#### 1.2.3.17 TagMean (Получение среднего значения тега за указанный интервал времени)

Находит среднее значение атрибута в течение указанного диапазона времени. TagAvg с опцией event-weighted дает тот же результат, что и TagMean. Рассмотрите возможность использования TagAvg вместо TagMean, особенно если в других анализах используется взвешенный по времени TagAvg.

Параметры

1. startTime типа object;
2. endTime типа object;
3. calculationBasis типа string (не обязательный, по умолчанию = "event").

TagMean(attname, starttime, endtime [, pctgood])

TagMean('Tag 1', '01-Jul-23', '30-Aug-23', 80)

TagMean('Tag 1', '01-Jul-23', '30-Aug-23', 80)

## Параметры

**attname**

атрибут с данными временных рядов (например, ссылка на данные PI Point), заключенный в одинарные кавычки

**starttime**

выражение времени, представляющее начало временного диапазона, заключенное в одинарные кавычки; может быть относительным временем (например, «-3 часа») относительно **endtime**

**endtime**

выражение времени, представляющее конец временного диапазона, заключенное в одинарные кавычки; может быть относительным временем (например, «+1 час») по отношению к **starttime**

**pctgood**

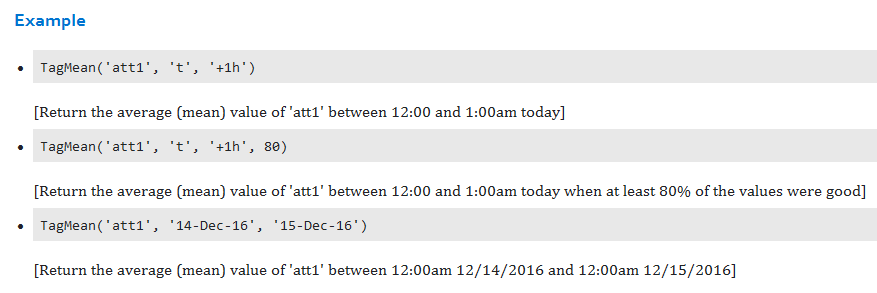
Опционально: минимальный процент goodvalue за указанный диапазон времени

## Результат

Среднее значение атрибута за заданный период времени. Обратите внимание, что среднее значение не взвешено по времени. Для средневзвешенного по времени вместо этого используйте TagAvg.

## Исключения

Если атрибут не имеет подходящих значений или минимум pctgood не достигнут за указанный диапазон времени, возвращается значение ошибки. В отличие от некоторых других суммирующих функций, TagMean не интерполирует какие-либо значения на границе. Таким образом, если в указанном интервале нет значения, возвращается значение ошибки.



#### 1.2.3.18 EventCount (Количество событий для атрибута в течение указанного диапазона времени.)

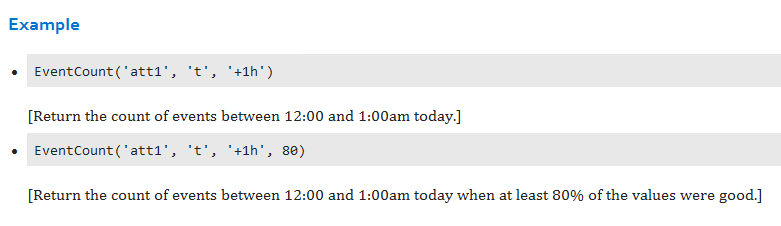
EventCount(attname, starttime, endtime [, pctgood])

EventCount('Tag 1',(Bom('\*-8h')+8\*60\*60), (Bod('\*-8h')+8\*60\*60))

EventCount('Tag 1', Bom('\*'.AddHours(-8)).AddMinutes(480), Bod('\*'.AddHours(-8)).AddMinutes(480))

Параметры

1. tagName типа string;
2. startTime типа object;
3. endTime типа object;



#### 1.2.3.19 HasChanged (Возвращает значение True, если с атрибутом произошло какое-либо событие за указанный период времени; в противном случае False.)

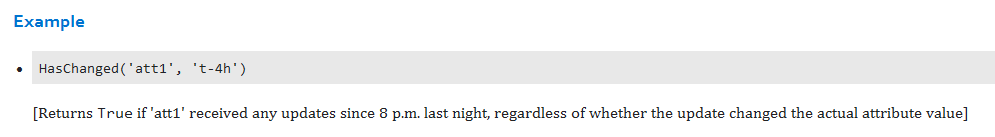
HasChanged(attname, t1)

HasChanged('Tag 1',(Bod('\*-8h')+8\*60\*60))

HasChanged('Tag 1', Bod('\*'.AddHours(-8)).AddMinutes(480))

Параметры

1. tagName типа string;
2. startTime типа object;



#### 1.2.3.20 StDev (Взвешенное по времени или по событию стандартное отклонение значений атрибута из указанного временного диапазона.)

StDev(attname, starttime, endtime [, pctgood, calculationBasis])

StDev('Tag 1',(Bom('\*-8h')+8\*60\*60), (Bod('\*-8h')+8\*60\*60), 80, "TimeWeighted")

StDev('Tag 1', Bom('\*'.AddHours(-8)).AddMinutes(480), Bod('\*'.AddHours(-8)).AddMinutes(480), 80, "TimeWeighted")

Параметры

1. tagName типа string;
2. startTime типа object;
3. endTime типа object;
4. Настраиваемый процент выборки значений good от общего количества значений
5. Опционально calculationBasis. EventWeighted или TimeWeighted

## Результат

Стандартное отклонение, взвешенное по времени или по событию, для значений атрибута из указанного временного диапазона

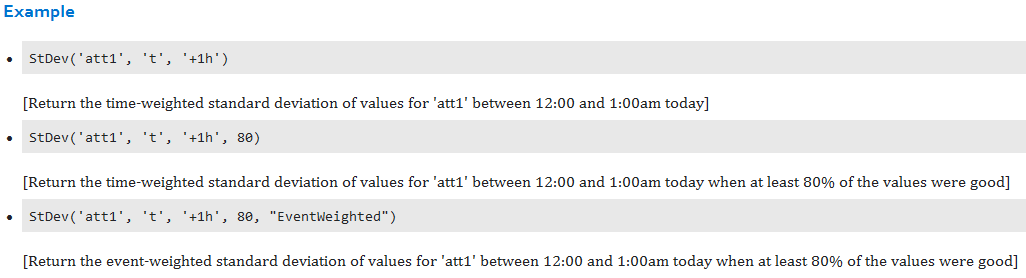
## Исключения

Если атрибут не имеет подходящих значений или минимум pctgood не достигнут за заданный диапазон времени, возвращается ошибка

## Примечания

* BadValue исключаются из расчета StDev.
* Чтобы настроить дополнительный параметр, необходимо указать любой предыдущий дополнительный параметр.

Если атрибут имеет очень мало хороших значений в течение определенного периода времени, результат этой функции может быть недостоверным. Используйте функцию PctGood, чтобы узнать, какой процент значений является хорошим.



#### 1.2.3.21 PrevEvent (Временная метка последнего записанного значения до указанного времени.)

PrevEvent(attname, t1)

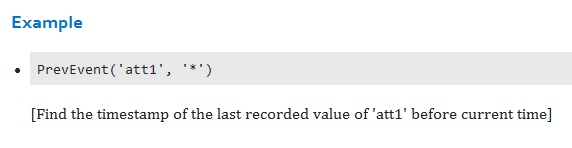
PrevEvent('Tag 1', '\*')

PrevEvent('Tag 1','\*')

Параметры

1. tagName типа string;
2. Time типа object;

Выход: String



#### 1.2.3.22 Abs (значение по модулю)

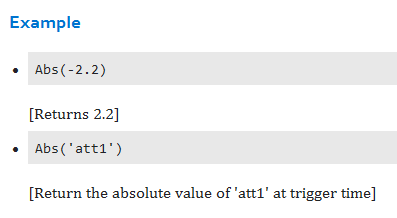
Abs(x)

Abs(TagVal('Tag 1','\*'))

Abs(TagVal('Tag1','\*').Value)

Параметры

1. input типа object;
2. Time типа object;



#### 1.2.3.23 RoundFrac (Округлите числовое значение от x до y десятичных знаков.)

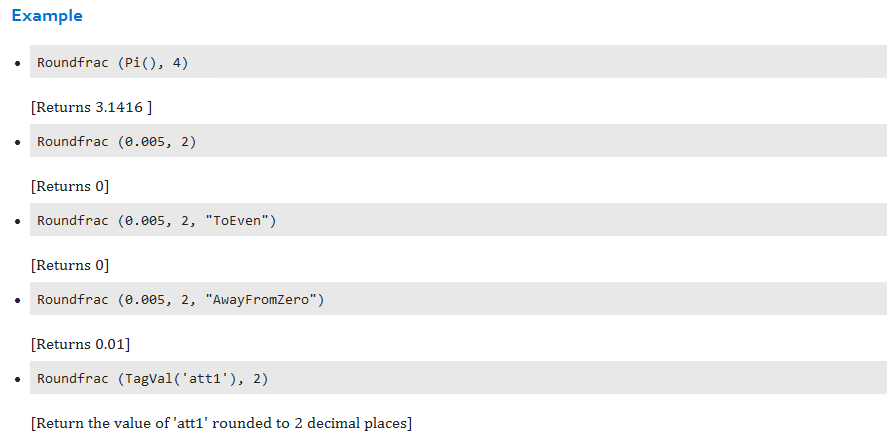
Roundfrac(x, y [,roundingMode] )

Roundfrac(TagVal('Tag 1','\*'), 2)

Roundfrac(TagVal('Tag1','\*').Value), 2)

Параметры

1. input типа object;
2. Переменная типа int, настраиваемое количество символов после запятой;
3. Метод округления. ToEven – до целых, AwayFromZero – округление в сторону значение больше изначального



#### 1.2.3.24 Round (Округление числа или времени до ближайшей единицы.)

Round(x [, unit] [, roundingMode])

Round(TagVal('Tag 1','\*'), 0.1)

Round(TagVal('Tag1','\*').Value), 0.1)

Параметры

1. input типа object;
2. Переменная типа int, настраиваемое количество символов после запятой;
3. Метод округления. ToEven – до целых, AwayFromZero – округление в сторону значение больше изначального

## Аргументы

x

Целое число, действительное число, выражение времени или интервал времени

unit

Опционально. Размер единицы измерения, до которой нужно округлить. Если x — число, единица измерения должна быть числом. Если x является выражением времени или периодом времени, единица измерения должна быть выражением времени. Если единица измерения опущена, функция Round округляет до ближайшего четного целого числа (для чисел) или секунды (для выражений времени).

roundingMode

Опционально. Тип применяемого округления. Использование строки «AwayFromZero» применяет округление значений средней точки от нуля. При отсутствии этого параметра или использовании «ToEven» применяется округление до целого по умолчанию.

## Результат

Ближайшее к x значение, кратное единице. Возвращает тот же тип данных, что и x. Более подробную информацию см. в следующих примерах.

## Исключения

Если x является строкой или если единица измерения имеет неправильный тип данных, возвращается значение ошибки.

## Примечания

В системе используется округление, при котором число округляется до ближайшего четного числа.

Если x — время, а единица измерения опущена, эта процедура не имеет никакого эффекта: время указывается с точностью до секунды.

Единица измерения аргумента, если она существует, переносится в результат.

## Пример

• Round(12.499)

[Returns 12 (округление до целого)]

• Round(12.5)

[Returns 12 (округление до чётного целого)]

• Round(12.5, "AwayFromZero")

[Returns 13 (округление в большую сторону до целого)]

• Round (13.5)

[Returns 14 (округление до целого)]

• Round(13.45, 0.1)

[Returns 13.4 (округление до десятых в дробной части)]

• Round(13.45, 0.1, "AwayFromZero")

[Returns 13.5 округление до десятых в большую сторону в дробной части)]

• Round(1285, 10)

[Returns 1280 (округление до числа кратного 10)]

• Round(1285, 10, "AwayFromZero")

[Returns 1290] Округление в большую сторону до целого числа кратного 10

• Round(1285, 10, "ToEven")

[Returns 1280 (округление до целого чётного числа кратного 10)]

• Round('14-Dec-16 11:47:16')

[Returns 14-Dec-16 11:47:16 (округление до секунды)]

• Round('14-Dec-16 12:30', '+1h')

[Returns 14-Dec-97 12:00:00 (округление до часа)]

• Round('18:47:15'-'15:00:09')

[Returns 03:47:06 (округление до секунды)]

• Round('18:45:40'-'15:15:10','+1m')

[Returns 03:30:00 (округление до минуты)]

#### 1.2.3.25 Concat (объединение двух объектов String)

Concat(s1, s2 [, ... sn])

Concat("12", "-31", "-23", " shut", "down")

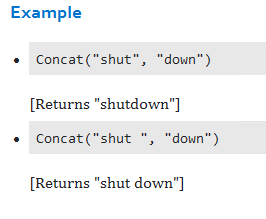
Concat("12", "-31", "-23", " shut", "down")

Параметры

1. input типа object перечисленные через запятую  
   Concat("12", "-31", "-23", " shut", "down")

Результат

12-31-23shutdown

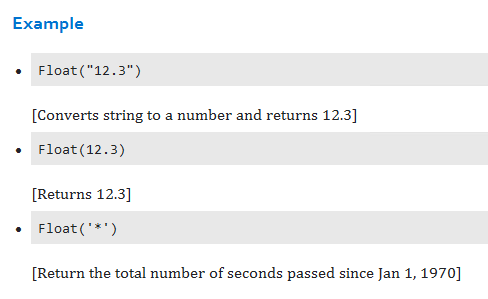


#### 1.2.3.26 Float (конвертация String в число)

Float(x)

Float((if(BadVal(TagVal('Concat2','\*')), 0, TagVal('Concat2','\*').Value)))

Если x не является числом или числовой строкой, возвращается Calc Failed



#### 1.2.3.27 Int (отображение целых значений)

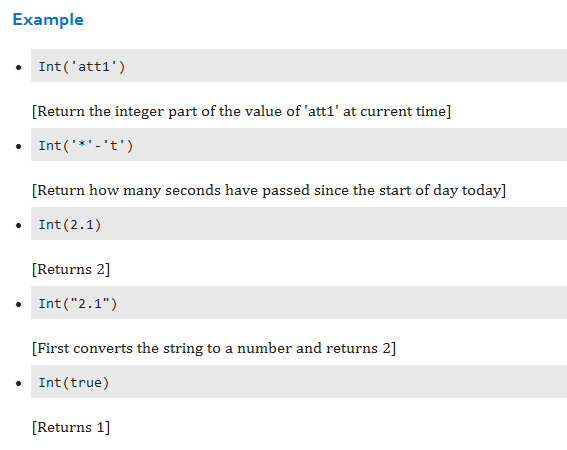
Int(x)

Int('Tag 1')

Int(TagVal('Tag1','\*').Value))

Возвращает целую часть целого или действительного числа.

Int('\*'-'t') – количество секунд с начала дня



#### 1.2.3.29 RecordedValues (Значения в определенном временном диапазоне на основе заданного пользователем типа границы.)

RecordedValues(attname, starttime, endtime [, boundarytype])

RecordedValues('Tag 1', '\*-10m', '\*')

RecordedValues('Tag 1', '\*'.AddMinutes(-10), '\*')

Параметры

1. tagName типа string;
2. startTime типа object;
3. endTime типа object;

## Аргументы

boundarytype

строка, указывающая поведение получения данных в конечные точки указанного времени в двойных кавычках:

"Inside" - вернуть ближайшие записанные значения внутри запрошенного диапазона времени в качестве первого и последнего значений

"Outside" - вернуть ближайшие записанные значения за пределами запрошенного диапазона времени в качестве первого и последнего значений

"Interpolated" - создать интерполированное значение в конечных точках запрошенного диапазона времени, если записанное значение не существует в это время

Примечание: По умолчанию режим "Inside"

## Результат

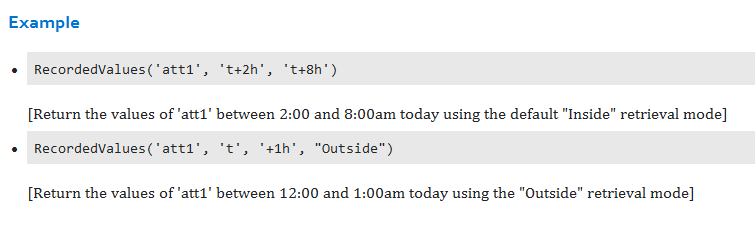
Массив значений, полученных в указанном диапазоне

## Исключение

Если атрибут не поддерживает вызовы диапазона, функция возвращает ошибку с указанием

## Примечания

* Если время начала раньше, чем время окончания, результирующие значения будут располагаться в порядке возрастания времени, в противном случае они будут располагаться в порядке убывания времени.
* Если значения недоступны, но вы указали типы границ «Внешние» или «Интерполированные», возвращается специальное значение (NoData).
* Эта функция извлекает до 500 000 значений одновременно.

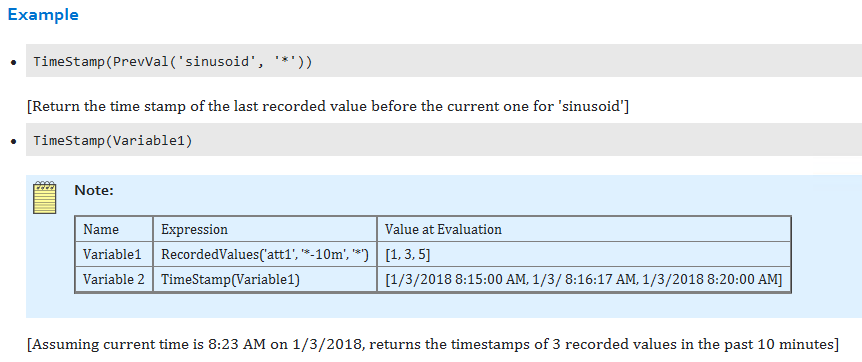


#### 1.2.3.30 TimeStamp (Возвращает метку времени для одного значения с меткой времени или массива значений с меткой времени.)

TimeStamp(x)

TimeStamp(RecordedValues('Tag 1', '\*-10m', '\*'))

TimeStamp(RecordedValues('Tag 1', '\*'.AddMinutes(-10), '\*'))



#### 1.2.3.31 ParseTime (Перевод выражения времени PI в метку времени. Используйте регулярное выражение времени в одинарных кавычках для повышения производительности)

ParseTime(s1)

Это новая функция можно применить такой же синтаксис

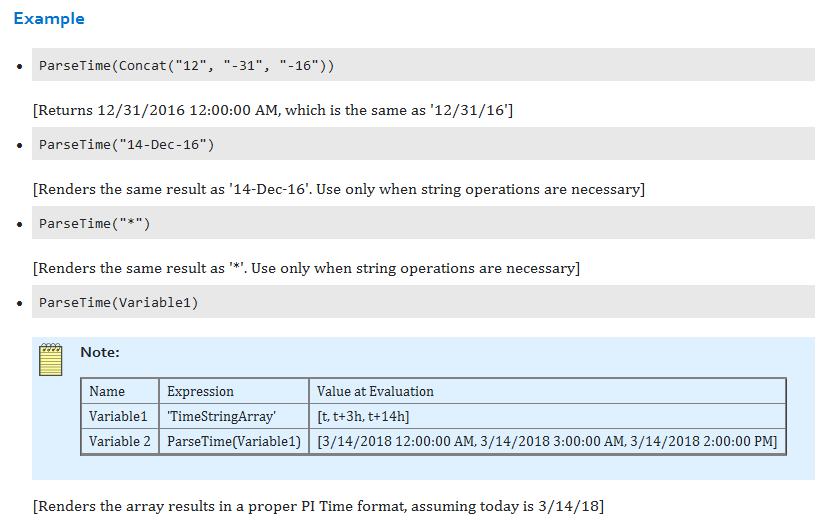
ParseTime(Concat("12", "-31", "-23"))

ParseTime(Concat("12", "-31", "-16"))

ParseTime("14-Dec-16")

ParseTime('\*')

ParseTime([ParseTime 1])



*1.2.3.32 Exit (Остановка дальнейших вычислений)*

IF (if badval('Tag 1') then 0 else TagVal('Tag 1')) < 100 THEN 'Tag 1' ELSE Exit()

(if(BadVal(TagVal('Tag1','\*')), 0, TagVal('Tag1','\*').Value)) > 0 ? TagVal('Tag1','\*').Value : Exit

Останавливает дальнейшие расчёты если выполняется условие Exit

*1.2.3.33 Tagname (Получение имени тега для связанного с ним атрибута)*

TagName(attname)

TagName('Tag 1')

## Аргументы

attname

Имя атрибута с датарефернсом TSDB, в одинарных кавычках

## Результат

Имя тега связанного с attname

## Исключения

Error если тега не существует

#### 1.2.3.34 PrevVal (Последнее записанное значения до указанного времени.)

PrevVal(attname, t1)

PrevVal('Tag 1')

PrevVal('Tag 1','\*')

Параметры

1. tagName типа string;
2. Time типа object;

Выход: Double

#### 1.2.3.35 Abs (Значение по модулю)

Abs(x)

Abs('Tag 1')

Abs(Tagval('Tag 1', '\*').Value)

Параметры

1. tagName типа string;
2. Time типа object;

Результат

Значение по модулю

Исключения

Error в случае если значение не является int или real

Выход: Double

*Новые функции КЧХК*

TagName

Concat

PrevVal

TimeStamp

Roundfrac

Float

Exit

HasChanged

EventCount