

#pragma once

```
SCFloatArrayRef CCISMA_S(SCFloatArrayRef In, SCFloatArrayRef SMAOut, SCFloatArrayRef CCIOut, int Index, int Length, float Multiplier);
SCFloatArrayRef CCI_S(SCFloatArrayRef In, SCFloatArrayRef MAOut, SCFloatArrayRef CCIOut, int Index, int Length, float Multiplier, unsigned int MovingAverageType);
float HighestHigh(SCFloatArrayRef In, int StartIndex, int Length);
SCFloatArrayRef HighestHigh_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef Highest_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
float LowestLow(SCFloatArrayRef In, int StartIndex, int Length);
SCFloatArrayRef Lowest_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef LowestLow_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
float TrueRange(SCBaseDataRef BaseDataIn, int Index);
SCFloatArrayRef TrueRange_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, int Index);
SCFloatArrayRef AverageTrueRange_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef TROut, SCFloatArrayRef ATROut, int Index, int Length, unsigned int MovingAverageType);
SCFloatArrayRef OnBalanceVolume_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, int Index);
SCFloatArrayRef OnBalanceVolumeShortTerm_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, SCFloatArrayRef OBVTemp, int Index, int Length);
SCFloatArrayRef MovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, unsigned int MovingAverageType, int Index, int Length);

void Stochastic_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef FastKOut, SCFloatArrayRef FastDOut, SCFloatArrayRef SlowDOut, int Index, int FastKLength, int FastDLength, int SlowDLength, unsigned int MovingAverageType);

void Stochastic2_S(SCFloatArrayRef InputDataHigh, SCFloatArrayRef InputDataLow, SCFloatArrayRef InputDataLast, SCFloatArrayRef FastKOut, SCFloatArrayRef FastDOut, SCFloatArrayRef SlowDOut, int Index, int FastKLength, int FastDLength, int SlowDLength, unsigned int MovingAverageType);

SCFloatArrayRef ExponentialMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);

void CalculateRegressionStatistics(SCFloatArrayRef In, double &Slope, double &Y_Intercept, int Index, int Length);
void CalculateLogLogRegressionStatistics(SCFloatArrayRef In, double &Slope, double &Y_Intercept, int Index, int Length);

SCFloatArrayRef LinearRegressionIndicator_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);

SCFloatArrayRef LinearRegressionIndicatorAndStdErr_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef StdErr, int Index, int Length);

SCFloatArrayRef AdaptiveMovAvg_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length, float FastSmoothConst, float SlowSmoothConst);
SCFloatArrayRef SimpleMovAvg_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef SimpleMovAvgSkipZeros_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef WildersMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef WeightedMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef HullMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef InternalArray1, SCFloatArrayRef InternalArray2, SCFloatArrayRef InternalArray3, int Index, int Length);
SCFloatArrayRef TriangularMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef InternalArray1, int Index, int Length);
SCFloatArrayRef VolumeWeightedMovingAverage_S(SCFloatArrayRef InPrice, SCFloatArrayRef InVolume, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef MovingMedian_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef Temp, int Index, int Length);
void InternalStandardDeviation(SCFloatArrayRef In, float& out, int start_idx, int length);
void GetStandardDeviation(SCFloatArrayRef In, float& Out, int StartIndex, int Length);
SCFloatArrayRef StandardDeviation_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef Ergodic_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int LongEMALength, int ShortEMALength, float Multiplier, SCFloatArrayRef InternalArray1, SCFloatArrayRef InternalArray2, SCFloatArrayRef InternalArray3, SCFloatArrayRef InternalArray4, SCFloatArrayRef InternalArray5, SCFloatArrayRef InternalArray6);
SCFloatArrayRef Keltner_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef In, SCFloatArrayRef KeltnerAverageOut, SCFloatArrayRef TopBandOut, SCFloatArrayRef BottomBandOut, int Index, int KeltnerMALength, unsigned int KeltnerMAType, int TrueRangeMALength, unsigned int
```

```

TrueRangeMAType, float TopBandMultiplier, float BottomBandMultiplier,
    SCFloatArrayRef InternalArray1, SCFloatArrayRef InternalArray2);
float WellesSum(float In, int Index, int Length, SCFloatArrayRef Out);
SCFloatArrayRef WellesSum_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
void DirectionalMovementTrueRangeSummation(SCBaseDataRef BaseDataIn, int Index, int Length,
    SCFloatArrayRef InternalTrueRangeSummation, SCFloatArrayRef InternalPosDM,
    SCFloatArrayRef InternalNegDM);

void DMI_S(SCBaseDataRef BaseDataIn, int Index, int Length, int DisableRounding,
    SCFloatArrayRef PosDMIOut, SCFloatArrayRef NegDMIOut, SCFloatArrayRef DiffDMIOut,
    SCFloatArrayRef InternalTrueRangeSummation, SCFloatArrayRef InternalPosDM, SCFloatArrayRef
    InternalNegDM);

SCFloatArrayRef DMIDiff_S(SCBaseDataRef BaseDataIn,
    int Index,
    int Length,
    SCFloatArrayRef Out,
    SCFloatArrayRef InternalTrueRangeSummation, SCFloatArrayRef InternalPosDM, SCFloatArrayRef
    InternalNegDM);

SCFloatArrayRef ADX_S(SCBaseDataRef BaseDataIn,
    int Index,
    int DXLength, int DXMovAvgLength,
    SCFloatArrayRef Out,
    SCFloatArrayRef InternalTrueRangeSummation,
    SCFloatArrayRef InternalPosDM,
    SCFloatArrayRef InternalNegDM,
    SCFloatArrayRef InternalDX);

SCFloatArrayRef ADXR_S(SCBaseDataRef BaseDataIn,
    int Index,
    int DXLength, int DXMovAvgLength, int ADXRInterval,
    SCFloatArrayRef Out,
    SCFloatArrayRef InternalTrueRangeSummation,
    SCFloatArrayRef InternalPosDM,
    SCFloatArrayRef InternalNegDM,
    SCFloatArrayRef InternalDX,
    SCFloatArrayRef InternalADX);

SCFloatArrayRef RSI_S(SCFloatArrayRef In,
    SCFloatArrayRef RsiOut,
    SCFloatArrayRef UpSumsTemp,
    SCFloatArrayRef DownSumsTemp,
    SCFloatArrayRef SmoothedUpSumsTemp,
    SCFloatArrayRef SmoothedDownSumsTemp,
    int Index, unsigned int AveragingType, int Length);

SCFloatArrayRef SmoothedMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef SmoothedAverageOut, int Index, int
    Length);

SCFloatArrayRef MACD_S(SCFloatArrayRef In, SCFloatArrayRef FastMAOut, SCFloatArrayRef SlowMAOut,
    SCFloatArrayRef MACDOut, SCFloatArrayRef MACDMAOut, SCFloatArrayRef MACDDiffOut, int Index, int
    FastMALength, int SlowMALength, int MACDMALength, int MovAvgType);

SCFloatArrayRef TEMA_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef InternalArray1, SCFloatArrayRef
    InternalArray2, SCFloatArrayRef InternalArray3, int Index, int Length);

SCFloatArrayRef BollingerBands_S(SCFloatArrayRef In, SCFloatArrayRef Avg, SCFloatArrayRef
    TopBand, SCFloatArrayRef BottomBand, SCFloatArrayRef StdDev, int Index, int Length, float Multiplier, int MovAvgType);

SCFloatArrayRef BollingerBands_StandardDeviationOfAverage_S(SCFloatArrayRef In, SCFloatArrayRef Avg,
    SCFloatArrayRef TopBand, SCFloatArrayRef BottomBand, SCFloatArrayRef StdDev, int Index, int Length, float
    Multiplier, int MovAvgType);

void Summation(SCFloatArrayRef in, float& out, int start_idx, int length);

```

```

float GetSummation(SCFloatArrayRef In,int Index,int Length);
double GetMovingAverage(SCFloatArrayRef in, int start_indx, int length);

double GetVariance(SCFloatArrayRef InData, int StartIndex, int Length);

SCFloatArrayRef AccumulationDistribution_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, int Index);

SCFloatArrayRef CumulativeSummation_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index);
SCFloatArrayRef ArmsEaseOfMovement_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, int VolumeDivisor, int
Index);
SCFloatArrayRef ChaikinMoneyFlow_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, SCFloatArrayRef
InternalArray, int Index, int Length);
SCFloatArrayRef Summation_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef Dispersion_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);
float GetDispersion(SCFloatArrayRef In, int Index, int Length);
SCFloatArrayRef EnvelopePercent_S(SCFloatArrayRef In, SCFloatArrayRef Out1, SCFloatArrayRef Out2, float pct, int
Index);

SCFloatArrayRef VerticalHorizontalFilter_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index, int Length);

SCFloatArrayRef RWI_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out1, SCFloatArrayRef Out2,
SCFloatArrayRef TrueRangeArray, SCFloatArrayRef LookBackLowArray, SCFloatArrayRef
LookBackHighArray, int Index, int Length);

SCFloatArrayRef UltimateOscillator_S(SCBaseDataRef BaseDataIn,
SCFloatArrayRef Out,
SCFloatArrayRef CalcE,
SCFloatArrayRef CalcF,
SCFloatArrayRef CalcG,
SCFloatArrayRef CalcH,
SCFloatArrayRef CalcI,
SCFloatArrayRef CalcJ,
SCFloatArrayRef CalcK,
SCFloatArrayRef CalcL,
SCFloatArrayRef CalcM,
SCFloatArrayRef CalcN,
SCFloatArrayRef CalcO,
SCFloatArrayRef CalcP,
SCFloatArrayRef CalcQ,
int Index, const int Length1, const int Length2, const int Length3);

SCFloatArrayRef WilliamsAD_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, int Index);

SCFloatArrayRef WilliamsR_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, int Index, int Length);
SCFloatArrayRef WilliamsR2_S(SCFloatArrayRef InputDataHigh, SCFloatArrayRef InputDataLow, SCFloatArrayRef
InputDataLast, SCFloatArrayRef Out, int Index, int Length);

int GetIslandReversal_S(SCBaseDataRef BaseDataIn, int Index);

SCFloatArrayRef Oscillator_S(SCFloatArrayRef In1, SCFloatArrayRef In2, SCFloatArrayRef Out, int Index);

float GetTrueHigh(SCBaseDataRef BaseDataIn, int Index);

float GetTrueLow(SCBaseDataRef BaseDataIn, int Index);

float GetTrueRange(SCBaseDataRef BaseDataIn, int Index);

float GetRange(SCBaseDataRef BaseDataIn, int Index);

float GetCorrelationCoefficient(SCFloatArrayRef In1, SCFloatArrayRef In2, int Index, int Length);

int NumberOfBarsSinceHighestValue(SCFloatArrayRef In, int Index, int Length);

int NumberOfBarsSinceLowestValue(SCFloatArrayRef In, int Index, int Length);

```

SCFloatArrayRef PriceVolumeTrend_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, **int** Index);

SCFloatArrayRef Momentum_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef TRIX_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef InternalEma_1, SCFloatArrayRef InternalEma_2, SCFloatArrayRef InternalEma_3, **int** Index, **int** Length);

SCFloatArrayRef Parabolic_S(s_Parabolic & ParabolicData);

SCFloatArrayRef ResettableZigZag_S(SCFloatArrayRef InputDataHigh, SCFloatArrayRef InputDataLow, SCFloatArrayRef ZigZagValues, SCFloatArrayRef ZigZagPeakType, SCFloatArrayRef ZigZagPeakIndex, **int** StartIndex, **int** Index, **float** ReversalPercent, **float** ReversalAmount, SCStudyInterfaceRef sc);

SCFloatArrayRef ResettableZigZag2_S(SCFloatArrayRef InputDataHigh, SCFloatArrayRef InputDataLow, SCFloatArrayRef ZigZagValues, SCFloatArrayRef ZigZagPeakType, SCFloatArrayRef ZigZagPeakIndex, **int** StartIndex, **int** Index, **int** NumberOfBars, **float** ReversalAmount, SCStudyInterfaceRef sc);

float GetArrayValueAtNthOccurrence(SCFloatArrayRef TrueFalseIn, SCFloatArrayRef ValueArrayIn, **int** Index, **int** NthOccurrence);

SCFloatArrayRef AroonIndicator_S(SCFloatArrayRef FloatArrayInHigh, SCFloatArrayRef FloatArrayInLow, SCFloatArrayRef OutUp, SCFloatArrayRef OutDown, SCFloatArrayRef OutOscillator, **int** Index, **int** Length);

SCFloatArrayRef Demarker_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, SCFloatArrayRef DemMax, SCFloatArrayRef DemMin, SCFloatArrayRef SmaDemMax, SCFloatArrayRef SmaDemMin, **int** Index, **int** Length);

SCFloatArrayRef EnvelopeFixed_S(SCFloatArrayRef In, SCFloatArrayRef Out1, SCFloatArrayRef Out2, **float** FixedValue, **int** Index);

int IsSwingHighAllowEqual_S(SCStudyInterfaceRef sc, **int** AllowEqual, **int** Index, **int** Length);

int IsSwingLowAllowEqual_S(SCStudyInterfaceRef sc, **int** AllowEqual, **int** Index, **int** Length);

SCFloatArrayRef AwesomeOscillator_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef TempMA1, SCFloatArrayRef TempMA2, **int** Index, **int** Length1, **int** Length2);

int CalculatePivotPoints

(**float** PriorOpen
, **float** PriorHigh
, **float** PriorLow
, **float** PriorClose
, **float** CurrentOpen
, **float**& PivotPoint
, **float**& PivotPointHigh
, **float**& PivotPointLow
, **float**& R_5
, **float**& R1, **float**& R1_5
, **float**& R2, **float**& R2_5
, **float**& R3
, **float**& S_5
, **float**& S1, **float**& S1_5
, **float**& S2, **float**& S2_5
, **float**& S3
, **float** &R3_5
, **float** &S3_5
, **float**& R4
, **float**& R4_5
, **float**& S4
, **float**& S4_5
, **float**& R5
, **float**& S5
, **float**& R6
, **float**& S6

```
, float& R7
, float& S7
, float& R8
, float& S8
, float& R9
, float& S9
, float& R10
, float& S10
, int FormulaType
);
```

```
int CalculateDailyOHLC
(SCStudyInterfaceRef sc
, const SCDatetimeMS& CurrentBarTradingDayDate
, int InNumberOfDaysBack
, int InNumberOfDaysToCalculate
, int InUseSaturdayData
, int InUseThisIntradayChart
, int InDailyChartNumber
, SCGraphData& DailyChartData
, SCDatetimeArray& DailyChartDateTimes
, int UseDaySessionOnly
, float& Open
, float& High
, float& Low
, float& Close
, float& Volume
, int IncludeFridayEveningSessionWithSundayEveningSession
, int InUseSundayData
);
```

```
int CalculateDailyPivotPoints
(SCStudyInterfaceRef sc
, int IntradayChartDate
, int FormulaType
, int DailyChartNumber
, SCGraphData& DailyChartData
, SCDatetimeArray& DailyChartDateTimeArray
, int NumberOfDaysToCalculate
, int UseSaturdayData
, int UseThisChart
, int UseManualValues
, float UserOpen
, float UserHigh
, float UserLow
, float UserClose
, int UseDaySessionOnly
, float& PivotPoint
, float& PivotPointHigh
, float& PivotPointLow
, float& R_5
, float& R1, float& R1_5
, float& R2, float& R2_5
, float& R3
, float& S_5
, float& S1, float& S1_5
, float& S2, float& S2_5
, float& S3
, float & R3_5
, float & S3_5
, float& R4
, float&R4_5
, float& S4
, float& S4_5
, float& R5
```

```
, float& S5
, float& R6
, float& S6
, float& R7
, float& S7
, float& R8
, float& S8
, float& R9
, float& S9
, float& R10
, float& S10
, int UseDailyChartForSettlementOnly
);
```

```
SCFloatArrayRef Slope_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index);
```

```
SCFloatArrayRef CalculateAngle_S(SCFloatArrayRef InputArray, SCFloatArrayRef OutputArray, int Index, int Length,
float ValuePerPoint);
```

```
SCFloatArrayRef DoubleStochastic_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Out, SCFloatArrayRef MovAvgIn,
SCFloatArrayRef MovAvgOut, SCFloatArrayRef MovAvgIn2, SCFloatArrayRef Unused, int Index, int Length, int
MovAvgLength, int MovAvgType);
```

```
double GetStandardError(SCFloatArrayRef In, int Index, int Length);
```

```
SCFloatArray& StandardError_S(SCFloatArrayRef In, SCFloatArray& Out, int Index, int Length);
```

```
SCFloatArrayRef CumulativeDeltaVolume_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Open, SCFloatArrayRef
High, SCFloatArrayRef Low, SCFloatArrayRef Close, int Index, int ResetCumulativeCalculation);
```

```
SCFloatArrayRef CumulativeDeltaTicks_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Open, SCFloatArrayRef High,
SCFloatArrayRef Low, SCFloatArrayRef Close, int Index, int ResetCumulativeCalculation);
```

```
SCFloatArrayRef CumulativeDeltaTickVolume_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef Open, SCFloatArrayRef
High, SCFloatArrayRef Low, SCFloatArrayRef Close, int Index, int ResetCumulativeCalculation);
```

```
void Vortex_S(SCBaseDataRef BaseDataIn, SCFloatArrayRef TrueRangeOut, SCFloatArrayRef VortexMovementUpOut,
SCFloatArrayRef VortexMovementDownOut, SCFloatArrayRef VMPlusOut, SCFloatArrayRef VMMinusOut, int Index, int
VortexLength);
```

```
void HeikinAshi_S(SCBaseDataRef BaseDataIn,
int Index,
int Length,
SCFloatArrayRef OpenOut, SCFloatArrayRef HighOut,
SCFloatArrayRef LowOut, SCFloatArrayRef LastOut,
int SetCloseToCurrentPriceAtLastBar);
```

```
void InverseFisherTransform_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef CalcArray1,
SCFloatArrayRef CalcArray2, int Index, int HighestLowestLength, int MovingAverageLength, int MovAvgType);
```

```
void InverseFisherTransformRSI_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef RSIArray1,
SCFloatArrayRef RSIArray2, SCFloatArrayRef RSIArray3, SCFloatArrayRef RSIArray4, SCFloatArrayRef RSIArray5,
SCFloatArrayRef CalcArray1, SCFloatArrayRef CalcArray2, int Index, int RSILength, int InternalRSIMovAvgType, int
RSIMovingAverageLength, int MovingAverageOfRSIType);
```

```
void MovingAverageCumulative_S(SCFloatArrayRef In, SCFloatArrayRef Out, int Index);
```

```
void CalculateCumulativeLogLogRegressionStatistics_S(SCFloatArrayRef In, double &Slope, double &Y_Intercept,
SCFloatArrayRef Array_Sum_x, SCFloatArrayRef Array_Sum_x2, SCFloatArrayRef Array_Sum_x_2, SCFloatArrayRef
Array_Sum_y, SCFloatArrayRef Array_Sum_xy, int Index);
```

```
SCFloatArrayRef CumulativeStdDev_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef Array_XSquared,
SCFloatArrayRef Array_Mean, SCFloatArrayRef Array_MeanOfSquares, int Index);
```

```
SCFloatArrayRef CumulativeHurstExponent_S(SCFloatArrayRef In, SCFloatArrayRef Out, SCFloatArrayRef
```


Array_Average, SCFloatArrayRef Array_MeanAdjustedData, SCFloatArrayRef Array_CumulativeDeviation, SCFloatArrayRef Array_MinMax, SCFloatArrayRef Array_StdDev, SCFloatArrayRef Array_RescaledRange, SCFloatArrayRef Array_XSquared, SCFloatArrayRef Array_Mean, SCFloatArrayRef Array_MeanOfSquares, SCFloatArrayRef Array_Sum_x, SCFloatArrayRef Array_Sum_x2, SCFloatArrayRef Array_Sum_x_2, SCFloatArrayRef Array_Sum_y, SCFloatArrayRef Array_Sum_xy, **int** Index);

void CalculateHurstExponent(SCFloatArrayRef In_X, SCFloatArrayRef In_Y, **double** &HurstExponent, **int** Index, **int** Length);

SCFloatArrayRef HurstExponentNew_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** LengthIndex);

void GetMinMaxValuesFromArray(**float*** Array, **int** ArrayLength, **float** &MinValue, **float** &MaxValue);

SCFloatArrayRef T3MovingAverage_S

(SCFloatArrayRef InputArray
 , SCFloatArrayRef OutputArray
 , SCFloatArrayRef CalcArray0
 , SCFloatArrayRef CalcArray1
 , SCFloatArrayRef CalcArray2
 , SCFloatArrayRef CalcArray3
 , SCFloatArrayRef CalcArray4
 , SCFloatArrayRef CalcArray5
 , **float** Multiplier
 , **int** Index
 , **int** Length);

SCFloatArrayRef ExampleFunction_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef ArnaudLegouxMovingAverage_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length, **float** Sigma, **float** Offset);

SCFloatArrayRef ExponentialRegressionIndicator_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef InstantaneousTrendline_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef CyberCycle_S(SCFloatArrayRef In, SCFloatArrayRef Smoothed, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef FourBarSymmetricalFIRFilter_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index);

SCFloatArrayRef SuperSmoother2Pole_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef SuperSmoother3Pole_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef ZeroLagEMA_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef Butterworth2Pole_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef Butterworth3Pole_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef DominantCyclePeriod_S(SCFloatArrayRef In, SCFloatArrayRef InstPeriod, SCFloatArrayRef Q1, SCFloatArrayRef I1, SCFloatArrayRef PhaseChange, SCFloatArrayRef Temp, SCFloatArrayRef MedianPhaseChange, SCFloatArrayRef DominantCycle, SCFloatArrayRef Out, **int** Index, **int** MedianLength);

SCFloatArrayRef LaguerreFilter_S(SCFloatArrayRef In, SCFloatArrayRef L0, SCFloatArrayRef L1, SCFloatArrayRef L2, SCFloatArrayRef L3, SCFloatArrayRef Out, **int** Index, **float** DampingFactor);

SCFloatArrayRef DominantCyclePhase_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index);

SCFloatArrayRef LinearRegressionSlope_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);

SCFloatArrayRef LinearRegressionIntercept_S(SCFloatArrayRef In, SCFloatArrayRef Out, **int** Index, **int** Length);