

```

#
# Study: FW_MOBO

# Adapted from the ToS BollingerBandsSMA study.
# Portions copyright 2007 by thinkorswim, inc.
# The study developer is not affiliated with thinkorswim, inc.
#
# Developed by ToS User: davidinc (studiessupport@pc3i.com)
# If you find any problems please let me know and refer
# to the revision number and date. Thanks.
#
#hint:<b>FW_MOBO</b>\nThis is a David Elliott (aka Firstwave) modification of Bollinger Bands.\n\n The first arrow
that plots may not be an actual BreakOUT / BreakDOWN, due to not knowing the MoboUP / MoboDOWN status prior to
the first (leftmost) candle. This study sets the first arrow based on the first close above the upper Mobo Band or first close
below the lower Mobo Band.\n\n<b>Input Parameters:</b>\n <b>colored mobo:</b> yes to color the bands, no to turn it
off\n <b>colored fill:</b> yes to turn colored-cloud on, no to turn it off\n <b>show alerts:</b> yes to turn the pop-up box
alerts on, no to turn it off\n <b>sound type:</b> select the sound you want, select nosound to turn it off. Show alerts must
also be yes to hear a sound. The window this study is running in must be open (not minimized) for the alert functionality
to work. \n<b>price, displace, length, num dev dn, num dev up:</b> use the defaults.
#
#
#
# Revision history:
# Rev# Date      Change Description
# -----
# 00 2009-01-06 Original
# 01 2009-02-04 Correct constant Num_Dev_Dn.
# 02 2009-12-09 Add BreakOut / BreakDown arrows. Added colored MOBO bands
#                to show MOBO Up/Down. Added colored fill between the bands.
# 03 2010-01-23 Added Pop-up / Audio "alert" functionality for MoBo
#                BreakOUT / BreakDOWN.
#
input ColoredMobo = yes;
input ColoredFill = yes;
input BreakArrows = yes;
input ShowAlerts = yes;
input SoundType = {default "NoSound", "Bell", "Chimes", "Ding", "Ring"};
input price = close;
input displace = 0;
input length = 10;
input Num_Dev_Dn = -0.8;
input Num_Dev_Up = +0.8;

def lastBar = if (IsNaN(close), 1, 0);
def sDev = stdev(data = price[-displace], length = length);
def MidLine = Average(data = price[-displace], length = length);
def LowerBand = MidLine + num_Dev_Dn * sDev;
def UpperBand = MidLine + num_Dev_Up * sDev;

def MoboStatus =
  if close > UpperBand then 2 # Mobo Up
  else
    if close < LowerBand then -2 # Mobo Down
    else 0 ; # between the bands

rec BreakStatus = compoundValue(1,
  if BreakStatus[1] == MoboStatus or MoboStatus == 0 then BreakStatus[1]
  else

```

```
if MoboStatus == 2 then 2
else -2, 0);
```

```
plot MidlineP = Midline;
MidLineP.SetDefaultColor(GetColor(1));
# David doesn't use the midline so it is hidden by default
MidLineP.hide();
```

```
plot UpperBandP = UpperBand;
UpperBandP.AssignValueColor (
  if !ColoredMobo then Color.White
  else
    if BreakStatus[0] == 2 then Color.Green
    else Color.Red);
UpperBandP.SetLineWeight(1);
```

```
plot LowerBandP = LowerBand;
LowerBandP.AssignValueColor (
  if !ColoredMobo then Color.White
  else
    if BreakStatus[0] == 2 then Color.Green
    else Color.Red);
LowerBandP.SetLineWeight(1);
```

```
# Breakout/down arrows.
plot BreakOutArrow =
  if BreakArrows then
    if BreakStatus[0] == BreakStatus[1] then double.NAN
    else if BreakStatus[0] == 2 then
      close else double.NAN
    else double.NAN;
BreakOutArrow.SetPaintingStrategy(PaintingStrategy.Arrow_Up);
BreakOutArrow.SetDefaultColor(Color.Red);
BreakOutArrow.SetLineWeight(3);
```

```
plot BreakDownArrow =
  if BreakArrows then
    if BreakStatus[0] == BreakStatus[1] then double.NAN
    else if BreakStatus[0] == -2 then
      close else double.NAN
  else double.NAN;
BreakDownArrow.SetPaintingStrategy(PaintingStrategy.Arrow_Down);
BreakDownArrow.SetDefaultColor(Color.Green);
BreakDownArrow.SetLineWeight(3);
```

```
# colored clouds
plot GreenUpper = if ColoredFill and BreakStatus[0] == 2 then
  UpperBandP else double.NAN;
GreenUpper.SetDefaultColor(CreateColor(0, 100, 0));
plot GreenLower = if ColoredFill and BreakStatus[0] == 2 then
  LowerBandP else double.NAN;
GreenLower.SetDefaultColor(CreateColor(0, 100, 0));
AddCloud (GreenUpper, GreenLower, CreateColor(0, 100, 0), color.red);
```

```
plot RedUpper = if ColoredFill and BreakStatus[0] == -2 then
  UpperBandP else double.NAN;
RedUpper.SetDefaultColor(CreateColor(200, 0, 0));
plot RedLower = if ColoredFill and BreakStatus[0] == -2 then
```

```
LowerBandP else double.NaN;
RedLower.SetDefaultColor(CreateColor(200, 0, 0));
AddCloud (RedUpper, RedLower, CreateColor(200, 0, 0), color.green);

alert(ShowAlerts and BreakStatus[1] == 2
 and (BreakStatus[1] <> BreakStatus[2]), "MoBo BreakOUT", Alert.Bar, SoundType);
alert(ShowAlerts and BreakStatus[1] == -2
 and (BreakStatus[1] <> BreakStatus[2]), "MoBo BreakDOWN", Alert.Bar, SoundType);

### the end
```