Модуль "Expressions"
РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Содержание

1 Введение

Модуль-фильтр "Expressions" для наших программ (например, Advanced Serial Data Logger) применяется для быстрого вычисления выражений и модификации значений переменных, которые сформировал парсер. Это мощный инструмент, легкий в использовании.
Поддерживает различные типы данных: строка, дата, время, десятичное число, переменные и логические данные. Применяются: арифметические и логические операторы, многоуровневые скобки, встроенные математические, строковые, логические функции и определенные пользователем переменные.

Он поддерживает следующие операции и функции:

Математические операции:

- \+ : Сложение
- \- : Вычитание
- * : Умножение
- / : Деление
- ^ : Степень (для основания - только положительные числа)

Математические функции: ABS, ATAN, COS, EXP, LN, ROUND, SIN, SQRT, SQR, TRUNC и многие другие;
Строковые функции: COPY, REPLACE, POS и многие другие;
Логические функции: AND, OR, XOR и т.д.

2 Системные требования

Для установки Expressions должны быть выполнены следующие требования:

Операционная система: Windows 2000 SP4 и выше, включая версии для 32-х и 64-х битных систем, а также серверные версии операционных систем. Наличие установленного последнего сервис-пака желательно.
Свободное дисковое пространство: Рекомендуется не менее 5 MB свободного дискового пространства.
Специальные требования для доступа: Вы должны начать работу как пользователь с правами Администратора, чтобы установить данный модуль.

Необходимо наличие главного приложения (ядра), например Advanced Serial Data Logger.

Примечания для Microsoft Vista и выше:

Поскольку наше программное обеспечение производит запись в реестр и папку Program Files, то необходимо выполнение следующих условий:

1. Вам нужны права Администратора, чтобы запускать наше программное обеспечение
2. Иконка нашего программного обеспечения будет находиться на рабочем столе;
3. Windows Vista спросит Ваше разрешение продолжить установку.

ПРИМЕЧАНИЕ. Вы можете единожды настроить учетную запись пользователя, чтобы больше видеть приведенный выше диалог. Решение этой проблемы - искать в Google.

3 Установка Expressions

1. Закройте главное приложение (например, Advanced Serial Data Logger), если оно запущено;
2. Скопируйте программу на ваш жесткий диск;
3. Запустите программу установки модуля, кликнув два раза на имени файла в проводнике Windows;
4. Следуйте инструкциям программы установки. Обычно достаточно нажать несколько раз кнопку "Дальше";
5. Запустите главное приложение. В случае успешной установки название модуля появится в окне настройки, на закладке "Модули".

Если модуль совместим с программой, то его название и номер версии будет отображаться в списке модулей. Примеры установленных модулей можно посмотреть на рис.1-2. Некоторые типы модулей требуют дополнительной настройки. Для этого достаточно выбрать модуль из списка и нажать кнопку "Настроить" рядом со списком. Процедура настройки модуля описана в следующих главах.

Некоторые типы модулей видны на закладке "Лог-файл". Для их настройки необходимо выбрать модуль из списка "Тип файла" и нажать кнопку "Дополнительно".

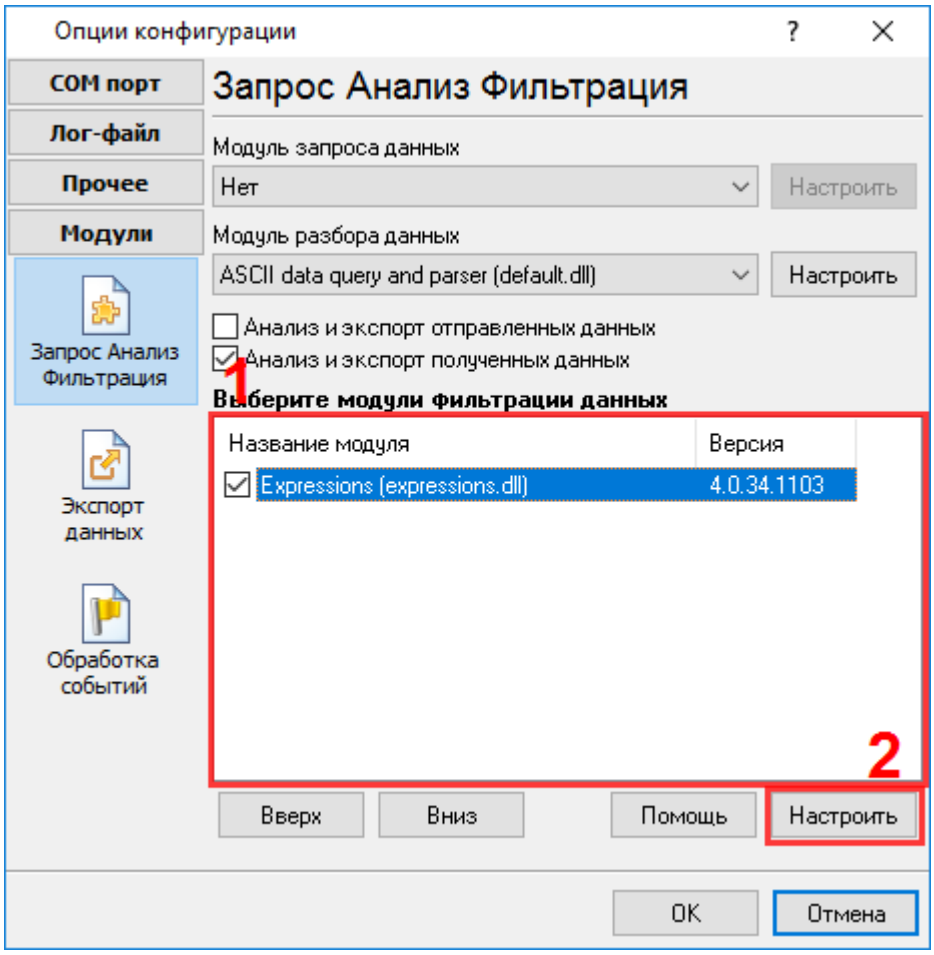

Рис.1. Пример установленного модуля

4 Глоссарий

Plug-in - модуль

Главная программа - программная оболочка, которая использует данный модуль. Например: Advanced Serial Data Logger

Основная программа - см. "Главная программа".

Парсер - модуль, который обрабатывает поток данных, выделяя из него пакеты данных и переменные из пакетов данных. Затем эти переменные используются в модулях экспорта данных.

Ядро - см. "Главная программа".

5 Руководство пользователя

5.1 Конфигурация

Конфигурация модуля очень проста (Рис.1). Вы можете добавить одно или более выражений в окне редактирования. Одна строка содержит одно выражение. Каждое выражение должно иметь следующий формат:

VARIABLE_NAME=EXPRESSION
Где VARIABLE_NAME - имя новой или существующей переменной синтаксического анализатора и EXPRESSION - любой набор функций и/или переменных.

Этот модуль может использовать все существующие переменные синтаксического анализатора (Вы можете выбрать эти переменные в группе "Переменные"). Вы можете прочитать о переменных синтаксического анализатора в файле подсказки синтаксического анализатора. К тому же, дополнительно, здесь Вы можете определить новые переменные и присвоить результат выражений для этих переменные. Поэтому определенные пользователем переменные могут использоваться в модулях экспорта данных как регулярная переменная анализатора.

Все функции, операторы и переменные расположены в выпадающем списке. Если Вы хотите добавить какую-нибудь функцию, тогда Вы должны:

1. Поместить курсор в окно редактирования;
2. Выбрать соответствующую категорию (Дата и Время например);
3. Выбрать функцию, которую Вы должны вставить в окно редактирования (Дата например);
4. Щелкнуть кнопкой "Add / Добавить".

Позже Вы можете напечатать ваши выражения непосредственно, не используя выпадающие списки.

Все поддерживаемые функции описаны в следующих разделах.

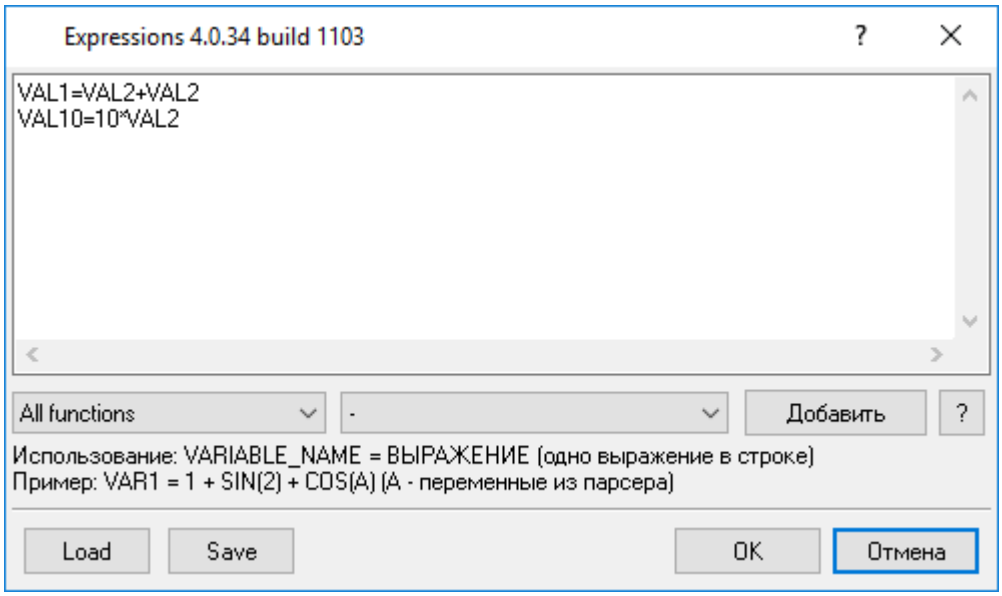

Рис.1. Окно конфигурации модуля

5.2 Поддерживаемые функции

5.2.1 Математические функции

ABS(X) - возвращает абсолютное значение аргумента X. Значение X - тип выражения - целое или реальное.

ARCCOS(X) - возвращает арккосинус (обратный косинус) X. Значение X должно быть между -1 и 1. Возвращаемая величина - в диапазоне [0..Pi], в радианах.

ARCCOSH(X) - возвращает обратный гиперболический косинус X. Значение X должно быть больше или равно 1.

ARCSIN(X) - возвращает обратный синус X. Значение X должно быть между -1 и 1. Возвращаемая величина в диапазоне [-Pi/2..Pi/2], в радианах.

ARCSINH(X) - возвращает обратный гиперболический синус X.

ARCTAN2(X) - вычисляет ArcTan (Y/X), и возвращает угол в правильном квадранте. Значения X и Y должны быть между -2^64 и 2^64. Кроме того, значение X не может быть равно 0. Возвращаемая величина будет лежать в диапазоне от -Pi до Pi, в радианах.

ARCTANH(X) - возвращает обратный гиперболический тангенс X. Значение X должно быть между -1 и 1 (включительно).

CEIL(X) - Ceil вычисляет, самое малое целое число большее или равное X. Значение X не может быть больше MaxInt.

Например:
Ceil(-2.8) = -2
Ceil(2.8) = 3
Ceil(-1.0) = -1

CLIP(X, Min, Max) - Возвращает Min, если X <= Min; возвращает Max, если X >= Max, иначе возвращает X.
Например:
CLIP(2, 3, 4) = 3
CLIP(3, 2, 4) = 3
CLIP(4, 2, 3) = 3

COS(X) - используют, чтобы вычислять косинус X, где X - угол в радианах;

COSH(X) - используют, чтобы вычислить гиперболический косинус X, где X - угол в радианах;

COTAN(X),
COTG(X) - Cotan или Cotg используют, чтобы получить котангенс X. Котангенс вычисляется по формуле - 1 / Tan(X);

DEG(X) - используют, чтобы углы, выраженные в радианах, преобразовывать в соответствующие величины в градусах, где градусы = (180 / Pi) радиан;

EXP(X) - возвращает значение E в степени X, где E - основание натуральных логарифмов;

FLOOR(X) - используют, чтобы получить округленное целое число. Округление производится в сторону уменьшения, т.е. итог меньше или равняется X.
например:
Floor(-2.8) = -3
Floor(2.8) = 2
Floor(-1.0) = -1
Примечание: Значение X должна быть меньше, чем максимальное вещественное число.

FRAC(X) - возвращает дробную часть аргумента X. X - реальный тип данных. Результатом является дробная часть X; то есть, Frac(X) = X - Int(X).

HEX(X) - Преобразовывает значение X в шестнадцатеричный код.

LN(X) - Возвращает натуральный логарифм (Ln(e) = 1) выражения X. X - реальный тип данных.

LOG(Base, X) - Возвращает логарифм числа X по заданному основанию Base.

POW(Base, Exponent), POWER(Base, Exponent) - Возводит Base в степень Exponent, где Exponent - любое значение. Для значений Exponent дробных или больше, чем 65535, значение Base должно быть больше 0;

POWLN2(X) - Возводит Ln2 в степень X, где X - любое значение;

RAD(X) - используют, чтобы углы, выраженные в градусах, преобразовывать в соответствующие величины в радианах, где радианы = (Pi / 180) градусов.

RANDOM(X) - Возвращает случайное число в пределах диапазона 0 <= X < Диапазон. Если диапазон не определен, результатом является произвольное число реального типа в пределах диапазона 0 <= X < 1.

Примечание: Поскольку реализация функции **RANDOM(X)** может отличаться между версиями модуля, мы не рекомендуем использовать **RANDOM(X)** для шифрования или других целей, которые требуют воспроизведения последовательности псевдослучайных чисел.

ROUND(X) - эта функция преобразует величины реального типа в тип целой величины (округляет). X - выражения реального типа. **ROUND(X)** возвращает величину типа Int64, которая округлена до ближайшего целого. Если число X - это половина между двумя целыми числами, результатом является четное число. Этот метод округления часто называют "Banker's Rounding".

SIGN(X) - используется, чтобы проверить признак числового значения. Возвращается признак числа.

0 если значение нулевое.
1 если значение больше нуля.
-1 если значение меньше нуля.

SIN(X) - используется, чтобы вычислить синус X, где X - угол в радианах;

SINH(X) - используется, чтобы вычислить гиперболический синус X;

SQR(X) - возвращает квадрат аргумента. X - выражения с плавающей точкой. Тип результат, такой же, как X, - квадрат X, или X*X.

SQRT(X) - X - выражения с плавающей точкой. Результат - квадратный корень X.

TAN(X), TG(X) - Tan или Tg возвращает тангенс X. Tan(X) = Sin(X) / Cos(X).

TRUNC(X) - эта функция преобразует величины реального типа в тип целой величины (округляет). X - выражения реального типа. TRUNC(X) возвращает величину типа Int64, которая округлена до целого числа путем отбрасывания дробной части.

5.2.2 Операторы

- - вычитание

***** - умножение

/ - деление

^ или ****** - возводит базовое число в любую степень. Для дробных значений степени или больше, чем 65535, базовое значение должно быть больше 0. Пример: x**y

+ - сложение

< - меньше

<= - меньше или равно

<> - не равно

= - равно

> - больше

>= - больше или равно

AND - логическое AND, тип операндов: целое. Тип результата: целое. Например: X and Y

DIV - деление до целого. Величина деления x div y является величиной x/y округленной до ближайшего целого.

MOD - остаток. **MOD** возвращает остаток, полученный при делении операндов. Другими словами, x mod y = x - (x div y) * y.

OR - логическое OR, тип операндов: целое. Тип результата: целое. Например: X or Y

SHL - разрядный сдвиг влево, тип операндов: целое. Тип результата: целое. Например: X shl 2

SHR - разрядный сдвиг вправо, тип операндов: целое. Тип результата: целое. Например: X shr 2

XOR - логическое XOR, тип операндов: целое. Тип результата: целое. Например: X xor Y

5.2.3 Строковые функции

FIRSTLINE(S) - копирует все символы с начала до символа CR или LF.

REMOVECHAR(S, Char) - Возвращает строку с удаленным одним символом. Где **Char** - удаленный символ, **S** - исходная строка, в которой будет удален символ. Операция поиска строки чувствительна к регистру символов.

REMOVENONPRINT(S) - Возвращает строку **S** со всеми удаленными непечатными символами (с кодом ASCII < 32).

REPLACE(S, OldPattern, NewPattern) - Возвращает строку с заменой одной подстроки другой подстрокой. **REPLACE** заменяет подстроки определенной **OldPattern** с подстрокой определенной **NewPattern**. **S** - исходная строка, чьи подстроки должны измениться. **OldPattern** - подстрока, которая должна быть заменена **NewPattern**. **NewPattern** - подстрока, которая заменяет **OldPattern**. Операция поиска строки ***не****чувствительна* к падежам.

REPLACECHAR(S, OldChar, NewChar) - Возвращает строку с заменой одной подстроки другой подстрокой.
REPLACECHAR заменяет подстроки определенной **OldChar** с подстрокой определенной **NewChar**. **S** - исходная строка, чьи подстроки должны измениться. **OldChar** - подстрока, которая должна быть заменена **NewChar**. **NewChar** - подстрока, которая заменяет **OldChar**. Операция поиска строки *чувствительна* к падежам.

SUBSTR(S, Index, Count), STRCOPY(S, Index, Count), COPY(S, Index, Count) - Возвращает подстроку строки. **Copy** возвращает подстроку или подмассив, содержащие **Count** символов или элементов, начинающихся в **S** позиции **[Index]**. Если в **Count** указано больше символов или элементов чем доступно в массиве, то возвращены будут все символы или элементы из **S** начинающиеся с позиции **[Index]** и до конца строки. Если **Index** больше, чем длина **S**, **Copy** возвращает пустую строку или массив. Подстрока является уникальной копией (то есть, **S** не запоминается; если элементы массива - указатели или объекты, они также не копируются).

STRPOS(Substr, S), POS(Substr, S) - поиск подстроки **Substr** в строке **S**. Тип выражений **Substr** и **S** - строка. **Pos** ищет **Substr** в пределах **S** и возвращает значение позиции подстроки в виде целого числа, которое является индексом первого знака **Substr** в пределах **S**. Функция **Pos** чувствительна к регистру символов. Если **Substr** не обнаружен, **Pos** возвращает ноль.

TRIMLEFT(S), **LTRIM(S)** - отбрасывает находящиеся в начале строки пробелы и управляющие символы.

TRIMRIGHT(S), RTRIM(S) - отбрасывает находящиеся в конце строки пробелы и управляющие символы;

TRIM(S) - отбрасывает находящиеся в начале и конце строки пробелы и управляющие символы.

5.2.4 Дата и время

DATE() - Возвращает текущую дату. Тип возвращаемой величины - DateTime.

DATE(S) - Возвращает значение даты, типа DateTime, преобразованное из строки S. Формат строки S - 'DD.MM.YYYY'. Пример: DATE('15.01.2007')

DATE(Y,M,D) - Возвращает значение даты, типа DateTime, преобразованное из величин Y (год), M (месяц), D (день) (целые значения). Пример: DATE(2007, 1, 15)

DAY(X) - Возвращает значение дня из величины X. Значение X должно быть типа DateTime.

GOMONTH(X,Y) - Добавляет месяцы Y к дате X. Y может быть отрицательной или положительной целой величиной. Значение X должно быть типа DateTime.

MONTH(X) - Возвращает значение месяца из величины X. Значение X должно быть типа DateTime.

NOW - Возвращает текущую дату и время. Тип возвращаемой величины - DateTime.

TIME() - Возвращает текущее время. Тип возвращаемой величины - DateTime.

TIME(S) - Возвращает значение времени, преобразованного из строки S. Формат строки S - 'HH:NN'. Пример: TIME('15:21'). Тип возвращаемой величины - DateTime.

TIME(H,M,S,MS) - Возвращает значение даты, преобразованного из величин H (час), M (минуты), S (секунды) и MC (миллисекунды). Пример: TIME(15, 21, 0, 0). Тип возвращаемой величины - DateTime.

YEAR(X) - Возвращает значение года из величины X. Значение X должно быть типа DateTime.

5.2.5 Прочие функции

Функции условий

IIF(X,Y,Z) - если логическое условие X является Истиной, тогда определяется и возвращается Y, в противном случае определяется и возвращается Z.

NVL(X,Y) - если переменная X не существует, не задана или ее значение равно NULL (пустое значение), то функция вернет значение Y.

DISCARD_DATA_PACKET_IF(X,Y) - если логическое условие X является Истиной, тогда обрабатываемый пакет данных будет проигнорирован. Данные из этого пакета не будут обрабатываться модулями фильтрации, расположенными ниже по списку и всеми модулями экспорта данных. Если задан опциональный параметр Y, то в лог сообщений программы будет выведено соответствующее сообщение.

Пример:

DISCARD_DATA_PACKET_IF(VAR > 10, "Value is too big")

GENERATE_EVENT_IF(X,Y,N1,V1,N2,V2) - если логическое условие X является Истиной, тогда программа сгенерирует внутреннее событие с именем Y, которое может быть обработано модулями обработки событий. Если заданы дополнительные параметры N1, V1 .. Nn, Vn, то эти значения будут переданы как параметры события.

SEND_EVENT_IF - является синонимом GENERATE_EVENT_IF.

Пример:

GENERATE_EVENT_IF(VAR > 10, "VAR_TOO_BIG_EVENT", "VAR_NAME", "VAR", "VAR_VALUE", VAR)

REDIRECT_DATA_IF(X, Y) - если логическое условие X является Истиной, тогда обрабатываемый пакет данных будет перенаправлен в другую конфигурацию с именем Y, в которой он пройдет всю цепочку модулей фильтрации и экспорта данных. Обработка пакета данных в данной конфигурации будет продолжена. Если дальнейшая обработка не требуется, то сразу за этой функцией можно вызвать DISCARD_DATA_PACKET_IF.

Пример:

REDIRECT_DATA_IF(VAR > 10, "COM2")
DISCARD_DATA_PACKET_IF(1=1)

SEND_BYTE_IF(X, Y) - если логическое условие X является Истиной, тогда байт Y будет отправлен в источник данных (COM порт или TCP порт). Обратите внимание, что не все логгеры и источники данных могут отправлять данные.

SEND_DATA_IF(X, Y) - если логическое условие X является Истиной, тогда **строка** Y будет отправлена в источник данных.

SEND_DATA_TO_DATA_SOURCE_IF(X, Z, Y) - если логическое условие X является Истиной, тогда **строка** Y будет отправлена в источник данных Z.

Пример:

SEND_DATA_TO_DATA_SOURCE_IF(VAR > 10, "COM2", "Data string" + CHR(13) + CHR(10))

Обратите внимание, что вы должны создать и настроить конфигурацию с именем Y. Т.е. она должна присутствовать в выпадающем списке в главном окне логгера.

Основные функции

MAX(A,B) - используют, чтобы сравнивать две числовые величины. MAX возвращает бОльшую величину.

MIN(A,B) - используют, чтобы сравнивать две числовые величины. MIN возвращает меньшую величину.

SUM(A,B) - Значение равно A+B, где A и B может быть строкой или числом.

Прочие функции

BYTETOSTR(X) - преобразует 1 байт во множестве X в величину байта без знака.

DOUBLETOSTR(X) - преобразует 8 байт в массиве X в двоичный код.

DOUBLETOSTRBE(X) - преобразует 8 байт в порядке "Big-endian" в массиве X в двоичный код

INT64TOSTR(X) - преобразует 8 байт в массиве X в целое число для 64 битовых величин.

INT64TOSTRBE(X) - преобразует 8 байт в порядке "Big-endian" в массиве X в целое число для 64 битовых величин.

LONGINTTOSTR(X) - преобразует 4 байта в массиве X в целое число для 32 битовых величин.

LONGINTTOSTRBE(X) - преобразует 4 байта в порядке "Big-endian" в массиве X в целое число для 32 битовых величин.

LONGWORDTOSTR(X) - преобразует 4 байта в массиве X в десятичное число без знака для 32 битовых величин.

LONGWORDTOSTRBE(X) - преобразует 4 байта в порядке "Big-endian" в массиве X в десятичное число без знака для 32 битовых величин.

SINGLETOSTR(X) - преобразует 4 байта в массиве X в переменную величину.

SINGLETOSTRBE(X) - преобразует 4 байта в порядке "Big-endian" в массиве X в переменную величину.

SMALLINTTOSTR(X) - преобразует 2 байта в массиве X в десятичное число для 16 битовых величин.

SMALLINTTOSTRBE(X) - преобразует 2 байта в порядке "Big-endian" в массиве X в десятичное число для 16 битовых величин.

WORDTOSTR(X) - преобразует 2 байта в массиве X в десятичное число без знака для 16 битовых величин.

WORDTOSTRBE(X) - преобразует 2 байта в порядке "Big-endian" в массиве X в десятичное число без знака для 16 битовых величин.

Функции не попавшие в документацию

Описание остальных функций можно найти в Google по запросу:

pascal "имя_функции"

или

delphi "имя_функции"

6 Проблемы?

6.1 Возможные проблемы

Модуль отсутствует в списке или помечен как "не установлен" - убедитесь, что модуль был установлен нужную папку. Все модули должны располагаться в подпапке "Plugins" в папке с программой. Также необходимо убедиться, что модуль совместим с вашей версией логгера. Если модуль несовместим, то сообщение об этом должно появиться в логе сообщений программы в главном окне.

Нет данных для публикации (экспорта) – никаких данных не было передано для экспорта. Решение: настройте парсер, убедитесь, что в парсере объявлена одна или несколько переменных.

Ошибка при присваивании значения переменной или параметра %s [%s] – ошибка обычно возникает, если данные не соответствуют указанному формату. Например, формат даты или времени не соответствует данным.

Ошибка при формировании параметра (%s) – программа не может преобразовать переменную одного формата в другой. Это проблема возникает, когда вы пытаетесь произвести экспорт переменной одного типа данных в столбец базы данных, который имеет другой несовместимый тип данных. Для исправления ошибки измените тип данных в парсере или в модуле экспорта данных.

Не удалось соединиться %s. (%s) – ошибка возникает, когда модуль не может подсоединиться к базе данных. Проверьте ваши настройки соединения в модуле экспорта данных.

Не удалось отсоединиться %s. (%s) – ошибка возникает, когда модуль не может отсоединиться от базы данных. Обычно, ошибка возникает при нарушении связи с базой данных (нет сети, база данных недоступна).

Если у вас возникли другие проблемы, то пишите на support@aggsoft.ru. Все решим в кратчайшие сроки.

Замечание: в тексте ошибки выше, выражение "%s" будет заменено на дополнительные данные.