EquityShort);
vars EquityLP = series(LowPass(EquityCurve,75));
if(EquityLP[0] < LowPass(EquityLP,100) && falling(EquityLP))
Lots = -1;
else
Lots = 1;
}

if (NumOpenShort + NumOpenLong == 0)
{
if (SMAFast[0] > SMASlow[0])
{
Margin = 0.5 * OptimalFLong * Capital * sqrt(max(1, Balance/Capital));
enterLong();
}
else
{
Margin = 0.5 * OptimalFShort * Capital * sqrt(max(1, Balance/Capital));
enterShort();
}
}

Lots = LotsBackup;
}

function run()
{
set(PARAMETERS+FACTORS);

BarPeriod = 1440;
LookBack = 200;
```

```
StartDate = 20010101;
EndDate = 20160601;
MonteCarlo = 1000;
Confidence = 100;
Capital = 10000;
NumWFOCycles = 10;

asset("USD/JPY");
TradeYenTrader();
}
```

Simple profit system - EUR/USD

Listing 29: Simple profit system

```

function run()
{

  BarPeriod = 60; // 1 hour bars
  LookBack = 20;
  StartDate = 2010;
  EndDate = 20160804;

  //NumWFOCycles = 6;
  //set(LOGFILE);
  //Capital = 5000;
  //Margin = 0.001* OptimalF * Capital * sqrt
      (1 + ProfitClosed/Capital);
  Hedge = 2;
  Lots=1;
  NumOptCycles = 2;
  NumSampleCycles = 2;


  set(PARAMETERS+FACTORS);
  var StopL = optimize(3,1,10) * ATRS(
      optimize(3,1,10));
  var TakeProfitL=optimize(3,1,10) * ATRS(
      optimize(3,1,10));

  var   StopS = optimize(3,1,10) * ATRS(
      optimize(3,1,10));
  var   TakeProfitS=optimize(3,1,10) * ATRS(
      optimize(3,1,10));

  if (hour() == 13){
    Stop=StopL;
    TakeProfit=TakeProfitL;
    enterLong();
  }

  if (hour() == 21 ){
    Stop=StopL;
    TakeProfit=TakeProfitL;
    enterShort();
  }

}
```

Profitable System - EUR/USD

Listing 30: Profitable System

```
function run()
{
  NumCores = -2;    // use multiple cores (
      Zorro S only)
  BarPeriod = 60; // 1 hour bars
  LookBack = 1240;  // needed for Fisher()
  StartDate = 2010;
  EndDate = 2015;   // fixed simulation
      period

  set(PARAMETERS);  // generate and use
      optimized parameters and factors

  Stop = 5 * ATR(90);
  TakeProfit=4* ATR(30);

  var nWLTOP=optimize(30,5,35);
  var nWLDOWN=optimize(70,60,100);

  vars aOpen = series(price());
  vars aAlma = series(ALMA(aOpen,
  optimize(8.61,8,30),
  optimize(6.15,6,19),
  0.8));

  vars MMI_Raw = series(MMI(aOpen,optimize
      (300,100,500)));
  vars MMI_Smooth = series(LowPass(MMI_Raw
      ,500));

  vars aEma1 = series(EMA(aOpen,optimize
      (12.6,12,100)));

  vars aWI1 = series(WillR(optimize
      (19.9,18,49)));


  if (rising(MMI_Smooth)  &&  aWI1[0] >
      nWLTOP*-1 && rising(aAlma) &&rising(
      aEma1)){
    enterLong();
  }else if ( falling(MMI_Smooth) && aWI1[0] <
       nWLDOWN*-1 && falling(aAlma)&&falling(
      aEma1)){
  enterShort();
}


}
```

Mean Variance Optimization

Listing 31: Mean Variance Optimization System

```
//
    ////////////////////////////////////////////
// Mean Variance Optimization
    ////////////////////////
//
    ////////////////////////////////////////////


//#define DAYS      252   // 1 year
#define DAYS      6*22 // 6 Months
//#define DAYS      4*22 // 4 Months
//#define DAYS      2*22 // 2 Months

#define WEIGHTCAP   .25 // Cap 0.15 - 0.5
    Range
#define NN        50    // max number of
    assets
#define LEVERAGE    4 // 1:4 leverage

//
    ////////////////////////////////////////////


function run()
{
  BarPeriod = 1440;
  LookBack = DAYS;
  NumYears = 7;
  set(PRELOAD); // allow extremely long
      lookback period
  set(LOGFILE);
  Verbose = 0;
  set(watch);

  //  AssetList = "ETF2016-OK.csv";
  AssetList = "AssetsZ8.csv";

  string    Names[NN];
  string    Symbols[NN]; // Store the ISIN
      Code
  vars      Returns[NN];
  var     Means[NN];
  var     Covariances[NN][NN];
  var     Weights[NN];
  static int  OldLots[NN];

  var TotalCapital = slider
      (1,1000,1000,50000,"Capital","Total
      capital to distribute");
  var VFactor = slider(2, 10 ,0, 100,"Risk","
      Variance factor");

  int N = 0;
  while(Names[N] = loop(Assets))
```

```
  {

    if(is(INITRUN) && strstr(Names[N], "#")==
        NULL) {
      assetHistory(Names[N], FROMLYAHOO);
      Symbols[N] = Symbol; // Store the isin
          code for quick referenze
    }

    if(strstr(Names[N], "#")== NULL && is(
        RUNNING)) {
      asset(Names[N]);
      Returns[N] = series((priceClose(0)-
          priceClose(1))/priceClose(1));
    }
    if(strstr(Names[N], "#")!= NULL && is(
        RUNNING)) Returns[N] = series(0);

    if(N++ >= NN) break;
  }

  if(tdm() == 1 && !is(LOOKBACK)){
    int i,j;
    for(i=0; i<N; i++) {
      Means[i] = Moment(Returns[i],LookBack
          ,1);
      for(j=0; j<N; j++)
      Covariances[N*i+j] = Covariance(Returns
          [i],Returns[j],LookBack);
    }
    var BestVariance = markowitz(Covariances,
        Means, N, WEIGHTCAP);
    var MinVariance = markowitzReturn(0,0);
    markowitzReturn(Weights,MinVariance+
        VFactor/100.*(BestVariance-MinVariance
        ));

    //    change the portfolio composition
        according to new weights
    for(i=0; i<N; i++)
    if (strstr(Names[i], "#")== NULL){
      asset(Names[i]);
      MarginCost = priceClose()/LEVERAGE;
      int NewLots = TotalCapital*Weights[i]/
          MarginCost;
      if(NewLots > OldLots[i])
      enterLong(NewLots-OldLots[i]);
      else if(NewLots < OldLots[i]) exitLong
          (0,0,OldLots[i]-NewLots);
      //          printf("\n%s - %s:  OldLots
          : %d NewLots: %d %.0f$",Names[i],
          Symbols[i], OldLots[i], NewLots );
      OldLots[i] = NewLots;
    }
  }
}
```

Intraday seasonality in FX market Based on this paper

Listing 32: Intraday seasonality in FX market System

```
var session1TZ = WET;
var session1Start = 8;
var session1End = 16;
var session2TZ = ET;
var session2Start = 11; // = 16 WET
var session2End = 17;

function tradeIS()
{
if(dow() >= 1 && dow() <= 5) {
if(NumOpenShort == 0 && lhour(session1TZ) ==
    session1Start)
enterShort();
if(NumOpenShort > 0 && lhour(session1TZ) >=
    session1End)
exitShort();
if(NumOpenLong == 0 && lhour(session2TZ) ==
    session2Start)
enterLong();
if(NumOpenLong > 0 && lhour(session2TZ) >=
    session2End)
exitLong();
}
}

function run()
{
StartDate = 2004;
UnstablePeriod = 0;
LookBack = 0;
BarPeriod = 1;
while(asset(loop("EUR/USD")))
tradeIS();
set(LOGFILE);
}
```

One Night Stand System

Listing 33: One Night Stand System

```

function tradeOneNightStand() {

  vars Price = series(price());
  vars SMA10 = series(SMA(Price, 10));
  vars SMA40 = series(SMA(Price, 40));

  //Stop = 90 * PIP;

  var BuyStop,SellStop;

  BuyStop = HH(10) + 1*PIP;
  SellStop = LL(10) - 1*PIP;

  if (dow() == 5 && NumOpenLong == 0 &&
      NumPendingLong == 0 && SMA10[0] > SMA40
      [0])
  enterLong(0,BuyStop);
  else if (dow() == 5 && NumOpenShort == 0 &&
       NumPendingShort == 0 && SMA10[0] <
      SMA40[0])
  enterShort(0,SellStop);

  if (dow() != 5 && dow() != 6 && dow() != 7)
       {
    exitLong();
    exitShort();
  }

}


function run() {

  BarPeriod = 1440;

  while(asset(loop("USD/CHF", "USD/JPY" , "
      GBP/USD", "EUR/USD")))
  tradeOneNightStand();

}
```

Listing 34: Portfolio trading Trend Counter Trend and Huck Trend

```
// Portfolio trading: Trend, Counter Trend
    and Huck Trend ////////////////////

function tradeTrend()
{
  TimeFrame = 1;
  var *Price = series(price());
  var *Trend = series(LowPass(Price,optimize
      (250,100,1000)));
  Stop = optimize(2,1,10) * ATR(100);

  if(strstr(Algo,":L") and valley(Trend))
  enterLong();
  else if(strstr(Algo,":S") and peak(Trend))
  enterShort();
}

function tradeCounterTrend()
{
  var *Price = series(price());
  var Threshold = optimize(1.0,0.5,2,0.1);
  var *DomPeriod = series(DominantPeriod(
      Price,30));
  var LowPeriod = LowPass(DomPeriod,500);
  var *HP = series(HighPass(Price,LowPeriod*
      optimize(1,0.5,2)));
```

```
  var *Signal = series(Fisher(HP,500));
  Stop = optimize(2,1,10) * ATR(100);

  if(strstr(Algo,":L") and crossUnder(Signal
      ,-Threshold))
  enterLong();
  else if(strstr(Algo,":S") and crossOver(
      Signal,Threshold))
  enterShort();
}

function HuckTrend()
{

  TimeFrame = 1;
  var *Price = series(price());
  var *LP5 = series(LowPass(Price,5));
  var *LP10 = series(LowPass(Price,optimize
      (10,6,20)));
  var *RSI10 = series(RSI(Price,10));
  Stop = optimize(5,1,10)*ATR(30);
  int crossed = SkillLong[0];
  int Delay = 3;


  if(crossOver(LP5,LP10))
  crossed = Delay;
  else if(crossUnder(LP5,LP10))
  crossed = -Delay;

  if(strstr(Algo,":L") and (crossed > 0 &&
      crossOver(RSI10,50))) {
    enterLong();
    crossed = 0;
  } else if(strstr(Algo,":S") and (crossed <
      0 && crossUnder(RSI10,50))) {
  enterShort();
  crossed = 0;
} else
SkillLong[0] -= sign(crossed);
}

function run()
{
  set(PARAMETERS+FACTORS+LOGFILE); // use
      optimized parameters and reinvestment
      factors
  BarPeriod = 240;  // 4 hour bars
  LookBack = 500; // needed for Fisher()
  NumWFOCycles = 8; // activate WFO
  NumBarCycles = 4; // 4 times oversampling

  if(ReTrain) {
    UpdateDays = 30;  // reload new price
        data from the server every 30 days
    SelectWFO = -1; // select the last cycle
        for re-optimization
  }

  // portfolio loop
  while(asset(loop("EUR/USD","USD/CHF")))
  while(algo(loop("TRND:L","TRND:S","CNTR:L",
      "CNTR:S","HuckTrend:L","HuckTrend:S")))
  {
    // set up the optimal margin
    if(Train)
    Lots = 1;
    else if(strstr(Algo,":L") and
        OptimalFLong > 0) {
      Lots = 1;
      Margin = clamp((WinLong-LossLong) *
          OptimalFLong/2, 50, 10000);
    } else if(strstr(Algo,":S") and
        OptimalFShort > 0) {
    Lots = 1;
    Margin = clamp((WinShort-LossShort) *
        OptimalFShort/2, 50, 10000);
  } else
  Lots = 0; // switch off trading

  if(strstr(Algo,"TRND"))
  tradeTrend();
  else if(strstr(Algo,"CNTR"))
  tradeCounterTrend();
  else if(strstr(Algo,"HuckTrend"))
  HuckTrend();

}

PlotWidth = 1000;
PlotHeight1 = 320;
}
```

Listing 35: System

```
// http://www.opserver.de/ubb7/ubbthreads.php
    ?ubb=showflat&Number=414386&page=1
function run(){

  set(LOGFILE);

  int FastPeriod = 8;
  int SlowPeriod = 21;
  int SignalPeriod = 9;

  BarPeriod = 15;
  vars Close = series(priceClose());
  TimeFrame = 1;

  MACD(Close,FastPeriod,SlowPeriod,
      SignalPeriod);
  vars MainLine = series(rMACD);
  vars SignalLine = series(rMACDSignal);
  vars Hist = series(rMACDHist);

  MACD(Close,FastPeriod*2,SlowPeriod*2,
      SignalPeriod*2);
  vars MainLine30 = series(rMACD);
  vars SignalLine30 = series(rMACDSignal);
  vars Hist30 = series(rMACDHist);

  MACD(Close,FastPeriod*4,SlowPeriod*4,
      SignalPeriod*4);
  vars MainLine60 = series(rMACD);
  vars SignalLine60 = series(rMACDSignal);
  vars Hist60 = series(rMACDHist);

  MACD(Close,FastPeriod*16,SlowPeriod*16,
      SignalPeriod*16);
  vars MainLine240 = series(rMACD);
  vars SignalLine240 = series(rMACDSignal);
  vars Hist240 = series(rMACDHist);



  if(crossOver(MainLine,SignalLine) &&
      MainLine30[0] > SignalLine30[0]
  && MainLine30[0] > SignalLine60[0] &&
      MainLine240[0] > SignalLine240[0]){


    exitShort();
    enterLong();
  }

  if(crossUnder(MainLine,SignalLine) &&
      MainLine30[0] < SignalLine30[0]
  && MainLine30[0] < SignalLine60[0] &&
      MainLine240[0] < SignalLine240[0]){


    exitLong();
    enterShort();
  }


}
```

System

Listing 36: System

```
//
function run ()


{

  StartDate = 20110601;
  EndDate = 20130120;
  BarPeriod = 1440;
```

```
  Stop = 50*PIP;
  Trail = 40*PIP;
  TrailLock = 1;


  vars day_low = series(priceLow());
  vars day_high = series(priceHigh());
  vars day_close = series(priceClose());
  vars EMA50 = series(EMA(day_close, 50));




  if (day_close[0] < day_low[1] && day_close
      [0] < EMA50[1])
  enterShort();


  plot ("EMA50", EMA50[0], 0, RED);

}
```

Listing 37: System

```
// Build 0004

// #define ASSETLOOP "USD/CHF", "EUR/USD", "
    GBP/USD", "USD/CAD", "AUD/USD", "USD/JPY",
     "XAU/USD", "XAG/USD", "NAS100", "SPX500",
     "GER30", "US30", "UK100" //FOREX SET
#define ASSETLOOP "EUR/USD", "GBP/USD", "USD/
    CAD", "AUD/USD", "USD/JPY", "XAU/USD", "
    XAG/USD" // No Stock
//#define ASSETLOOP "USD/CHF", "EUR/USD","USD
    /JPY","XAU/USD","SPX500"//MIN FOREX SET
//#define ASSETLOOP "EUR/USD" //test asset

//---- Global VAR ----
int myCapital = 0;
var myMargin = 0;
var comp = 0;
//--------------------

function setSlider()
{
  myCapital = slider(1,2500,0,25000,"Capital"
      ," Initial Capital"); //used for
      compounding calculation
  myMargin = slider(2,50,0,500,"Margin","
      Initial Margin"); // fixed or initial
      Margin
  comp= slider(3,0,0,1,"Comp.", "0=Fixed
      Margin 1=Compound Margin");
}

function checkEquity()
{
  // equity curve trading: switch to phantom
      mode when the equity
  // curve goes down and is below its own
      lowpass filtered value

  if(Train) { Lots = 1; return; } // no
      phantom trades in training mode
  vars EquityCurve = series(ProfitClosed+
      ProfitOpen);
  vars EquityLP = series(LowPass(EquityCurve
      ,10));
  if(EquityLP[0] < LowPass(EquityLP,100) &&
      falling(EquityLP))
  Lots = -1; // drawdown -> phantom trading
  else
  Lots = 1; // profitable -> normal trading
}

function enterFSLong()
{
  if(comp == 1 && !is(TRAINMODE)) {
    Margin = OptimalFLong * myMargin * sqrt(1
        + max(0,(WinLong-LossLong)/myCapital)
        );
    enterLong();
  } else {
  Margin=myMargin;
  enterLong();
}
}

function enterFSShort()
{
  if(comp == 1 && !is(TRAINMODE)) {
    Margin = OptimalFShort * myMargin * sqrt
        (1 + max(0,(WinShort-LossShort)/
        myCapital));
    enterLong();
  } else {
  Margin=myMargin;
  enterLong();
}
}

function CLSTR()
{
  TimeFrame=24;

  Stop = 2*ATR(14);
  Trail = Stop;
  TrailLock = 10;

  checkEquity();

  var dayL = optimize(40,10,80);
  var dayS = optimize(40,10,80);

  if (priceHigh() >= HH(dayL)) enterFSLong();
  if (priceLow() <= LL(dayS)) enterFSShort();
}

function CNTR()
{
  TimeFrame = 4;
  Stop = optimize(4,2,8) * ATR(100);
  Trail = 4*ATR(100);

  vars Price = series(price());
  vars Filtered = series(BandPass(Price,
      optimize(30,20,40),0.5));
  vars Signal = series(Fisher(Filtered,500));
  var Threshold = optimize(1,0.5,1.5,0.1);

  checkEquity();

  if(crossUnder(Signal,-Threshold))
      enterFSLong();
  else if(crossOver(Signal,Threshold))
      enterFSShort();
}

function TRND()
{
  TimeFrame = 1;
  Stop = optimize(4,2,8) * ATR(100);
  Trail = 0;

  vars Price = series(price());
  vars Trend = series(LowPass(Price,optimize
      (500,300,800)));

  checkEquity();

  if(valley(Trend)) enterFSLong();
  else if(peak(Trend)) enterFSShort();
}

function run()
{
  set(PARAMETERS+FACTORS); // generate and
      use optimized parameters and factors

  BarPeriod = 60; // 1 hour bars
  LookBack = 2500;  // needed for Fisher()
  NumWFOCycles = 6; // activate WFO

  StartDate = 2009;
  EndDate = 2014;
  Hedge=5;

  if(ReTrain) {
    UpdateDays = -1;  // update price data
        from the server
```

```
    SelectWFO = -1; // select the last cycle for re-optimization
    reset(FACTORS); // don't generate factors when re-training
  }
  NumWFOCycles = 6; // activate WFO

  setSlider();

  // portfolio loop
  while(asset(loop(ASSETLOOP)))
  while(algo(loop("TRND","CNTR","CLSTR")))
  {
    if(Algo == "TRND")
    TRND();

    if(Algo == "CNTR")
    CNTR();

    if(Algo == "CLSTR")
    CLSTR();
  }

  PlotWidth = 700;
  PlotHeight1 = 400;
  //ColorUp = ColorDn = ColorWin = ColorLoss = 0; // don't plot candles and trades
  //set(TESTNOW+PLOTNOW);
}
```

Listing 38: Enter a trade when the RSI12 crosses

```
// Zorro version
// enter a trade when the RSI12 crosses over 75 or under 25
function run()
{
  // get the RSI series
  vars Close = series(priceClose());
  vars rsi12 = series(RSI(Close,12));

  // set up stop / profit levels
  Stop = 200*PIP;
  TakeProfit = 200*PIP;

  // if rsi crosses over buy level, exit short and enter long
  if(crossOver(rsi12,75))
  reverseLong(1);
  // if rsi crosses below sell level, exit long and enter short
  if(crossUnder(rsi12,25))
  reverseShort(1);
}
```

Listing 39: First Hour Breakout System

```
// Zorro version
function run()
{

  set(TICKS+LOGFILE);

  BarPeriod = 30;
  LookBack  = 3;
  Hedge = 2;


  StartDate = 20100101;
  //  EndDate   = 20100201;

  asset("UK100");

  int marketstarthour   = 8;
  int marketstartminute = 0;
  int marketendhour     = 16;
  int marketendminute   = 30;

  vars hi       = series(priceHigh());
  vars lo       = series(priceLow());

  //Find the high low range of first hours trading
  vars enthigh    = series(MaxVal(hi,2));
  vars entlow     = series(MinVal(lo,2));
  vars Range      = series(enthigh[0]-entlow[0]);









  //  printf("\n%2.0d:%2.0d High %4.1f Low %4.1f enthigh %4.1f entlow %4.1f range %4.1f",hour(),minute(),hi[0],lo[0],enthigh[0],entlow[0],Range[0]);
  //

  //  if (NumPendingTotal == 1)   // Take only one trade per day OCO one cancels other Not sure if this was part of strategy or not
  //  {
    //    for(open_trades)
    //    {
      //      if (TradeIsPending) exitTrade(ThisTrade);
      //    }
    //  }

  if ((hour() == marketstarthour+1)
  and (minute() == marketstartminute))
  {

    //      printf("\nenthigh = %4.1f; entlow = %4.1f; Range = %4.1f; PIP = %4.1f",enthigh[0],entlow[0],Range[0],PIP);
    //

    if (Range[0] < 40*PIP)    // Don t trade if range is higher than 40 points
    {
      //        printf("\nRange %4.1f < 40 pips, entering pending trades at enthigh %4.1f & entlow %4.1f,price %4.1f",Range[0],enthigh[0],entlow[0],priceClose());

      EntryTime = 15;                       // Expire pending trades at 4:30

      enterLong(0,enthigh[0],Range[0],Range[0]);
      enterShort(0,entlow[0],Range[0],Range[0]);    //Buy on the breakout of the lowest low or the highest high of that first hour
    }

  }

  if (((hour() >= marketendhour)
  and (minute() >= marketendminute)) and (NumOpenTotal > 0))
  {
    //      printf("\nExiting Open Trades");
    exitLong();
    exitShort();
  }
}
```

Listing 40: First Hour Breakout System II

```
LookBack = 100;
set(TICKS);

vars Price = series(price());

StartMarket = 800;
EndMarket = 900;

var DH = 0, DL = 0;

if(lhour(UTC) > EndMarket) {
  DH = dayHigh (UTC,0);
  DL = dayLow (UTC,0);
}
```

```
var Range = DH-DL;
var Close = timeOffset(UTC,0,16,30);
var Open = timeOffset(UTC,0,8,00);
var StartTrade = timeOffset(UTC,0,9,00);
var Now = timeOffset(UTC,0,0,00);

if (Now > Close)
exitLong();

if (Now > Close)
exitShort();

asset("UK100");

Stop = Range;

TakeProfit = Range;

if (Range > 40*PIP)
Margin = 0;
```

Trademanagement Function

TradePriceOpen The ask price when the trade was opened. If the trade was not yet opened, it's the current price of the asset.

TradePriceClose

TradeUnits

TradeRoll

TradeProfit

TradeEntryLimit

TradeLots

TradeIsShort

TradeIsLong

Trading auxiallary functions

Listing 41: TrailingStop

```
int TrailingStop()
{
  // adjust the stop only when the trade is
      in profit
  if(TradeProfit > 0)
  // place the stop at the lowest bottom of
      the previous 3 candles
  TradeStopLimit = max(TradeStopLimit,LL(3));
  // plot a line to make the stop limit
      visible in the chart
  plot("Stop",TradeStopLimit,MINV,BLACK);
  // return 0 for checking the limits
  return 0;
}
```

Listings

Resources