

[Statistics](#) | [Charts](#) | [Trades](#) | [Formula](#) | [Settings](#) | [Symbols](#)

**Formula**

```

/*
File Name: MagicTrend_Indicator_1520.afl
Description: MagicTrend Indicator
Created Date: 20/09/2018
Version: 1.5.2.0
Author: Rakesh Solanki
Contact: admin@nascenttraders.com
Website: https://www.nascenttraders.com/
Copyright 2017-19 Rakesh Solanki. All Rights Reserved.

Afl codes contained in this file is exclusive property of Rakesh Solanki/www.nascenttraders.com and are protected under
the Copyright Act 1957 India. Unauthorised use, copy, reverse engineering, manipulation, modification, selling by
whatever means without the explicit permission of Rakesh Solanki/www.nascenttraders.com is criminal offence
and will attract legal action.
*/

SetBarsRequired( 100, 0 );
SetFormulaName( "MagicTrend Indicator" );
SetOption( "Initialequity", 1000000 );
SetOption( "AccountMargin", 0.25 );
SetOption( "MinShares", 1 );
MaxOpenPos = Param( "MaxOpenPos", 1, 1, 200, 1 );
SetOption( "MaxOpenPositions", MaxOpenPos );
SetOption( "PriceBoundChecking", 1 );
SetOption( "CommissionMode", 2 );
SetOption( "CommissionAmount", 20 );
PositionSize = -400 / ( MaxOpenPos );

_SECTION_BEGIN( "MagicTrend Indicator" );
P = Param( "Period", 20.0, 5.0, 50.0, 5.0 );
M = Param( "Multiplier", 3.0, 1.0, 5.0, 0.5 );

BarEnd = BarIndex() < LastValue( BarIndex() );
S = MagicTrend( P, M );

Buy = Cover = ( C >= S ) AND BarEnd;
Sell = Short = ( C < S ) AND BarEnd;
Buy = ExRem( Buy, Sell OR Short );
Sell = ExRem( Sell, Buy );
Short = ExRem( Short, Cover OR Buy );
Cover = ExRem( Cover, Short );

LTH = Param( "Line Tickness", 2, 0.1, 5, 0.1 );
MTL = ParamToggle( "MagicTrend Line", "Hide|Show", 1 );
TrendSignal = ParamToggle( "TrendSignal", "Arrow|Triangle", 1 );

if ( TrendSignal )
{
    UPShape = shapeUpArrow;
    DNShape = shapeDownArrow;
}
else
{
    UPShape = shapeSmallUpTriangle;
    DNShape = shapeSmallDownTriangle;
}

PlotShapes( Buy * shapeSquare, colorLime, 0, Min( S, L ), -10, 0 );
PlotShapes( Buy * shapeSquare, colorGreen, 0, Min( S, L ), -20, 0 );
PlotShapes( Short * shapeSquare, colorRed, 0, Max( S, H ), 10, 0 );
PlotShapes( Short * shapeSquare, colorOrange, 0, Max( S, H ), 20, 0 );
PlotShapes( Buy*UPShape, colorWhite, 0, Min( S, L ), -15 );
PlotShapes( Short*DNShape, colorWhite, 0, Max( S, H ), -15 );
SetChartOptions( 0, chartShowArrows | chartShowDates );
_N( Title = StrFormat( "{{NAME}} - {{INTERVAL}} [{{DATE}}] Open %g, Hi %g, Lo %g, Close %g ({{VALUES}})",
    , O, H, L, C, SelectedValue( ROC( C, 1 ) ) ) + " MTI: " + WriteVal( S, 1.2 ) );

if ( MTL )
{
    Plot( S, "STOP", IIf( C >= S, colorBrightGreen, colorRed ), styleStaircase | styleThick | styleSwingDots | styleNoTitle, Null, Null, 0, 0, LTH );
}

PlotOHLC( O, H, L, C, "Close", colorDefault, ParamStyle( "Price Style" ) | GetPriceStyle() | styleNoTitle );

_SECTION_END();

_SECTION_BEGIN( "Low Risk Entry Signal" );
PerRisk = PercentRisk( P, M );
LowRiskEntry = ParamToggle( "Low Risk Entry", "Show|Hide", 1 );
EntrySignalRisk = Param( "Entry Signal Risk", 40, 5, 100, 5 );

DownBar = ( C < Ref( C, -1 ) );
UpBar = ( C >= Ref( C, -1 ) );
LowRiskBuy = ( ( C >= S ) AND( PerRisk < EntrySignalRisk ) AND DownBar );
LowRiskShort = ( ( C < S ) AND( PerRisk < EntrySignalRisk ) AND UpBar );

LowRiskBuy = ExRem( LowRiskBuy, UpBar );
LowRiskShort = ExRem( LowRiskShort, DownBar );

LRESP = ParamToggle( "Entry Signal Position", "Bottom|Candle", 0 );
LRESPBUY = IIf( LRESP, LowestVisibleValue( L ), L );
LRESPSHORT = IIf( LRESP, LowestVisibleValue( L ), H );

if ( LowRiskEntry )
{
    PlotShapes( LowRiskBuy * shapeSquare, colorLime, 0, LRESPBUY, -10, 0 );
    PlotShapes( LowRiskBuy * shapeSquare, colorGreen, 0, LRESPBUY, -20, 0 );
    PlotShapes( LowRiskShort * shapeSquare, colorRed, 0, LRESPSHORT, 10, 0 );
    PlotShapes( LowRiskShort * shapeSquare, colorOrange, 0, LRESPSHORT, 20, 0 );
    PlotShapes( LowRiskBuy * shapeUpArrow, colorWhite, 0, LRESPBUY, -15 );
    PlotShapes( LowRiskShort * shapeDownArrow, colorWhite, 0, LRESPSHORT, -15 );
}

_SECTION_END();

_SECTION_BEGIN( "Alerts" );
// Alert Signals
TrendAlert = ParamToggle( "Trend Alert", "No|Yes", 1 );

```

```

LowRiskAlert = ParamToggle( "Entry Alert", "No|Yes", 1 );
AlertSound = ParamToggle( "Alert Sound", "Off|On", 0 );

if ( AlertSound )
{
    AlertIf( TrendAlert*Buy, "SOUND C:\\Windows\\Media\\notify.wav", "Trend Up: Buy @ " + WriteVal( C, 1.2 ), 1, 15, 0 );
    AlertIf( TrendAlert*Short, "SOUND C:\\Windows\\Media\\notify.wav", "Trend Down: Short @ " + WriteVal( C, 1.2 ), 2, 15, 0 );
    AlertIf( LowRiskAlert*LowRiskBuy, "SOUND C:\\Windows\\Media\\notify.wav", "Buy Entry @ " + WriteVal( C, 1.2 ), 1, 15, 0 );
    AlertIf( LowRiskAlert*LowRiskShort, "SOUND C:\\Windows\\Media\\notify.wav", "Short Entry @ " + WriteVal( C, 1.2 ), 2, 15, 0 );
}
else
{
    AlertIf( TrendAlert*Buy, "", "Trend Up: Buy @ " + WriteVal( C, 1.2 ), 1, 13, 0 );
    AlertIf( TrendAlert*Short, "", "Trend Down: Short @ " + WriteVal( C, 1.2 ), 2, 13, 0 );
    AlertIf( LowRiskAlert*LowRiskBuy, "", "Buy Entry @ " + WriteVal( C, 1.2 ), 1, 13, 0 );
    AlertIf( LowRiskAlert*LowRiskShort, "", "Short Entry @ " + WriteVal( C, 1.2 ), 2, 13, 0 );
}

_SECTION_END();

_SECTION_BEGIN( "Trade Report" );
y = Status( "pxchartheight" );
x = Status( "pxchartwidth" );
GfxSetOverlayMode( 1 );
GfxSetFont( "Verdana", 12, 800, True );
GfxSetBkMode( 0 );
GfxSetTextColor( ColorRGB( 18, 18, 18 ) );
GfxTextOut( "Created By: Rakesh Solanki", 0.45*x, 0.53*y );
GfxSetFont( "Verdana", 14, 800, True );
GfxSetBkMode( 1 );
GfxTextOut( "www.nascenttraders.com", 0.45*x, 0.5*y );

Open_Long = Flip( Buy, Sell );
Open_Short = Flip( Sell, Buy );
Buy_Price = ValueWhen( Buy, BuyPrice, 1 );
Short_Price = ValueWhen( Short, ShortPrice, 1 );
Sell_Price = IIf( Sell, ValueWhen( Sell, SellPrice, 1 ), Short_Price );
Cover_Price = IIf( Cover, ValueWhen( Cover, CoverPrice, 1 ), Buy_Price );
Profit = IIf( Open_Long, ( Close - Buy_Price ), IIf( Open_Short, ( Short_Price - Close ), 0 ) );
Last_Profit = IIf( Ref( Open_Long, -1 ) == 1 AND Open_Long != 1,
    ( Sell_Price - Buy_Price ), IIf( Ref( Open_Short, -1 ) == 1 AND Open_Short != 1, ( Short_Price - Cover_Price ), 0 ) );
Cum_Profit[0] = Last_Profit[0];
PreviousProfit = ValueWhen( Last_Profit != 0, Last_Profit, 1 );

for ( i = 1; i < BarCount; i++ )
{
    Cum_Profit[i] = Cum_Profit[i - 1] + Last_Profit[i];
}

Trade[0] = 0;

for ( i = 1; i < BarCount; i++ )
{
    if ( Buy[i] == 1 OR Short[i] == 1 )
    {
        Trade[i] = Trade[i - 1] + 1;
    }
    else
    {
        Trade[i] = Trade[i - 1];
    }
}

STR = ParamToggle( "Trade Report", "Show|Hide", 1 );
TRC = ParamList( "Trade Report Color", "Default|White|Black", 0 );

if ( STR == 1 )
{
    GfxSetFont( "Verdana", 10, 100 );
    GfxSetBkMode( 1 );

    if ( TRC == "White" )
    {
        GfxSetTextColor( ColorRGB( 255, 255, 255 ) );
    }
    else
    {
        if ( TRC == "Black" )
        {
            GfxSetTextColor( ColorRGB( 0, 0, 0 ) );
        }
        else
        {
            GfxSetTextColor( ColorRGB( 128, 128, 128 ) );
        }
    }

    GfxTextOut( ( "Total Trades : " + WriteVal( Trade, 3.0 ) ), 13, y - 125 );
    GfxTextOut( ( "Total Profit : " + WriteVal( Cum_Profit, 4.2 ) ), 13, y - 100 );
    GfxTextOut( ( "Previous Profit : " + WriteVal( PreviousProfit, 2.2 ) ), 13, y - 50 );
    GfxTextOut( ( "Price : " + WriteVal( Close, 2.2 ) ), 13, y - 75 );
    GfxTextOut( ( "Current Profit : " + WriteVal( Profit, 2.2 ) ), 13, y - 25 );
}

GfxSetFont( "Verdana", 12, 800, True );
GfxTextOut( "Last Price: " + WriteVal( LastValue( C ), 1.2, True ), 0.87*x, 0.0005*y );
_SECTION_END();

_SECTION_BEGIN( "Barometer" );
SBM = ParamToggle( "BaroMeter", "Show|Hide", 1 );

if ( SBM )
{
    y = Status( "pxchartheight" );
    x = Status( "pxchartwidth" );
    GfxSelectPen( colorBlack, 0.1 );

    if ( SelectedValue( C > S ) )
    {
        GfxSelectSolidBrush( colorDarkRed );
        GfxRoundRect( int( 0.03 * x ), int( 0.07 * y ), int( 0.15 * x ), int( 0.11 * y ), int( 0.11 * y ) - int( 0.07 * y ),
            int( 0.11 * y ) - int( 0.07 * y ) );
        GfxSelectSolidBrush( colorGreen );
    }
}

```

```
GfxRoundRect( int( 0.03 * x ), int( 0.07 * y ), int( 0.03 * x ) + ( int( 0.12 * x )
    * ( 1 - SelectedValue( PerRisk ) * 0.01 ) )
    , int( 0.11 * y ), int( 0.11 * y ) - int( 0.07 * y ) , int( 0.11 * y ) - int( 0.07 * y ) );
}
else
{
    GfxSelectSolidBrush( colorGreen );
    GfxRoundRect( int( 0.03 * x ), int( 0.07 * y ), int( 0.15 * x ), int( 0.11 * y ), int( 0.11 * y ) - int( 0.07 * y ) , int( 0.11 * y ) - int( 0.07 * y ) );
    GfxSelectSolidBrush( colorDarkRed );
    GfxRoundRect( int( 0.03 * x ), int( 0.07 * y ), int( 0.03 * x ) + ( int( 0.12 * x ) * ( 1 - SelectedValue( PerRisk ) * 0.01 ) )
        , int( 0.11 * y ), int( 0.11 * y ) - int( 0.07 * y ) ,
        int( 0.11 * y ) - int( 0.07 * y ) );
}
}

_SECTION_END();

_SECTION_BEGIN( "Signal Scan" );
Trade = WriteIf( C >= S, "Buy", "Short" );
Filter = ( PerRisk < EntrySignalRisk );
Color = IIf( C >= S, colorBrightGreen, colorRed );
TextColor = IIf( C >= S, colorBlack, colorWhite );
SetOption( "NoDefaultColumns", True );
AddTextColumn( Name(), "Full name", 77, TextColor, Color, 150 );
AddColumn( DateTime(), "Date", formatDateTime, TextColor, Color, 200 );
AddTextColumn( Trade, "Trend Signal", 77, TextColor, Color, 100 );
AddColumn( C, "CLOSE", 1.2, TextColor, Color, 100 );
AddColumn( PerRisk, "PercentRisk", 1.2, TextColor, Color, 100 );
_SECTION_END();
```