KASE STATWARE®
NINJATRADER
PUBLIC DATA MEMBERS

Kase StatWare v9.91

August 2018

Kase StatWare® offers public data series that are members of each indicator’s class. These data series may be used to build custom indicators and strategies.

The information and tables in this document show the members available for each indicators and what they represent. Each public member represents a data series that can be accessed like the data structures in NinjaTrader, so the most recent bar will be indexed as 0 and prior bars will be positive numbers increasing the farther back you would like to access. (i.e. [1] is one bar ago [2] is two bars ago etc.)

Kase DevStops

Used to access Warning Line, Dev1, Dev2, and Dev3 member series.

Syntax

```
KaseDevStops(double dev1Val, double dev2Val, double dev3Val, int numBars, int barsInFMA, int barsInSMA)
```

Input Descriptions

Input	Description	Default
double Dev1Val	The standard deviation multiplier used to calculate Dev1.	1
double Dev2Val	The standard deviation multiplier used to calculate Dev2.	2.2
double Dev3Val	The standard deviation multiplier used to calculate Dev3.	3.6
int numBars	The number of bars to be used for the ATR and Standard Deviation calculations.	30
int barsInFMA	Number of bars to be used for the Fast Moving Average	10
int barsInSMA	Number of bars to be used for the Slow Moving Average	21

Public Members

Series	Description	Expected Return
Warn	Warning Line values	The calculated value for the warning line
Dev1	Dev1 value	The calculated value given the dev1Value input and the standard deviation calculated over numBars.
Dev2	Dev2 value	The calculated value given the dev2Value input and the standard deviation calculated over numBars.
Dev3	Dev3 value	The calculated value given the dev3Value input and the standard deviation calculated over numBars.

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign Dev1, Dev2, and Dev3 values for most recent bar to variables
        double Dev1 = KaseDevStops(1, 2.2, 3.6, 30, 10, 21).Dev1[0];
        double Dev2 = KaseDevStops(1, 2.2, 3.6, 30, 10, 21).Dev2[0];
        double Dev3 = KaseDevStops(1, 2.2, 3.6, 30, 10, 21).Dev3[0];
}
```

KaseCD

Used to access the KaseCD Momentum Values, Divergence, and KCDPeak signals.

Syntax

KaseCD(bool showAllDivs, double slopeFilter, int tolerance, int numBars, double peakStdDev, double peakFixed, int shortCylce, int longCycle, int bridgeFilter, double peakFilter, bool calcInRealTime)

Input Descriptions

Input	Description	Default
bool showAllDivs	Used to delete divergences that have been overtaken by a more recent divergence. This is primarily used for the display of the indicator and has no effect on the returns of the indicator.	false
double slopeFilter	The percentage (0.01 = 1%) that a potential momentum peak's value must be versus surrounding bars to be considered a momentum peak.	0.01
int tolerance	Distance between price swing and momentum peak for divergences and PeakOuts to be valid.	2
int numBars	Number of bars between confirmed swings for divergences (lookback for divergence).	55
double peakStdDev	Standard deviations of momentum used to measure potential overbought/oversold conditions for KCDpeak signals.	2.00
double peakFixed	Fixed overbought/oversold line. The overbought/oversold line for PeakOuts is the lesser of the KCD_PeakFixed or the KCD_peakSTDEV.	90
Int shortCycle	Minimum lookback for dynamic range	8
Int longCycle	Maximum lookback for dynamic range	65
Int bridgeFilter	The number of swings that can take place for a divergence to be considered bridging.	2
double peakFilter	Minimum value that momentum must overcome to be considered a potential peak.	15
bool calcInRealTime	Treat each calculation as the closing calculation for the bar.	false

Public Members

Series	Description	Expected Return
KCDHisto	Momentum value of KaseCD	Momentum value for index requested.
BullKCDPeak	KaseCD Bullish Peak	1 if signal confirmed, 0 otherwise
BullWeakKCDPeak	Weak KaseCD Bullish Peak	1 if signal confirmed, 0 otherwise
BearKCDPeak	KaseCD Bearish Peak	1 if signal confirmed, 0 otherwise
BearWeakKCDPeak	Weak KaseCD Bearish Peak	1 if signal confirmed, 0 otherwise
BullDivergence	KaseCD Bullish Divergence	1 if signal confirmed, 0 otherwise
BullWeakDivergence	Weak KaseCD Bullish Divergence	1 if signal confirmed, 0 otherwise
BearDivergence	KaseCD Bearish Divergence	1 if signal confirmed, 0 otherwise
BearWeakDivergence	Weak KaseCD Bearish Divergence	1 if signal confirmed, 0 otherwise

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign KCDHisto and BullDivergence values for most recent bar to variables
        double KCDHistoVal = KaseCD(false, 0.01, 2, 55, 2, 90, 8, 65, 2, 15, false).KCDHisto[0];
        double BullDiv = KaseCD(false, 0.01, 2, 55, 2, 90, 8, 65, 2, 15, false).BullDivergence[0];
}
```

KasePO

Used to access the KasePO Momentum Values, Divergence, and KCDPeak signals.

Syntax

KasePO(bool showAllDivs, double slopeFilter, int tolerance, int numBars, double peakStdDev, double peakFixed, int shortCylce, int longCycle, int bridgeFilter, double peakFilter, bool calcInRealTime)

Input Descriptions

Input	Description	Default
bool showAllDivs	Used to delete divergences that have been overtaken by a more recent divergence. This is primarily used for the display of the indicator and has no effect on the returns of the indicator.	false
double slopeFilter	The percentage (0.01 = 1%) that a potential momentum peak's value must be versus surrounding bars to be considered a momentum peak.	0.01
int tolerance	Distance between price swing and momentum peak for divergences and PeakOuts to be valid.	2
int numBars	Number of bars between confirmed swings for divergences (lookback for divergence).	55
double peakStdDev	Standard deviations of momentum used to measure potential overbought/oversold conditions for KPOpeak signals.	2.25
double peakFixed	Fixed overbought/oversold line. The overbought/oversold line for PeakOuts is the lesser of the KPO_PeakFixed or the KPO_peakSTDEV.	200
Int shortCycle	Minimum lookback for dynamic range	8
Int longCycle	Maximum lookback for dynamic range	65
Int bridgeFilter	The number of swings that can take place for a divergence to be considered bridging. (See Bridging Illustration).	2
double peakFilter	Minimum value that momentum must overcome to be considered a potential peak.	0
bool calcInRealTime	Treat each calculation as the closing calculation for the bar.	false

Public Members

Series	Description	Expected Return
KPOHisto	Momentum value of KasePO	Momentum value for index requested.
BullKPOPeak	KasePO Bullish PeakOut	1 if signal confirmed, 0 otherwise
BullWeakKPOPeak	Weak KasePO Bullish PeakOut	1 if signal confirmed, 0 otherwise
BearKPOPeak	KasePO Bearish PeakOut	1 if signal confirmed, 0 otherwise
BearWeakKPOPeak	Weak KasePO Bearish PeakOut	1 if signal confirmed, 0 otherwise
BullDivergence	KasePO Bullish Divergence	1 if signal confirmed, 0 otherwise
BullWeakDivergence	Weak KasePO Bullish Divergence	1 if signal confirmed, 0 otherwise
BearDivergence	KasePO Bearish signal	1 if signal confirmed, 0 otherwise
BearWeakDivergence	Weak KasePO Bearish Divergence	1 if signal confirmed, 0 otherwise

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign KPOHisto and BullDivergence values for most recent bar to variables
        double KPOHistoVal = KasePO(false, 0.01, 2, 55, 2, 90, 8, 65, 2, 15, false).KPOHisto[0];
        double BullDiv= KasePO(false, 0.01, 2, 55, 2, 90, 8, 65, 2, 15, false).BullDivergence[0];
}
```

KEES

Used to access KEES status (dots) and signals (L and S).

Syntax

```
KEES(bool dots, int length, int multiplier, bool outsideReversals, bool outsideSames, bool calcInRealTime)
```

Input Descriptions

Input	Description	Default
bool dots	Turn on or off the permission dots for the indicator. If this is set to false, the KeesStatus return will always be 0.	true
int length	Lookback length for underlying synthetic longer bar length momentum filter.	9
int multiplier	Number of bars to use to create synthetic longer bar length for filter.	3
bool outsideReversals	Include special conditions for reversal outside bars.	true
bool outsideSames	Include special condition for outside bars that are in the same direction as the previous bar.	True
bool calcInRealTime	Treat each calculation as the closing calculation for the bar.	true

Public Members

Series	Description	Expected Returns
KeesEntry	KEES entry signal (negative = short, positive = long) (3 first class, 2 second class, 1 warning)	-3, -2, -1, 0, 1, 2, 3
KeesStatus	KEES bar status (negative = short, positive = long) (4 first class entry, 3 first class non-entry, 2 second class entry, 1 second class non-entry, 0 error)	-4, -3, -2, -1, 0, 1, 2, 3, 4

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign KeesEntry and KeesEntry values for most recent bar to variables
        double KeesEntry = KEES(true, 9, 3, true, true, true).KeesEntry[0];
        double KeesStatus = KEES(true, 9, 3, true, true, true).KeesStatus[0];
}
```

Kase Perm Stochastic

Used to access values of the Kase Permission Stochastic.

Syntax

```
KasePermSto_V99(int multiplier, int length, int overBought, int overSold)
```

Input Descriptions

Input	Description	Default
int length	Lookback for underlying synthetic longer bar length momentum filter.	9
int multiplier	Number of bars to use to create synthetic longer bar length for filter.	3
int overBought	set overbought line (does not affect returns)	85
int overSold	set oversold line (does not affect returns)	15

Public Members

Series	Description	Expected Return
PermissionD	value of PermD (akin to Stochastick %D)	0 to 100
PermissionK	value of PermK (akin to Stochastick %K)	0 to 100

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign PermissionD and PermissionK values for most recent bar to variables
        double PermissionD = KasePermSto_V99(9, 3, 85, 15).PermissionD [0];
        double PermissionK = KasePermSto_V99(9, 3, 85, 15).PermissionK [0];
}
```

Kase Perm Function

Used to access values of the Kase Permission Function.

Syntax

KasePermFunc_V99(int multiplier, int length, int overBought, int overSold)

Input Descriptions

Input	Description	Default
int length	Lookback for underlying synthetic longer bar length momentum filter.	9
int multiplier	Number of bars to use to create synthetic longer bar length for filter.	3
int overBought	set overbought line.	85
int overSold	set oversold line.	15

Public Members

Series	Description	Expected Return
PLong	status of longer bar length filter (1 long, 0 otherwise)	0(not set or short) 1 (long)
PShort	status of longer bar length filter (1 short, 0 otherwise)	0(not set or long) 1 (short)

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign PLong and PLong values for most recent bar to variables
        double PLong = KasePermFunc _V99(9, 3, 85, 15).PLong [0];
        double PShort = KasePermFunc _V99(9, 3, 85, 15).PShort [0];
}
```

RevAmounts

Used to acces reversal amount values used by the Kase DevStops (size of stop).

Syntax

KaseRevAmounts_V99(double dev1Val, double dev2Val, double dev3Val, int numBars)

Input Descriptions

Input	Description	Default
double Dev1Val	The standard deviation multiplier used to calculate Dev1.	1
double Dev2Val	The standard deviation multiplier used to calculate Dev2.	2.2
double Dev3Val	The standard deviation multiplier used to calculate Dev3.	3.6
int numBars	The number of bars to be used for the ATR and Standard Deviation calculations.	30

Public Members

Series	Description	Expected Return
Warn	Warning Line reversal value	The calculated value for the warning line
Dev1	Dev1 line reversal value	The calculated value given the dev1Value input and the standard deviation calculated over num bars.
Dev2	Dev2 line reversal value	The calculated value given the dev2Value input and the standard deviation calculated over num bars.
Dev3	Dev3 line reversal value	The calculated value given the dev3Value input and the standard deviation calculated over num bars.

Example with Default Inputs

```
protected override void OnBarUpdate()
{
        //assign Rev1, Rev2, and Rev3 values for most recent bar to variables
        double Rev1 = KaseRevAmounts _V99(1, 2.2, 3.6, 30).Rev1[0];
        double Rev2 = KaseRevAmounts _V99(1, 2.2, 3.6, 30).Rev2[0];
        double Rev3 = KaseRevAmounts _V99(1, 2.2, 3.6, 30).Rev3[0];
}
```

Kase Trend

Used to access the trend direction of trigger level of Kase Trend.

Syntax

```
KaseTrend(int lookback, double sensitivity, bool useClose)
```

Input Descriptions

Input	Description	Default
Int lookback	The number of bars to be used for the ATR and Standard Deviation calculations.	30
double sensitivity	The standard deviation multiplier used to calculate trigger level.	3
bool useClose	When true the trend direction will flip upon close beyond trigger level. When false the trend direction will flip upon touch of trigger level.	true

Public Members

Series	Description	Expected Return
TrendDirection	Direction of trend	0 = error, 1 = Bullish, -1 = Bearish
TriggerLevel	Price at which the trend trigger level is plotted	The calculated value given the sensitivity input and the standard deviation calculated over num bars.

Example with Default Inputs

```
protected override void OnBarUpdate()
{
	int trendDirection = KaseTrend(30, 3, true).TrendDirection[0];
	double triggerLevel = KaseTrend(30, 3, true).TriggerLevel[0];
}
```