```
#region Using declarations
using System;
using System.ComponentModel;
using System.Diagnostics;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Xml.Serialization;
using NinjaTrader.Cbi;
using NinjaTrader.Data;
using NinjaTrader.Gui.Chart;
#endregion
// This namespace holds all indicators and is required. Do not change it.
namespace NinjaTrader.Indicator
{
/// <summary>
/// Plots the upper band and lower bands on top of the price type
/// </summary>
[Description("MicroTrends ATR Volatility Bands with Price Support - version 7.0.0.21 - Copyright 2010 MicroTrends All Rights Reserved. Plots the upper band and lower bands on top of the price type - default close. ")]
public class MicroTrendsATRVolatilityBands : MTIndicatorBase
{
private double aTRMultiplier = 2.5;
private int aTRPeriod = 4;
private int aTRSmoothing =5;

private DataSeries dsUpperV=null; //upper vol band
private DataSeries dsLowerV=null; //lower vol band

/// <summary>
/// This method is used to configure the indicator and is called once before any bar data is loaded.
/// </summary>
protected override void Initialize()
{
this.Name="MT Bands ATR Volatility";
Add(new Plot(Color.Pink, PlotStyle.Line, "Upper Volatility Band"));
Add(new Plot(Color.Pink, PlotStyle.Line, "Lower Volatility Band"));
dsUpperV = new DataSeries(this);
dsLowerV = new DataSeries(this);
CalculateOnBarClose = true;
Overlay = true;
PriceTypeSupported = true;
ZOrder=-1;
}
/// <summary>
/// Called on each bar update event (incoming tick)
/// </summary>
protected override void OnBarUpdate()
{
if (CurrentBar<1 || BarsInProgress>0) return;
```

```
dsUpperV.Set(Input[0]+(aTRMultiplier * ATR(aTRPeriod)[0]));
dsLowerV.Set(Input[0]-(aTRMultiplier * ATR(aTRPeriod)[0]));
if(aTRSmoothing>1)
{
Values[0].Set(WMA(dsUpperV,this.aTRSmoothing)[0]);
Values[1].Set(WMA(dsLowerV,this.aTRSmoothing)[0]);
}
else
{
Values[0].Set(dsUpperV[0]);
Values[1].Set(dsLowerV[0]);
}

}

#region Properties

[Browsable(false)] // this line prevents the data series from being displayed in the indicator properties dialog, do not remove
[XmlIgnore()] // this line ensures that the indicator can be saved/recovered as part of a chart template, do not remove
public DataSeries UpperBand
{
get { Update();
return Values[0]; }
}

[Browsable(false)] // this line prevents the data series from being displayed in the indicator properties dialog, do not remove
[XmlIgnore()] // this line ensures that the indicator can be saved/recovered as part of a chart template, do not remove
public DataSeries LowerBand
{
get { Update();
return Values[1]; }
}

[Description("ATR Multiplier - The ATR Multiplier default 2.5")]
[GridCategory("Parameters")]
[Gui.Design.DisplayNameAttribute("ATR Multiplier")]
public double ATRMultiplier
{
get { return aTRMultiplier; }
set { aTRMultiplier = Math.Max(0, value); }
}

[Description("ATR Period - the period to calculate an average")]
[GridCategory("Parameters")]
[Gui.Design.DisplayNameAttribute("ATR Period")]
public int ATRPeriod
```

```
{
get { return aTRPeriod; }
set { aTRPeriod = Math.Max(0, value); }
}

[Description("ATR Smoothing Period - the period to smooth the ATR - n periods")]
[GridCategory("Parameters")]
[Gui.Design.DisplayNameAttribute("ATR Smoothing")]
public int ATRSmoothing
{
get { return aTRSmoothing; }
set { aTRSmoothing = Math.Max(1, value); }
}

#endregion
}
}
```