

MIMIC Formula Table

(From the MIMIC User Guide)



**12301 Research Blvd.
Building IV, Suite 410
Austin, TX 78759**

**U.S. Help Desk Phone: +1-800-546-9646 (or direct +1-512-697-3000), select ext. 3400
U.K. Help Desk Free Phone: 0800 032 6063
Europe Help Desk Phone: +44 20 7190 2947
Help Desk Email: support@lim.com
+1-512-697-3001 (Fax)**

Build Number: 4.1.0
Part Number: 087_44
Date: July 2, 2008

Copyright © 2000-2008 by Logical Information Machines, Inc.

Patented May, 1995 U.S. Patent No. 08/392, 612

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Logical Information Machines, Inc.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013.

Logical Information Machines, Inc.
120 North LaSalle Street
Suite 2150
Chicago, IL 60602

(312) 456-3000

Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

While every precaution has been taken in the preparation of this manual, we assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. Logical Information Machines, Inc. may revise this publication from time to time without notice.

Table of Contents

Formula Table	1
Overview	1
Formulas for Built-ins	2
% Move or Move	2
Absolute Value	2
Adjust Contract	3
Average	6
Bollinger High/Bollinger Low	6
Compound_return	7
Construct Date	7
Coregression	7
Coregression_intercept	8
Coregression_slope	9
Correlation	10
Count	10
Covariance	11
Dividend Adjusted Return	11
Ease of Move	11
Exponential Average	12
Extract Day	12
Extract Month	13
Extract Year	13
Greatest Integer	13
Highest	13
If-Then Statement	14
Inverse Log	14
Linear Regression	14
Lowest	16
Macd_osc	16
Macd_signal	16
Median	17
Midpoint or Mid_point	17
Move	18
Natural Log	18
Nearest Integer	18
Negative Value	19
Net Change	19
Net Percent Change	19
On Balance Volume	20
Periods in Range	20

Positive Value	20
Product	21
Range	21
Rel Strength	21
Select Contract	22
Simple_rsi	24
Smallest Integer	25
Standard Deviation	26
Standard Deviation Population	26
Sum	26
True Midpoint	27
True Range	27
Variance	28
Wilders_rsi	28
Williamsr	29
Formulas for Server Macros	30
_CoolingDegreeDays	30
_HeatingDegreeDays	30
_OffPeakAverage	31
_OnPeakAverage	31
_2x8Average	32
_2x16Average	33
_5x8Average	34
_5x16Average	35
_7x8Average	36
_7x16Average	36
AccDist	37
Aspect	38
AspectEquatorial	38
AspectHelio	39
AspectPlot	39
PlotAspectX	39
BackContract	40
BlackScholesOptionPrice	41
ChannelIndex	41
ChoppinessIndex	41
DeMarker	42
DeferredContract	42
DirMove	44
DirMoveIdx	44
FarContract	45
FrontContract	46
HistVolLog	47
HistVolPct	48
MoneyFlowIndex	48
NegVolIndex	48

NetHedgers	49
NetHedgersPct	49
NetLargeSpecs	49
NetLargeSpecsPct	50
NetSmallTraders	50
NetSmallTradersPct	50
NormalDist	51
RangeExpIndex	51
RealSum	51
Stochastics	52
Trendline	53
VolPriceTrend	53
Volatility	53
VolatilityIdx	54
WildersADX	54
WildersADXR	54
WildersDIMinus	55
WildersDIPlus	55
WildersDX	55
WildersRSI	56
WilliamsVolAD	56
Index	59

Formula Table

Overview

The following tables are provided to outline the structure or format of commonly used mathematical **Formulas** that may be typed directly into the **Formula** text field of the **Data Builder**. In addition, several examples of these **Formulas** have been provided in the body of each table.

Formulas for Built-ins

% Move or Move

Percent Move is the difference, expressed as a percentage, between a specified starting value and an ending specified value. Move calculates the difference between a specified starting value and an ending specified value.

$$\text{percentMove}(X, n) \equiv 100 \frac{X - X_n}{|X_n|}$$

If either X or X_n is a NaN, the immediately preceding value will be considered instead, i.e., X_{t-1} or X_{n+1} . If these values are also NaN, the percent move will be NaN as well. For example, when computing a 2 day percent move, if X is a NaN, but X_2 is not, then XMIM will return $X_{t-1} - X_2$ instead of $X - X_2$. But, if X_{t-1} is also NaN, then XMIM will return NaN.



Note that percentMove is negative precisely when the move is negative; i.e., the percent move from -10 to -12 is -20, not 20.

Constant (Number)	Time-Period (day/mth/yr)	"% move" or "move"	of	Series
4	day	% move	of	HO
1	month	move	of	CL
2	Quarter	% move	of	(HO - CL)
1	Month	move	of	(1 week average of HO)

Absolute Value

This study returns the absolute value for a specified Attribute. Where the value is a holiday or missing data series (expressed as NaN) the return will be NaN.

$$\text{absoluteValue}(X) \equiv |X|$$

If X is NaN, absoluteValue returns NaN as well.

"absolute_value"	of	Series
absolute_value	of	HO

Adjust Contract

MIMIC will adjust all studies using futures contract so that the artificial price jump associated with the expiration of a contract will not be factored into the study. For example, if a study using the front continuous contract captured a time period which included a roll or expiration date, by default, that price difference between the old or expiring contract and the new contract would be subtracted or backed out of the equation as of the expiration date. The `adjust_contract` function is used so that the jump or price difference associated with expiration (cost of carry) is not adjusted out of the study. There are four fields that are required to be specified within this study- the Attribute or series, the deferred number of the contract chosen (i.e., 1 represents the contract nearest to expiration or the front contract, 2 represents the second nearest contract to expiration, etc.), the number of days until expiration that the contract should be followed (i.e., 0 specifies that the contract will be active until expiration, 1 specifies that the contract will be active or followed up until 1 day before expiration, etc.) and contract month which is the parameter which specifies the contract month of interest (i.e., setting the field to 0 will disable this field thereby allowing the contract to roll to the regularly traded serial months, setting the field to 1 will specify only January contracts, etc.).

$$\text{adjustContract}(X, c, d, m) \equiv \hat{X}$$

where \hat{X} is computed by offsetting each relation of type futures in X with an appropriate amount so that the expiration-day boundaries do not introduce an artificial jump.

<code>adjust_contract</code>	“Series”	Contract Number	Days to Expiration	Month of Interest
<code>adjust_contract</code>	(HO,	1,	0,	0)
<code>adjust_contract</code>	(CL,	2,	5,	1)
<code>adjust_contract</code>	(NG,	3,	3,	0)



In the examples above the first example would create a series that when used in a **Formula** within another row of the worksheet would unadjust the continuous price series for the nearest to expire (or front) month. This series is set up to ‘roll’ on the expiration day. The next example, when used in a **Formula** within another row of the worksheet, would unadjust the continuous price series for the second nearest to expire (or back) January month (connects only January contracts). This series is set up to ‘roll’ five days before the expiration day of the contract. The last example, when used in a **Formula** within another row of the worksheet, would unadjust the continuous price series for the third nearest to expire (or back) contract. This series is set up to ‘roll’ three days before the expiration day of the contract.

The following shows further examples to demonstrate the use of adjust contract.

Example 1: Consider the trading day of 10/8/2002 for the following scenarios. The “prompt” crude oil contract would be Nov 2002, CL_2002X.

Data Table:

Date	LIM SYMBOL	adjust_contract version
10/8/2002	CL_2002Z - Close	adjust 1st December contract out adjust_contract (CL, 1, 0, 12)
	29.51	29.51
	CL_2003J - Close	adjust 6th contract month out adjust_contract (CL, 6, 0, 0)
	27.6	27.6
	CL_2004H - Close	adjust 2nd March contract out adjust_contract (CL, 2, 0, 3)
	23.86	23.86
	CL_2005F - Close	adjust 3rd January contract out adjust_contract (CL, 3, 2, 1)
	23.03	23.03

Let's look closer at each of these to see how the adjust_contract function chooses the correct contract.

December 2002 contract: This is the nearest December contract right now. The explicit LIM symbol is CL_2002Z. To ask for the closest contract month using adjust_contract you fill in adjust_contract (CL, 1, 0, 12), where:

- CL = base commodity symbol of choice.
- 1 = N nearest. 3 would imply the 3rd nearest, 6 the 6th nearest.
- 0 = This is "Number of days before Expiration". Setting this number to "0" indicates the series will locate the next nearest contract when the contract expires. Setting this number to "4" would mean that the contract should roll to the next one 4 days before expiration of the current contract.
- 12 = This is the contract month, 12 equals December, 5 equals May, 7 equals July.



Setting this number to "0" makes the parameter inactive, meaning the other parameters will determine the behavior.

April 2003 contract: This is the 6th nearest contract out. The explicit LIM symbol is CL_2003J. To ask for the literal N contract out using adjust_contract, you fill in adjust_contract (CL, 6, 0, 0), where:

- CL = base commodity symbol of choice.
- 6 = N nearest. 6 points to the 6th closest or nearest to expiration active contract.
- 0 = This is "Number of days before Expiration". Setting this to "0" indicates the series will locate the next nearest contract when the contract expires. Setting this to "4" would mean that the contract should roll to the next one 4 days before expiration of the current contract.

- 0 = Normally, this is where you put in the contract month. However, since we want the exact 6th closest contract, this is set to “0”.

March 2004 contract: This is the 2nd nearest March contract. The 1st nearest is the March 2003 contract. The 3rd nearest would be March 2005. To ask for the 2nd nearest March using `adjust_contract`, you would fill in `adjust_contract (CL, 2, 0, 3)` where:

- CL = base commodity symbol of choice.
- 2 = 2nd nearest
- 0 = This is “Number of days before Expiration”. Setting this to “0” indicates the series will locate the next nearest contract when the contract expires.
- 3 = March contract.

January 2005 contract: This is the 3rd nearest January contract. The closest or first nearest would be January 2003. To ask for the 3rd nearest using `adjust_contract`, you will fill in `adjust_contract (CL, 3, 2, 1)`, where:

- CL = base commodity symbol of choice
- 3 = 3rd nearest
- 2 = This is “Number of days before Expiration”. Setting this to “2” indicates the series will roll 2 days before the contract expires and choose the next 3rd nearest contract.
- 1 = January contract.

Example 2: More Real-World Examples

How do you ask for?

1. Spread between the 2nd December and the 1st December contract. Often times this is referred to as a Dec – Red Dec spread.

$$\text{adjust_contract (CL, 1, 0, 12)} - \text{adjust_contract (CL, 2, 0, 12)}$$

2. The spread between the 1st and 2nd nearby contracts.

$$\text{adjust_contract (CL, 1, 0, 0)} - \text{adjust_contract (CL, 2, 0, 0)}$$

3. The 3rd nearest November contract.

$$\text{adjust_contract (CL, 3, 0, 11)}$$

4. The 15th closest contract out.

$$\text{adjust_contract (CL, 15, 0, 0)}$$

Average

Calculates a simple moving average of values in the specified time period. When calculating an N day average, XMIM will display a moving N day average for any executable date; however, where there is a holiday or missing data point (expressed as NaN) a moving average of less than N days will be displayed. For example, if a 5 day average was calculated and our executable dates include December 25, a holiday, the study would calculate a 4 day average for this date because no data point exists for December 25. Likewise, on December 26 a 4 day average will be displayed because of the missing data associated with the holiday. On December 25, a 5 value average will display (NaN) because no data exists to perform the calculation. However on December 26, or the next date where data exists, the study include the data on December 26 and will search back for the other 4 values to perform the calculation (skipping over December 25). To ensure that N actual 'values' are used in the calculation the term value should be substituted instead of day in the formula.

$$\text{average}(X, n) \equiv \frac{\text{sum}(X, n)}{\text{count}(X, n)}$$

Constant (Number)	Time Period (day/mth/yr)	"average"	of	Series
5	value	Average	of	HO
3	month	Average	of	CL
2	quarter	Average	of	NG
1	year	Average	of	RB.UNL
4	week	Average	of	PN



The term "value" may be used to anchor 5 absolute values to be used in the calculation for the above example. If one of the five values were unavailable for the 5-day average, MIMIC would substitute a 4-day average. To ensure that 5 actual 'values' are used in the calculation the term value should be substituted instead of day in the formula. Also, on a day without data the use of 'value' in the formula would return a "NaN" (not a number, indicating that no values existed on that day for the calculation).

Bollinger High/Bollinger Low

Bollinger_high and Bollinger_low studies return values that are derived from a moving average plus (high) or minus some relative percent of the moving average. The relative value, subtracted or added to the moving average, is the standard deviation for the time window. These price bands can be defined so that they encompass most of the price activity with the ceiling (bollinger_high) coinciding with the relative highs and the floor (bollinger_low) coinciding with relative lows. These significant high and low thresholds can be used to measure overbought or oversold levels in the market. There are three fields that are required to be specified within this study- the Attribute or series, the relevant time period for the study and the width of the band (1 denotes 1 standard deviation, 2 denotes 2 standard deviations and 3 denotes 3 standard deviations).

$$\text{bollingerLow}(X, n, w) \equiv \text{average}(X, n) - w * \text{standardDeviation}(X, n)$$

$$\text{bollingerHigh}(X, n, w) \equiv \text{average}(X, n) + w * \text{standardDeviation}(X, n)$$

“bollinger_high/ bollinger_low”	Series	Constant (Number)	Time Period (day/mth/yr)	Constant (Number)
bollinger_high/bollinger_low	(HO,	1	day,	Standard Deviation)

Compound_return

Compound_return is the accumulated return of a specified relation column, between a specified start and end date. Since the formula relies on multiplication, all NaN values should be skipped at all times. If the relation column value is zero, then the value is 1. If the relation column is all NaN then the return value is NaN. For this study, it is practical to use % move as the attribute.

$$C_0 = 1$$

$$C_N = \left(1 + \frac{R_N}{100}\right) * C_{N-1}$$

where C = compound_return and R = relation column

“compound_return”	Constant (Number)	Time Period (day/mth/yr)	Attribute of Series	Time Period (day/mth/yr)	Time Period (day/mth/yr)
compound_return	1	day	% move of (GII.DELL.NASDAQ)	1 year ago	today

Construct Date

This study returns XMIM's mathematical value corresponding to the date month/day/year format.

$$\text{constructDate}(m, d, y) \equiv D$$

where D is an integer corresponding to the date *m/d/y* (month/day/year format). The number D is guaranteed to be in the standard date representation of XMIM, as given by the XMIM construct current date.

“construct_date”	Constant (Number)	Year
construct_date	(month, day,	1996)

Coregression

Returns the statistical co-regression of two series where X is the Independent variable and Y is the dependent variable.

$$\text{coregression}(X, Y, n) \equiv \alpha + \beta X$$

where α and β are the regression coefficients, computed as follows:

$$\Delta \equiv n \sum_{i=1}^n X_{n-i}^2 - \left(\sum_{i=1}^n X_{n-i} \right)^2$$

$$\alpha \equiv \frac{\sum_{i=1}^n X_{n-i}^2 \sum_{i=1}^n Y_{n-i} - \sum_{i=1}^n X_{n-i} \sum_{i=1}^n X_{n-i} * Y_{n-i}}{\Delta}$$

$$\beta \equiv \frac{n \sum_{i=1}^n X_{n-i} * Y_{n-i} - \sum_{i=1}^n X_{n-i} \sum_{i=1}^n Y_{n-i}}{\Delta}$$

As in the case for linear regression, the formulas above are unstable and prone to round-off errors, so we replace them with the mathematically equivalent ones below:

$$\beta \equiv \frac{1}{\sum_{i=1}^n \left(X_{n-i} - \sum_{j=1}^n X_{n-j} / n \right)^2} \sum_{i=1}^n \left(X_{n-i} - \sum_{j=1}^n X_{n-j} / n \right) Y_{n-i}$$

$$\alpha \equiv \frac{\sum_{i=1}^n Y_{n-i} - \beta \sum_{i=1}^n X_{n-i}}{n}$$

NaNs are handled exactly as in the case for linearRegression.

Constant (Number)	Time-Period (day/mth/yr)	“coregression”	of	Series
1	day	coregression	of	HO and CL

Coregression_intercept

Returns the statistical co-regression intercept of two series where X is the independent variable and Y is the dependent variable.

$$\text{coregression_intercept}(X, Y, n) \equiv \alpha$$

where α is the regression intercept, computed as follows:

$$\Delta \equiv n \sum_{i=1}^n X_{n-i}^2 - \left(\sum_{i=1}^n X_{n-i} \right)^2$$

$$\alpha \equiv \frac{\sum_{i=1}^n X_{n-i}^2 \sum_{i=1}^n Y_{n-i} - \sum_{i=1}^n X_{n-i} \sum_{i=1}^n X_{n-i} * Y_{n-i}}{\Delta}$$

As in the case for linear regression, the formulas above are unstable and prone to round-off errors, so we replace them with the mathematically equivalent ones below:

$$\beta \equiv \frac{1}{\sum_{i=1}^n \left(X_{n-i} - \sum_{j=1}^n X_{n-j} / n \right)^2} \sum_{i=1}^n \left(X_{n-i} - \sum_{j=1}^n X_{n-j} / n \right) Y_{n-i}$$

$$\alpha \equiv \frac{\sum_{i=1}^n Y_{n-i} - \beta \sum_{i=1}^n X_{n-i}}{n}$$

NaNs are handled exactly as in the case for linearRegression.

Constant (Number)	Time-Period (day/mth/yr)	“coregression_intercept”	of	Series
1	day	coregression_intercept	of	HO and CL

Coregression_slope

Returns the statistical co-regression slope of two series where X is the Independent variable and Y is the dependent variable.

$$\text{coregression_slope}(X, Y, n) \equiv \beta$$

where β is the regression slope, computed as follows:

$$\Delta \equiv n \sum_{i=1}^n X_{n-i}^2 - \left(\sum_{i=1}^n X_{n-i} \right)^2$$

$$\beta \equiv \frac{n \sum_{i=1}^n X_{n-i} * Y_{n-i} - \sum_{i=1}^n X_{n-i} \sum_{i=1}^n Y_{n-i}}{\Delta}$$

As in the case for linear regression, the formulas above are unstable and prone to round-off errors, so we replace them with the mathematically equivalent ones below:

$$\beta \equiv \frac{1}{\sum_{i=1}^n (X_{n-i} - \sum_{j=1}^n X_{n-j} / n)^2} \sum_{i=1}^n (X_{n-i} - \sum_{j=1}^n X_{n-j} / n) Y_{n-i}$$

NaNs are handled exactly as in the case for linearRegression.

Constant (Number)	Time-Period (day/mth/yr)	“coregression_slope”	of	Series
1	day	coregression_slope	of	HO and CL

Correlation

This study returns the statistical correlation coefficient between two series. This study will return values between the range of 1 (perfect correlation) and -1 (perfectly negative correlation).

$$\text{correlation}(X, Y, n) \equiv \frac{\sum_{i=0}^{n-1} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=0}^{n-1} (X_i - \bar{X})^2 \sum_{i=0}^{n-1} (Y_i - \bar{Y})^2}}$$

where \bar{X} is equal to average (X, n) and similarly for \bar{Y} . NaNs are handled exactly as in the case for covariance.

Constant (Number)	Time-Period (day/mth/yr)	“correlation”	of	Series
100	day	correlation	of	HO and CL
1	month	correlation	of	HO and NG
4	week	correlation	of	(HO + CL) and NG

Count

This study counts the number of occurrences for which a value is returned for a given time period. This function will not include holidays or other NaN (Not a Number) values in its count.

$$\text{count}(X, n) \equiv \sum_{i=0}^{n-1} \text{defined}(X_i)$$

where $\text{defined}(X)$ returns a 0 if X is NaN and 1 otherwise.

Constant (Number)	Time-Period (day/mth/yr)	“count”	of	Series
1	day	count	of	HO

Covariance

A statistical covariance measure is the measure of the relationship between two series whose values are observed over a moving time window.

$$\text{covariance}(X, Y, n) \equiv \frac{\text{sum}((X - \bar{X}) * (Y - \bar{Y}), n)}{n - 1}$$

where \bar{X} is equal to average (X, n) and similarly for \bar{Y} . As in the variance study, if n is less than or equal to 1, covariance returns NaN.

Constant (Number)	Time-Period (day/mth/yr)	"covariance"	of	Series
1	day	covariance	of	HO and CL

Dividend Adjusted Return

Dividend_adjusted_return is the overall percentage return of a dividend paying stock when it is assumed additional shares (or fractions of shares) are purchased with the dividend gains on Ex-Dividend date at the current stock's closing price. On each Ex-Dividend date, the amount of shares increases, but the initial investment remains the same. The formula assumes a purchase of 1 share at the start date at the then current closing price (this is the initial investment).

$$\left[\frac{((\text{accumulated_no_shares} * \text{current_share_price}) - \text{initial_investment})}{\text{initial_investment}} \right] * 100$$

"dividend_adjusted_return"	Attribute	of	Series	Time Period (day/mth/yr)	Time Period (day/mth/yr)
dividend_adjusted_return	(Close	of	GII.C.NYSE,	1 year ago	today)

Ease of Move

This study is derived from Richard Arms' Ease of Movement indicator and is a momentum indicator that incorporates range and volume to signal certain trends evolving within the markets. This indicator was developed in order to quantify volume and price changes in order to determine the ease, or lack thereof, with which price is able to move up or down. The formula for this study is the difference of the today's midpoint minus a previous midpoint multiplied by today's range all divided by today's volume.

$$\text{easeOfMove}(S) \equiv \frac{(M - M_1) * \text{range}(S)}{\text{Volume of } X}$$

where M is equal to midpoint (S) and M_1 is the previous value of midpoint (S).

“ease_of_move”	of	Series
ease_of_move	of	HO

Exponential Average

There exists a school of thought that says that using a simple moving average system, where all days are weighted equally, doesn't properly acknowledge the impact of more recent prices on future price movement. Hence, it is argued that it would be preferred to 'weight' the more recent days more heavily in a 'moving average' equation. This study is calculated as of the sum of the current price multiplied by a 'weight' or smoothing constant and the previous day's study is multiplied by $(1 - \text{weight})$. The weight of a prior day's price will decrease the further back in time, relevant to today's price, it is. There are two fields that are required to be specified within this study- the Attribute or series and weight. There are two acceptable ways to calculate weight: $1 / n$ (where n represents the number of days) or $2 / n + 1$. For example, if we were to use a 14 day equivalent, we could type in either $1 / 14$ or $2 / 14 + 1$ into the 'weight' field for our study.

$$\text{exponentialAverage}(X, w) \equiv w * X + (1 - w) \text{exponentialAverage}_1(X, w)$$

where w is restricted to lie in the range $w \in [0,1]$ and $\text{exponentialAverage}_1(X, w)$ is the previous value of $\text{exponentialAverage}(X, w)$ (or NaN if none). $\text{exponentialAverage}$ ignores any NaNs in X . That is, if a given value is NaN, then the $\text{exponentialAverage}$ is simply equal to the previous value of the $\text{exponentialAverage}$.

Although exponential average takes into account the entire history of X , it is sometimes useful to compare it to an “n-day” average. The closer w is to 1, the shorter term the exponential average becomes. From the formula, it is clear that for w equal to 1, the exponential average of X is precisely X . Conversely, the closer w is to zero, the longer term the exponential average becomes.

“exponential_average”	Series	Constant (Number)
exponential_average	(HO,	Weight)

Extract Day

This study returns XMIM's mathematical value corresponding to the day as encoded within the software.

$$\text{extractDay}(D) \equiv d$$

where d is the day encoded in XMIM date D . See the entry for “Construct Date” for an explanation.

“extract_day”	from	Date
extract_day	from	1996

Extract Month

This study returns XMIM's mathematical value corresponding to the month as encoded within the software.

$$\text{extractMonth}(D) \equiv m$$

where m is the month encoded in the XMIM date D . See the entry for “Construct Date” for an explanation.

“extract_month”	from	Date
extract_month	from	1996

Extract Year

This study returns XMIM's mathematical value corresponding to the year as encoded within the software.

$$\text{extractYear}(D) \equiv y.$$

where y is the year encoded in the XMIM date D . See the entry for “Construct Date” for an explanation.

“extract_year”	from	Date
extract_year	from	1996

Greatest Integer

This study returns a ceiling for a series using the some units specified as “steps”. There are two fields that are required to be specified within this study- the Attribute or series (this may also be a constant/number or variable) and step. This study will return a value that is greater or equal to the Attribute and that is a multiple of the number specified in the steps field. For example, if the Close of US was specified as the Attribute and ‘2’ was specified in the steps field, the study would return a number that was divisible by 2 and that was greater than the daily closing price.

$$\text{greatestInteger}(N, s) \equiv s * \left\lceil \frac{N}{s} \right\rceil$$

“greatest_integer”	Series	Constant (Number)
greatest_integer	(HO,	Step)

Highest

Returns the highest value in a data series during a specified period of time.

Identification of a succession of higher highs and higher lows is a technical method commonly used to designate an uptrend and may be used to identify certain trending patterns within your studies.

$$\mathit{highest}(X, n) \equiv \max_{i=0}^{n-1} X_i$$

This study will ignore any NaN values. If all the X values are NaN, it will return NaN. Otherwise, it will return the maximum of the non-NaN values.

Constant (Number)	Time Period (day/mth/yr)	“highest”	of	Series
1	day	highest	of	HO

If-Then Statement

This language structure lets you assign a **Series** based on time, date, event or price conditions. It creates a logical structure so that if the condition in the ‘If’ clause is true, the **Series** represented by the ‘Then’ clause is assigned on that date.

if	Condition	then	Series	endif
if	Date is NG expiration_day	then	HO	endif
if	HO is more than CL	then	CL	endif
if	1 week % move of HO is more than 10	then	1 week % move of HO	endif

Inverse Log

This study returns the inverse logarithm for a specified Attribute or series.

$$\mathit{inverseLog}(X) \equiv e^x$$

If X is NaN, its inverseLog is also NaN.

“inverse_log”	of	Series
Inverse_log	of	HO

Linear Regression

Returns a value for a given attribute based on a linear regression (fitting data to a line, such as $y=ax + b$) over a specified time period. Linear regression is a statistical method of following trends. According to

industry authorities, the results of linear regression studies are quite similar to the moving averages. This study performs a linear regression over a specified time period.

$$\text{linearRegression}(X, n) \equiv \alpha + \beta n$$

where α and β are the linear regression coefficients, computed as follows:

$$\Delta \equiv n \sum_{i=1}^n i^2 - \left(\sum_{i=1}^n i \right)^2$$

$$\alpha \equiv \frac{\sum_{i=1}^n i^2 \sum_{i=1}^n X_{n-i} - \sum_{i=1}^n i \sum_{i=1}^n i * X_{n-i}}{\Delta}$$

$$\beta \equiv \frac{n \sum_{i=1}^n i * X_{n-i} - \sum_{i=1}^n i \sum_{i=1}^n X_{n-i}}{\Delta}$$

The formulas stated above are the standard formulas for linear regression; however, they are not well-suited to a computer implementation, because they suffer from severe round-off errors. To alleviate this, XMIM implements the equivalent formulas below instead:

$$\beta \equiv \frac{1}{\sum_{i=1}^n \left(i - \sum_{j=1}^n j/n \right)^2} \sum_{i=1}^n \left(i - \sum_{j=1}^n j/n \right) X_{n-i}$$

$$\alpha \equiv \frac{\sum_{i=1}^n X_{n-i} - \beta \sum_{i=1}^n i}{n}$$

Note that some of the expressions above are actually simple functions of n , i.e.,

$$\sum_{i=1}^n i \equiv \frac{n(n+1)}{2}$$

In general, XMIM will pre-compute these expressions, and so the formulas above are greatly simplified (and largely unrecognizable). LinearRegression ignores any NaN values in X. That is, if 3 values are NaN in the computation of linearRegression(X, 10), XMIM computes linearRegression(X' 7) instead, where the X' are derived from X by “stripping” away the NaN values. Note in particular that the placement of NaNs throughout the X time series is not significant.

Constant (Number)	Time Period (day/mth/yr)	“linear_regression”	of	Series
1	day	linear_regression	of	HO

Lowest

Returns the lowest value in a data series during a specified period of time. This function is helpful in identifying lower highs and lower lows, characterizing an downward trend pattern.

$$\text{lowest}(X, n) \equiv \min_{i=0}^{n-1} X_i$$

This study will ignore any NaN values. If all the X values are NaN, it will return NaN. Otherwise, it will return the minimum of the non-NaN values.

Constant (Number)	Time Period (day/mth/yr)	“lowest”	of	Series
1	day	lowest	of	HO

Macd_osc

This study’s name is short hand for moving average convergence divergence oscillator. This indicator was developed by Gerald Appel and is widely employed to measure market momentum as well as detecting and following trends. This study measures the rate of convergence and divergence between two exponentially smoothed moving averages (or the rate of acceleration or deceleration between a faster or shorter period moving average and a slower or longer period moving average). There are three fields that must required to be specified within this study- the Attribute or series and two fields representing the weights for the faster and slower exponentially smooth moving averages. This study will return the difference between the two moving averages. A value of ‘0’ would indicate that the averages are exactly the same whereas a positive value (where the faster average was the first defined weight field) would indicate acceleration and a negative value would indicate deceleration.

$$\text{macdOsc}(X, w_1, w_2) \equiv \text{exponentialAverage}(X, w_1) - \text{exponentialAverage}(X, w_2)$$

For an explanation of the w_1 , w_2 and w_3 arguments, see the discussion about “Exponential Average”.

“macd_osc”	Series	Constant (Number)	Constant (Number)
macd_osc	(HO,	Weight 1	Weight 2)

Macd_signal

This study incorporates a ‘signal’ line into the study, macd_osc, listed above. Where the first two fields represent the faster and slower moving averages in the macd_osc study, the third weight represents the exponential weight of this macd_osc study. So in effect this study is an exponential study, as defined by the third weight, of macd_osc created by the first two weight fields. The basic method of incorporating this study

into a trading program is to look for buy signals when the `macd_osc` crosses above the `macd_signal` and to look for sell signals upon the reverse.

$$\text{macdSignal}(X, w_1, w_2, w_3) \equiv \text{exponentialAverage}(\text{macdOsc}(X, w_1, w_2), w_3)$$

For an explanation of the w_1 , w_2 , and w_3 arguments, see the discussion about “Exponential Average”.

“macd_signal”	Series	Constant (Number)	Constant (Number)	Constant (Number)
macd_signal	(HO,	Weight 1	Weight 2	Weight 3)

Median

Returns the median value in a data series. Median function first sorts the data series, and then returns the median value. If the data item number in the series is odd, it returns the middle data item in the series. Otherwise, it returns the middle two data item average. For example, if the data series is (10, 4, 7), the median value is 7. If the data series is (10, 2, 3, 27), the median value is $(3+10)/2=6.5$

$$\text{Median}(X, n) = y$$

This study will ignore any NaN values. If all the X values are NaN, it will return a NaN. Otherwise, it will return the non-NaN median value.

“median”	Constant (Number)	Time Period (day/mth/yr)	“median”	of	“Close”	of	Series
	3	day	median	of	Close	of	CL

Midpoint or Mid_point

This study returns the mathematical midpoint for a specified Attribute and is calculated by dividing the sum, of the High of the Attribute and the Low of the Attribute, by 2. A popular method of employing this study is to gauge a 50% market retracement (as defined as some point below the midpoint), off from some specified high, as potential entry points for a trading program. The syntax “midpoint” or “mid_point” may be used.

$$\text{Midpoint}\left(\frac{X+Y}{2}\right)$$

Where X is the High of the Attribute and Y is the Low of the Attribute.

“midpoint” or “mid_point”	of	Series
midpoint	of	HO
mid_point	of	HO

Move

Returns the difference between the start and end values over a specified time period.

$$\text{move}(X, n) \equiv X - X_n$$

If either X or X_n is a NaN, the immediately preceding value will be considered instead, i.e., X_{t-1} or X_{n+1} . If these values are also NaN, the move will be NaN as well. For example, when computing a 2 day move, if X is a NaN, but X_2 is not, then XMIM will return $X_{t-1} - X_2$ instead of $X - X_2$. But, if X_{t-1} is also NaN, then XMIM will return NaN.

Constant (Number)	Time-Period (day/mth/yr)	“move”	of	Series
1	day	move	of	HO

Natural Log

This study returns the natural logarithm for a specified Attribute or series.

$$\text{naturalLog}(X) \equiv \ln X$$

If X is NaN or less than or equal to 0, naturalLog returns NaN

“natural_log”	of	Series
natural_log	of	HO

Nearest Integer

This study will return either the smallest integer or the greatest integer, depending upon which value is closest to the value of the Attribute or series selected. This study may be used to calibrate where the value of the Attribute is relative to some ceiling (greatest integer) or floor (smallest integer).

$$\text{nearestInteger}(N, s) \equiv \text{greatestInteger}(N, s)$$

or

$$\text{nearestInteger}(N, s) \equiv \text{smallestInteger}(N, s)$$

depending on which of the two values is closer to N . If N is the midpoint of these two values, XMIM chooses greatestInteger (N, s) arbitrarily.

“nearest_integer”	Series	Constant (Number)
Nearest_integer	(HO	Step)

Negative Value

This study will only return the value of the Attribute specified where the value is a negative number. Where the value is positive or zero the value returned is '0'.

$$\text{negativeValue}(X) \equiv \min(X, 0)$$

Notice that this function never returns a positive number. If X is NaN, negativeValue returns NaN as well.

“negative_value”	of	Series
Negative_value	of	HO

Net Change

This study is the same as the “move study”, but the sign of the change is removed. Net change is the absolute value of the move.

$$\text{netChange}(X, n) \equiv |X - X_n|$$

This study treats NaNs analogously to move. It also performs the same special cases.

Constant (Number)	Time-Period (day/mth/yr)	“net_change”	of	Series
1	day	net_change	of	HO

Net Percent Change

This study returns the absolute value of the percentage move over some time period specified.

$$\text{netPercentChange}(X, n) \equiv 100 \left| \frac{X - X_n}{X_n} \right|$$

This study treats NaNs analogously to move. It also performs the same special cases.

Constant (Number)	Time-Period (day/mth/yr)	“net_percent_change”	of	Series
1	day	net_percent_change	of	HO

On Balance Volume

This study quantifies the trend in volume over time and is most closely identified with research conducted by Joseph Granville. There are several methods by which this study may be analyzed. The primary methods are trend assessment and divergence analysis. The study returns a cumulative value representing the running total equated to the sum of (daily or periodic volume multiplied by 1, where the daily or periodic move is positive, or by -1, where the move is negative). This study is a ‘running’ calculation based on market direction and volume. The only field required to be specified for the study is the Attribute.

$$\text{onBalanceVolume}(S) \equiv \text{onBalanceVolume}_1(S) + \text{sign} * \text{Volume of } S$$

where *sign* is 1 if Close of S is up and -1 otherwise. The expression onBalanceVolume1(S) refers to the previous value of onBalanceVolume(S).

"on_balance_volume"	of	Series
on_balance_volume	of	HO

Periods in Range

This study returns the number of periods (days, weeks, etc.) in some specified window time. The most interesting use of this study is to calibrate the number of periods or days between two given conditions. For example, the periods in range from the time when some value crosses above a moving average to when the value crosses below. This example could assist in identifying the period of time a certain trade would be active based on a moving average system.

$$\text{periodsInRange}(n) \equiv \#n$$

where $\#n$ counts the number of periods (days, weeks, etc.) in the time period n . If this time period is specified as a from/to range, and the boundaries are reversed (i.e., the “from” occurs after the “to”), periodsInRange returns a negative number.

"periods_in_range"	Constant (Number)	Time Period (day/mth/yr)
periods_in_range	1	day

Positive Value

This study will only return the value of the Attribute specified where the value is a positive number. Where the value is negative or zero the value returned is ‘0’.

$$\text{positiveValue}(X) \equiv \max(X, 0)$$

If X is NaN, `positiveValue` returns NaN as well.

“positive_value”	of	Series
positive_value	of	HO

Product

This study will return the product (the multiplication of some values to each other) of those values of an Attribute as defined over some period of time. Where there exists a holiday or missing data (expressed by NaN) the equation is multiplied by ‘1’. There are two fields that must be specified within this study- the Attribute or series and time period.

$$product(X, n) \equiv \prod_{i=0}^{n-1} X_i$$

This study will ignore any values that are NaN. That is, it returns the product of the non-NaN values in the given range. If all the X values are NaN, product returns a NaN.

“product”	Series	Time Period (day/mth/yr)
product	(HO,	day)

Range

This study will return the range for some specified Attribute by subtracting the High of the Attribute - the Low of the Attribute. Trading ranges, performed for certain key periods (i.e. calendar month or 6 months), are used to establish the upper and lower boundaries that are essential to defining levels of support and resistance. Breakouts from established trading ranges provide important trading signals.

$$range(S) \equiv \text{High of } S - \text{Low of } S$$

“range”	of	Series
range	of	HO

Rel Strength

This study returns the difference between the period percent move of one data series versus another defined series. This study is useful for attribution analysis (research that defines what portion of a security’s price is

‘attributable’ to the broad market moves). The most interesting use of this study is to compare the price of a security or market sector versus its comparable broad market index to see whether the security or sector is outpacing or under-performing the index.

$$\text{relStrength}(S, I, n) \equiv \text{percentMove}(S, n) - \text{percentMove}(I, n)$$

“rel_strength”	Sec Series	Index Series	Constant (Number)
rel_strength	(HO,	CL,	NumDays)

Select Contract

The MIM will adjust all studies using futures contract so that the artificial price jump associated with the expiration of a contract will not be factored into the study. For example, if a study using the front continuous contract captured a time period which included a roll or expiration date, by default, that price difference between the old or expiring contract and the new contract would be subtracted or backed out of the equation as of the expiration date. The select_contract function is used so that the jump or price difference associated with expiration (cost of carry) is not adjusted out of the study. This is accomplished by ‘anchoring’ the current contract that satisfies our expiration requirements (i.e. if the front contract is the March 1998 contract, the studies will be performed exclusively on this contract and will ignore all previous contracts that meet our expiration requirement). There are four fields that are required to be specified within this study- the Attribute or series (select your futures series), the deferred number of the contract chosen (i.e., 1 represents the contract nearest to expiration or the front contract, 2 represents the second nearest contract to expiration, etc.), the number of days until expiration that the contract should be followed (i.e., 0 specifies that the contract will be active until expiration, 1 specifies that the contract will be active or followed up until 1 day before expiration, etc.) and contract month which is the parameter which specifies the contract month of interest (i.e., setting the field to 0 will disable this field thereby allowing the contract to roll to the regularly traded serial months, setting the field to 1 will specify only January contracts, etc.).

$$\text{selectContract}(X, c, d, m) \equiv \hat{X}$$

where \hat{X} is computed by replacing all references to a relation in X with the corresponding “current” contract. The current contract is selected as in adjustContract.

“select_contract”	Series	Constant (Number)	Constant (Number)	Constant (Number)
select_contract	(HO,	DeferredNum,	NumDays,	ContractMonth)

The following shows further examples to demonstrate the use of select_contract.

Example 1:

Consider the trading day of 10/8/2002 for the following scenarios.

The “prompt” crude oil contract would be Nov 2002, CL_2002X .

Data Table

Date	LIM SYMBOL	adjust_contract version
10/8/2002	CL_2002Z - Close	adjust 1st December contract out select_contract (CL, 1, 0, 12)
	29.51	29.51
	CL_2003J - Close	adjust 6th contract month out select_contract (CL, 6, 0, 0)
	27.6	27.6
	CL_2004H - Close	adjust 2nd March contract out select_contract (CL, 2, 0, 3)
	23.86	23.86
	CL_2005F - Close	adjust 3rd January contract out select_contract (CL, 3, 2, 1)
	23.03	23.03

Let's look closer at each of these to see how the select_contract function chooses the correct contract.

December 2002 contract: This is the nearest December contract right now. The explicit LIM symbol is CL_2002Z. To ask for the closest contract month using select_contract you fill in select_contract (CL, 1, 0, 12), where:

- CL = base commodity symbol of choice.
- 1 = N nearest. 3 would imply the 3rd nearest, 6 the 6th nearest.
- 0 = This is "Number of days before Expiration". Setting this number to "0" indicates the series will locate the next nearest contract when the contract expires. Setting this number to "4" would mean that the contract should roll to the next one 4 days before expiration of the current contract.
- 12 = This is the contract month, 12 equals December, 5 equals May, 7 equals July.



Setting this number to "0" makes the parameter inactive, meaning the other parameters will determine the behavior.

April 2003 contract: This is the 6th nearest contract out. The explicit LIM symbol is CL_2003J. To ask for the literal N contract out using select_contract, you fill in select_contract (CL, 6, 0, 0), where:

- CL = base commodity symbol of choice.
- 6 = N nearest. 6 points to the 6th closest or nearest to expiration active contract.
- 0 = This is "Number of days before Expiration". Setting this to "0" indicates the series will locate the next nearest contract when the contract expires. Setting this to "4" would mean that the contract should roll to the next one 4 days before expiration of the current contract.

- 0 = Normally, this is where you put in the contract month. However, since we want the exact 6th closest contract, this is set to “0”.

March 2004 contract: This is the 2nd nearest March contract. The 1st nearest is the March 2003 contract. The 3rd nearest would be March 2005. To ask for the 2nd nearest March using `select_contract`, you would fill in `select_contract (CL, 2, 0, 3)` where:

- CL = base commodity symbol of choice.
- 2 = 2nd nearest
- 0 = This is “Number of days before Expiration”. Setting this to “0” indicates the series will locate the next nearest contract when the contract expires.
- 3 = March contract.

January 2005 contract: This is the 3rd nearest January contract. The closest or first nearest would be January 2003. To ask for the 3rd nearest using `select_contract`, you will fill in `select_contract (CL, 3, 2, 1)`, where:

- CL = base commodity symbol of choice
- 3 = 3rd nearest
- 2 = This is “Number of days before Expiration”. Setting this to “2” indicates the series will roll 2 days before the contract expires and choose the next 3rd nearest contract.
- 1 = January contract.

Example 2:

```
SHOW
  NGjan: select_contract ( NG, 1, 0, 0 )
  NGfeb: select_contract ( NG, 2, 0, 0 )
  NGmar: select_contract ( NG, 3, 0, 0 )
  avg:
    (select_contract ( NG, 1, 0, 0 ) + select_contract ( NG, 2, 0, 0 ) +
     select_contract ( NG, 3, 0, 0 )
    ) / 3
WHEN
  Date is December
```

Simple_rsi

Simple relative strength index is a variation of Welles Wilder's Relative Strength Index indicator and is a popular overbought-oversold measure. These measures are intended to reflect situations when prices have risen or fallen too sharply and are thus vulnerable to a market reaction. Wilder's version uses a form of exponential smoothing where `simple_rsi` uses a simple moving average. This study compares the relative strength of prices gains (closes that are higher than the previous day's close) versus price losses (closes

that are lower than the previous day's close) by averaging these values for some specified period of time. Simple_rsi may be performed for any time period. Wilder originally suggested 14 days, however, technicians sometimes prefer a shorter/more sensitive period of time such as 5 or 7 days. Industry standards derive the overbought and oversold indicators at levels that range around 70 (overbought) and 30 (oversold).

$$\text{simpleRSI}(X, n) \equiv 100 - \frac{100}{1 + U/D}$$

where U and D are the average of the 1-period up-and down-moves of X, respectively, over the period n . That is, U and D are given by

$$U \equiv \text{average}(\text{positiveValue}(\text{move}(X, 1)), n)$$

$$D \equiv -\text{average}(\text{negativeValue}(\text{move}(X, 1)), n)$$



Note in the definition of D that D is always positive, since negativeValue is never positive.

"simple_rsi"	Series	Constant (Number)	Time Period (day/mth/yr)
simple_rsi	(HO,	1	day)

Smallest Integer

This study returns the floor for a series using units specified as 'steps'. There are two fields that are required to be specified within this study- the Attribute or series (this may also be a constant/number or variable) and step. This study will return a value that is lesser or equal to the Attribute and that is a multiple of the number specified in the steps field. For example, if the Close of US was specified as the Attribute and '2' was specified in the steps field, the study would return a number that was divisible by 2 and that was lesser than or equal to the daily closing price.

$$\text{smallestInteger}(N, s) \equiv s * \left\lfloor \frac{N}{s} \right\rfloor$$

That is, smallestInteger returns the floor of N using the units in s . For example, $\text{smallestInteger}(5, 2)$ is equal to 4, because 4 is the largest multiple of 2 less than or equal to 5. Similarly, $\text{smallestInteger}(5.3, .25)$ is equal to 5.25.

"smallest_integer"	Series	Constant (Number)
smallest_integer	(HO,	step)

Standard Deviation

This study provides the standard deviation of a population based on the sample as defined by the values included in the specified time period. Standard deviation measures the spread from the mean (or measures how far from the center or mean the data tends to range). This study is a commonly used statistic used to measure the degree of dispersion in the data.

$$\text{standardDeviation}(X, n) \equiv \sqrt{\text{variance}(X, n)}$$

This corresponds to the standard deviation used for population samples.

Constant (Number)	Time Period (day/mth/yr)	“standard_deviation”	of	Series
1	month	standard_deviation	of	HO
1	quarter	standard_deviation	of	(2 * CL)
2	year	standard_deviation	of	1 day move of HO

Standard Deviation Population

Returns a moving population standard deviation of values in the time period (uses n as a divisor) based on the population itself, that is, the values in the period.

$$\text{standardDeviationPopulation}(X, n) \equiv \sqrt{\frac{\text{sum}((X - \text{average}(X, n))^2, n)}{n}}$$

For efficiency and stability, this formula is actually computed as follows:

$$\text{standardDeviationPopulation}(X, n) \equiv \sqrt{\frac{\text{sum}(X^2, n) - \text{sum}(X, n) * \text{average}(X, n)}{n}}$$

Constant (Number)	Time Period (day/mth/yr)	“standard_deviation_population”	of	Series
1	day	standard_deviation_population	of	HO

Sum

Returns the sum total of all the values in the series during some specified time period. Holidays and missing values (expressed as Nan) have no effect on the sum. As with any study, if you want the sum to always include a set number of periods or days than use the term ‘value’ in lieu of day.

$$\text{sum}(X, n) \equiv \sum_{i=0}^{n-1} X_i$$

This study will ignore any values that are NaN. That is, it returns the sum of all the non-NaN values in the period n . If all the X values are NaN, sum returns a NaN.

Constant (Number)	Time Period (day/mth/yr)	“sum”	of	Series
1	day	sum	of	HO

True Midpoint

This study is very similar to the 'midpoint' study except 'High' is defined as the true High or the greater of either today's or yesterday's High and 'Low' is defined as the true Low or lesser of either today's or yesterday's Low. For most days, the true High will be identical to today's High and the true Low will be identical to today's low. The differences, where the previous day's value is selected, will occur on downside gap days (the range is below yesterday's range) or conversely upside gap days (the range is above yesterday's range). By using the 'true' value, we can factor in the downside or upside gap in price.

$$\text{trueMidpoint}(S) \equiv \frac{H + L}{2}$$

where H is the higher of the High of S and the previous Close of S , and L is the lower of the Low of S and the previous Close of S .

“true_midpoint”	of	Series
true_midpoint	of	HO

True Range

The True Range indicator is the greatest of the following:

- The current High less the current Low
- The absolute value of: current High less the previous Close
- The absolute value of: current Low less the previous Close

$\text{trueRange}(S) = \text{maximum}((\text{High}-\text{Low}), (|\text{High}-\text{Close 1 day ago}|), (|\text{Low}-\text{Close 1 day ago}|))$

“true_range”	of	Series
true_range	of	HO

Variance

This study returns the statistical variance of a population based on the values specified in the time period. Variance is the square of the standard deviation, which is a measure of data dispersion. A data series that has values that are widely dispersed will have a high standard deviation and variance.

$$\text{variance}(X, n) \equiv \frac{\text{sum}((X - \text{average}(X, n))^2, n)}{n - 1}$$

We choose to use a different but equivalent formula to compute the above expression, so that it can be computed in a single pass. The actual formula used is:

$$\text{variance}(X, n) \equiv \frac{\text{sum}(X^2, n) - \text{sum}(X, n) * \text{average}(X, n)}{n - 1}$$

In the cases where there are less than two non-NaN values in n , variance returns NaN.

Constant (Number)	Time Period (day/mth/yr)	“variance”	of	Series
1	day	variance	of	HO

Wilders_rsi

Returns the Welles Wilder’s Relative Strength Index indicator, a popular overbought-oversold measure. This study is very similar to `simple_rsi`, except Wilder’s version uses a form of exponential smoothing where `simple_rsi` uses a simple moving average.

This study compares the relative strength of prices gains (closes that are higher than the previous day’s close) versus price losses (closes that are lower than the previous day’s close) by using an exponentially smooth average of these values for some specified period of time.

Like `simple_rsi`, `wilders_rsi` may be performed for any time period. Wilder originally suggested 14 days, however, technicians sometimes prefer a shorter/more sensitive period of time such as 5 or 7 days. This study will range between 0 and 100.

Industry standards derive the overbought and oversold indicators at levels that range around 70 (overbought) and 30 (oversold).

$$\text{wildersRSI}(X, w) \equiv 100 - \frac{100}{1 + U / D}$$

where U and D are the exponential average of the 1-period up-and down-moves of X , respectively, using the weight w . That is, U and D are given by

$$U \equiv \text{exponentialAverage}(\text{positiveValue}(\text{move}(X, 1)), w)$$

$$D \equiv -\text{exponentialAverage}(\text{negativeValue}(\text{move}(X, 1)), w)$$

For an explanation of the weight w , see the discussion in [“Exponential Average”](#).

$$\text{williamsR}(S) \equiv 100 \frac{\text{Close of } S - \text{Low of } S}{\text{High of } S - \text{Low of } S}$$

“wilders_rsi”	Series	Constant (Number)
wilders_rsi	(HO,	weight)

Williamsr

This study is a momentum indicator and is used to calibrate the position of the Close relative to the High for some period. Successive values of 100 for the study may indicate an upward trend since the price of a series closes at its High. Conversely, successive values of 0 may indicate a downward trend since the price of the series closes at its Low.

“williamsr”	of	Series
williamsr	of	HO

Formulas for Server Macros



When creating formulas with server macros, use the symbol names. Using Row Labels with server macros when constructing formulas is not supported.

_CoolingDegreeDays

Originally designed to evaluate energy demand and consumption, degree days are based on how far the average temperature departs from a human comfort level of 65 °F*. Simply put, each degree of temperature *above* 65 °F is counted as one “cooling degree day”. For example, a day with an average temperature of 80 °F will have 15 cooling degree days. Source: www.weather2000.com/dd_glossary.html.

_CoolingDegreeDays (Series)

Example:



The Series input is always the symbol for a city.

```
SHOW
  cdd: _CoolingDegreeDays ( HOUSTON.TX )
WHEN
  Date is August
```

The query above will calculate the daily cooling degree days during the month of August in Houston,TX . As you can see it gets quite warm.

“_CoolingDegreeDays”	City
_CoolingDegreeDays	(HOUSTON.TX)

_HeatingDegreeDays

Originally designed to evaluate energy demand and consumption, degree days are based on how far the average temperature departs from a human comfort level of 65 °F*. Simply put, each degree of temperature *below* 65 °F is counted as one “heating degree day”. For example, a day with an average temperature of 60 °F will have 5 heating degree days. Source: www.weather2000.com/dd_glossary.html.

_HeatingDegreeDays (Series)

Example:



The Series input is always the symbol for a city.

```
SHOW
  hdd: _HeatingDegreeDays ( CHICAGO.OHARE.IL )
WHEN
  Date is February
```

The query above will calculate the daily heating degree days during the month of February in Chicago,IL . Bring a jacket!

"_HeatingDegreeDays"	City
_HeatingDegreeDays	(CHICAGO.OHARE.IL)

_OffPeakAverage

This macro is designed to calculate the Off Peak Average as it relates to hourly power prices. Many ISO data feeds are available from LIM. (PJM, NYISO, NEPOOL, IMO, and many European ones as well).

_OffPeakAverage (Series)

Example:



The Series input will be the symbol for a particular zone/node. Run this using “hourly” execution units.

```
SHOW
  1: _OffPeakAverage ( PJM.WESTERNHUB.HUB.RT )
WHEN
  Date is within 1 week
```

The query above will calculate the Off Peak Average for the real-time LMP prices on the Western Hub area of PJM. There are some assumptions being made in the calculation. These have been collected from the industry and seem to be standard. Namely, on NERC holidays the Off Peak Average uses all hours in the day. On normal days (non-NERC holidays) hours 1-7 and hour 24 is used in the calculation.

"_OffPeakAverage"	Series
_OffPeakAverage	(PJM.WESTERNHUB.HUB.RT)

_OnPeakAverage

This macro is designed to calculate the On Peak Average as it relates to hourly power prices. Many ISO datafeeds are available from LIM. (PJM, NYISO, NEPOOL, IMO, and many European ones as well).

`_OnPeakAverage` (Series)

Example:



The Series input will be the symbol for a particular zone/node. Run this using “hourly” execution units.

```
SHOW
  1: _OnPeakAverage ( NYISO_DUNWOD_TWRT )
WHEN
  Date is within 3 weeks
AND
  Date is from Monday to Friday
AND
  Time is 03:00 pm
```

The query above will calculate the On Peak Average for the real-time LMP prices in ZONE I (Dunwoodie) of NYISO. There are some assumptions being made in the calculation. These have been collected from the industry and seem to be standard. The On Peak average is calculated during Monday to Friday only. Hours 8 – 23 are considered in the average. Notice that you can have 1 number per day returned if you anchor the WHEN condition with a time. This is only an anchor and will not affect the results of the query.

<code>“_OnPeakAverage”</code>	Series
<code>_OnPeakAverage</code>	<code>(NYISO_DUNWOD_TWRT)</code>

These macros below (`_2x8Average` through `_7x16Average`) will only be relevant when looking at intraday data (e.g. hourly) in the electricity markets. The typical sources of this data from LIM are the Independent System Operators (ISO’s) in North America, which include PJM, (Pennsylvania Jersey Maryland), NYISO (New York), NEISO (New England) and MISO (Midwest) – in particular the macros will only work in those markets that use LMP pricing methods. The user should note that different US regions have different concepts of peak and off-peak and that generally speaking, the Western region treats Saturday as a peak period and the Eastern region does not.

These types of calculation may also be relevant for European Electricity markets with intra-day pricing as well, although definitions of peak and off-peak pricing again will vary by region.

`_2x8Average`

This macro is designed to calculate the average for Saturday and Sunday from 12am – 6pm and 11pm.

`_2x8Average` (Series)

Example:



The Series input will be the symbol for a particular zone/node. The column “ImpVal” is assigned to each Series in the macro script. Therefore only symbols with the column, LmpVal can be used in this macro. Run this using “hourly” execution units.

```
SHOW
  1: _2X8Average ( A418.138KV.TR7212.PJMRT )
WHEN
  Date is within 1 week
```

The query above will calculate the average for hours 1-7 and hour 24 on Saturday and Sunday for the real-time LMP prices for the Pnode ID: 32408579 for PJM. Notice in the macro code below, there are some assumptions being made in the calculation. These have been collected from industry and seem to be standard.

“_2x8Average”	Series
_2x8Average	(A418.138KV.TR7212.PJMRT)

Macro Code:

```
ATTR MACRO _2X8Average ( SECURITY PricePoint)
RETURN
IF Date is Saturday
OR Date is Sunday
THEN average from 12:00 am to 11:00 pm of
IFTime is from 12:00 am to 06:00 am
OR Time is 11:00 pm
THEN lmpval of PricePoint ENDIF ENDIF
ENDMACRO
```

_2x16Average

This macro is designed to calculate the average for Saturday and Sunday from 7am – 10pm.

_2x16Avaerage (Series)

Example:



The Series input will be the symbol for a particular zone/node. The column “ImpVal” is assigned to each Series in the macro script. Therefore only symbols with the column, LmpVal can be used in this macro. Run this using “hourly” execution units.

```
SHOW
  1: _2X16Average ( NEPOOL.Z.CONNECTICUT.DA )
WHEN
  Date is within 3 months
  AND
  Time is 04:00 pm
  AND
  Date is from Saturday to Sunday
```

The query above will calculate the average for hours 8-23 on Saturday and Sunday for the real-time LMP prices for the New England ISO LMP LOAD ZONE Day Ahead Market. Notice in the macro code below, there are some assumptions being made in the calculation. These have been collected from industry and seem to be standard.

“_2x16Average”	Series
_2x16Average	(NEPOOL.Z.CONNECTICUT.DA)

Macro Code:

```
ATTR MACRO _2X16Average ( SECURITY PricePoint )
RETURN
IF Date is Saturday
OR Date is Sunday
THEN average from 07:00 am to 10:00 pm of lmpval of
PricePoint ENDIF
ENDMACRO
```

_5x8Average

This macro is designed to calculate the average for Monday through Friday from 12am-6am and 11pm.

_5x8Average (Series)

Example:



The Series input will be the symbol for a particular zone/node. The column “lmpVal” is assigned to each Series in the macro script. Therefore only symbols with the column, LmpVal can be used in this macro. Run this using “hourly” execution units.

```
SHOW
1: _5X8Average ( NYISO_GENESE_DA )
WHEN
  Date is within 2 years
AND
  Date is from Monday to Friday
```

The query above will calculate the average for hours 1-7 and hour 24 Monday thru Friday for the real-time LMP prices for the NYISO Load Zone Genesee Real Time Market. Notice in the macro code below, there are some assumptions being made in the calculation. These have been collected from industry and seem to be standard.

“_5x8Average”	Series
_5x8Average	(NYISO_GENESE_DA)

Macro Code:

```
ATTR MACRO _5X8Average ( SECURITY PricePoint)
RETURN
IF Date is from Monday to Friday
THEN average from 12:00 am to 11:00 pm of IF
Time is from 12:00 am to 06:00 am
OR
Time is 11:00 pm
THEN Impval of PricePoint
ENDIF
ENDMACRO
```

_5x16Average

This macro is designed to calculate the average for Monday through Friday from 7am – 10pm.

_5x16Average (Series)

Example:



The Series input will be the symbol for a particular zone/node. The column “ImpVal” is assigned to each Series in the macro script. Therefore only symbols with the column, LmpVal can be used in this macro. Run this using “hourly” execution units.

```
SHOW
1: _5X16Average ( DELAWARECT11.AGGREGATE.PJMDA )
WHEN
range of DELAWARECT11.AGGREGATE.PJMDA is less than 0.25
AND
Date is from Monday to Friday
```

The query above will calculate the average for hours 8-23 on Monday through Friday for the real-time LMP prices for the Day Ahead Locational Marginal Price in Delaware. Notice in the macro code below, there are some assumptions being made in the calculation. These have been collected from industry and seem to be standard.

“_5x16Average”	Series
<u>_5x16Average</u>	(DELAWARECT11.AGGREGATE.PJMDA)

Macro Code:

```
ATTR MACRO _5X16Average ( SECURITY PricePoint)
RETURN
IF Date is from Monday to Friday
THEN average from 07:00 am to 10:00 pm of Impval of
PricePoint
ENDIF
```

ENDMACRO

_7x8Average

This macro is designed to calculate the average for Monday through Friday from 12am-6am and 11pm.

`_7x8Average (Series)`

Example:



The Series input will be the symbol for a particular zone/node. The column “ImpVal” is assigned to each Series in the macro script. Therefore only symbols with the column, LmpVal can be used in this macro. Run this using “hourly” execution units.

```
SHOW
  1: _7X8Average ( NYISO_ALLEGHENY_COGEN_DA )
WHEN
  Date is before 7/27/2001
```

The query above will calculate the average for hours 1-7 and hour 24 on Sunday thru Saturday for the real-time LMP prices for the NYISO Generator ALLEGHENY___COGEN Real Time Market . Notice in the macro code below, there are some assumptions being made in the calculation. These have been collected from industry and seem to be standard.

“_7x8Average”	Series
<code>_7x8Average</code>	<code>(NYISO_ALLEGHENY_COGEN_DA)</code>

Macro Code:

```
ATTR MACRO _7X8Average ( SECURITY PricePoint)
RETURN
average from 12:00 am to 11:00 pm of IF
{Time is from 12:00 am to 06:00 am
OR
Time is 11:00 pm}
THEN lmpval of PricePoint
ENDIF
ENDMACRO
```

_7x16Average

This macro is designed to calculate the average for Monday through Friday from 7am – 10pm.

`_7x16Average (Series)`

Example:



The Series input will be the symbol for a particular zone/node. The column “ImpVal” is assigned to each Series in the macro script. Therefore only symbols with the column, LmpVal can be used in this macro. Run this using “hourly” execution units.

```
SHOW
  1: _7X16Average ( NEPOOL.LD.ASHCREEK115.DA )
WHEN
  Date is 1 week before any holiday
```

The query above will calculate the average for hours 8-23 on Sunday through Saturday for the real-time LMP prices for the New England ISO LMP NETWORK NOD Day Ahead Market. Notice in the macro code below, there are some assumptions being made in the calculation. These have been collected from industry and seem to be standard.

“_7x16Average”	Series
_7x16Average	(NEPOOL.LD.ASHCREEK115.DA)

Macro Code:

```
ATTR MACRO _7X16Average ( SECURITY PricePoint )
RETURN
average from 07:00 am to 10:00 pm of lmpval of
PricePoint
ENDMACRO
```

AccDist

The Accumulation/Distribution is a momentum indicator that associates changes in price with the daily range. Divergences between the Accumulation/Distribution and the security’s price imply a change is imminent. When a divergence does occur, prices usually change to confirm the Accumulation/Distribution. For example, if the indicator is moving up and the security’s price is going down, prices will probably reverse and start going up. If the days price change is positive then the difference in the daily high and low price is added to the total, and conversely if the daily change is negative then the daily range is subtracted from the total.

AccDist (Series)

Example:

“AccDist”	Series
AccDist	(DELL)

Macro Code:

```

COLUMN MACRO AccDist ( SECURITY Sec )
RETURN
  Sum from begin_of_time to today of
    if Close of Sec > Close of Sec 1 unit ago then
      absolute_value ( Close of Sec -
        if Low of Sec < Close of Sec 1 unit ago
          then Low of Sec
          else Close of Sec 1 unit ago
        endif )
    else
      if Close of Sec < Close of Sec 1 unit ago then
        0 -
          absolute_value ( Close of Sec -
            if High of Sec > Close of Sec 1 unit ago
              then High of Sec
              else Close of Sec 1 unit ago
            endif )
      else
        0
      endif
    endif
ENDMACRO

```

Aspect

Returns the difference in ecliptic longitude degrees between any two planets (except Earth). Choose planets (including Sun & Moon) relations from the “Other: Astronomy” menu in the **Search Database** window.

Aspect (Planet1, Planet2)

Example:

Aspect (MERCURY, SATURN)

“Aspect”	Planet 1	Planet 2
Aspect	(MERCURY,	SATURN)

AspectEquatorial

Returns the difference in equatorial right ascension between any two planets (except Earth, the Sun, or the Moon).

AspectEquatorial (Planet1, Planet2)

Example:

AspectEquatorial (MERCURY, SATURN)

“AspectEquatorial”	Planet 1	Planet 2
AspectEquatorial	(MERCURY,	SATURN)

AspectHelio

Returns the difference in heliocentric longitude between any two planets.

AspectHelio (Planet1, Planet2)

Example:

AspectHelio (MERCURY, SATURN)

"AspectHelio"	Planet 1	Planet 2
AspectHelio	(MERCURY,	SATURN)

AspectPlot

The following are all used in the same manner. These all plot vertical lines on the chart on days when any astrological aspect is in force. Astrologers are especially interested in the locations of solar system objects relative to other solar system objects - numbers called "aspect" angles with the centers of the Earth or Sun (depending on which coordinate system, geocentric or heliocentric, is used) located at the vertex of the measured angle. Significant aspect angles have names associated with them listed below in order of supposed importance and influence, along with the Orb - the number of degrees of tolerance often used with the aspect, and the usual symbols used to denote the angles in printed Ephemeris tables.

No. Glyphs	Aspect	Abbrev.	Aspect Angle	Orb (tolerance)	Description of Standard
1	Conjunct	(Con)	0.00	+/- 7 deg.	Circle with extending line
2	Opposite	(Opp)	180.00	+/- 7 deg.	Two circles joined by line
3	Square	(Squ)	90.00	+/- 7 deg.	Quadrilateral
4	Trine	(Tri)	120.00	+/- 7 deg.	Triangle
5	Sextile	(Sex)	60.00	+/- 6 deg.	Six pointed asterisk
6	Inconjunct	(Inc)	150.00	+/- 3 deg.	'K' rotated clockwise
7	Semisextile	(Ssx)	30.00	+/- 3 deg.	'K' rotated counter-clockwise
8	Semisquare	(Ssq)	45.00	+/- 3 deg.	Acute angle
9	Sesquisquare	(Ses)	135.00	+/- 3 deg.	Square with extending lines

Minor Aspects (not included in macro set - use "AspectX" macros which let you specify any arbitrary aspect angle.)

PlotAspectX

Plots vertical lines on the chart on days when any astrological aspect is in force.

PlotAspectX (Planet1, Planet2, aspect, orb)

Example:

PlotAspectX (MARS, NEPTUNE, 90, 0)

“PlotAspectX”	Planet 1	Planet 2	Aspect (Number)	Orb (Number)
PlotAspectX	(MARS,	NEPTUNE,	90,	0)

The following are all used in the same manner:

- PlotConjunct
- PlotInconjunct
- PlotOpposite
- PlotSemisextile
- PlotSemisquare
- PlotSesquisquare
- PlotSextile
- PlotSquare
- PlotTrine

These all plot vertical lines on the chart on days when any astrological aspect is in force.

Example:

PlotConjunct (SATURN, MARS, 0)

“PlotConjunct”	Planet 1	Planet 2	Orb (Number)
PlotConjunct	(SATURN,	MARS,	0)

BackContract

Returns the price of the back contract. This functionality is now built into the language itself. The default selection for a parent futures contract (i.e., US) is the front continuous contract. The button labeled ‘continuous’ located directly above the Symbol Browser may be toggled to ‘back’ to perform the same feature. This macro is included for backward compatibility with older versions of the XMIM software.

BackContract (Series, NumDays, ContractMth)

Example:

BackContract (NG, 0, 0)

“BackContract”	Series	NumDays (Number)	ContractMth (Number)
BackContract	(HO,	1,	1)

BlackScholesOptionPrice

Computes the option price based on the Black Scholes model for a European option. The interest rate must be expressed as a decimal fraction. For example, 2% will be 0.02. The sigma argument must also be expressed as decimal fraction: e.g. 16% is 0.16. The days to expiration are in actual days and will be divided by 365 to convert to fractions of a year. For a 252 day year adjust the days input by 252/365.

BlackScholesOptionPrice (@underlying_price, @strike, @interest_rate, @sigma, @days_expiration, @is_call)

Example:

```
BlackScholesOptionPrice ( Close of SP_2004Z, 1100,FRH_DGS1 / 100,
(30 day std_dev of Close of SP_2004Z) / 100,
periods_in_range from today to SP_2004Z.C1100 last_data_day, 1 )
```

@underlying_price	@strike	@interest_rate	@sigma	@days_expiration	@is_call
Close of SP_2004Z,	1100,	FRH_DGS1/100,	(30 day std_dev of Close of SP_2004Z) / 100,	periods_in_range from today to SP_2004Z.C1100 last_data_day,	1

ChannelIndex

Returns an index of bollinger bands, a statistical version of the stochastic.

Donald Lambert developed this price momentum indicator - Commodity Channel Index (CCI). It shows how the most recent close relates to the standard deviation range over the recent time period.

ChannelIndex (Series, TimePeriod)

Example:

```
ChannelIndex (SP, 22 days)
```

"ChannelIndex"	Series	Constant (Number)	Time Period (day/mth/yr)
ChannelIndex	(HO,	22	days)

ChoppinessIndex

Returns an indicator that measures the sum of all the ranges in a time period to the range over the time period as a measure of the trendiness or choppiness of a price series. Inspired by chaos theory, it is similar to Wilder's Directional Movement Index but with simpler calculations and less lag. The Choppiness Index varies between 0 and 100, the higher the index the choppier the price action is and the lower the index the more trending the price action

ChoppinessIndex (Series, TimePeriod)

Example:

ChoppinessIndex (JY, 10 days)

“ChoppinessIndex”	Series	Constant (Number)	Time Period (day/mth/yr)
ChoppinessIndex	(HO,	22	days)

DeMarker

Returns Tom DeMark’s Oscillator. The DeMarker identifies potential price bottoms and tops. The indicator fluctuates between 0 and 1.

DeMarker (Series, TimePeriod)

Example:

DeMarker (Close of SPX, 15 days)

“DeMarker”	Attribute	of	Series	Constant (Number)	Time Period (day/mth/yr)
DeMarker	(Close	of	HO,	22	days)

DeferredContract

This study provides a generalized macro for dealing with futures contracts. The DeferredContract function is used to identify certain back contract relationships within the software.

There are two methods to identify a futures contract:

1. You may identify the ‘specific’ contract (i.e., the March 1998 30 year U.S. Treasury Bond contract or US_1998H)
2. You may identify a ‘generic’ contract based upon the contracts expiration (i.e., the front continuous adjusted 30 year U.S. Treasury Bond contract or US). If a user wanted to test a trading strategy that involved the sixth contract on the board, or some contract expiring later than the front, far and back contracts (these selections are available under the ‘continuous’ button above the Symbol Browser), DeferredContract would be used to define contract.

There are four fields that are required to be specified within this study:

1. The Attribute or series (select your futures series).

2. The deferred number of the contract chosen (i.e., 1 represents the contract nearest to expiration or the front contract, 2 represents the second nearest contract to expiration, etc.).
3. The number of days until expiration that the contract should be followed (i.e., 0 specifies that the contract will be active until expiration, 1 specifies that the contract will be active or followed up until 1 day before expiration, etc.).
4. Contract month that is the parameter that specifies the contract month of interest (i.e., setting the field to 0 will disable this field thereby allowing the contract to roll to the regularly traded serial months, setting the field to 1 will specify only January contracts, etc.).

DeferredContract (Series, DeferredNum, NumDays, ContractMth)

Example 1:

```
DeferredContract (US, 3, 0, 0)
```

"DeferredContract"	Attribute	Deferred Number	NumDays till expire (Number)	Contract Month (Number)
DeferredContract	(US,	3,	0,	0)

The following shows further examples to demonstrate the use of DeferredContract.

Example 2:

As the macro code below shows, the deferred contract macro is using the `adjust_contract` built-in. Therefore, the properties that are required entry for the `adjust contract` are the same for the deferred contract.

Assuming the 10/8/2002 trade date, the prompt/front contract for the NYMEX natural gas futures is the November 2002 contract (NG_2002X). If we wanted to use the deferred contract macro to modify the continuous NG contract to show the 5th contract out instead of the prompt contract, we could use the following syntax:

```
SHOW
1: DeferredContract ( NG, 5, 0, 0 )
```

This will modify NG to the 5th contract out. (Prompt +4). This will now show the price for the March 2003 NG contract. (NG_2003H)

- NG= base commodity symbol of choice.
- 5 = N nearest. 5 points to the 5th closest or nearest to expiration active contract.
- 0 = This is "Number of days before Expiration". Setting this to "0" indicates the series will locate the next nearest contract when the contract expires. Setting this to "4" would mean that the contract should roll to the next one 4 days before expiration of the current contract.
- 0 = Normally, this is where you put in the contract month. However, since we want the exact 6th closest contract, this is set to "0".

*Any of the 4 parameters can be modified to adjust the continuous contract as described in the `adjust_contract` built-in definition.

Here is an example of the data:

Date	Day	Deferred	NG_2003H
10/08/2002	Tue	4.023	4.023

Example 3:

```
SHOW
twoSixAvg:
  (DeferredContract ( NG, 2, 0, 0 ) + DeferredContract ( NG, 3, 0, 0 )
  + DeferredContract ( NG, 4, 0, 0 ) + DeferredContract ( NG, 5, 0,
  0 ) + DeferredContract ( NG, 6, 0, 0 )
  ) / 5
frontNG: NG
WHEN
  Date is within 2 years
AND
  NG is DEFINED
```

Macro Code:

```
ATTR MACRO DeferredContract ( ATTR Series, CONSTANT DeferredNum = 1,
  CONSTANT NumDays = 0, CONSTANT ContractMth = 0 )
RETURN
  adjust_contract ( Series, DeferredNum, NumDays, ContractMth )
ENDMACRO
```

DirMove

DirMove returns the Welles Wilder's Directional Movement indicator. To be used in conjunction with DirMoveIdx (below), which returns the Welles Wilder's Directional Movement Index. DirMove uses an exponential average in its calculation, while DirMoveIdx uses simple moving average.

Example:

DirMove (Close of MSFT, 0.071428571)

"DirMove"	Attribute	of	Series	Constant Weight
DirMove	(Close	of	HO,	0.071428571)

DirMoveIdx

Returns the Welles Wilder's Directional Movement Index.

Example:

DirMoveIdx (Close of INTC, 78 days)

“DirMoveldx”	Attribute	of	Series	Constant (Number)	Time Period (day/mth/yr)
DirMoveldx	(Close	of	HO,	78	days)

FarContract

Returns the price of the third nearest-to-delivery futures contract in a futures market. This functionality is now built into the language itself. The default selection for a parent futures contract (i.e., US) is the front continuous contract. The button labeled ‘continuous’ located directly above the Symbol Browser may be toggled to ‘far’ to perform the same feature. This macro is included for backward compatibility with older versions of the XMIM software.

FarContract (Series, NumDays, ContractMth)

Example 1:

```
FarContract (Close of CL, 4, 11)
```

“FarContract”	Attribute	of	Series	NumDays (Number)	ContractMth (Number)
FarContract	(Close	of	HO,	4,	11)

Example 2:

As the macro code below shows, The FarContract macro is using the adjust_contract built-in. Therefore, the properties that are required entry for the adjust contract are the same for the far contract with the exception of the second adjust_contract parameter which indicates the *N* month out. This is always set to 3.

A user can utilize the FarContract macro to modify a continuous contract to pull the far (3rd) closest contract to expiration.

Assuming the 10/8/2002 trade date, the prompt/front contract for the NYMEX natural gas futures is the November 2002 contract (NG_2002X). If we wanted to use the FarContract macro to modify the continuous NG contract to show the 3rd contract out instead of the prompt contract, we could use the following syntax:

```
SHOW
1: FarContract ( NG, 0, 0 )
```

This will always modify NG to the 3rd contract out. (Prompt +2). This will now show the price for the January 2003 NG contract. (NG_2003F).

- NG= base commodity symbol of choice.
- 0 = This is “Number of days before Expiration”. Setting this to “0” indicates the series will locate the next nearest contract when the contract expires. Setting this to “4” would mean that the contract should roll to the next one 4 days before expiration of the current contract.
- 0 = Normally, this is where you put in the contract month. However, since we want the exact 6th closest contract, this is set to “0”.

*Any of the 3 parameters can be modified to adjust the continuous contract as described in the `adjust_contract` built-in definition.

Here is an example of the data:

```
Date      Day      Far      NG_2003F
10/08/2002  Tue      4.243      4.243
```

Example 3:

```
SHOW
macro: FarContract ( CL, 0, 0 )
xmimlanguage: far CL
WHEN
    Date is 2001
```

Macro Code:

```
ATTR MACRO FarContract (ATTR Series, CONSTANT NumDays = 0,
                        CONSTANT ContractMth = 0)
RETURN
    adjust_contract (Series, 3, NumDays, ContractMth)
ENDMACRO
```

FrontContract

Returns the price of the nearest-to-delivery futures contract in the futures market. This functionality is now built into the language itself. The default selection for a parent futures contract (i.e. US) is the front continuous contract. The button labeled ‘continuous’ located directly above the Symbol Browser may be toggled to ‘front’ to perform the same feature. This macro is included for backward compatibility with older versions of the XMIM software.

`FrontContract (Series, NumDays, ContractMth)`

Example 1:

```
FrontContract (Close of NG, 4, 11)
```

“FrontContract”	Attribute	of	Series	NumDays (Number)	ContractMth (Number)
FrontContract	(Close	of	HO,	4,	11)

Example 2:

As the macro code below shows, the `FrontContract` macro is using the `adjust_contract` built-in. Therefore, the properties that are required entry for the `adjust_contract` are the same for the `FrontContract` macro with the exception of the second `adjust_contract` parameter which indicates the *N* month out. This is always set to 1.

A user can utilize the `FrontContract` macro to modify a continuous contract to pull the prompt or closest contract to expiration.

Assuming the 10/8/2002 trade date, the prompt/front contract for the NYMEX natural gas futures is the November 2002 contract (NG_2002X). If we wanted to use the FrontContract macro to pull the front contract, we could use the following syntax:

```
SHOW
  Front: FrontContract ( NG, 0, 0 )
```

This will always modify NG to the front contract (Prompt). This will now show the price for the November 2002 NG contract. (NG_2002X)

- NG= base commodity symbol of choice.
- 0 = This is “Number of days before Expiration”. Setting this to “0” indicates the series will locate the next nearest contract when the contract expires. Setting this to “4” would mean that the contract should roll to the next one 4 days before expiration of the current contract.
- 0 = Normally, this is where you put in the contract month. However, since we want the exact 6th closest contract, this is set to “0”.

*Any of the 3 parameters can be modified to adjust the continuous contract as described in the adjust_contract built-in definition.

Here is an example of the data:

Date	Day	Front	NG_2002X
10/08/2002	Tue	3.862	3.862

Example 3:

```
SHOW
  macro: FrontContract ( US, 0, 0 )
  xmimlanguage: front US
WHEN
  Date is 2002
```

Macro Code:

```
ATTR MACRO FrontContract (ATTR Series, CONSTANT NumDays = 0,
  CONSTANT ContractMth = 0)
  RETURN
adjust_contract (Series, 1, NumDays, ContractMth)
ENDMACRO
```

HistVolLog

Returns the historical volatility of a series: the standard deviation of the logarithmic price changes measured at regular intervals of time. The default time period is 20 days. The default number of units per year is 250.

HistVolLog (Series, NumUnits, UnitsPerYear)

Example:

```
HistVolLog (Close of front CL, 20 days, 250)
```

"HistVolLog"	Attribute	of	Series	NumUnits (Number)	Time Period (day/mth/yr)	UnitsPer Year (Number)
HistVolLog	(Close	of	HO,	20	days,	250)

HistVolPct

Returns the historical volatility of a series: the standard deviation of the percent price changes measured at regular intervals of time. The default time period is 20 days. The default number of units per year is 250.

HistVolPct (Series, NumUnits, UnitsPerYear)

Example:

HistVolPct (Close of front NG, 20 days, 250)

"HistVolPct"	Attribute	of	Series	NumUnits (Number)	Time Period (day/mth/yr)	UnitsPer Year (Number)
HistVolPct	(Close	of	HO,	20	days,	250)

MoneyFlowIndex

The Money Flow Index measures the amount of money flowing in and out of a security. It is an oscillator not unlike RSI that incorporates Volume into its calculation. The values range between 0 and 100. Market tops generally occur around 80 and bottoms around 20.

MoneyFlowIndex (Series, TimePeriod)

"MoneyFlowIndex"	Series	NumUnits (Number)	Time Period (day/mth/yr)
MoneyFlowIndex	(HO,	50	days)

NegVolIndex

Returns an accumulated sum of the close-to-close percent moves of the security when the volume is lower than the volume one period ago.

NegVolIndex (Series)

Example:

NegVolIndex (INTC)

“NegVolIndex”	Series
NegVolIndex	(HO)

NetHedgers

Returns the net open interest held by traders in the commercial hedgers category of futures traders in a market. The calculation is net long minus net short positions.

NetHedgers (Series)

Example:

NetHedgers (NG)

“NetHedgers”	Futures Contract
NetHedgers	(HO)

NetHedgersPct

Returns the net open interest holdings of the commercial hedger category as a percentage of total open interest in the market.

NetHedgersPct (Series)

Example:

NetHedgersPct (SP)

“NetHedgersPct”	Futures Contract
NetHedgersPct	(HO)

NetLargeSpecs

Returns the net open interest held by traders in the large speculator category of futures traders in a market. That is, the net long minus net short positions.

NetLargeSpecs (Series)

Example:

NetLargeSpecs (JY)

“NetLargeSpecs”	Futures Contract
NetLargeSpecs	(HO)

NetLargeSpecsPct

Returns the net open interest holdings of the large speculator category as a percentage of total open interest in the market.

NetLargeSpecsPct (Series)

Example:

NetLargeSpecsPct (US)

“NetLargeSpecsPct”	Futures Contract
NetLargeSpecsPct	(HO)

NetSmallTraders

Returns the remaining net open interest held by traders with positions smaller than reporting thresholds.

NetSmallTraders (Series)

Example:

NetSmallTraders (HO)

“NetSmallTraders”	Futures Contract
NetSmallTraders	(HO)

NetSmallTradersPct

Returns the net open interest of the Small Traders category as a percentage of total open interest.

NetSmallTradersPct (Series)

Example:

NetSmallTradersPct (HO)

“NetSmallTradersPct” Futures Contract

NetSmallTradersPct (HO)

NormalDist

NormalDist(n) returns the percent of area under the normal distribution curve commonly called the z-table. The parameter n is the number of standard deviations from the mean. The macro NormalDist(n) returns the percent of the curve below parameter n. This is commonly used to compute probability.

Example:

```
NormalDist (5)
```

“NormalDist” Number of Standard Deviations from the Mean (Number)

NormalDist (5)

RangeExpIndex

Returns Tom DeMark's REI2 indicator as explained in his book *“The New Science of Technical Analysis”*.

RangeExpIndex (Series, TimePeriod)

Example:

```
RangeExpIndex (GE, 15 days)
```

“RangeExpIndex” Series Constant (Number) Time Period (day/mth/yr)

RangeExpIndex (HO, 15 days)

RealSum

Here Counter is a number which is the # of days to start the look back. Returns the sum of the last N days.

RealSum (Attribute, Counter)

Example:

```
RealSum (DELL, 3 )
```

is the same as

sum from 2 days ago to today of DELL

Both are considering 3 days of data for the calculation.

"RealSum"	Series	Counter (N) (Number)
RealSum	(HO,	3)

Stochastics

KStochastic

Stochastics is a momentum or price velocity indicator developed by George C. Lane. This macro is sometimes referred to as "%K" in the market. This is the normal stochastic.

KStochastic (Security, TimePeriod)

Example:

KStochastic (DELL, 18 days)

"KStochastic"	Series	Constant (Number)	Time Period (day/mth/yr)
KStochastic	(HO,	18	days)

SDStochastic

This macro is an average of the "SKStochastic" macro.

SDStochastic (Security, TimePeriod, TimePeriod2)

Example:

SDStochastic (MPPP, 24 days, 10 days, 6 days)

"SDStochastic"	Series	Constant (Number)	Time Period1	Constant (Number)	Time Period2	Constant (Number)	Time Period3
SDStochastic	(HO,	24	days,	10	days,	6	days)

SKStochastic

This macro is sometimes referred to as "%D" in the market. It is the slower moving stochastic.

SKStochastic (Security, TimePeriod, TimePeriod2)

Example:

SKStochastic (OAT, 14 days, 4 days)

"SKStochastic"	Series	Constant (Number)	Time Period1 (day/mth/yr)	Constant (Number)	Time Period2 (day/mth/yr)
SKStochastic	(HO,	14	days,	4	days,

Trendline

Returns drawing of a line to connect a series of peaks or troughs on a Chart. Both uptrends and downtrends in a market can be defined in terms of trendlines.

Trendline (Price1, Date1, Price2, Date2)

"Trendline"	Price 1	Date 1	Price 2	Date 2
Trendline	(21.20,	1/1/2001	17.85,	2/3/2002)

VolPriceTrend

Returns an index which takes the distance of the close of the day from the midpoint of the day, multiplies that times the volume and sums the results over time.

VolPriceTrend (Series, TimePeriod)

Example:

VolPriceTrend (DELL, 68 days)

"VolPriceTrend"	Sec Series	Constant (Number)	Time Period (day/mth/yr)
VolPriceTrend	(HO,	68	days)

Volatility

Returns the high of the period divided by the low.

Volatility (Series)

Example:

Volatility (RAD)

“Volatility”	Sec Series
Volatility	(HO)

VolatilityIdx

Returns the exponential moving average of the true range.

Refer to Exponential Average notes for weight info.

VolatilityIdx (Security, Weight)

Example:

VolatilityIdx (IO, 0.0714)

“VolatilityIdx”	Sec Series	Weight
VolatilityIdx	(HO,	0.0714)

WildersADX

The familiar average directional movement index is simply the directional movement index (WildersDX) averaged for the time period chosen for study related to the previous day’s average directional movement index.

WildersADX (Series, Weight)

Example:

WildersADX (JO, 0.071428571)

“WildersADX”	Sec Series	Weight
WildersADX	(HO,	0.071428571)

WildersADXR

Returns an exponentially smoothed version of the WildersADX study discussed above.

WildersADXR (Series, Weight)

Example:

WildersADXR (CME.DB, 0.071428571)

“WildersADX”	Series	Weight
WildersADX	(HO,	0.071428571)

WildersDIMinus

Returns a momentum indicator typically used to identify downtrend directional behavior in the market and developed by Welles Wilder. This study is calculated by dividing the exponentially Positive Directional Movement (those days when the difference between today’s low minus the previous day’s low is greater than the difference between today’s high minus the previous day’s high) by the smoothed True Range.

WildersDIMinus (Series, Weight)

Example:

```
WildersDiMinus ( DS, 0.071428571)
```

“WildersDIMinus”	Series	Weight
WildersDIMinus	(HO,	0.071428571)

WildersDIPlus

Returns a momentum indicator typically used to identify up-trend directional behavior in the market and developed by Welles Wilder. This study is calculated by dividing the exponentially smoothed Positive Directional Movement (those days when the difference between today’s high minus the previous day’s high is greater than the difference between today’s low minus the previous day’s low) by the smoothed True Range.

WildersDIPlus (Series, Weight)

Example:

```
WildersDIPlus ( CNDL, 0.071428571)
```

“WildersDIPlus”	Series	Weight
WildersDIPlus	(HO,	0.071428571)

WildersDX

WildersDX returns the equivalent to Welles Wilder’s Directional Movement equation. This calculation divides the difference between DI+ and DI- (true directional movement) and sum of DI+ and DI- (total percentage of directional movement), and then multiplies this value by 100%.

The directional movement index (DX) is thus an accounting or a scoring between 0 - 100 of the power of the trend, be the trend up or down, a high score means the market is strongly trending (up or down) and a low score bespeaks of weak or no trend.

WildersDX (Series, Weight)

Example:

WildersDX (BO, 0.071428571)

"WildersDX"	Series	Weight
WildersDX	(BO,	0.071428571)

WildersRSI

Returns the Welles Wilder's Relative Strength Index indicator, a popular overbought-oversold measure. This study is very similar to `simple_rsi`, except Wilder's version uses a form of exponential smoothing where `simple_rsi` uses a simple moving average.

This study compares the relative strength of prices gains (closes that are higher than the previous day's close) versus price losses (closes that are lower than the previous day's close) by using an exponentially smooth average of these values for some specified period of time.

Like `simple_rsi`, `wilders_rsi` may be performed for any time period. Wilder originally suggested 14 days, however, technicians sometimes prefer a shorter/more sensitive period of time such as 5 or 7 days. This study will range between 0 and 100.

Industry standards derive the overbought and oversold indicators at levels that range around 70 (overbought) and 30 (oversold).

WildersRSI (Series, Weight)

Example:

WildersRSI (Close of MSFT, 0.095)

"WildersRSI"	Attribute	of	Series	Weight
WildersRSI	(Close	of	HO,	0.095)

WilliamsVolAD

Returns the Larry Williams Volume Accumulation-Distribution indicator.

WilliamsVolAD (Security, TimePeriod)

Example:

WilliamsVolAD (HO, 17 days)

“WilliamsVolAD”	Sec Series	Constant (Number)	Time Period (day/mth/yr)
WilliamsVolAD	(HO,	17	days)

Index

Symbols

% Move, 2
 _2x16Average, 33
 _2x8Average, 32
 _5x16Average, 35
 _5x8Average, 34
 _7x16Average, 36
 _7x8Average, 36
 _CoolingDegreeDays, 30
 _HeatingDegreeDays, 30
 _OffPeakAverage, 31
 _OnPeakAverage, 31

A

Absolute Value, 2
 AccDist, 37
 Adjust Contract, 3
 Aspect, 38
 AspectEquatorial, 38
 AspectHelio, 39
 AspectPlot, 39
 Average, 6

B

BackContract, 40
 BlackScholesOptionPrice, 41
 Bollinger High/Bollinger Low, 6
 Built-ins, 2

C

ChannelIndex, 41
 ChoppinessIndex, 41
 Compound_return, 7
 Construct Date, 7
 Coregression, 7
 Coregression_intercept, 8
 Coregression_slope, 9
 Correlation, 10
 Count, 10

Covariance, 11

D

DeferredContract, 42
 DeMarker, 42
 DirMove, 44
 DirMoveIdx, 44
 Dividend Adjusted Return, 11

E

Ease of Move, 11
 Exponential Average, 12
 Extract Day, 12
 Extract Month, 13
 Extract Year, 13

F

FarContract, 45
 Formula Table, 1
 Formulas
 for Built-ins, 2
 FrontContract, 46

G

Greatest Integer, 13

H

Highest, 13
 HistVolLog, 47
 HistVolPct, 48

I

If-Then Statement, 14
 Inverse Log, 14

L

Linear Regression, 14
 Lowest, 16

M

Macd_osc, 16
 Macd_signal, 16
 Median, 17
 Mid_point, 17
 Midpoint, 17

MoneyFlowIndex, 48
Move, 2, 18

N

Natural Log, 18
Nearest Integer, 18
Negative Value, 19
NegHedgersPct, 49
NegVolIndex, 48
Net Change, 19
Net Percent Change, 19
NetHedgers, 49
NetLargeSpecs, 49
NetLargeSpecsPct, 50
NetSmallTraders, 50
NetSmallTradersPct, 50
NormalDist, 51

O

On Balance Volume, 20

P

Periods in Range, 20
PlotAspectX, 39
Positive Value, 20
Product, 21

R

Range, 21
RangeExpIndex, 51
RealSum, 51
Rel Strength, 21

S

Select Contract, 22
Server Macros, 30
Simple_rsi, 24
Smallest Integer, 25
Standard Deviation, 26
Standard Deviation Population, 26
Stochastics, 52
Sum, 26

T

Trendline, 53
True Midpoint, 27

True Range, 27

V

Variance, 28
Volatility, 53
VolatilityIdx, 54
VolPriceTrend, 53

W

Wilders_rsi, 28
WildersADX, 54
WildersADXR, 54
WildersDIMinus, 55
WildersDIPlus, 55
WildersDX, 55
WildersRSI, 56
Williamsr, 29
WilliamsVolAD, 56