



Версия 5.1

# Руководство Пользователя

© ООО «Энергокруг », 2020. Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотографирование, магнитную запись или иные средства копирования или сохранения информации, без письменного разрешения владельцев авторских прав.

Все упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат своим законным владельцам.

---

## **ООО «Энергокруг »**

РОССИЯ, 630049, г.Новосибирск, ул. Галушака 2А, офис 317/1

Тел. +7 (8412) 55-64-95

Общие вопросы: [info@scadadatarate.ru](mailto:info@scadadatarate.ru), [info@energokrug.ru](mailto:info@energokrug.ru)

Техническая поддержка: [support@scadadatarate.ru](mailto:support@scadadatarate.ru)

<http://www.ScadaDataRate.ru/>

<http://www.energokrug.ru/>

## ОБ ЭТОЙ КНИГЕ

**SCADA *DataRate*™** – универсальное средство мониторинга, контроля и управления производственными процессами. Использование ***DataRate*** позволит Вам легко и эффективно:

- Построить автоматизированную систему (АСУ ТП), осуществляющую контроль и управление на базе интеллектуальных датчиков, контроллеров и компьютеров
- Создать операторский интерфейс для отображения и сохранения информации с устройств, поддерживающих обмен данными по технологии OPC DA/HDA/A&E
- Визуализировать данные из базы данных системы управления производством
- Организовать систему отчетности
- Производить удаленный мониторинг и управление системой с использованием Web-технологий.

Руководство пользователя **SCADA *DataRate*™** приведено в данной книге.

Название книги	Название части	Содержание
<b>SCADA <i>DataRate</i>™</b>  Руководство Пользователя  Обозначение документа:  ИГТЯ.20500W- 05.10-И2.3.1	<u>Часть 1</u>  <b>ОСНОВЫ <i>DataRate</i></b>  Руководство Пользователя ИГТЯ. 20500W-05.10-И2.3.2	1. Интегрированная среда разработки 2. Объектная модель 3. Динамическая визуализация данных
	<u>Часть 2</u>  <b>ОБЪЕКТЫ И СЕРВИСЫ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ</b>  Руководство Пользователя ИГТЯ. 20500W-05.10-И2.3.3	4. События и тревоги (алармы) 5. Расписания 6. История процесса. Тренды 7. Модуль отчетов
	<u>Часть 3</u>  <b>ОБМЕН ДАННЫМИ И СРЕДСТВА ИНТЕГРАЦИИ</b>  Руководство Пользователя ИГТЯ. 20500W-05.10-И2.3.4	8. Обмен данными 9. Источники данных и параметры 10. Система ограничения доступа 11. Среда исполнения 12. Среда исполнения клиент 13. Web-контроль 14. Резервирование 15. Техническая поддержка

Информация, содержащаяся в данной книге, не может гарантировать абсолютную точность и полноту приводимых сведений. Это связано с возможными человеческими или техническими ошибками, допущенными в процессе подготовки информации, а также с политикой совершенствования и развития ***DataRate***.

ООО «Энергокруг» не несет ответственности за возможные ошибки, связанные с использованием данной информации.

Надеемся, что SCADA ***DataRate*** позволит Вам успешно разрабатывать и эксплуатировать автоматизированные системы контроля и управления.

С уважением, ООО «Энергокруг»







Версия 5.1

## ЧАСТЬ 2

# **ОБЪЕКТЫ И СЕРВИСЫ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ**

Руководство Пользователя

Объекты и сервисы промышленной автоматизации.

Руководство Пользователя./3-е изд

© ООО «Энергокруг », 2020. Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотографирование, магнитную запись или иные средства копирования или сохранения информации, без письменного разрешения владельцев авторских прав.

Все упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат своим законным владельцам.

---

## **ООО «Энергокруг »**

РОССИЯ, 630049, г.Новосибирск, ул. Галуцака 2А, офис 317/1

Тел. +7 (8412) 55-64-95

Общие вопросы: [info@scadadatarate.ru](mailto:info@scadadatarate.ru), [info@energokrug.ru](mailto:info@energokrug.ru)

Техническая поддержка: [support@scadadatarate.ru](mailto:support@scadadatarate.ru)

<http://www.ScadaDataRate.ru/>

<http://www.energokrug.ru/>

## СОДЕРЖАНИЕ

	Стр.
<b>4 СОБЫТИЯ И ТРЕВОГИ (АЛАРМЫ)</b>	<b>4-1</b>
4.1 Протокол событий	<b>4-3</b>
4.1.1 Настройки Протокола событий	4-4
4.1.2 Графический примитив Протокол событий	4-25
4.1.3 События элемента проекта	4-28
4.1.4 Регистрация событий в протоколе	4-34
4.1.5 Печать сообщений о событиях из скрипта	4-39
4.1.6 Экспорт протокола событий из скрипта	4-44
4.1.7 Чтение протокола событий из скрипта	4-49
4.2 Сигнализация	<b>4-49</b>
4.2.1 Установка сигнального подсостояния	4-53
4.2.2 Сигнализация контроля уровня	4-55
4.2.3 Дискретная сигнализация	4-59
4.2.4 Сигнализация отклонения	4-60
4.2.5 Пользовательская сигнализация	4-60
4.2.6 Сигнализация скорости изменения	4-60
4.2.7 Квитирование состояния	4-61
4.2.8 Сводное сигнальное состояние. Привязка к сигнализации	4-62
4.2.9 Доступ к сигнализации	4-64
4.2.10 Пример сигнализации в проекте	4-67
4.2.11 Журнал тревог	4-70
4.2.12 Окно активной сигнализации	4-83
4.3 Оповещения	<b>4-87</b>
4.3.1 Менеджер оповещений	4-87
4.3.2 Создание, привязка аргумента и срабатывание оповещений	4-97
4.3.3 Звуковое оповещение	4-100
4.3.4 Почтовое оповещение	4-101
4.3.5 SMS оповещение	4-104
<b>5 РАСПИСАНИЯ</b>	<b>5-1</b>
5.1 Папки расписаний	5-1
5.2 Создание расписания	5-2
5.3 Привязка к расписанию	5-3
5.4 Редактирование свойств расписания	5-4
5.4.1 Расписание с однократной активацией	5-4
5.4.2 Расписание с повторяющейся активацией	5-4
<b>6 ИСТОРИЯ ПРОЦЕССА. ТРЕНДЫ</b>	<b>6-9</b>
6.1 Менеджер трендов	<b>6-10</b>
6.1.1 Группы настроек истории	6-10
6.1.2 Проверка истории	6-21
6.1.3 Перья Менеджера трендов	6-25
6.2 Тренды объектов	<b>6-31</b>
6.2.1 Мастер создания перьев для объекта	6-31
6.3 Тренды тегов	<b>6-32</b>
6.3.1 Мастер создания перьев для тега	6-33
6.3.2 Настройка трендирования тегов	6-33
6.3.3 Настройка трендирования атрибутов	6-34
6.3.4 Выбор тегов по маске	6-35
6.3.5 Просмотр истории тега объекта	6-35
6.3.6 Удаление точки из тренда	6-37
6.4 Графический элемент Тренд	<b>6-39</b>
6.4.1 Создание графического элемента Тренд	6-39

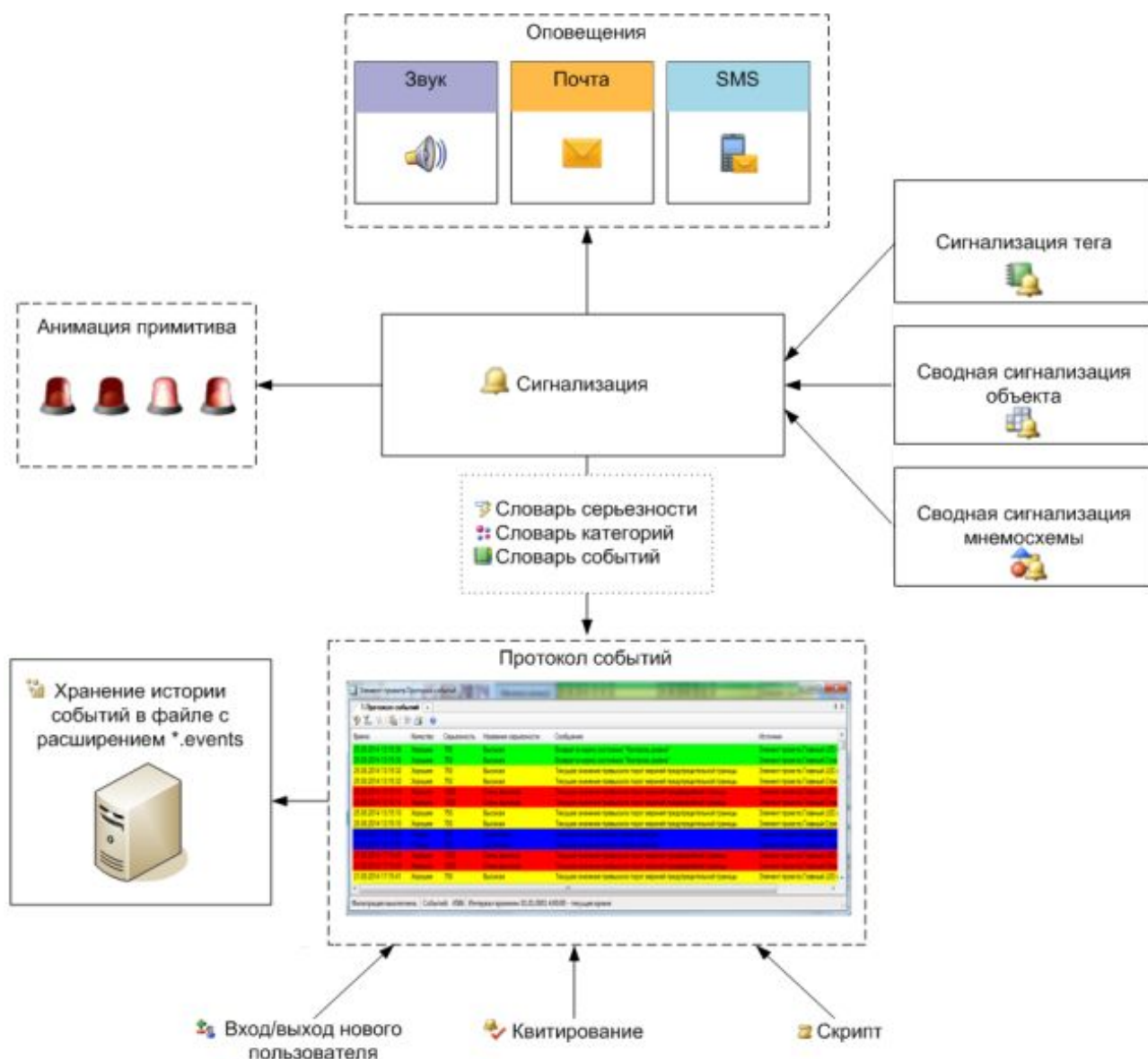
6.4.2	Общие настройки	6-40
6.4.3	Настройка координатной сетки	6-40
6.4.4	Добавление/Удаление перьев	6-42
6.4.5	Настройка перьев	6-47
6.4.6	Управление графическим элементом Тренд	6-57
6.5	Графический элемент Табличный тренд	<b>6-63</b>
6.5.1	Настройка запроса данных	6-64
6.5.2	Просмотр истории	6-66
6.5.3	Световое перо	6-67
6.5.4	Свойства колонки Перо	6-68
6.5.5	Свойства колонки Время	6-70
6.6	Аналитический тренд	<b>6-71</b>
6.6.1	Создание аналитического тренда	6-72
6.6.2	Область тренда	6-73
6.6.3	Настройки тренда	6-81
6.6.4	Настройки перьев	6-86
6.6.5	Сохранение изображения тренда	6-102
6.6.6	Изменение данных скриптом	6-103
6.6.7	Источники данных привязок	6-105
6.7	Режимный тренд	<b>6-107</b>
6.7.1	Интервальный режимный тренд	6-108
6.7.2	Полномасштабный режимный тренд	6-113
6.8	Доступ к трендам из скрипта	<b>6-119</b>
6.8.1	Чтение значений из трендов	6-119
6.8.2	Обновление данных	6-121
6.8.3	Запись значений в тренд	6-122
<b>7</b>	<b>МОДУЛЬ ОТЧЕТОВ</b>	<b>7-1</b>
7.1	Шаблон отчета	<b>7-2</b>
7.1.1	Секции отчета	7-6
7.1.2	Элементы отчета	7-10
7.1.3	Встроенные рисунки	7-43
7.1.4	Форматирование элементов отчета	7-44
7.2	Разбивка отчета на страницы	<b>7-48</b>
7.3	Предварительный просмотр	<b>7-49</b>
7.4	Взаимодействие с пользователем	<b>7-51</b>
7.5	Автоматическое изменение размера и позиционирование	<b>7-55</b>
7.6	Использование выражений	<b>7-56</b>
7.6.1	Создание выражений	7-57
7.6.2	Работа с выражениями	7-59
7.6.3	Применение глобальных коллекций в выражениях	7-62
7.6.4	Примеры выражений	7-66
7.6.5	Использование функций в выражениях	7-69
7.7	Менеджер отчетов	<b>7-85</b>
7.7.1	Хранение снимков данных	7-86
7.7.2	Печать отчетов	7-89
7.7.3	Рассылка отчетов	7-90
7.7.4	Публикация в папку	7-93
7.7.5	Публикация на FTP сервер	7-96
7.8	Генерация отчета	<b>7-99</b>

## 4 СОБЫТИЯ И ТРЕВОГИ (АЛАРМЫ)

### Определения

**Состояние** описывает переменные свойства объекта и включает множество стабильных значений параметров объекта. Состояние может существовать в течение некоторого времени (например, секунд, минут, часов, дней или недель).

**Сигнализация (тревога, alarm)** – аварийное состояние, т.е. специальный случай какого-либо состояния, требующего специального внимания.



Сигнализация используется для уведомления оператора о состояниях системы или объекта, которые потенциально могут привести к проблемам.

Как правило, сигнализации настраиваются на срабатывание в том случае, если некоторая характеристика системы или объекта выйдет за установленный предел.

**Сводная сигнализация** (*summary alarm*) – обобщенный признак текущего состояния сигнализации (активная в данный момент сигнализация).

**Сводное сигнальное состояние** – обобщенная сигнализация для объекта или вида объекта. Используется для оповещения.

Проект или элемент проекта могут иметь несколько Сводных сигнальных состояний, объект или вид объекта – только одно.

**Исторические сигнализации** (*historical alarms*) – сигнализации, неактивные в данный момент. Обычно они хранятся в базе сигнализаций. Например, может потребоваться просмотреть сводку всех текущих сигнализаций, ожидающих подтверждения, в то время как вся остальная информация об аварийных состояниях представляет лишь исторический интерес и поэтому является менее экстренной.

**Квитирование** – подтверждение состояния. Обычно оператор квитирует аварийное состояние.

**Подсостояние** – множество взаимоисключающих состояний, определяющих одно, "главное", состояние.

Например, сигнализация **Контроль уровня** включает следующие подсостояния:

- **ВАГ** – верхняя аварийная граница
- **ВПГ** – верхняя предупредительная граница
- **НПГ** – нижняя предупредительная граница
- **НАГ** – нижняя аварийная граница.

**Событие** – любое единичное изменение, факт возникновения которого имеет значение и должен быть зафиксирован (зарегистрирован) в системе с меткой времени его появления. Событие является мгновенным: как только оно произошло, оно исчезает.

Событие может быть связано с сигнализацией. Один из видов событий – переход в состояние сигнализации или выход из него. Событие может представлять собой действие оператора (например, изменение конфигурации системы, задание уставок) и в этом случае оно не связано с сигнализацией.

События визуализируются в виде сообщений о состоянии системы.

### **ВНИМАНИЕ!!!**

Таким образом, Сигнализация представляет собой состояние, а уведомление о сигнализации – событие.

Изменение состояния может представлять собой, например, вход элемента системы в состояние тревоги, выход из него, а также – квитирование.

**Сообщение** – структурированная текстовая строка, которая отображает для пользователя информацию об одном или нескольких логически связанных событиях.

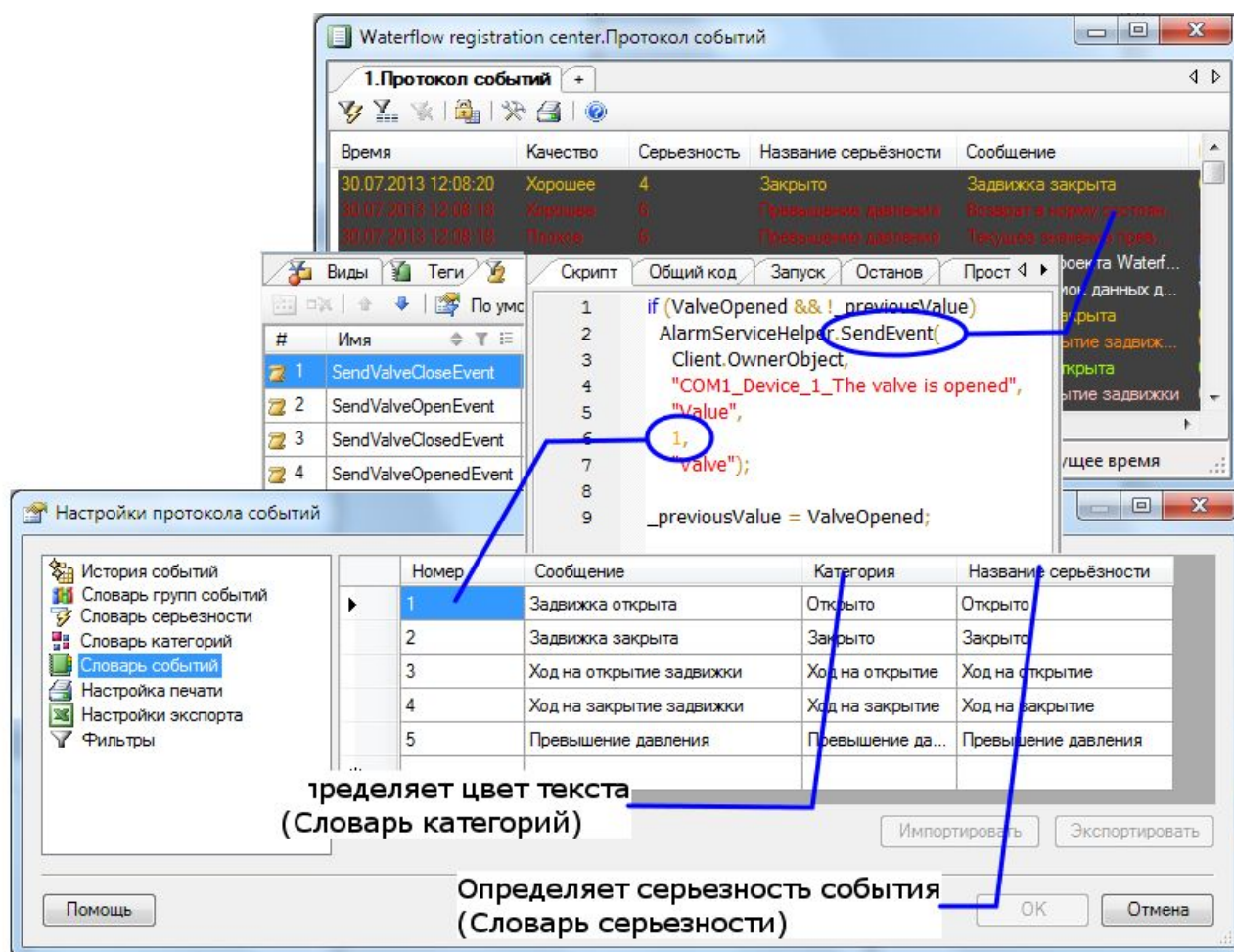
**Сообщения о событиях** регистрируются в протоколе событий и отображаются в примитиве **Протокол событий** и в окне сообщений **Элемента проекта.Протокол событий**.

## 4.1 Протокол событий

Для формирования и отображения потока событий **DataRate** предоставляет:

- Системный объект **Протокол событий** – конфигурирование, отображение и формирование истории событий (архива сообщений)
- Графический примитив **Протокол событий** – настройка и отображение протокола
- Библиотечный скрипт **Event Log** – посылка сообщений в протокол событий
- **Словарь групп событий** – словарь типов событий (**Связанные с состоянием, Трекинга, Простые**)
- **Словарь серьезности** – словарь важности событий
- **Словарь категорий** – словарь категорий событий
- **Словарь событий** – словарь сообщений

Взаимосвязь этих компонент показана на следующем рисунке



Номер сообщения в вызове функции **SendEvent** определяет текст сообщения и цвета его отображения в протоколе событий. Скрипт запускается при изменении значения атрибута тега, к которому привязан аргумент скрипта **ValveOpened**.



### Ведение протокола событий

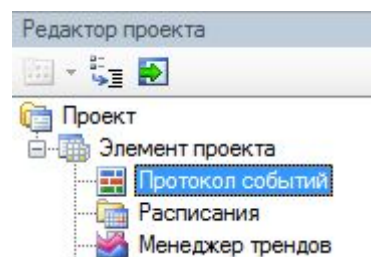
Для формирования протокола событий следует:

- 1 Сформировать словари серьезности, категорий и событий (или использовать системные словари **DataRate**)
- 2 Создать скрипты (на основе системной библиотеки) для отправки сообщений в протокол событий
- 3 Привязать аргументы скриптов к изменяемым атрибутам тегов.

#### ВНИМАНИЕ!!!

**Скрипт запускается только при изменении значения атрибута!**

Для ведения протокола событий используется системный сервис **Протокол событий**, входящий в состав элемента проекта.



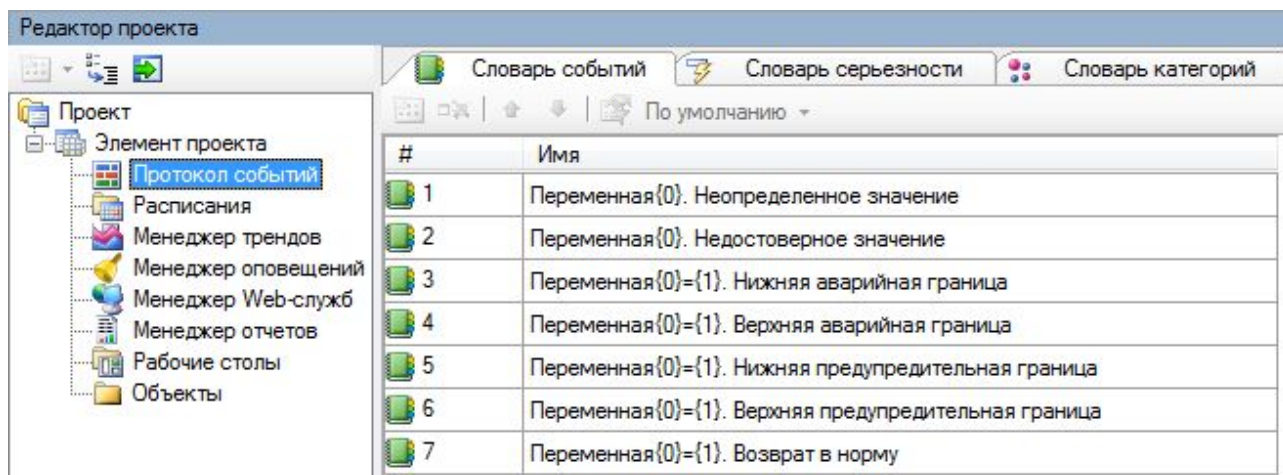
### Визуализация протокола событий

Для визуализации протокола событий используются графический примитив **Протокол событий** и окно сообщений **Элемента проекта.Протокол событий**.

#### 4.1.1 Настройки Протокола событий

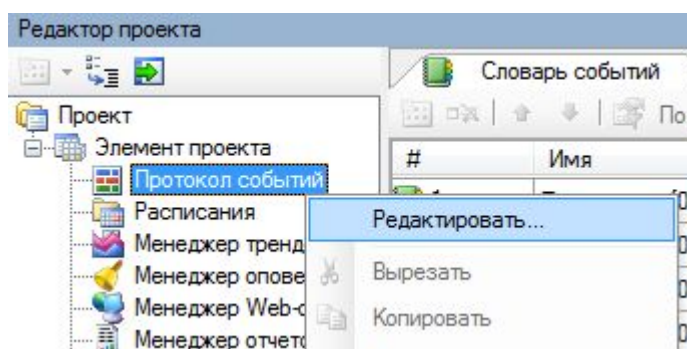
Для того чтобы настроить **Протокол событий** следует:

- 1 Выбрать **Протокол событий** в структуре проекта (в общем случае, **Проект/Элемент проекта/Протокол событий**)

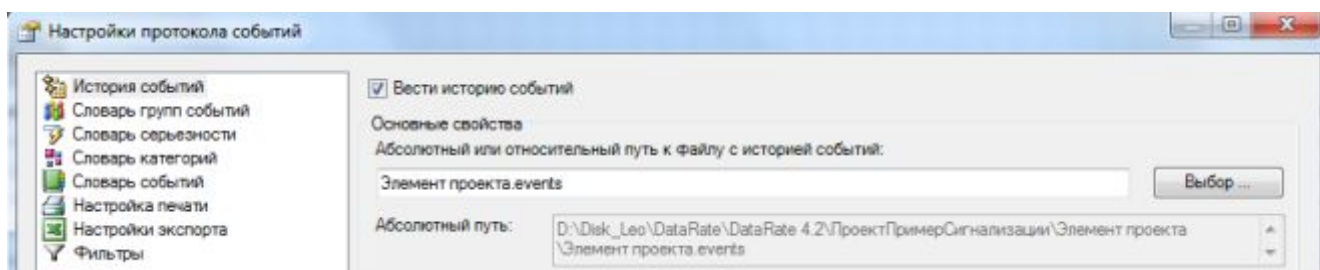


- 2 Вызвать окно **Настройки протокола событий** двойным щелчком левой клавиши мыши по **Протокол событий** в дереве проекта или выбрать в контекстном меню протокола событий пункт **Редактировать**.





- 3 В левой части появившегося окна **Настройки протокола событий** выбрать необходимый элемент и задать в правой части окна соответствующие настройки.



Для подтверждения сделанных изменений нужно нажать кнопку «**ОК**», для отмены – кнопку «**Отмена**».

Заккрытие окна при помощи кнопки в заголовке также отменяет сделанные изменения.

Для вызова справки по выбранной настройке нажмите кнопку «**Помощь**».

#### 4.1.1.1 Настройки хранения истории

Вкладка **История событий** предназначена для настройки параметров создания и хранения архива (истории событий):

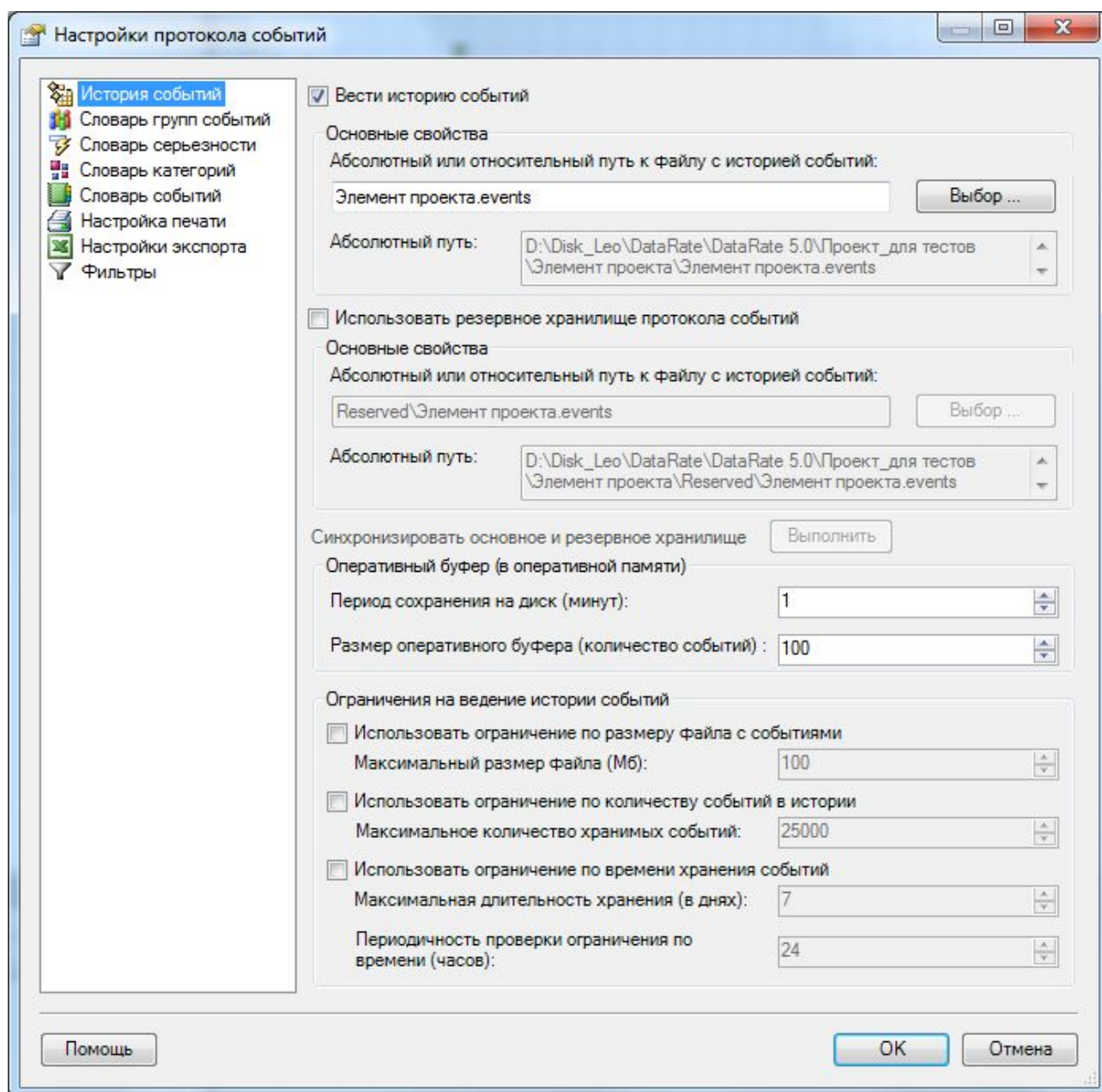
- **Размера файла с историей** (архива)
- **Размера оперативного буфера**
- **Периода сохранения на диск.**

#### **ВНИМАНИЕ!!!**

Если вести историю событий не нужно, снимите галочку «Вести историю событий»!

В этом случае в режиме исполнения принятые события будут храниться только в оперативном буфере (размер буфера задается параметром **Размер оперативного буфера**) и периодически удаляться при приходе новых событий в случае переполнения буфера.

Сохранение на диск производиться не будет!



Галочка **Вести историю событий** установлена по умолчанию. В этом случае принятые события будут сохраняться в файле на жестком диске в соответствии с заданными настройками.

### **ВНИМАНИЕ!!!**

Рассматриваемые ниже настройки доступны только при включенном режиме **Вести историю событий** (за исключением параметра **Размер оперативного буфера**).

### **Основные свойства**

Группа настроек **Основные свойства** предназначена для задания пути к файлу с историей:

- **Абсолютный или относительный путь к файлу с историей событий** – путь можно задать вручную, заполнив поле, или выбрать с помощью стандартного диалогового окна выбора файлов, нажав на кнопку **Выбор...** справа от текстового поля

- **Абсолютный путь** – полный путь к файлу (текстовое поле **только для чтения**).

### Оперативный буфер

Группа настроек **Оперативный буфер** определяет:

- **Период сохранения на диск** – периодичность (в минутах), с которой события будут сохраняться на диск, не зависимо от того, на сколько заполнен оперативный буфер событий.
- **Размер оперативного буфера** – максимальное количество событий, которое могут одновременно находиться в буфере.  
Если ведется история, то при заполнении буфера происходит сброс событий на диск.

При переходе проекта из режима выполнения в режим останова, все несохраненные события из буфера сбрасываются в файл на диске.

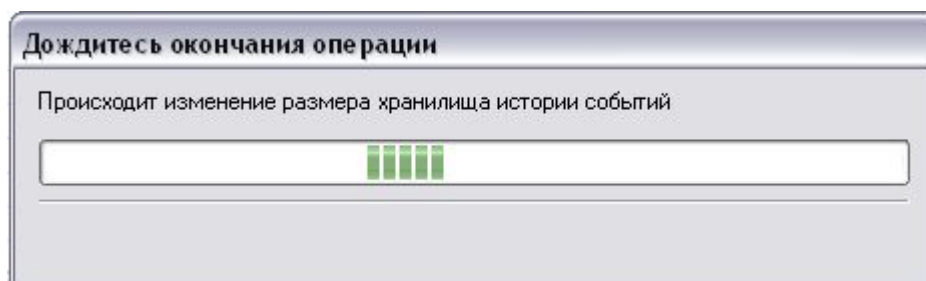
#### **ВНИМАНИЕ!!!**

**Рекомендуем оставить настройки оперативного буфера по умолчанию.  
Настройки влияют на производительность приложения.**

### Ограничение на ведение истории событий

Группа настроек **Ограничение на ведение истории событий** позволяет настроить ограничения на размер файла с историей:

- **Использовать ограничение по размеру файла с событиями** – задает максимальный размер файла с историей на диске.  
Если количество принятых событий превысит ограничение, запись событий будет происходить циклически, затирая более старые события.
- **Использовать ограничение по количеству событий в истории** – задает максимальное количество событий, хранимых в истории.  
Если количество принятых событий превысит ограничение, запись событий будет происходить циклически, затирая более старые события.
- **Использовать ограничение по времени хранения событий** – задает максимальную глубину хранения событий в днях.
- **Периодичность проверки ограничения по времени** – периодичность в часах, с которой будет проверяться ограничение по времени хранения событий и удаляться события, не удовлетворяющие данному ограничению.  
Во время проверки и удаления событий пользовательский интерфейс будет заблокирован и выдано следующее сообщение.



#### **ВНИМАНИЕ!!!**

**Ограничения по размеру файла с историей и по количеству событий не могут устанавливаться совместно с ограничением по времени хранения.**

### Резервирование протокола событий

Для настройки резервирования протокола событий необходимо установить флаг **Использовать резервное хранилище протокола событий**, при этом станет доступной группа настроек резервного хранилища протокола событий.

Настройки позволяют настраивать/просматривать значения следующих параметров:

- **Абсолютный или относительный путь к файлу с историей событий** - путь можно задать вручную, заполнив поле, или выбрать с помощью стандартного диалогового окна выбора файлов, нажав на кнопку **Выбор...** справа от текстового поля.
- **Абсолютный путь** – полный путь к файлу (текстовое поле **только для чтения**).

Кроме того, имеется возможность синхронизации основного и резервного хранилищ. Для этого необходимо воспользоваться опцией **Синхронизировать основное и резервное хранилище** и нажать кнопку **Выполнить**

**ВНИМАНИЕ!**

**Механизм резервирования протокола событий имеет ряд особенностей:**

- Основное и резервное хранилище имеют общие ограничения на ведение истории событий
- Пути к файлам основного и резервного хранилищ должны отличаться
- В случае невозможности сохранения основного протокола событий в процессе работы проекта, сохранение резервного также не возможно
- В случае невозможности сохранения резервного протокола событий, основное продолжит работу.

### 4.1.1.2 Словарь групп событий

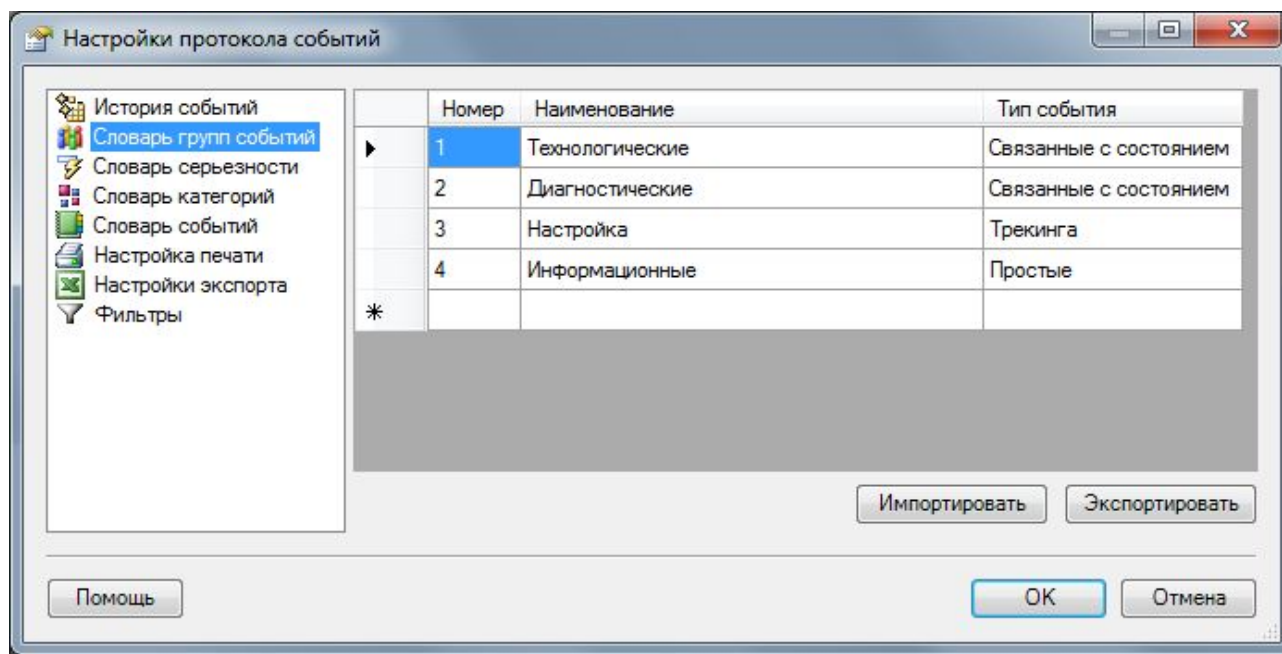
**Словарь групп событий** предназначен для описания групп, события которых соответствуют возможным состояниям автоматизируемого процесса.

Событий группируются в соответствии с типом события (согласно спецификации OPC Alarms and Events):

- **Связанные с состоянием**
- **Трекинга**
- **Простые**

### Как настроить группы событий

Для настройки групп событий выберите пункт **Словарь групп событий** в левой части окна **Настройки протокола событий** и задайте необходимые значения.



По умолчанию в **Словаре групп событий** установлены следующие группы:

- **Технологические**
- **Диагностические**
- **Настройка**
- **Информационные**

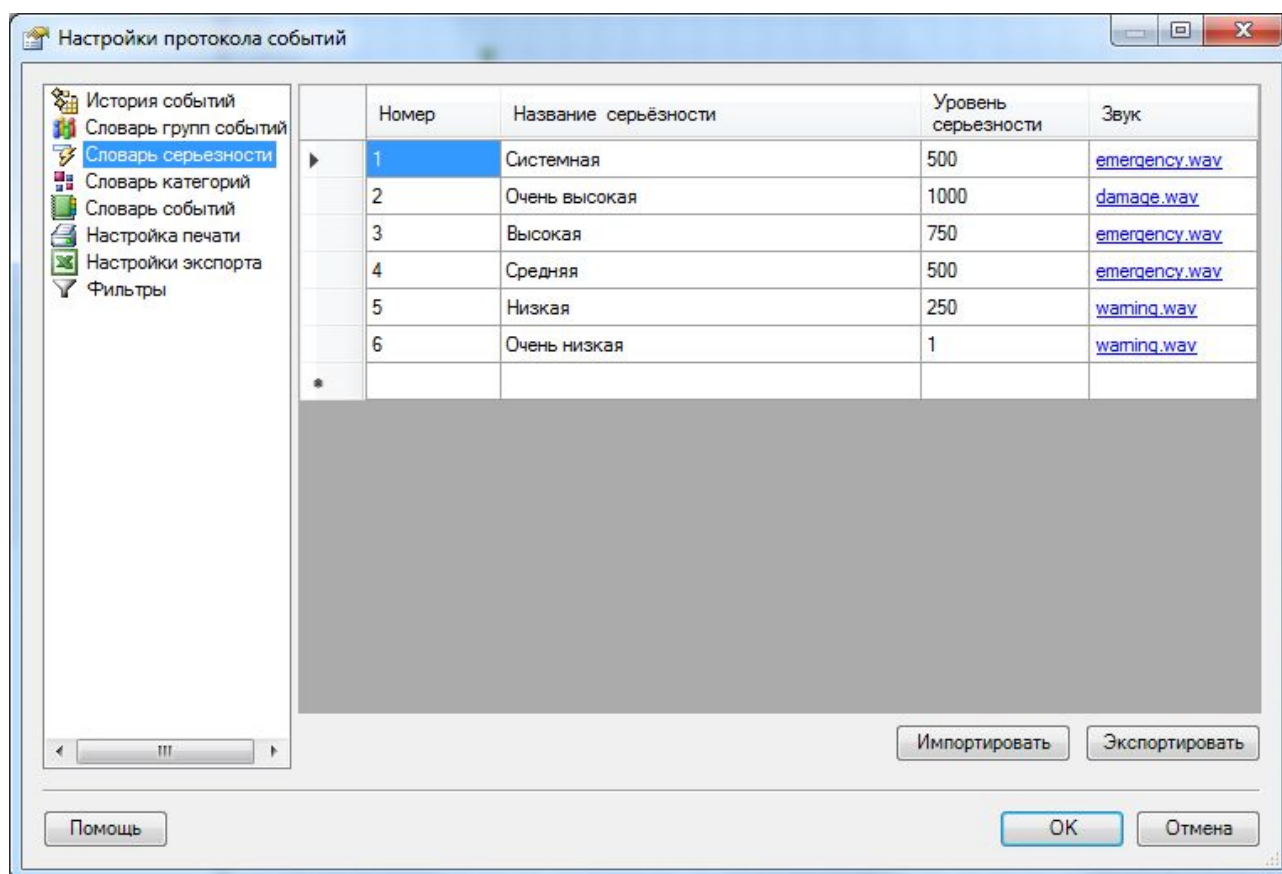
#### 4.1.1.3 Словарь серьезности

**Словарь серьезности** предназначен для задания уровня серьезности событий.

Уровень серьезности может использоваться для фильтрации событий, а также для сигнализаций системы.

#### Как настроить уровни серьезности событий

Для настройки уровней серьезности выберите пункт **Словарь серьезности** в левой части окна **Настройки протокола событий** и задайте необходимые значения в правой части.



#### Звуковое сопровождение

Каждой серьезности можно поставить в соответствие определенный звуковой файл.

Имя файла задается в столбце **Звук**.

Если звуковой файл не задан, то в столбце **Звук** отображается ссылка с текстом **<нет>**.

Для задания звука необходимо нажать на ссылку в столбце **Звук** и в открывшемся окне выбрать файл.

Для удаления звука из колонки можно нажать на ссылку с именем файла и в появившемся диалоговом окне нажать кнопку **Отмена** или выделить ячейку с именем файла и нажать на клавиатуре клавишу **Delete**.



### **ВНИМАНИЕ!!!**

Нельзя удалить или задать другое имя для серьезности Системная (под номером один в словаре).

Для серьезности Системная можно только изменить значение Уровень серьезности.

### **Добавление новой записи в словарь серьезности**

Для добавления новой записи следует:

- 1 Поставит курсор на поле со звездочкой
- 2 Ввести **название серьезности**
- 3 Задать **уровень серьезности** (от 1 до 1000)
- 4 Выбрать или задать имя звукового файла (если это необходимо).

### **Сохранение и загрузка словаря серьезности**

Для того чтобы **Словарь серьезности** можно было использовать в другом проекте или в другом **Протоколе событий**, либо просто создать резервную копию словаря, сохраните словарь на диске. Для этого:

- 1 Нажмите кнопку **Экспортировать**
- 2 В появившемся окне **Сохранить** укажите имя файла

### **ВНИМАНИЕ!!!**

**Не обязательно сохранять словарь серьезности отдельно.**

**В любом случае он хранится в проекте вместе с Протоколом событий до момента удаления элемента проекта из проекта.**

Для загрузки сохраненного словаря серьезности следует нажать кнопку **Импортировать**.

#### 4.1.1.4 Словарь категорий

**Словарь категорий** определяет основные типы (**категории**) событий.

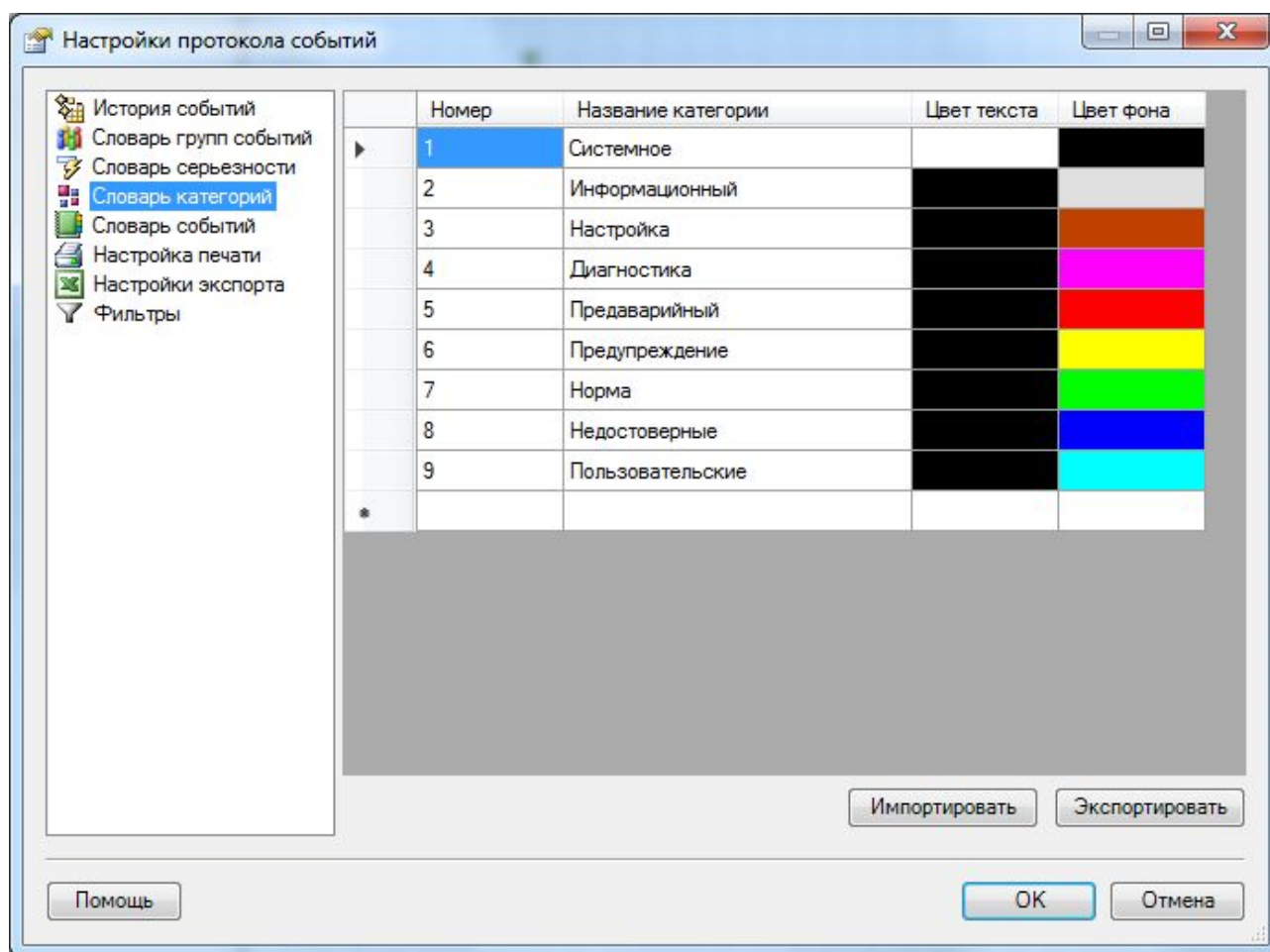
**Категория** определяет с какими "происшествиями" в контролируемом процессе связаны события. Например, это могут быть события, связанные с состояниями объекта контроля и управления; события, связанные с изменениями внутрисистемных переменных (tracking) или другие события, сообщения о которых являются просто информационными.

Словарь категорий определяет для каждого события:

- **название категории**
- **цвет текста и цвет фона сообщения** о данном событии.

### **Как задать категорию события**

Выберите пункт **Словарь категорий** в левой части окна **Настройки протокола событий** для задания категорий



### Добавление новой записи в словарь категорий

Для добавления новой записи следует:

- 1 Поставить курсор на поле со звездочкой
- 2 Ввести **название категории**
- 3 Задать **цвет текста** и **цвет фона**

#### Рекомендуемые цвета категорий событий:

- **Норма** – зеленый
- **Предупредительные** – желтый
- **Предаварийные** – красный
- **Диагностика** – фиолетовый
- **Настройка** – коричневый
- **Информационные** – белый
- **Пользовательские** – голубой

#### **ВНИМАНИЕ!!!**

Нельзя удалить или задать другое имя для категории «Системная» (под номером один в словаре), можно только изменить значения Цвет текста и Цвет фона.

### Сохранение и загрузка словаря категорий.

Для того чтобы **Словарь категорий** можно было использовать в другом проекте или в другом Протоколе событий, либо просто создать резервную копию словаря, сохраните словарь на диске. Для этого следует:

- 1 Нажать кнопку **Экспортировать**
- 2 В появившемся окне **Сохранить** указать имя файла

### ВНИМАНИЕ!!!

Не обязательно сохранять словарь серьезности отдельно.

В любом случае он хранится в проекте вместе с Протоколом событий до момента удаления элемента проекта из проекта.

Для загрузки словаря категорий следует нажать кнопку **Импортировать**.

#### 4.1.1.5 Словарь событий

**Словарь событий** предназначен для описания возможных **событий** в управляемом или наблюдаемом процессе.

Данные словаря используются при формировании протокола событий.  
"Пересылка" сообщений в протокол событий осуществляется из скрипта.

### ВНИМАНИЕ!!!

Для регистрации (записи) события из словаря событий в протокол событий необходимо, чтобы был создан или загружен ранее созданный хотя бы один словарь событий.

### Характеристики события

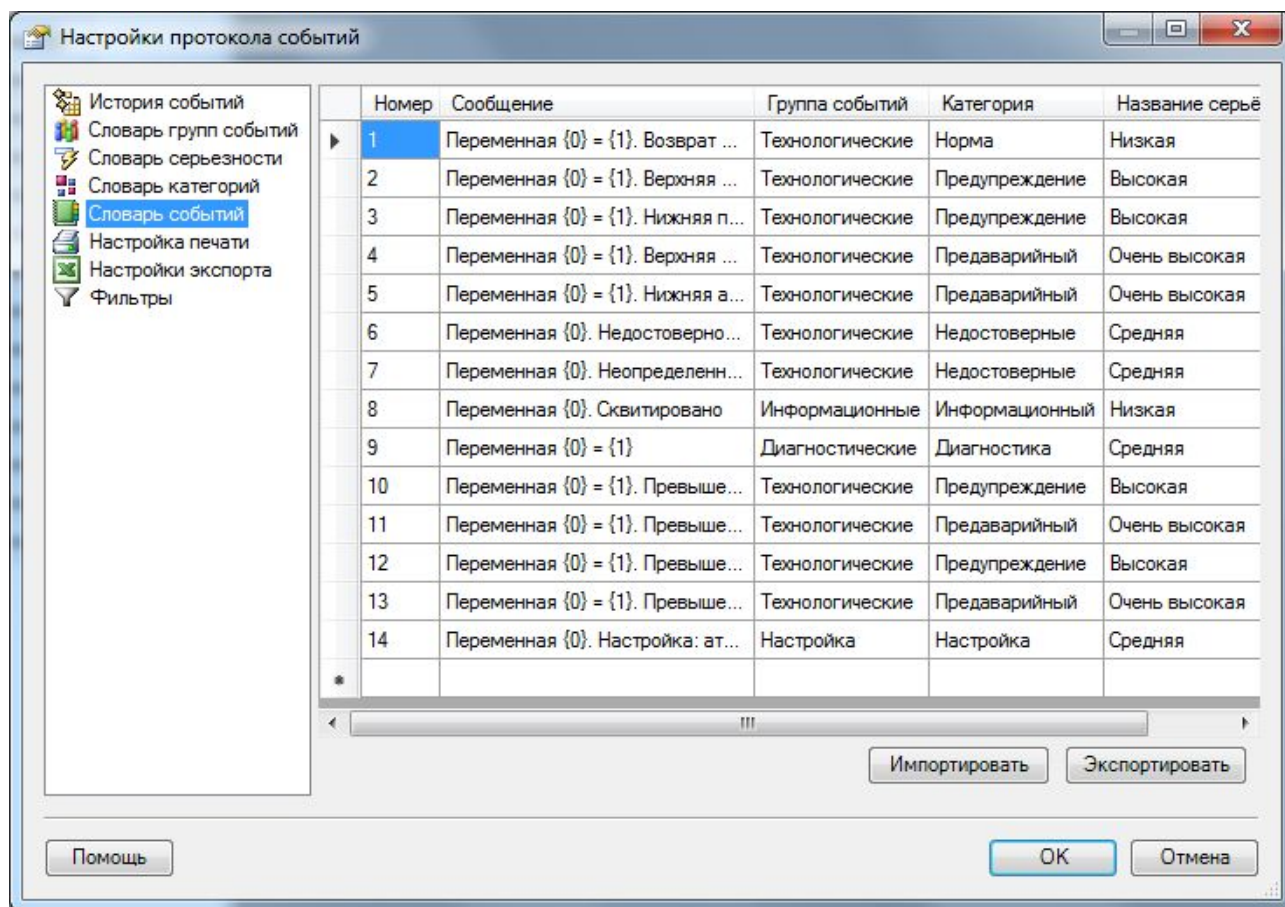
Каждое событие описывается следующими характеристиками

№	Имя	Описание
1	Номер в словаре	Номер в словаре событий. Данный номер указывается в программе Пользователя для регистрации в протоколе событий сообщения о событии.
2	Сообщение	Текст сообщения о свершившемся событии. Может содержать параметры. Параметры задаются в соответствие со стандартом задания параметров строк .NET Framework.
3	Группа событий	Выбирается из списка групп событий в Словаре групп событий
4	Категория	По категории события определяется его "раскраска" в примитиве <b>Протокол события</b> и окне <b>События элемента проекта</b> . Выбирается в словаре категорий.
5	Название серьезности	Название серьезности события. Выбирается из списка серьезностей в словаре серьезности.

### Как задать описание события

Для настройки данного словаря выберите пункт **Словарь событий** в левой части окна **Настройки протокола событий** и задайте описания событий.





### Добавление новой записи в словарь событий

Для добавления новой записи следует:

- 1 Поставить курсор на поле со звездочкой
- 2 Ввести текст сообщения
- 3 Выбрать из списка тип (категорию) события
- 4 Выбрать из списка серьезность

### ВНИМАНИЕ!!!

Номер записи в словаре генерируется автоматически. Автоматически контролируется его уникальность.

Изменять номер записи нельзя!

Для того чтобы новый тип события появился в списке выбора типов событий нужно добавить этот тип в словарь категорий событий.

Для того чтобы новое значение серьезности появилось в списке выбора серьезности нужно добавить эту серьезность в словарь серьезности.

### Сохранение и загрузка словаря событий.

Для того чтобы **Словарь событий** можно было использовать в другом проекте или в другом **Протоколе событий**, либо просто создать резервную копию словаря, сохраните словарь на диске:

- Нажмите кнопку **Экспортировать**
- В появившемся окне **Сохранить** укажите имя файла.

### ВНИМАНИЕ!!!

Не обязательно сохранять словарь событий отдельно. В любом случае он хранится в проекте вместе с Протоколом событий до момента удаления элемента проекта из проекта.

Для загрузки словаря событий, следует нажать кнопку **Импортировать**.

### 4.1.1.6 Системные события

**Системные события** – это специальные события, информирующие об изменении внутреннего состояния **DataRate**.

Список системных событий может изменяться в зависимости от версии **DataRate**.

#### **ВНИМАНИЕ!!!**

**Системные события нельзя добавлять, удалять и редактировать.**

Для **DataRate** определен следующий список системных событий:

- **Вход пользователя [имя пользователя]** – происходит в момент входа пользователя в Среду Исполнения
- **Выход пользователя [имя пользователя]** – происходит в момент выхода пользователя из Среды Исполнения
- **Удаленный вход пользователя [имя пользователя]** – происходит в момент входа пользователя в Среду Исполнения Клиент
- **Удаленный выход пользователя [имя пользователя]** – происходит в момент выхода пользователя из Среды Исполнения Клиент
- **Открытие проекта [название проекта]** – происходит в момент открытия проекта
- **Закрытие проекта [название проекта]** – происходит в момент закрытия проекта.

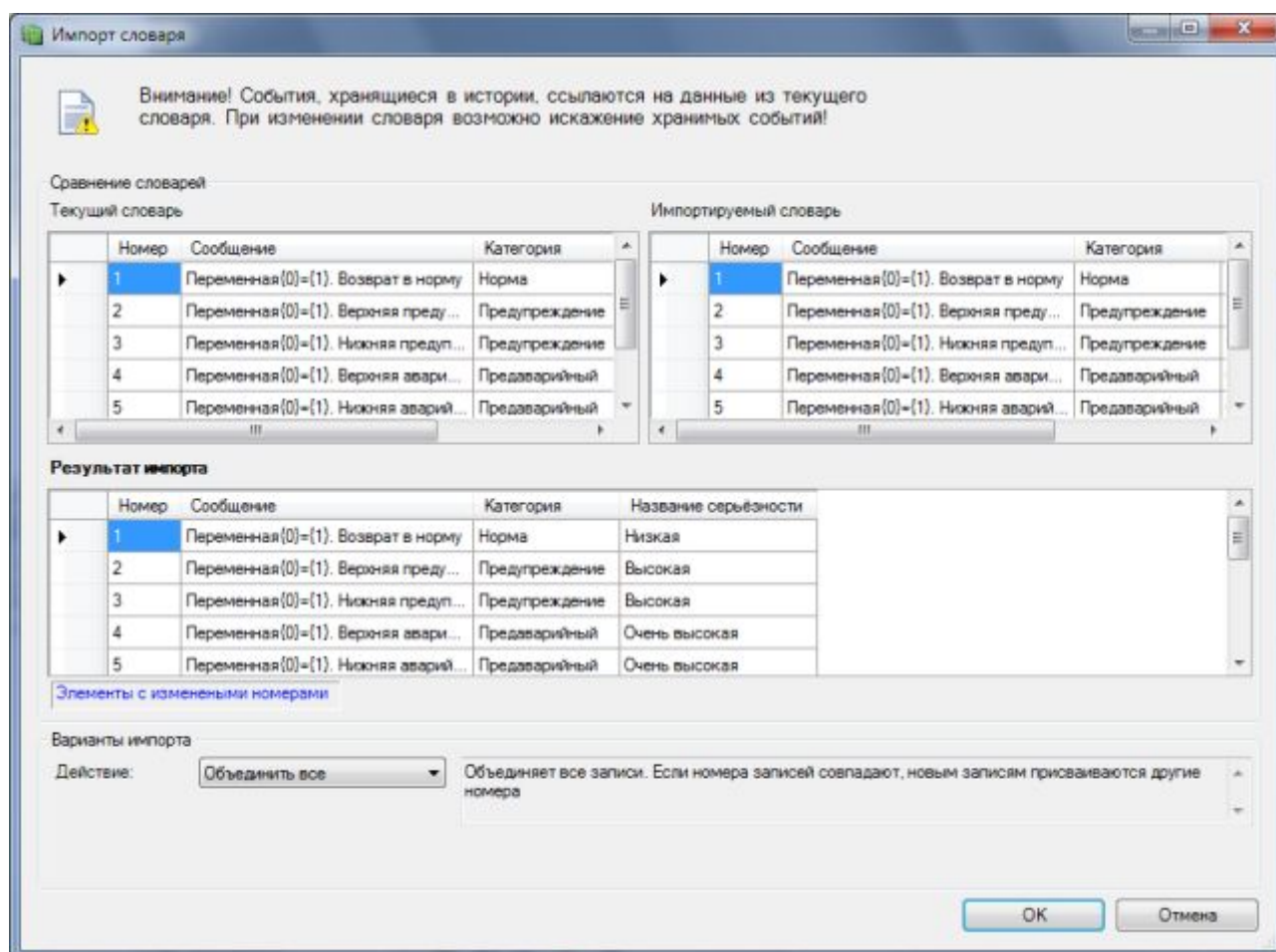
### 4.1.1.7 Импорт словаря

#### **ВНИМАНИЕ!!!**

При импорте словаря следует помнить, что если для данного проекта уже существует файл с историей событий, то изменение или удаления записей из словарей может привести к нарушению целостности данных, хранящихся в истории событий.

Для импорта словаря следует:

- 1 Нажать на кнопку **Импортировать** в окне **Настройки протокола событий**
- 2 В открывшемся окне загрузки файлов выбрать импортируемый словарь (**Словарь событий.edic**)
- 3 В появившемся после выбора словаря окне **Импорт словаря** задать параметры загрузки и формирования результирующего словаря



### Сравнение словарей

Группа **Сравнение словарей** содержит три таблицы:

- **Текущий словарь** – содержит записи из словаря, находящегося в Протоколе событий;
- **Импортируемый словарь** – содержит записи, находящиеся в импортируемом словаре
- **Результат импорта** – показывает, как будет выглядеть словарь после импорта, в зависимости от выбранного действия.

Синим цветом подсвечиваются события из импортируемого словаря, номера которых изменены при импорте.

### Варианты импорта

Группа **Варианты импорта** содержит выпадающий список с вариантами импорта. Существует **четыре варианта импорта**:

- **Объединить все** – объединяет записи текущего словаря и импортируемого. Если в импортируемом словаре есть записи с такими же номерами как в текущем словаре, номера этих записей заменяются другими.
- **Объединить без перезаписи** – объединяет записи текущего словаря и импортируемого. Если в импортируемом словаре есть записи с такими же номерами как в текущем словаре, то эти записи не добавляются в результирующий словарь.
- **Объединить с перезаписью** – объединяет записи текущего словаря и импортируемого. Если в импортируемом словаре есть записи с такими же

номерами как в текущем словаре, то записи из текущего словаря заменяются импортируемыми.

- **Заменить все** – заменяет текущий словарь импортируемым.

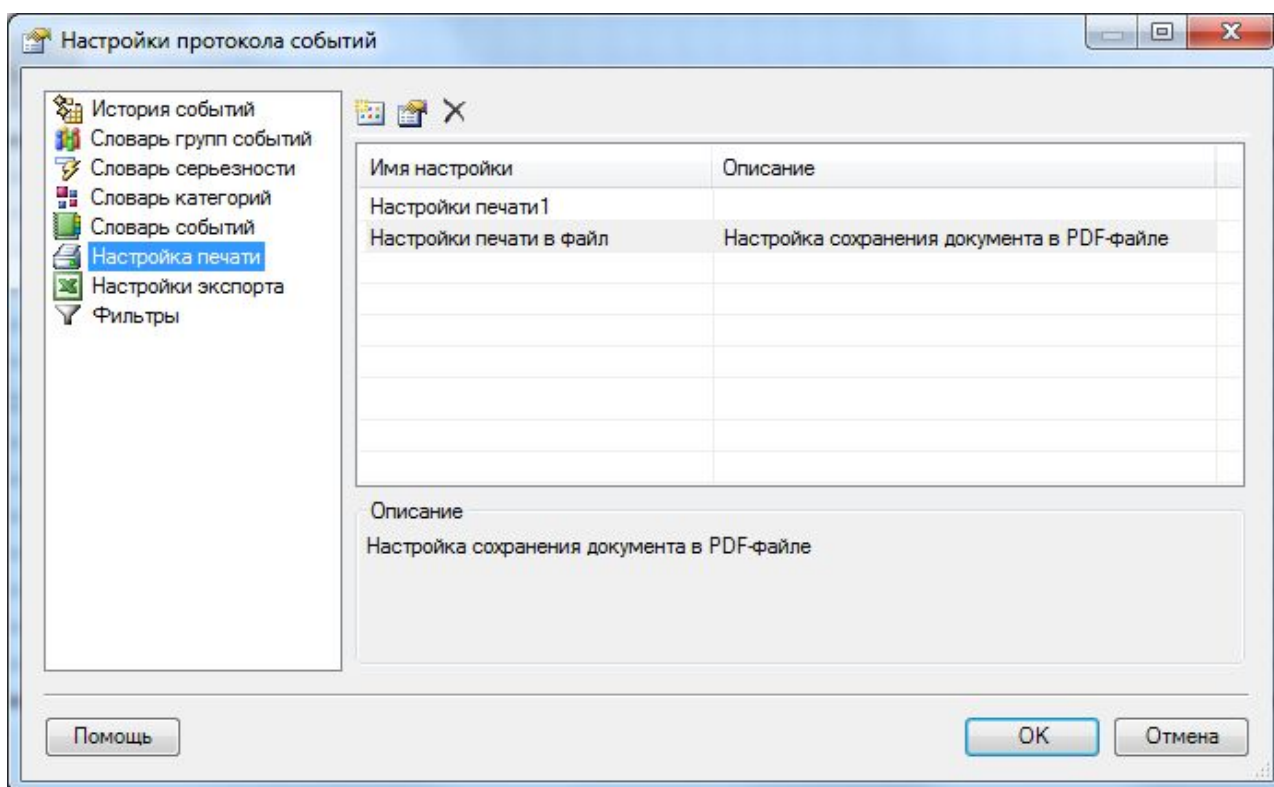
Текстовое поле справа от выпадающего списка содержит подробное описание выбранного действия.

По умолчанию установлено действие **Объединить все**.

### 4.1.1.8 Настройки печати протокола событий

**Настройки печати** – это предустановленные параметры для печати протокола событий.

Настроек может быть несколько, и все они отображаются в списке настроек.



Для редактирования настроек печати используются кнопки на панели инструментов в окне **Настройки протокола событий**:

- вызывает окно настройки параметров печати и добавляет установленные параметры в список настроек
- открывает окно с выбранными настройками печати
- удаляет выделенные настройки печати.

### 4.1.1.9 Окно Настройка печати

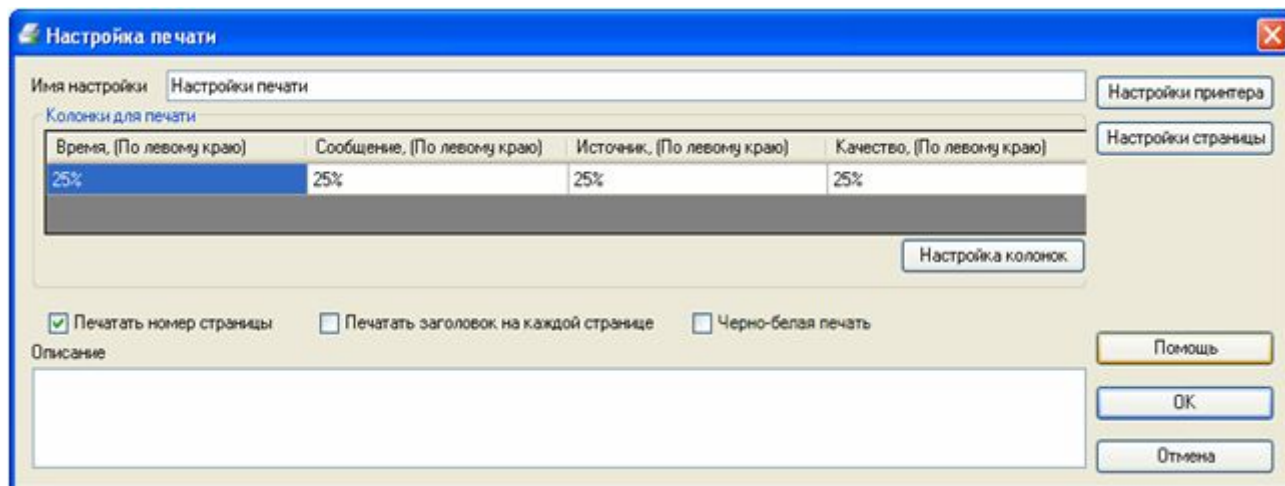
Окно **Настройка печати** предназначено для редактирования параметров печати протокола событий.

Данные настройки используются в скриптах при выводе на печать сообщений о событиях.

Окно **Настройки печати** включает:

- **Имя настройки** – уникальное имя настройки. Отображается в окне **Настройки протокола событий**

- **Описание** – область описания настройки печати
- **Печатать номер страницы** – задает вывод номера страницы при печати
- **Печатать заголовок на каждой странице** – задает печать заголовка на каждой странице
- **Черно-белая печать** – задание печати с использованием оттенков серого цвета
- **Настройки принтера** – кнопка вызова окна задания параметров принтера, используемого при печати протокола событий. Подробное описание настраиваемых параметров приведено в разделе **Редактор скриптов/Окно Печать**.



- **Параметры страницы** – кнопка вызова окна задания параметров страницы, используемой при печати протокола событий. Подробное описание настраиваемых параметров приведено в разделе **Редактор скриптов/Окно Параметры страницы**
- **Колонки для печати** – панель отображения выбранных колонок для печати и их настроек.



В заголовке каждой колонки отображается имя колонки и выравнивание.

В ячейке таблицы отображается ширина колонки в процентах от общей ширины таблицы. При изменении ширины колонки автоматически пересчитывается ширина всех колонок.

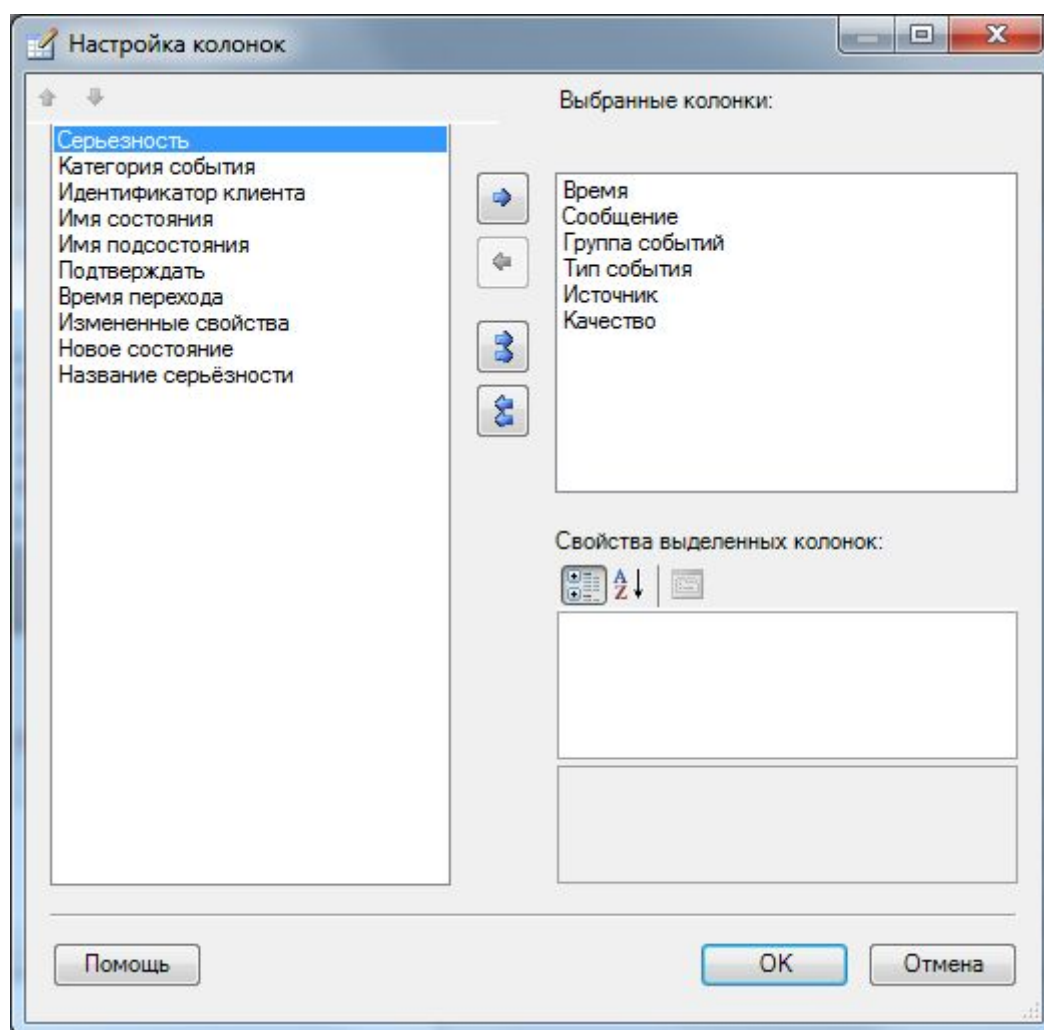
Ширину колонки можно изменить путем ввода с клавиатуры значения в ячейку.

### Настройка колонок для печати

Для выбора колонок и задания выравнивания следует:

- 1 Щелкнуть на кнопке **Настройка колонок**
- 2 В появившемся окне **Настройка колонок для печати** выбрать колонки и задать выравнивание текста при печати





## Выбор колонок

В левой части окна располагается список **Доступные колонки**, из которого можно выбирать колонки для вывода на печать. В правой части окна находится список **Выбранные колонки**, содержащий колонки для печати.

Для добавления колонок нужно выделить требуемые колонки в списке **Доступные колонки** и нажать на кнопку со стрелкой вправо.

Для удаления колонок нужно выделить требуемые колонки в списке **Выбранные колонки** и нажать кнопку со стрелкой влево.

**Порядок колонок в протоколе событий** можно настроить с помощью стрелок (вверх/вниз) на панели инструментов над списком выбранных колонок.

## Свойства

Панель **Свойства выделенных колонок** позволяет редактировать выравнивание текста колонок, выделенных в списке **Выбранные колонки**:

- По центру
- По левому краю
- По правому краю.

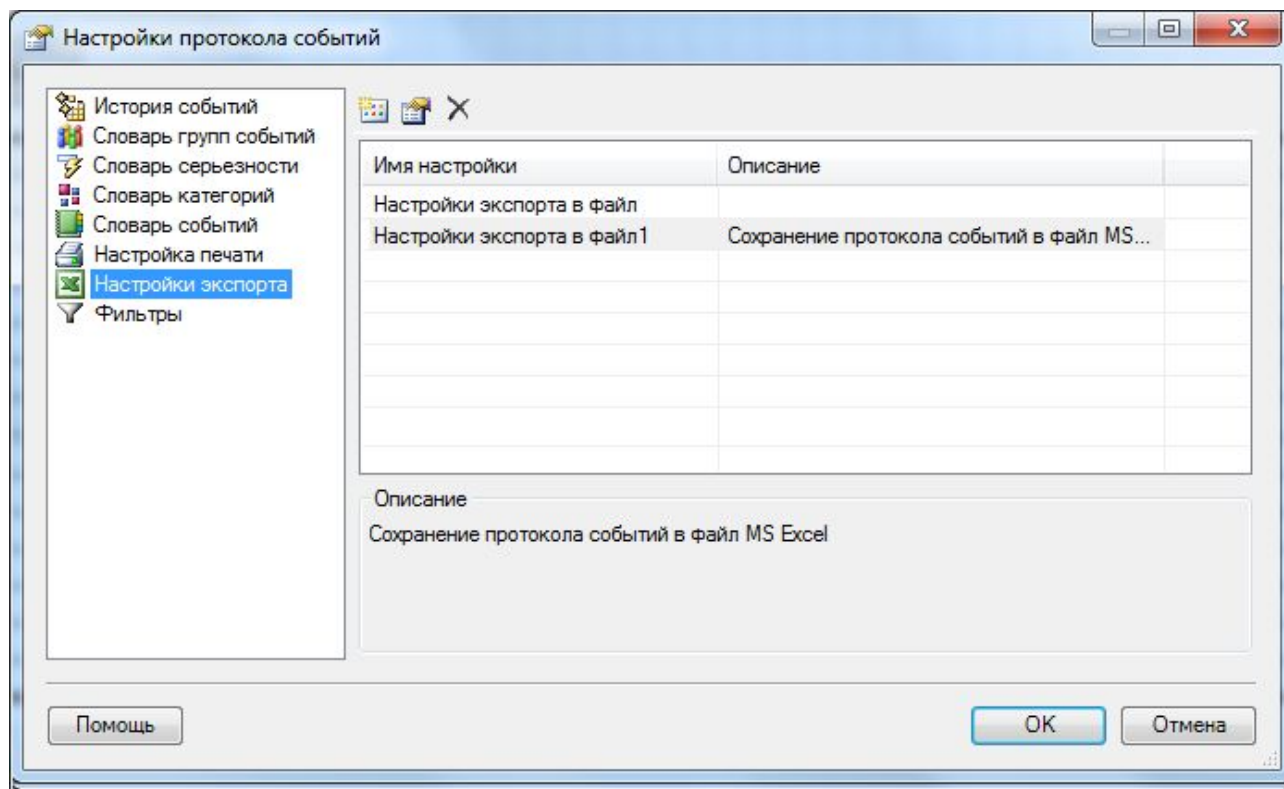
Для подтверждения сделанных изменений нужно нажать кнопку **ОК**, для отмены – кнопку **Отмена**.

Для вызова справки по выбранной настройке нажмите кнопку **Помощь**.




Закрытие окна при помощи кнопки в заголовке отменяет сделанные изменения.

## 4.1.1.10 Настройки экспорта

**Настройки экспорта** – это предустановленные параметры для сохранения сообщений Протокола событий в файл форматов: CSV, MS Excel, XML (для чтения MS Excel). Настроек может быть несколько и все они отображаются в списке настроек



Для редактирования настроек используются кнопки на панели инструментов в окне **Настройки протокола событий**:

-  – вызывает окно **Настройка экспорта** и добавляет установленные параметры в список настроек
-  – открывает окно с выбранными настройками экспорта
-  – удаляет выделенные настройки.

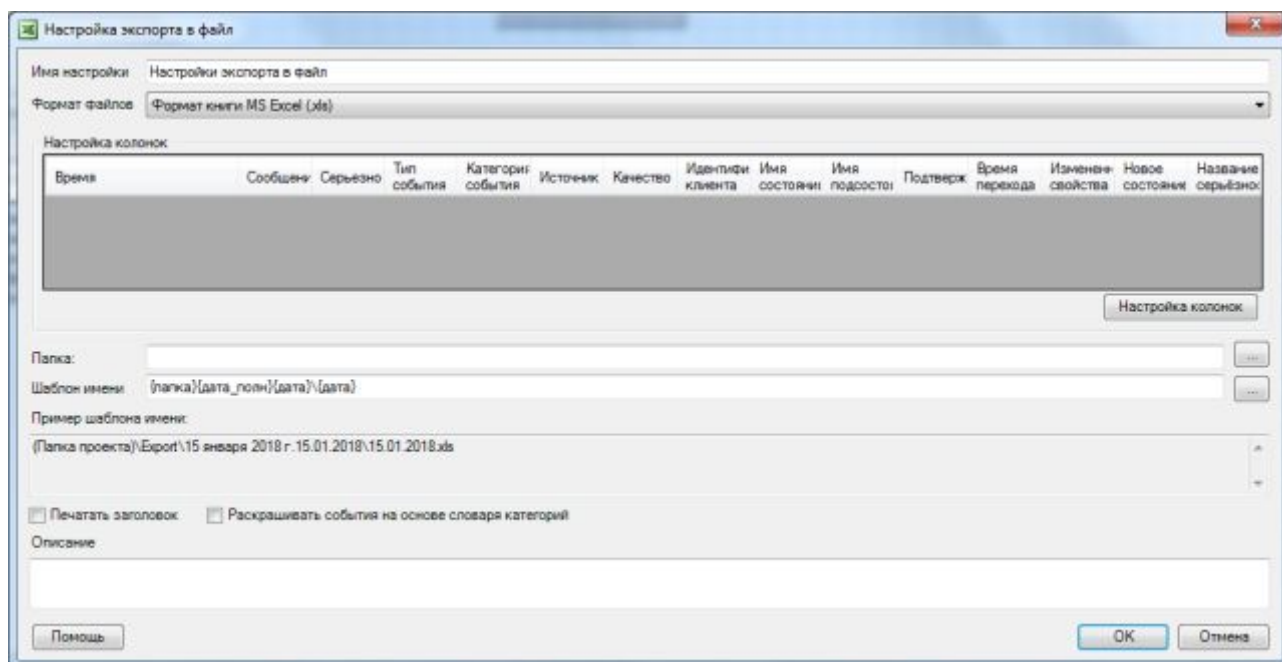
**Окно настройки экспорта и окно настройки колонок**

Окно **Настройка экспорта в файл** предназначено для редактирования параметров экспорта протокола событий.

Окно **Настройка экспорта** включает:

- **Имя настройки** – уникальное имя настройки. Отображается в окне Настройки протокола событий.
- **Формат файлов** – формат выходного файла
- **Описание** – область описания настройки
- **Колонки протокола событий** – панель отображения выбранных колонок для печати и их настроек (для задания колонок нажать на кнопку Настройка колонок).
- **Папка, Шаблон имени** – путь и наименование файла для экспорта протокола событий
- **Печать заголовка** – формировать строку заголовка колонок в файле экспорта

- **Раскрашивать события на основе словаря категорий** – признак для отображения цвета символов и фона сообщений протокола событий соответственно заданным в Словаре категорий
- **Описание** – текст описания данной настройки экспорта
- **Формат файла** – тип файла, который будет сформирован:
  - CSV значение-разделитель (.csv)
  - Книга MS Excel (.xls)
  - XML для MS Excel (.xml)



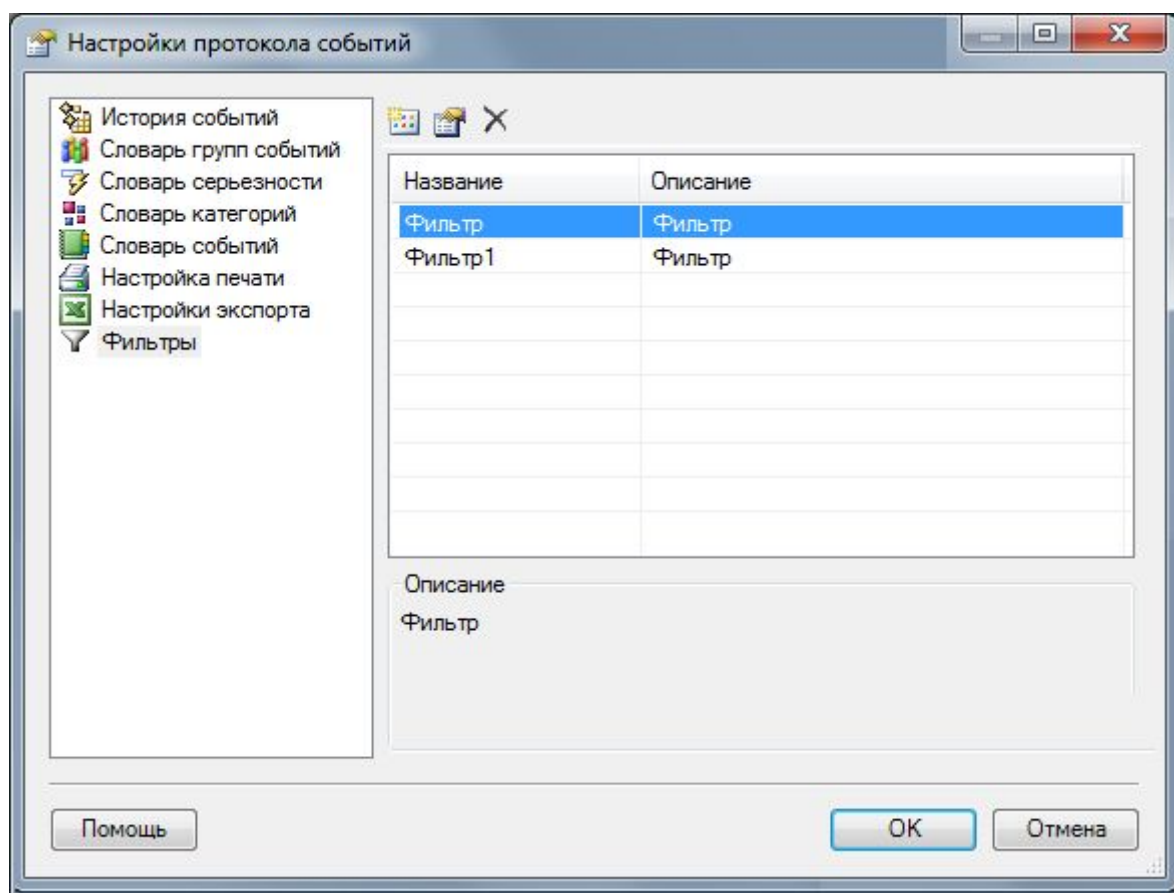
Окно **Настройка колонок** (аналогично окну настройки колонок для печати) предназначено для выбора и настройки колонок протокола событий, определенных для экспорта.

#### 4.1.1.11 Фильтры протокола событий




**Фильтры** – это предустановленная настройка условий фильтрации для протокола событий.

Фильтров может быть несколько, и все они отображаются в списке фильтров.





Для редактирования настроек используются кнопки на панели инструментов вкладки **Фильтры** в окне **Настройки протокола событий**:

-  – открывает окно создания и редактирования нового фильтра, добавляет созданный фильтр в список
-  – открывает окно редактирования фильтра
-  – удаляет выделенный фильтр.

### Окно редактирования фильтра

Окно **Редактирование фильтра** предназначено для изменения имени, описания и условий фильтрации.

**Название фильтра** – это уникальное имя фильтра для протокола событий элемента проекта.

**Описание фильтра** – позволяет задать понятийное описание условий фильтрации.

Редактирование фильтра

Название фильтра:

Описание фильтра:

Состав фильтра: ☐ Объединение ☒ Пересечение

Атрибут события: **Серьёзность** (выбрано)

Условие фильтрации: Серьёзность в диапазоне от 1 до 1000

Доступные атрибуты события:

- Сообщение
- Серьёзность
- Название серьёзности
- Тип события
- Группа событий
- Категория события
- Источник
- Качество
- Идентификатор клиента
- Имя состояния
- Имя подсостояния
- Подтверждать
- Время перехода
- Изменённые свойства
- Новое состояние

OK Отмена

Фильтр может содержать одно или несколько условий фильтрации.

Каждое условие фильтрации выполняет сравнение заданного атрибута события с соответствующим атрибутом возникающих событий.

**Атрибут события** определяет содержание условия фильтрации:

- |                              |  |
|------------------------------|--|
| <b>Сообщение</b>             | – текст, который описывает событие;  |
| <b>Серьёзность</b>           | – диапазон значений серьёзности события;   |
| <b>Название серьёзности</b>  | – текст, который определяет серьёзность события в строковом виде (например, системная, высокая, очень низкая);                         |
| <b>Тип события</b>           | – набор типов событий (простые события, трекинга, связанные с состоянием);   |
| <b>Категория события</b>     | – набор категорий событий из словаря категорий;  |
| <b>Источник</b>              | – текст, который определяет объект сгенерировавший уведомление о событии (например, это может быть имя тега);                          |
| <b>Качество</b>              | – качество основных элементов данных, на которых основано данное состояние (хорошее, плохое, неопределённое);                          |
| <b>Идентификатор клиента</b> | – идентификатор клиента, который подтвердил состояние;   |
| <b>Имя состояния</b>         | – имя соответствующего OPC состояния;  |
| <b>Имя подсостояния</b>      | – имя активного в настоящее время OPC подсостояния;  |
| <b>Подтверждать</b>          | – индикатор, отражающий необходимость подтверждения;   |
| <b>Время перехода</b>        | – интервал времени, задающий время перехода в состояние или подсостояние, которое связано с этим уведомлением;                         |
| <b>Изменённые свойства</b>   | – набор изменённых свойств (изменение состояния подтверждения, доступности, качества, серьёзности, подсостояния, сообщения, атрибута); |
| <b>Новое состояние</b>       | – набор новых состояний (доступно, активно, квитируется).  |





**Состав фильтра** – определяет объединение (логическая операция «ИЛИ») или пересечение (логическая операция «И») условий фильтрации при выборке в протокол событий.

При объединении условий фильтрации событие будет показано пользователю, если любое из условий выполняется. По умолчанию используется пересечение условий фильтрации, то есть событие будет показано пользователю, если все условия выполняются.

### ВНИМАНИЕ!!!

Поскольку фильтрация событий из протокола событий происходит в порядке появления условий фильтрации в фильтре, то порядок условий фильтрации может сказываться на времени фильтрации. Для сокращения времени фильтрации необходимо ставить первыми те условия фильтрации, которые, вероятно, отфильтруют наибольшее число событий из протокола.

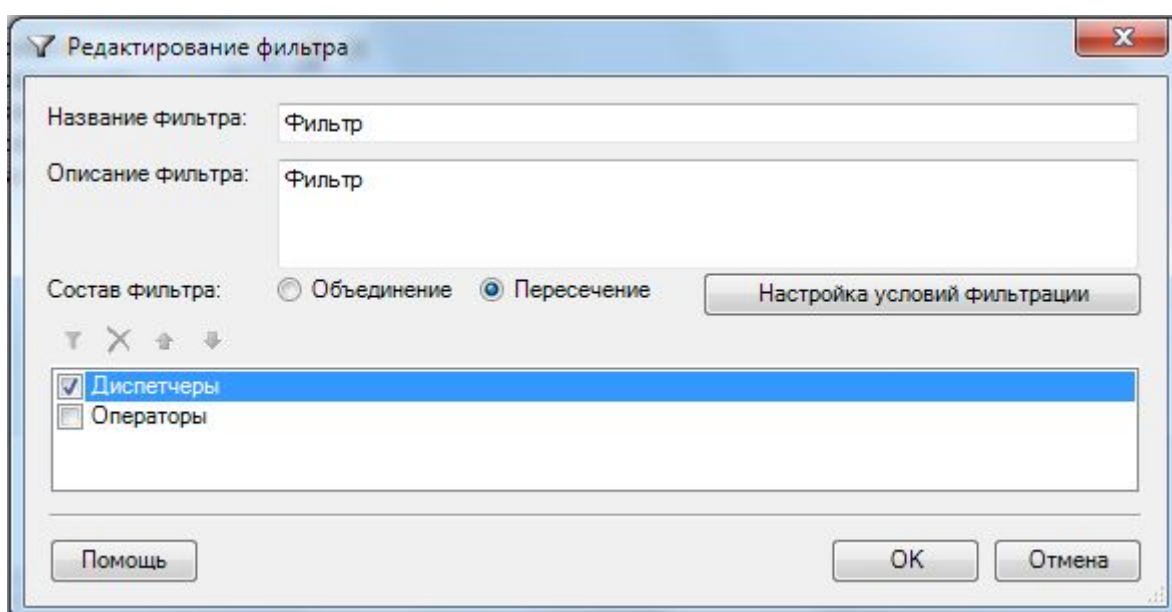
Для работы с коллекцией условий фильтрации используются кнопки панели инструментов **Состав фильтра**:

-  – добавление нового условия фильтрации;
-  – удаления выбранного условия фильтрации;
-  – перемещение выбранного условия фильтрации вверх;
-  – перемещение выбранного условия фильтрации вниз.

После добавления нового условия фильтрации нажатием кнопки **панели инструментов** необходимо выбрать атрибут события для условия фильтрации из выпадающего списка. После этого необходимо настроить условие фильтрации нажатием кнопки мыши в соответствующую ячейку в колонке **Условие фильтрации** и изменением соответствующих полей.

После редактирования условия фильтрации можно просмотреть настройку, выполнив переход (нажатием кнопки мыши) на ячейку в колонку **Атрибут события**.

Каждому фильтру **должен быть сопоставлен** набор ролей, для которых этот фильтр применим. Для этого необходимо нажать кнопку **Настройка ролей** и отметить допустимые роли галочкой. Для возврата к настройке условий фильтрации надо нажать кнопку **Настройка условий фильтрации**.



Для примитива **Протокол событий** и источника данных **Протокол событий** значения условий фильтрации могут быть параметризованы.

### **ВНИМАНИЕ!!!**

При изменении набора фильтров в элементе проекта или при изменении набора условий фильтрации в каком-либо фильтре настройки параметризации для примитива и источника данных **Протокол событий** будут сброшены, потребуется их перенастройка.

После нажатия на кнопку **ОК** созданный или изменённый фильтр сохранится в списке фильтров протокола событий.

Для отмены сделанных изменений необходимо нажать на кнопку **Отмена**.

Условия фильтрации, которые связаны с временным периодом, называются **интерактивными условиями фильтрации**, то есть изменяющимися в диалоге с пользователем. Фильтры, содержащие интерактивные условия фильтрации, также называются интерактивными.

Редактирование интерактивных фильтров (условий фильтрации) возможно в **режиме исполнения** в диалоговых окнах **Выбора фильтров**.

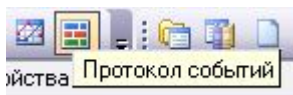
### 4.1.2 Графический примитив Протокол событий

Графический примитив **Протокол событий** предназначен для отображения протокола событий на мнемосхеме.

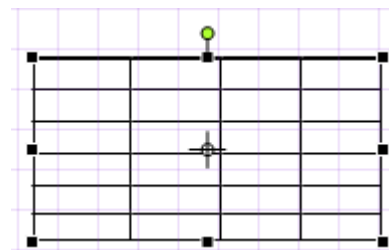
#### Создание

Для создания примитива **Протокол событий** следует:

1. Выбрать иконку создания протокола событий на панели инструментов или в меню



2. Щелкнуть левой клавишей мыши по мнемосхеме, где будет размещен протокол, и, удерживая левую клавишу мыши, задать начальные размеры таблицы для вывода протокола
3. После добавления примитива **Протокол событий** на мнемосхему произвести его настройку



#### 4.1.2.1 Основные настройки

Основные настройки примитива **Протокол событий** располагаются в разделах: **Основные**, **Положение** и **Таблица**.

<b>Основные</b>	
Имя объекта	Протокол событий1
Тип	Протокол событий
Видимость/мигание	Видим
Период обновления	1000
Обновлять асинхронно	Нет
Фильтры	(Коллекция)
<b>Положение</b>	
Смещение слева	50
Смещение сверху	40
Ширина	460
Высота	324
Угол поворота	0
Центр вращения X	0,5
Центр вращения Y	0,5
<b>Таблица</b>	
Ширина рамки	1
Цвет рамки	Черный
Ширина линий сетки	1
Цвет линий сетки	Черный
Цвет фона	Прозрачный
Высота строки	14
Ширина колонки	40
Шрифт	Arial; 10

– имя примитива **Протокол событий** на мнемосхеме

– период обновления протокола событий (**кратно 250 мс**)

– фильтры, выбранные для фильтрации протокола событий

– позиция примитива на мнемосхеме

– размеры таблицы

– настройки вращения примитива

– ширина рамки обводки примитива

– цвет рамки обводки

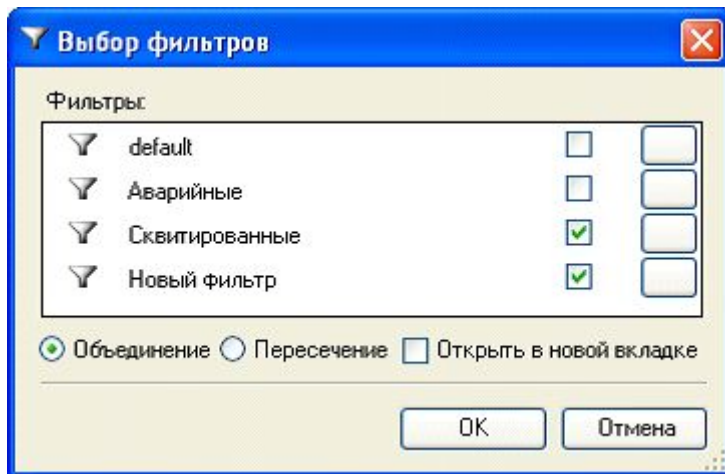
– шрифт текста сообщения о событии

Настройки примитива **Протокол событий** аналогичны настройкам графического примитива **Таблица**.

## Фильтрация сообщений

Для того чтобы указать категории сообщений, которые будут отображаться в примитиве **Протокол событий**, следует:

- 1 Выделить свойство **Фильтры** и щелкнуть левой клавишей мыши по кнопке в строке
- 2 В появившемся окне **Выбор фильтров** задать применяемые фильтры (поставить галочки напротив имен фильтров).



### ВНИМАНИЕ!!!

**Настройка выбранных фильтров доступна только в режиме редактирования.**

Если используется несколько фильтров, то при фильтрации событий следует задать их объединение или пересечение.

**При объединении фильтров** (логическая операция "ИЛИ") сообщение включается в протокол событий, если любой из фильтров работает.

**При пересечении фильтров** (логическая операция "И") сообщение включается в протокол событий, если все фильтры работают. По умолчанию используется пересечение фильтров.

В режиме исполнения доступно редактирование интерактивных условий фильтрации в выбранных фильтрах.

## 4.1.2.2 Колонки и строки

Настройка строк и колонок примитива **Протокол событий** производится в разделах **Протокол событий** и **Форматирование**.

<b>Протокол событий</b>	
Число строк	6
Вывод сверху вниз	Да
Цвет текста по умол.	Черный
Цвет фона по умолч	Белый
<b>Форматирование</b>	
Колонки	Количество: 5
Заголовки колонок	Количество: 0

– максимально число отображаемых сообщений  
– направление вывода сообщений. Сверху вниз – последнее полученное сообщение будет на самом верху.

– выбор колонок, отображаемых в примитиве  
– настройка заголовков колонок.

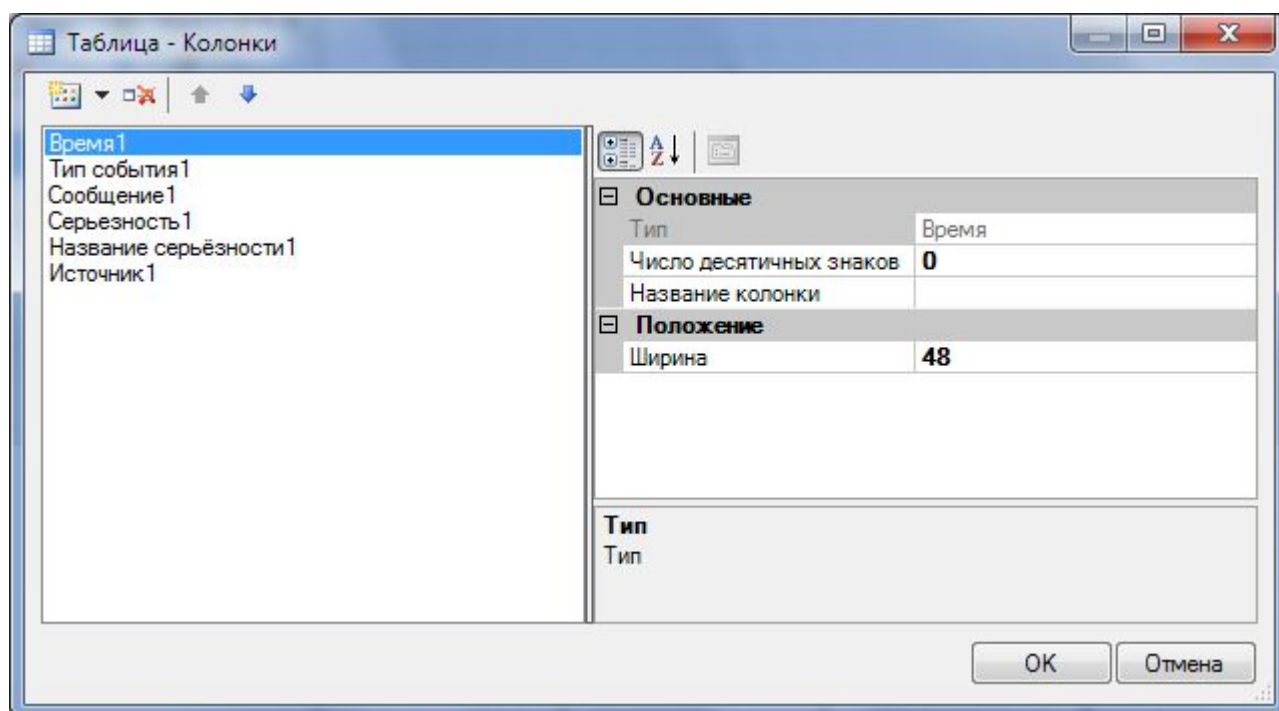
Подробное описание, как это сделать дано в разделе **Графические примитивы/Таблица**


## Настройка колонок

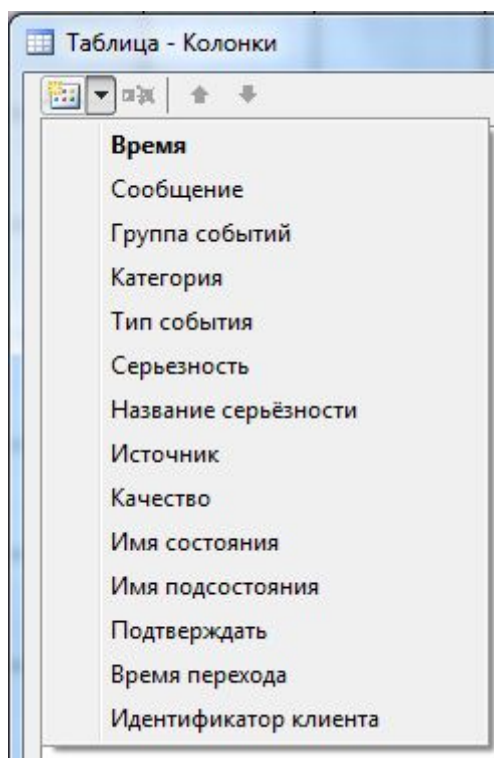
Чтобы войти в режим редактирования колонок следует:

- 1 Выделить свойство **Колонки** и нажать на кнопку в строке
- 2 В появившемся окне **Таблица-Колонки** выбрать необходимые колонки и настроить их размеры (свойство **Ширина**, значение в пикселах).





Для того чтобы добавить колонки для вывода расширенной информации, следует раскрыть выпадающий список рядом с кнопкой .



**Группа событий** – группа события (Технологические, Диагностические, Настройка, Информационные) из Словаря групп событий

**Категория** – категория сообщения.

**Качество** – качество основных элементов данных, на которых основано данное состояние.

**Имя состояния** – имя соответствующего OPC состояния.

**Имя подсостояния** – имя активного в настоящее время OPC подсостояния.

**Подтверждать** – индикатор, отражающий необходимость подтверждения.

**Время перехода** – время перехода в состояние или подсостояние, которое связано с этим уведомлением.

**Идентификатор клиента** – идентификатор OPC-клиента, который подтвердил состояние

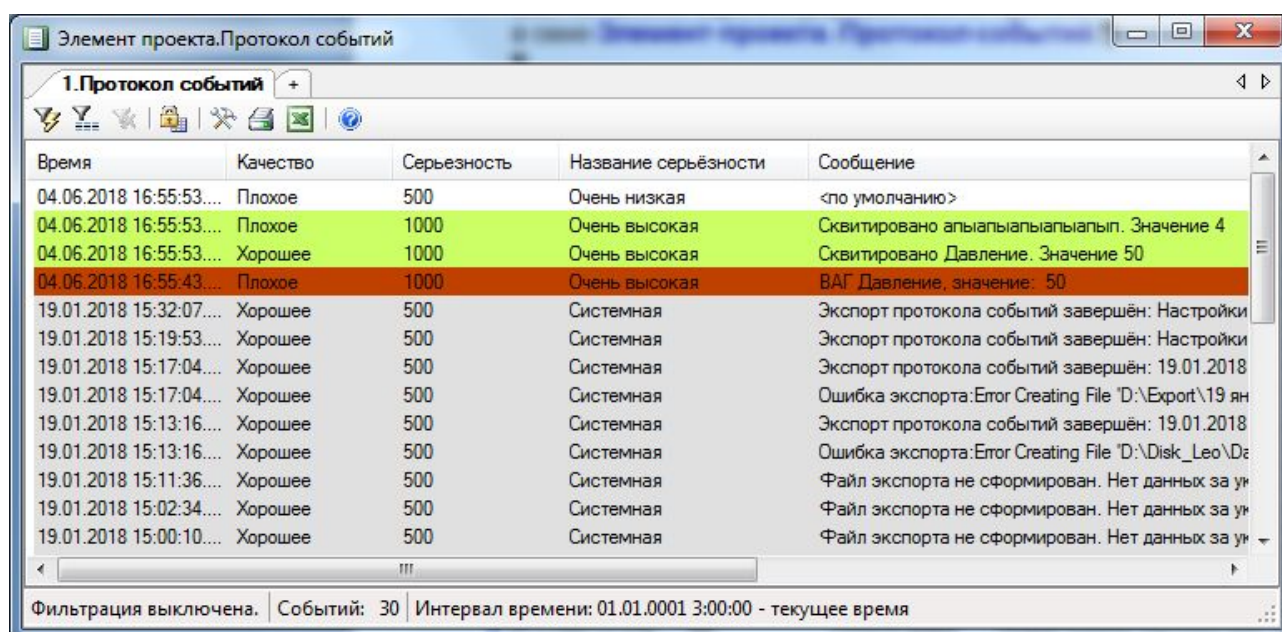
По умолчанию примитив **Протокол событий** создается с 6-ю колонками:

- **Время** - время возникновения и регистрации события в протоколе событий.
- **Тип события** - тип события (связанное с состоянием, связанное с трекингом, простое).
- **Сообщение** - текст, который описывает событие.

- **Серьезность** - серьезность события, безотлагательность реакции на событие (значение в диапазоне 1..1000).
- **Название серьезности** - текст, который задается в словаре серьезности в поле название серьезности.
- **Источник** - ссылка на объект, который сгенерировал уведомление о событии (например, это может быть имя тега).

### 4.1.3 События элемента проекта

Все события, зарегистрированные в протоколе событий элемента проекта, отображаются в окне **Элемент проекта. Протокол событий**.



Для работы с вкладками окна протокола событий используйте горячие клавиши:

**Ctrl+T** - создать новую вкладку,

**Ctrl+W** - закрыть активную вкладку.

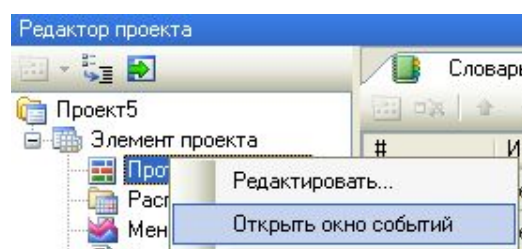
#### 4.1.3.1 Вызов окна просмотра событий элемента проекта

Окно **Элемент проекта. Протокол событий** можно вызвать двумя способами:

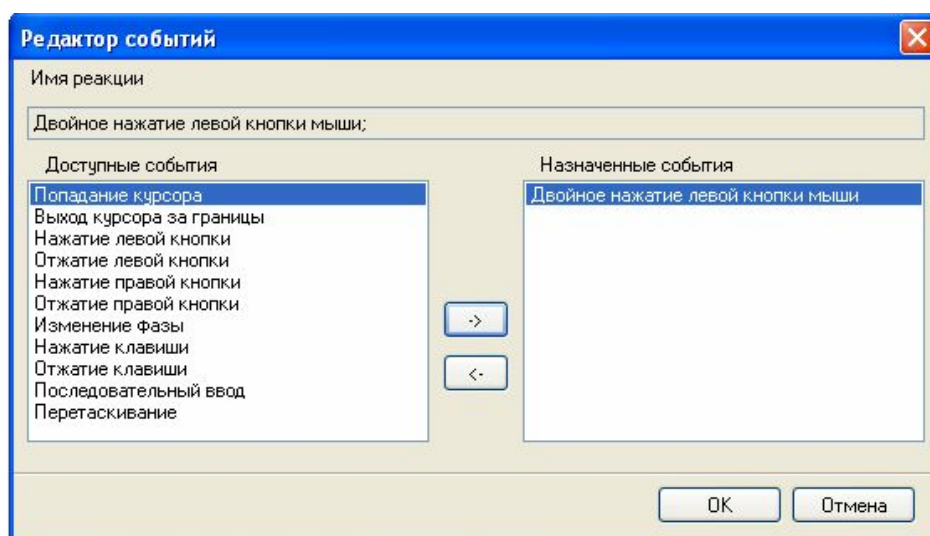
- **В режиме имитации:** в контекстном меню системного объекта **Протокол событий** выбрать команду **Открыть окно событий**
- **В режиме выполнения:** активизировать (например, щелчком мыши) графический примитив, для которого назначена реакция **Окно протокола событий**.

В этом случае в среде разработки следует:

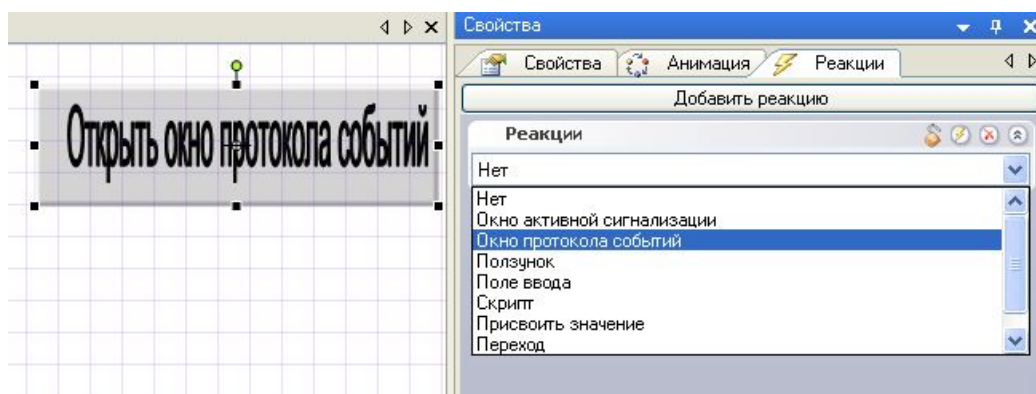
- 1 Создать графический примитив (например, прямоугольник)
- 2 Выбрать подходящее событие для его «активизации» – нажатие клавиши или кнопки мыши



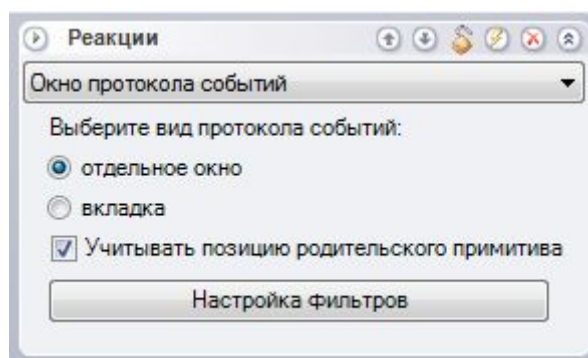




- 3 Назначить созданному примитиву реакцию **Окно протокола событий**



- 4 Задать отображение протокола событий (в отдельном окне или сделать вкладкой в главном окне)



После запуска проекта на выполнение при двойном щелчке мыши (для данного примера) по прямоугольнику появится окно **Элемент проекта. Протокол событий**

#### 4.1.3.2 Панель инструментов







Для управления отображением протокола событий используйте кнопки панели инструментов окна **Элемент проекта. Протокол событий**:



– настроить промежуток времени. Открывает диалоговое окно настройки промежутка времени.



– настроить фильтры. Открывает диалоговое окно выбора фильтров.

-  – сбросить все фильтры (доступно, когда активен хотя бы один фильтр). Выключает все включенные фильтры.
-  – «заморозить» прием событий. При нажатии прекращает прием событий от протокола событий, при этом состояние кнопки становится «нажато». При повторном нажатии или при включении/выключении фильтров восстанавливает прием событий, кнопка возвращается в состояние «не нажато».
-  – настроить колонки таблицы. Открывает окно настройки колонок таблицы.
-  – печать событий. Открывает окно настройки печати.
-  – открывает окно экспорта протокола событий в файл
-  – помощь. Открывает справку.

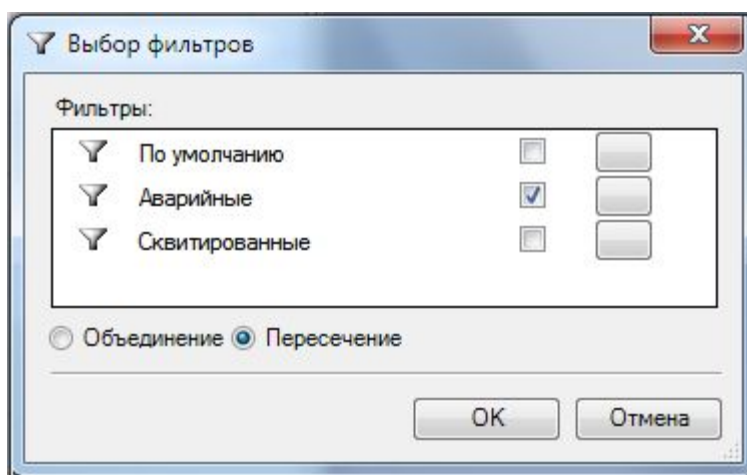
Каждая из кнопок панели инструментов действует в текущей вкладке окна.

### 4.1.3.3 Выбор фильтров

Окно **Выбор фильтров** предназначено для выбора доступных фильтров и редактирования интерактивных условий фильтрации.

Фильтр выбирается галочкой напротив имени фильтра. Интерактивные условия фильтрации можно редактировать в окне редактирования фильтра, нажав на кнопку рядом с галочкой. Если используется несколько фильтров, то при фильтрации событий следует задать их объединение или пересечение.


**При объединении фильтров** (логическая операция "ИЛИ") сообщение включается в протокол событий, если любой из фильтров работает.



**При пересечении фильтров** (логическая операция "И") сообщение включается в протокол событий, если все фильтры работают. По умолчанию используется пересечение фильтров.

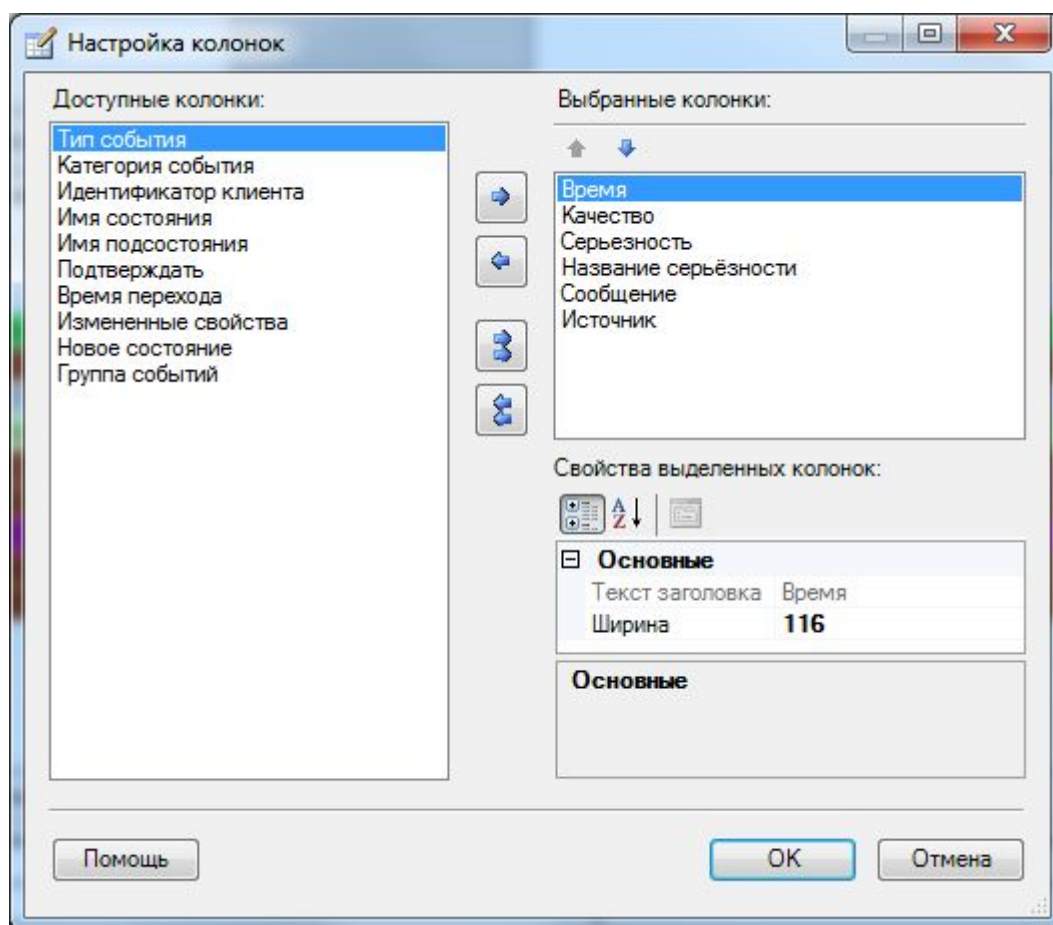
Фильтры могут быть применены к текущей вкладке или к новой вкладке выбором галочки **Открыть в новой вкладке** (доступно только в окне протокола событий).

### 4.1.3.4 Настройка колонок

Окно **Настройка колонок** вызывается кнопкой  – **Настройка колонок таблицы** – на панели инструментов окна **Элемент проекта. Протокол событий**.

**Для добавления** колонок, которые будут отображаться в протоколе событий, следует выделить требуемые колонки в списке **Доступные колонки** и нажать кнопку со стрелкой вправо.

Свойства колонок, выделенных в списке **Выбранные колонки**, можно редактировать в панели **Свойства выделенных колонок**.



Стрелки (**вверх/вниз**) на панели инструментов над списком выбранных колонок позволяют настроить порядок следования колонок в протоколе событий.


**Для удаления** колонок из протокола событий следует выделить нужные колонки в списке **Выбранные колонки** и нажать кнопку со стрелкой влево.

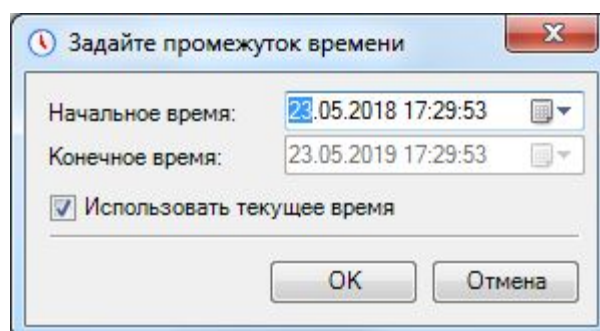
Для подтверждения сделанных изменений нужно нажать кнопку **OK**, для отмены – кнопку **Отмена**. Закрытие окна при помощи кнопки в заголовке также отменяет сделанные изменения.

Для вызова справки нажмите кнопку **Помощь**.


#### 4.1.3.5 Фильтрация событий по времени

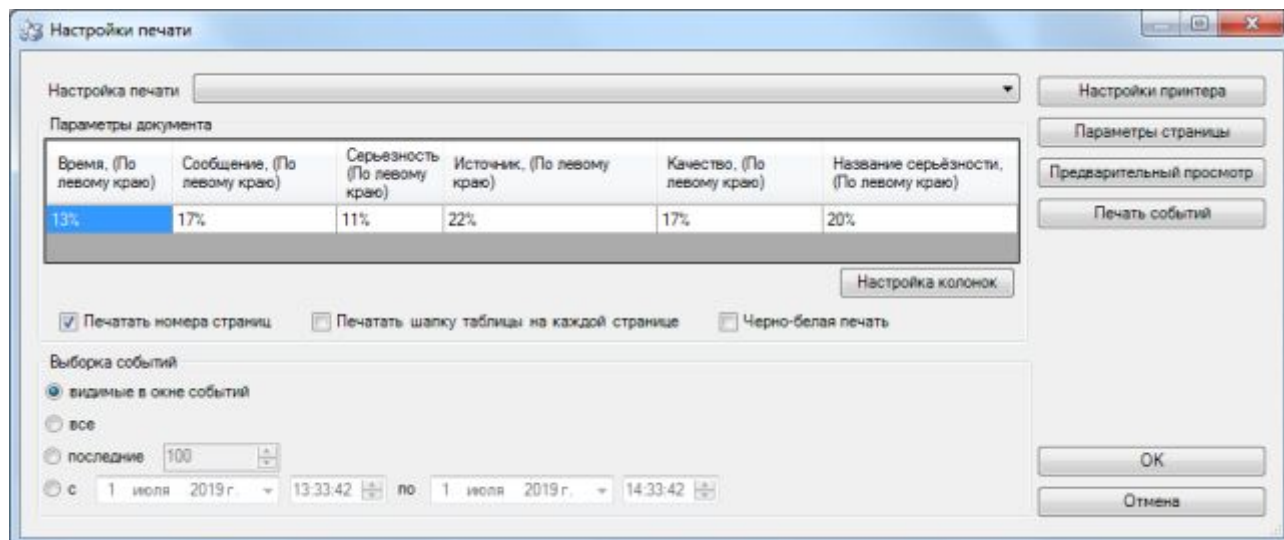
Для настройки фильтра по времени следует:

- 1 Нажать на кнопку  на панели инструментов окна **Элемент проекта. Протокол событий**
- 2 В появившемся окне **Задайте промежуток времени** ввести начальное и конечное время для фильтрации событий
- 3 Для задания в качестве конечного времени текущего времени установить галочку **Использовать текущее время**.



### 4.1.3.6 Печать событий

Окно **Настройки печати** вызывается при нажатии на кнопку  – **Печать событий** – на панели инструментов окна **Элемент проекта. Протокол событий**.



#### Параметры документа

- Группа настроек **Параметры документа** предназначена для настройки отображения выбранных колонок документа и их параметров. В заголовке каждой колонки отображается имя колонки и выравнивание. В ячейке таблицы отображается ширина колонки в процентах. При изменении ширины колонки автоматически пересчитывается ширина всех колонок. Ширину колонки можно изменить путем ввода с клавиатуры значения в ячейку. **Для выбора колонок и задания выравнивания** нужно щелкнуть на кнопке **Настройка колонок** и в появившемся окне **Настройка колонок** выбрать колонки и задать выравнивание текста при печати.
- Печатать номер страницы** – задает вывод номера страницы при печати
- Печатать шапку на каждой странице** – задает печать заголовка на каждой странице
- Черно-белая печать** – задание печати с использованием оттенков серого цвета

#### Выборка событий

Группа настроек **Выборка событий** предназначена для выбора **способа формирования выборки событий** для печати.

Можно указать один из четырех способов выборки событий для печати:

- Видимые в окне событий** – события, видимые в окне событий на момент печати или предварительного просмотра
- Все** – все события, зарегистрированные в протоколе событий
- Последние [количество]** – последние события протокола событий. Количество событий задается в поле, которое становится доступным при выборе этого способа
- С [начальная дата, время] по [конечная дата, время]** – события за заданный промежуток времени. Временной интервал задается в элементах управления выбора даты и времени, которые становятся доступны при выборе этого способа.

**Фильтрация** при выборке событий задается в соответствии с **настройками фильтров** в окне **Элемент проекта. Протокол событий**.


### Элементы управления

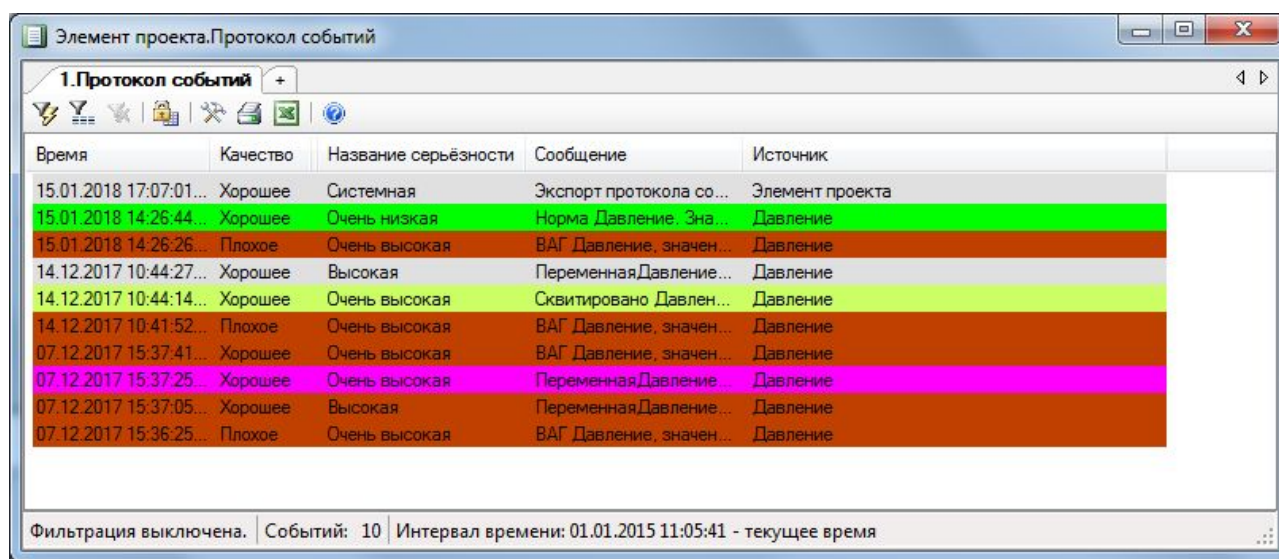
Для управления настройками печати протокола событий используются следующие кнопки:

- **Настройки принтера** – вызывает стандартное диалоговое окно **Настройки принтера**
- **Параметры страницы** – вызывает стандартное диалоговое окно **Параметры страницы**
- **Предварительный просмотр** – вызывает стандартное диалоговое окно **Предварительный просмотр**
- **Печать событий** – печатает события в соответствии с заданными настройками
- **ОК** – сохраняет текущие **настройки печати** и закрывает диалоговое окно.
- **Отмена** – отменяет все сделанные изменения и закрывает диалоговое окно.

#### 4.1.3.7 Экспорт протокола событий

Для экспорта протокола событий следует:

- 1 Нажать на кнопку **Экспорт в файл**  на панели инструментов окна протокола событий



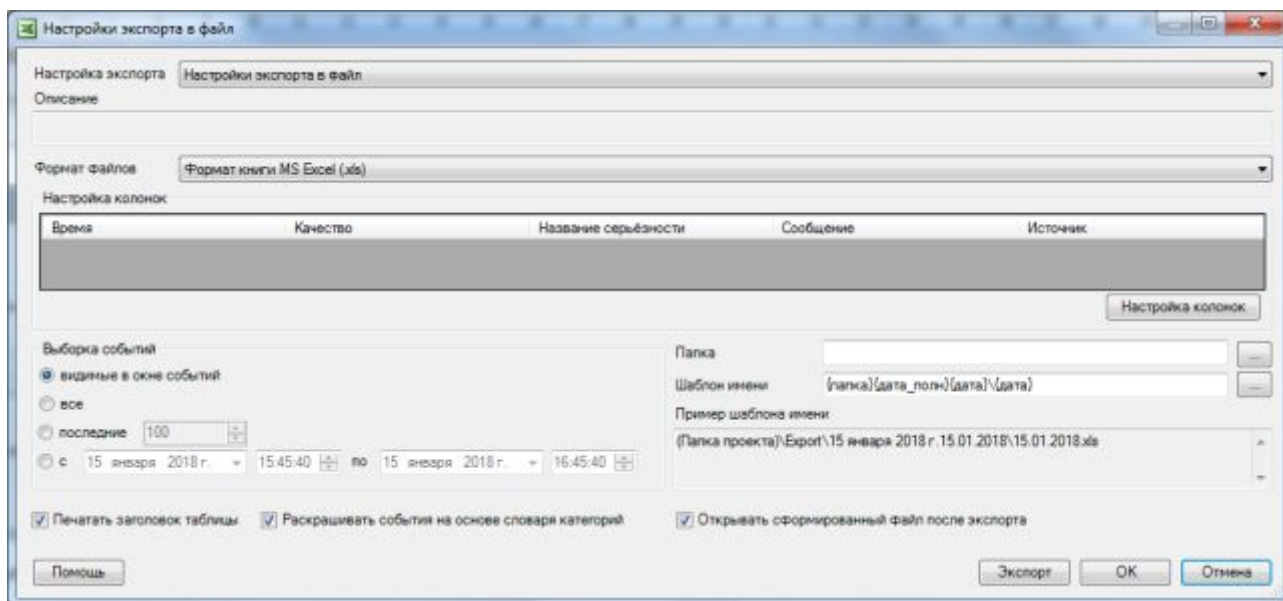
- 2 В открывшемся окне **Настройка экспорта в файл** выбрать формат файла, задать требуемые значения параметров и при необходимости настроить колонки (нажать на кнопку **Настроить колонки**).
- 3 Нажать на кнопку **Экспорт**

Сохранение в файл происходит в асинхронном режиме.

По окончании процесса формируется событие в протокол событий.

Пример сформированного файла (MS Excel) протокола событий приведен ниже.





	A	B	C	D	E
1	Время	Качество	Название серьезности	Сообщение	Источник
2	15.01.2018 17:07:01.817	Хорошее	Системная	Экспорт протокола событий завершен: 15.01.2018.xls	Элемент проекта
3	15.01.2018 14:26:44.404	Хорошее	Очень низкая	Норма Давление. Значение: 1	Давление
4	15.01.2018 14:26:26.303	Плохое	Очень высокая	ВАГ Давление, значение: 10	Давление
5	14.12.2017 10:44:27.045	Хорошее	Высокая	Переменная Давление=5. Верхняя предупредительная граница	Давление
6	14.12.2017 10:44:14.336	Хорошее	Очень высокая	Сквитировано Давление. Значение: 15	Давление
7	14.12.2017 10:41:52.677	Плохое	Очень высокая	ВАГ Давление, значение: 15	Давление
8	07.12.2017 15:37:41.586	Хорошее	Очень высокая	ВАГ Давление, значение: 15	Давление
9	07.12.2017 15:37:25.656	Хорошее	Очень высокая	Переменная Давление= 15. Нижняя аварийная граница	Давление
10	07.12.2017 15:37:05.436	Хорошее	Высокая	Переменная Давление= 5. Нижняя предупредительная граница	Давление
11	07.12.2017 15:36:25.436	Плохое	Очень высокая	ВАГ Давление, значение: 10	Давление

## 4.1.4 Регистрация событий в протоколе

Для регистрации событий в протоколе событий используются **функции отправки сообщений**.

Эти функции вызываются в скриптах, подключаемых в реакциях графических примитивов.

### Функции отправки сообщений

Для отправки сообщений в Протокол событий используются функции:

- **AlarmServiceHelper.SendSimpleEvent** – отправка сообщения о произвольном пользовательском событии.
- **AlarmServiceHelper.SendTrackingEvent** – отправка сообщения о событии, связанных с произошедшими изменениями.

Для использования этих функций в скриптах, в редакторе скриптов во вкладке **Пространства имен** добавьте ссылку на **Krug.ObjectModel.AlarmServiceInterfaces**



## Регистрация пользовательского события из словаря событий

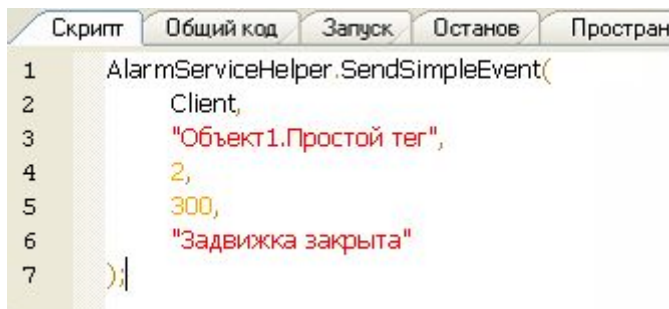
```
void SendSimpleEvent (object client, string source, int eventCategory, int severity,
                     string message);
```

где

**client** – объект (владелец скрипта), возвращающий сервис IAlarmManager  
**source** – источник события (например "Объект1.Простой тег"), который сгенерировал уведомление о событии  
**eventCategory** – категория, которой принадлежит данное событие  
**severity** – серьезность (важность) события. Целое значение из диапазона [1...1000]  
**message** – текст, который описывает событие

### Пример

```
AlarmServiceHelper.SendSimpleEvent(Client, "Объект1.Простой тег", 2, 300,
"Задвижка закрыта");
```



```
void SendSimpleEvent (object client, string source, int eventCategory, int severity,
                     int groupId, string message);
```

где

**client** – объект (владелец скрипта), возвращающий сервис IAlarmManager  
**source** – источник события (например "Объект1.Простой тег"), который сгенерировал уведомление о событии  
**eventCategory** – категория, которой принадлежит данное событие  
**severity** – серьезность (важность) события  
**groupId** – группа событий  
**message** – текст, который описывает событие

```
void SendSimpleEvent (object client, string source, int messageIdInDictionary,
                     params object[] data);
```

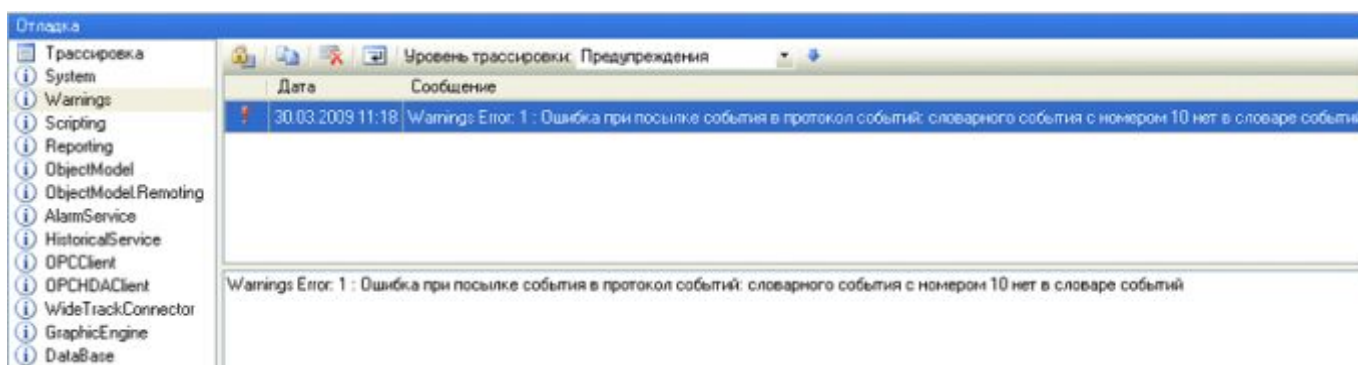
где

**client** – объект, возвращающий сервис IAlarmManager  
**source** – объект, который сгенерировал уведомление о событии  
**messageIdInDictionary** – номер события в словаре событий  
**data** – данные, которые передаются в текст сообщения.  
 Если параметры отсутствуют, никакие значения задавать не нужно, если параметры есть – перечислите их через запятую.

Если события с данным номером нет в словаре событий, то в окне **Отладка** в журнале регистрации сообщений с именем **Warnings** (предупреждения) появится сообщение:

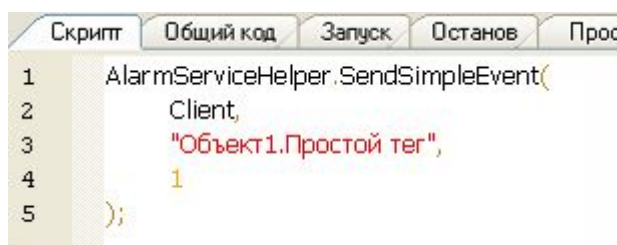
**Warnings Error: 1: Ошибка при посылке события в протокол событий: словарного события с номером [MessageID] нет в словаре событий.**





### Пример

```
AlarmServiceHelper.SendSimpleEvent(Client, "Объект1.Простой тег", 1);
```



### Регистрация произошедших изменений

```
void SendTrackingEvent (object client, string source, int eventCategory,
                        int severity, string message, string actorID);
```

где

**client** – объект (владелец скрипта), возвращающий сервис IAlarmManager

**source** – источник события (например "Объект1.Простой тег"), который сгенерировал уведомление о событии

**eventCategory** – категория, которой принадлежит данное событие

**severity** – серьезность (важность) события

**message** – текст, который описывает событие

**actorID** – идентификатор клиента, который инициализировал действие, приводящее к событию, имеющему отношение к трэкингу

```
void SendTrackingEvent (object client, string source, int eventCategory,
                        int severity, int groupId, string message, string actorID);
```

где

**client** – объект (владелец скрипта), возвращающий сервис IAlarmManager

**source** – источник события (например "Объект1.Простой тег"), который сгенерировал уведомление о событии

**eventCategory** – категория, которой принадлежит данное событие

**severity** – серьезность (важность) события

**groupId** – группа событий

**message** – текст, который описывает событие

**actorID** – идентификатор клиента, который инициализировал действие, приводящее к событию, имеющему отношение к трэкингу

```
void SendTrackingEvent (object sender, string source, int messageIdInDictionary,
                        string actorID, params object[] data);
```

где

**sender** – объект - причина события  
**source** – источник события (например "Объект1.Простой тег"),  
 который сгенерировал уведомление о событии  
**source** – источник события (например "Объект1.Простой тег"),  
 который сгенерировал уведомление о событии  
**messageIdInDictionary** – номер события в словаре событий  
**actorID** – идентификатор клиента, который инициализировал  
 действие, приводящее к событию, имеющему отношение к трэкингу  
**data** – данные, которые передаются в текст сообщения  
 (необязательные параметры). Если параметры отсутствуют, никакие значения  
 задавать не нужно, если параметры есть – перечислите их через запятую.

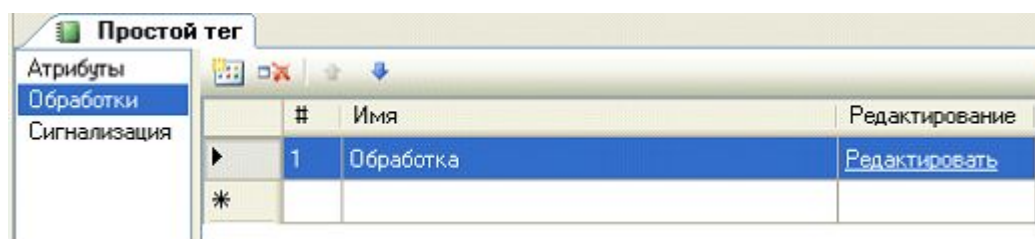
```
void SendTrackingEvent(object client, string tagName, string attributeName,
                      int messageIdInDictionary, params object[] data);
```

где

**client** – объект (владелец скрипта), возвращающий сервис IAlarmManager  
**tagName** – имя тэга  
**attributeName** – имя атрибута тэга  
**messageIdInDictionary** – номер события в словаре событий  
**data** – данные, которые передаются в текст сообщения  
 (необязательные параметры). Если параметры отсутствуют, никакие значения задавать  
 не нужно, если параметры есть – перечислите их через запятую

### Пример. Обработка тэга

**DataRate** предоставляет возможность назначить для тэга алгоритм его обработки.



В качестве алгоритма обработки выступает скрипт.

Используем следующий скрипт, как обработку тэга.

```
System.String Source = String.Format("{0}.{1}.{2}.{3}", ProjectName,
                                     ObjectName, TagName, AttributeName);
System.Int32 MessageID = 10;
System.String ActorID = TagName;
AlarmServiceHelper.SendTrackingEvent(Client, Source, MessageID, ActorID,
                                     OldValue.AsString, NewValue.AsString);
```

```

Скрипт      Пространства имен
1      // источник события - например, полное имя атрибута
2      System.String Source =
3          String.Format("{0}.{1}.{2}.{3}", ProjectName, ObjectName, TagName, AttributeName);
4      // сообщение в словаре событий под десятым номером
5      // описано так "Изменение значения атрибута с {0} на {1}" с серьезностью "Системная"
6      System.Int32 MessageID = 10;
7      // идентификатор клиента - например, имя тега
8      System.String ActorID = TagName;
9      // регистрация изменения
10     AlarmServiceHelper.SendTrackingEvent(
11         Client,
12         Source,
13         MessageID,
14         ActorID,
15         OldValue.AsString,
16         NewValue.AsString);
    
```

или

```

System.Int32 MessageID = 10;
AlarmServiceHelper.SendTrackingEvent(Client, TagName, AttributeName,
                                     MessageID, OldValue.AsString,
                                     NewValue.AsString);
    
```

```

Скрипт      Пространства имен
1      // сообщение в словаре событий под десятым номером
2      // описано так "Изменение значения атрибута с {0} на {1}" с серьезностью "Системная"
3      System.Int32 MessageID = 10;
4      // регистрация изменения
5      AlarmServiceHelper.SendTrackingEvent(
6          Client,
7          TagName,
8          AttributeName,
9          MessageID,
10         OldValue.AsString,
11         NewValue.AsString);
    
```

Обработка привязана к аргументу **Атрибут**.

Аргументы -				
Имя в скрипте	Имя	Тип	Запускать скрипт	Привязка
Атрибут	Атрибут	Double	<input checked="" type="checkbox"/>	NA

При изменении значения атрибута (с 0 на 1)

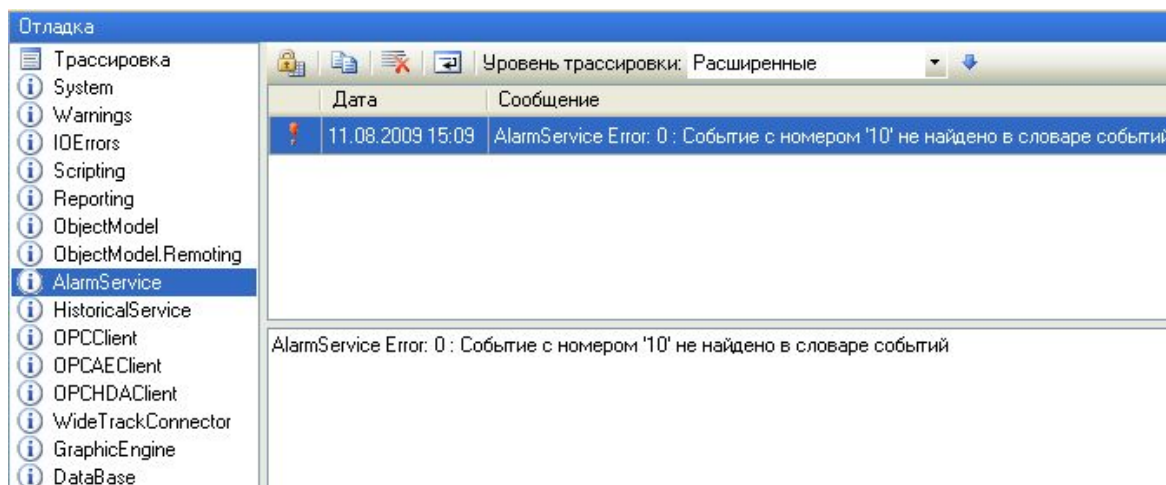
Простой тег					
Простой тег.Обработка					
Объект.Простой тег					
	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение
	Атрибут	Double		<input type="checkbox"/>	1

в протоколе событий появится сообщение:

Время	Источник	Сообщение	Идентификатор клиента
11.08.2009 14:56:56	Элемент проекта.Объект.Простой тег.Атрибут	Изменение значения атрибута с 0 на 1	Простой тег

Если в скрипте неправильно указан номер сообщения (**MessageID**) в словаре событий, то в окне **Отладка** в журнале регистрации сообщений с именем **AlarmService** появится сообщение:

**AlarmService Error: 0 : Событие с номером не найдено в словаре событий.**



#### Пример. Обрыв соединения

Для того чтобы занести в протокол событий сообщение об обрыве соединения OPC AE коннектора с OPC AE сервером для объекта OPC AE Коннектор следует:

1 Создать скрипт с текстом:

```
if (IsError)
{
    AlarmServiceHelper.SendTrackingEvent(
        Client,
        "OPC AE Коннектор",
        1,
        500,
        "Ошибка при работе коннектора с сервером",
        "DataRate System");
}
```

2 Добавить аргумент с именем **IsError** и привязать его к атрибуту **IsError** тега **System** текущего коннектора

3 Для аргумента выставить признак **Запускать скрипт**.

При потере соединения с OPC AE сервером в протокол событий будет добавлено сообщение об ошибке.

#### 4.1.5 Печать сообщений о событиях из скрипта

Для печати протокола событий из скрипта используются специальные методы класса **AlarmServiceHelper**. Для того чтобы методы этого класса стали доступны, в редакторе скриптов во вкладке **Пространства имен** добавьте ссылку на **Krug.ObjectModel.AlarmServiceInterfaces**.

### Печать последних сообщений о событиях

```
AlarmServiceHelper.PrintLastEvents(object client, int count,  
                                   string printSettingsName)
```

где

<b>Client</b>	– владелец скрипта
<b>Count</b>	– количество печатаемых сообщений о событиях
<b>printSettingsName</b>	– название используемой группы настроек или null, если настройки будут запрашиваться у пользователя.

### Печать последних сообщений о событиях, отфильтрованных по заданным свойствам

```
AlarmServiceHelper.PrintLastEvents(object client, int count,  
                                   IEventNotificationFilter[] filters,  
                                   string printSettingsName)
```

где

<b>Client</b>	– владелец скрипта
<b>Count</b>	– количество печатаемых сообщений о событиях
<b>Filters</b>	– массив фильтров для фильтрации событий по определенным свойствам
<b>printSettingsName</b>	– название используемой группы настроек или null, если настройки будут запрашиваться у пользователя.

### Печать всех сообщений протокола событий

```
AlarmServiceHelper.PrintAllEvents(object client, string printSettingsName)
```

где

<b>Client</b>	– владелец скрипта
<b>printSettingsName</b>	– название используемой группы настроек или null, если настройки будут запрашиваться у пользователя.

### Печать всех сообщений протокола событий, отфильтрованных по заданным свойствам

```
AlarmServiceHelper.PrintAllEvents(object client, IEventNotificationFilter[] filters,  
                                   string printSettingsName)
```

где

<b>Client</b>	– владелец скрипта
<b>Filters</b>	– массив фильтров для фильтрации событий по определенным свойствам
<b>printSettingsName</b>	– название используемой группы настроек или null, если настройки будут запрашиваться у пользователя.

### Печать сообщений о событиях в заданном промежутке времени

```
AlarmServiceHelper.PrintEvents(object client, DateTime startTime,  
                               DateTime endTime, int maxCountLimit,  
                               string printSettingsName)
```

где

<b>Client</b>	– владелец скрипта
<b>StartTime</b>	– начальное время запрашиваемого интервала;
<b>EndTime</b>	– конечное время запрашиваемого интервала;
<b>maxCountLimit</b>	– ограничение на количество возвращаемых событий или -1, если без ограничений

**printSettingsName** – название используемой группы настроек или null, если настройки будут запрашиваться у пользователя.

### Печать сообщений о событиях в заданном промежутке времени, отфильтрованных по заданным свойствам

```
AlarmServiceHelper.PrintEvents(object client, DateTime startTime,
                               DateTime endTime, EventFilter[] filters,
                               int maxCountLimit, string printSettingsName)
```

где

**Client** – владелец скрипта  
**StartTime** – начальное время запрашиваемого интервала  
**EndTime** – конечное время запрашиваемого интервала  
**Filters** – массив фильтров для фильтрации событий по определенным свойствам  
**maxCountLimit** – ограничение на количество возвращаемых событий или -1, если без ограничений  
**printSettingsName** – название используемой группы настроек или null, если настройки будут запрашиваться у пользователя.

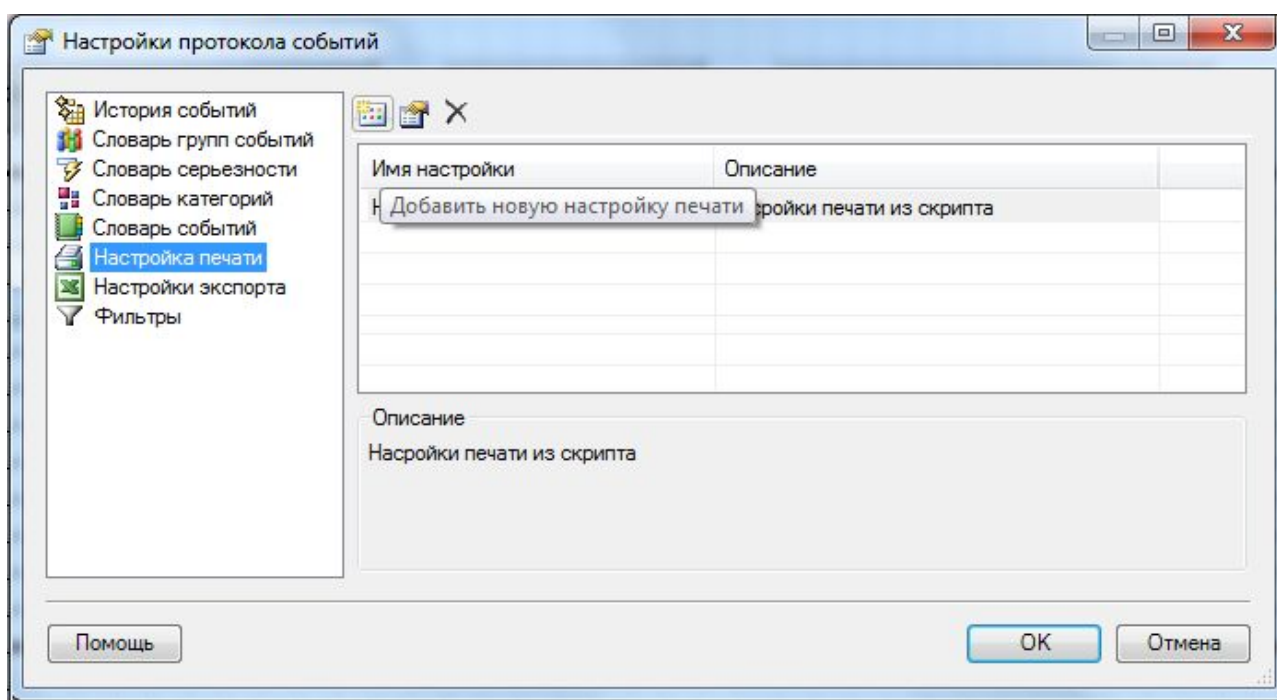
### ВНИМАНИЕ!!!

Название группы настроек для параметра **printSettingsName** в скрипте нужно брать из списка настроек печати протокола событий. Если заданное в скрипте название группы настроек не будет найдено в списке настроек печати протокола событий или будет пустым, то печать событий не произойдет и в журнал регистрации (лог) будет записано соответствующее предупреждение.

### Пример.

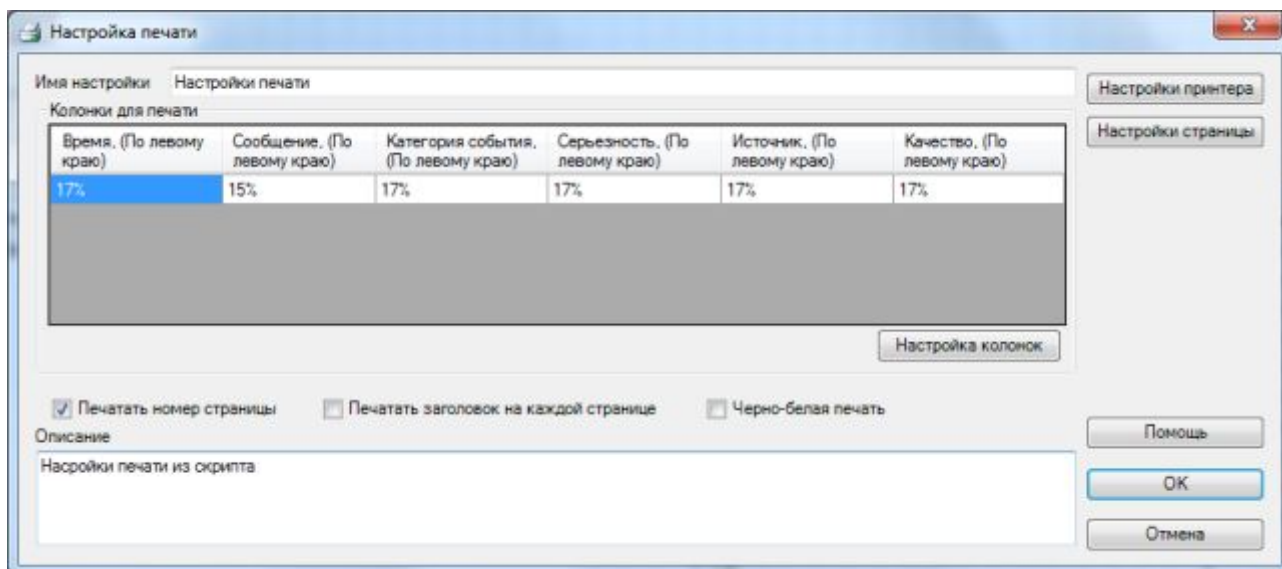
Добавим в проект, с установленным элементом **Протокол событий**, настройки и графический элемент активизации печати Протокола событий:

- 1 В окне **Редактор проекта** выбрать элемент **Протокол событий**, правой кнопкой мыши вызвать контекстное меню и выбрать пункт меню **Редактировать...**

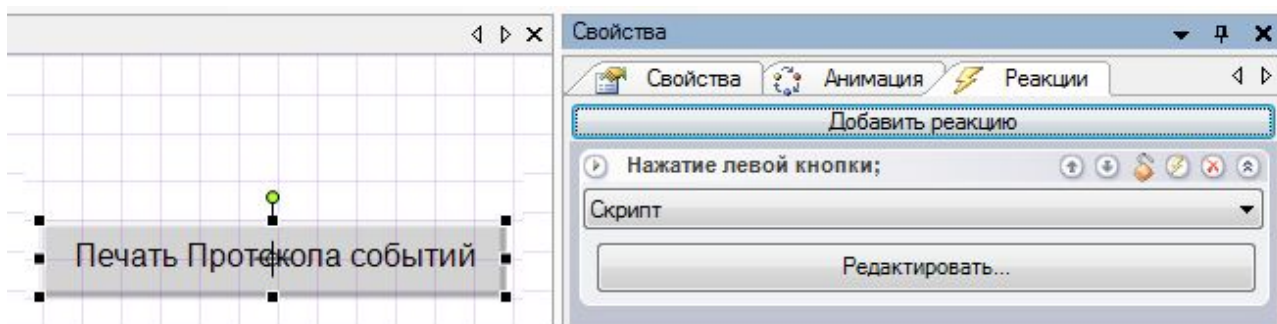




- 2 В открывшемся окне **Настройки протокола событий** выбрать **Настройка печати** и нажать на кнопку **Добавить новую настройку печати**
- 3 В открывшемся окне **Настройка печати** задать необходимые параметры: настройки принтера (в примере: **Foxit Reader PDF Printer**), настройки страницы (**Альбомная**), колонки (**Настройка колонок**) и описание (в примере "Настройки печати из скрипта"), и нажать на кнопку **ОК**



- 4 Для управления печатью создать новый объект (например, с именем **Объект**) с видом **Мнемосхема** и добавить на мнемосхему примитив **Объёмный прямоугольник** с текстом **Печать Протокола событий**
- 5 Настроить реакцию **Скрипт** объемного прямоугольника на нажатие левой кнопки мыши



- 6 Отредактировать текст скрипта

### ВНИМАНИЕ!

Проверьте, что на вкладке **Пространства имён** добавлены:

- **Krug.ObjectModel.AlarmServiceInterfaces**
- **Krug.ObjectModel.AlarmService.Filtration**

```

1 //Начальное время 1.12.2017 10:00.
2 DateTime startTime = new DateTime(2017, 12, 1, 10, 0, 0, DateTimeKind.Local);
3
4 // Конечное время 15.1.2018 17:00.
5 DateTime endTime = new DateTime(2018, 1, 15, 17, 0, 0, DateTimeKind.Local);
6
7 // Создание массива фильтров.
8 IEventNotificationFilter[] filters = new IEventNotificationFilter[1];
9
10 // Создание фильтра Серьёзность
11 //(типы фильтруемых значений серьёзности от 200 до 1000)
12 // и добавление в массив фильтров.
13 SeverityFilterCondition severityFilterCondition = new SeverityFilterCondition();
14 severityFilterCondition.MinLevel = 1;
15 severityFilterCondition.MaxLevel = 1000;
16 filters[0] = severityFilterCondition;
17
18 try
19 {
20     // Печать событий с соответствующими настройками.
21     AlarmServiceHelper.PrintEvents(Client, startTime, endTime, filters, 10, "Настройки печати");
22 }
23
24 catch (Exception ex)
25 {
26     // Если произошла ошибка, показывать сообщение с её описанием.
27     MessageBox.Show(ex.Message);
28 }
29

```

- 7 Сохранить проект, запустить его на исполнение и щелкнуть по созданной кнопке **Печать Протокола событий**

Контроль уровня: ВАГ - 10; ВПГ - 5; НПГ - 5; НАГ - 10

Поле ввода:

Заданное значение: 4

Окно активной сигнализации    Квитировать    Печать Протокола событий

Время	Событие	Элемент проекта	Состояние
15.01.2018 17:07:01	Экспорт протокола событий завершен: 15.01.2018.xls		Хорошее
15.01.2018 14:28:44	Норма Давление. Значение 1	Давление	Хорошее
15.01.2018 14:28:26	ВАГ Давление. значение: 10	Давление	Плохое
14.12.2017 10:44:27	Переменная Давление-5. Верхняя предупредительная граница	Давление	Хорошее
14.12.2017 10:44:14	Сквитировано Давление. Значение 15	Давление	Хорошее
14.12.2017 10:41:52	ВАГ Давление. значение: 15	Давление	Плохое
07.12.2017 15:37:41	ВАГ Давление. значение: 15	Давление	Хорошее
07.12.2017 15:37:25	Переменная Давление-5. Нижняя аварийная граница	Давление	Хорошее
07.12.2017 15:37:05	Переменная Давление-5. Нижняя предупредительная граница	Давление	Хорошее
07.12.2017 15:36:25	ВАГ Давление. значение: 10	Давление	Плохое

- 8 В результате сформируется файл (в настройке печати "**Настройки печати**" установлена печать в PDF-файл), имя которого включает названия проекта и элемента проекта, а также текст "Протокол событий": **ПроектПримерСигнализации. Элемент проекта.Протокол событий.pdf...**

ПроектПримерСигнализации. Элемент проекта.Протокол событий						
Время	Сообщение	Категория события	Серьезность	Источник	Качество	
15.01.2018 17:07:01.817	Экспорт протокола событий завершен: 15.01.2018.xls	Информационный	500	Элемент проекта	Хорошее	
15.01.2018 14:26:44.404	Норма Давление. Значение: 1	Норма	500	Давление	Хорошее	
15.01.2018 14:26:26.303	ВАГ Давление, значение: 10	Настройка	1000	Давление	Плохое	
14.12.2017 10:44:27.045	ПеременнаяДавление=5. Верхняя предупредительная граница	Информационный	750	Давление	Хорошее	
14.12.2017 10:44:14.336	Сквитировано Давление. Значение: 15	Квитирование	1000	Давление	Хорошее	
14.12.2017 10:41:52.677	ВАГ Давление, значение: 15	Настройка	1000	Давление	Плохое	
07.12.2017 15:37:41.580	ВАГ Давление, значение: 15	Настройка	1000	Давление	Хорошее	
07.12.2017 15:37:25.856	ПеременнаяДавление=15. Нижняя аварийная граница	Диагностика	1000	Давление	Хорошее	
07.12.2017 15:37:05.436	ПеременнаяДавление=5. Нижняя предупредительная граница	Настройка	750	Давление	Хорошее	
07.12.2017 15:36:25.436	ВАГ Давление, значение: 10	Настройка	1000	Давление	Плохое	

### ВНИМАНИЕ!

Данный скрипт можно запускать и по расписанию

#### 4.1.6 Экспорт протокола событий из скрипта

Для экспорта протокола событий из скрипта используются специальные методы класса **AlarmServiceHelper**.

Для того чтобы методы этого класса стали доступны, в редакторе скриптов во вкладке **Пространства имен** добавьте ссылку на **Krug.ObjectModel.AlarmServiceInterfaces**.

**Экспорт событий в заданном промежутке времени, отфильтрованных по заданным свойствам**

```
AlarmServiceHelper.ExportEvents(object client, string printSettingsName, DateTime
startTime, DateTime endTime,
IEventNotificationFilter[] filterOptions)
```

где

**client** — клиент скрипта;  
**printSettingsName** — название используемой группы настроек;  
**startTime** — начальное время запрашиваемого интервала;  
**endTime** — конечное время запрашиваемого интервала;  
**filterOptions** — массив фильтров для фильтрации событий по определенным свойствам;

**Экспорт заданного количество событий, начиная с указанной даты назад, отфильтрованных по заданным свойствам**

`AlarmServiceHelper.ExportEvents(object client, string printSettingsName, DateTime endTime, int eventsCount, IEventNotificationFilter[] filterOptions)`

где

**client** — клиент скрипта;  
**printSettingsName** — название используемой группы настроек;  
**endTime** — конечное время запрашиваемого интервала;  
**eventsCount** — количество экспортируемых событий;  
**filterOptions** — массив фильтров для фильтрации событий по определенным свойствам.

**Экспорт всех событий, зарегистрированных протоколом событий, отфильтрованных по заданным свойствам**

`AlarmServiceHelper.ExportEvents(object client, string printSettingsName, IEventNotificationFilter[] filterOptions)`

где

**client** — клиент скрипта;  
**printSettingsName** — название используемой группы настроек;  
**filterOptions** — массив фильтров для фильтрации событий по определенным свойствам;

**Экспорт событий в заданном промежутке времени, отфильтрованных по заданным свойствам, в произвольном формате в произвольный файл**

`AlarmServiceHelper.ExportEvents(object client, DateTime startTime, DateTime endTime, string exportPath, ExportFileFormats format, IEventNotificationFilter[] filterOptions)`

где

**client** — клиент скрипта;  
**startTime** — начальное время запрашиваемого интервала;  
**endTime** — конечное время запрашиваемого интервала;  
**format** — формат выходного файла;  
**filterOptions** — массив фильтров для фильтрации событий по определенным свойствам.

**Экспорт всех событий, отфильтрованных по заданным свойствам, в произвольном формате в произвольный файл.**

`AlarmServiceHelper.ExportEvents(object client, string exportPath, ExportFileFormats format, IEventNotificationFilter[] filterOptions)`

где

**client** — клиент скрипта;  
**exportPath** — каталог для экспорта;  
**format** — формат выходного файла;  
**filterOptions** — массив фильтров для фильтрации событий по определенным свойствам.



### ВНИМАНИЕ!

Название группы настроек для параметра **printSettingsName** в скрипте нужно брать из списка настроек экспорта протокола событий.

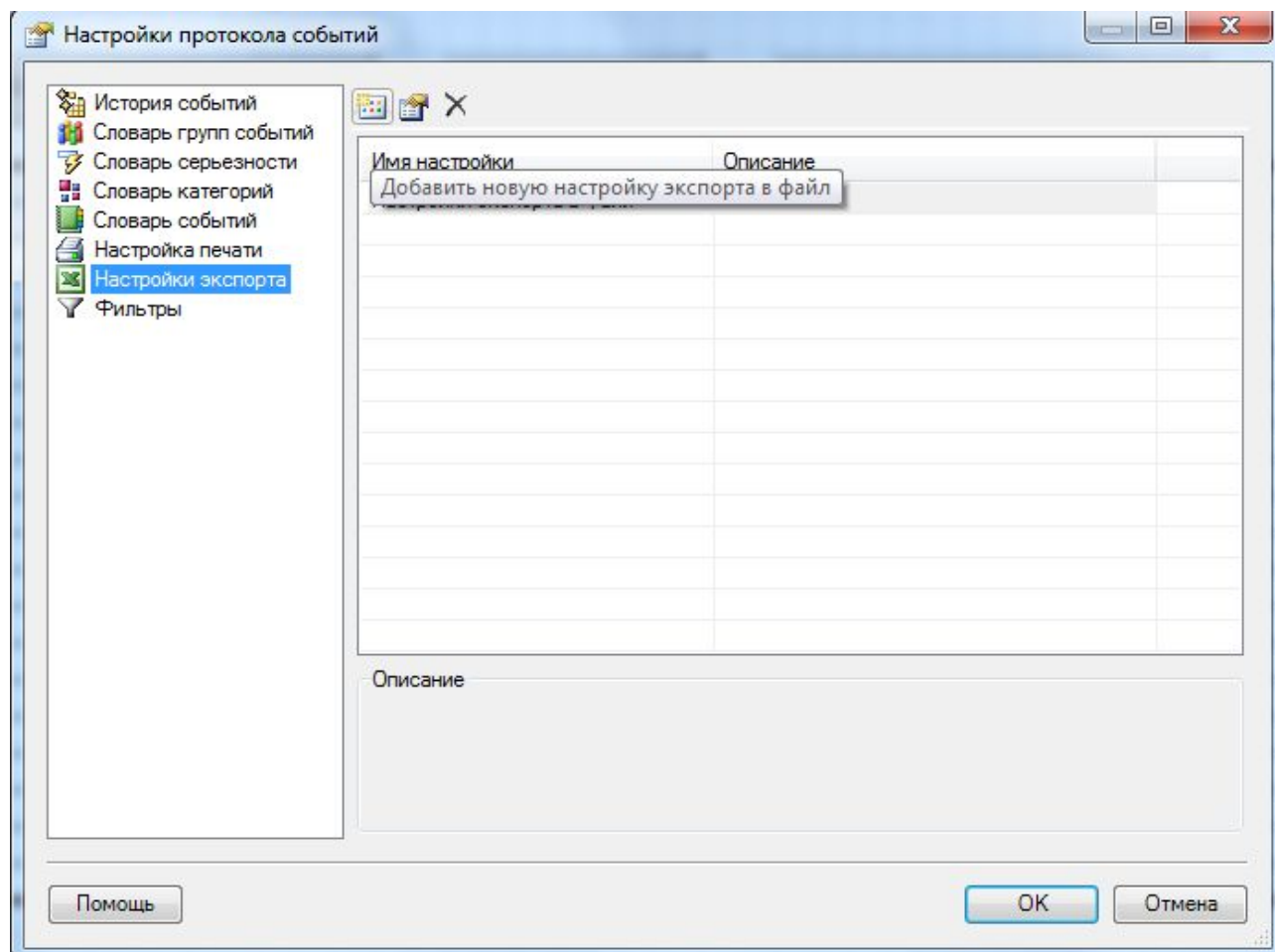
Если заданное в скрипте **название группы настроек** не будет найдено в **списке настроек экспорта протокола событий** или будет пустым, то экспорт событий не произойдет и в журнал регистрации (лог) будет записано соответствующее предупреждение.

### Пример

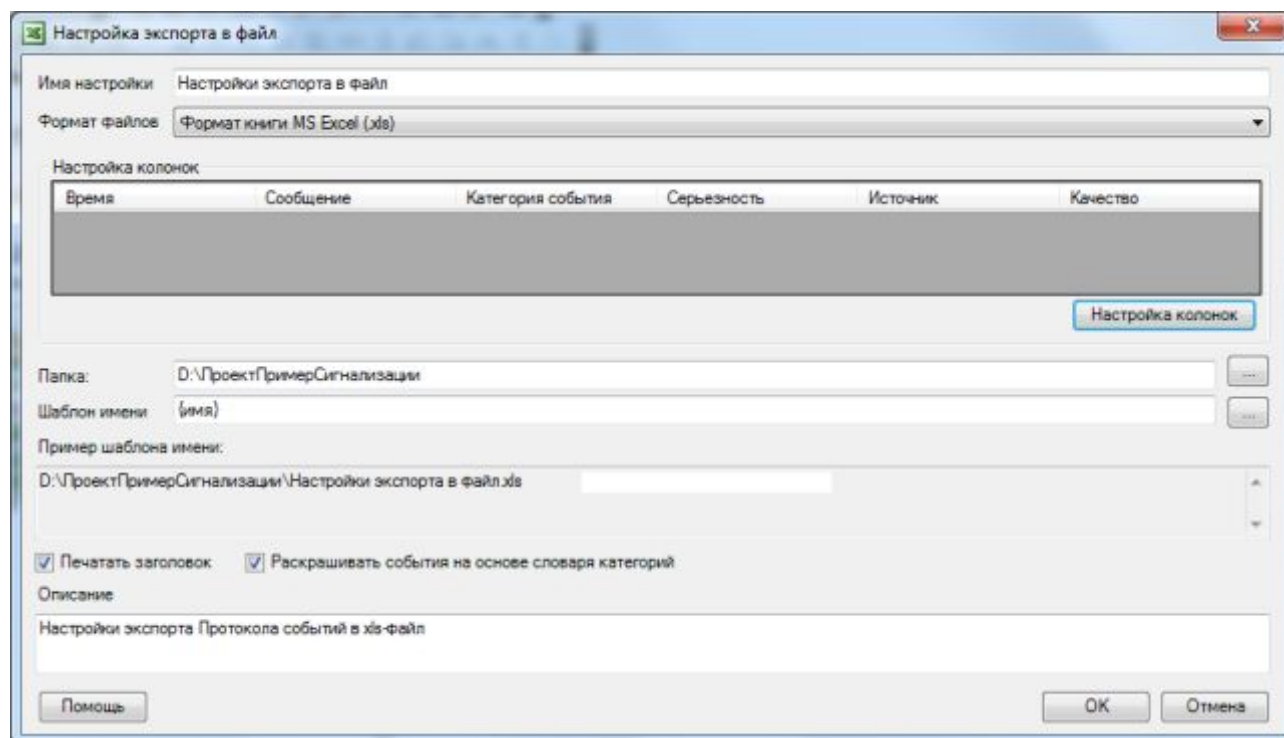
Добавим в проект, с установленным элементом **Протокол событий**, настройки и графический элемент активизации экспорта Протокола событий.

Сначала определим настройки экспорта:

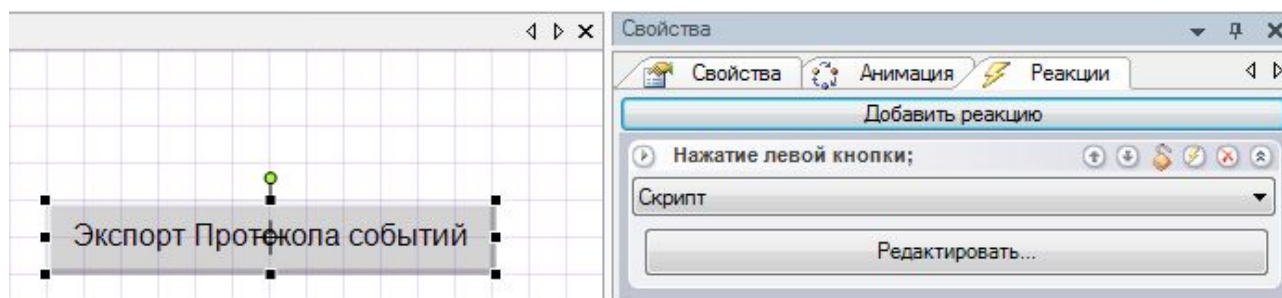
- 1 В окне **Редактор проекта** выбрать элемент **Протокол событий**, правой кнопкой мыши вызвать контекстное меню и выбрать пункт меню **Редактировать...**
- 2 В открывшемся окне **Настройки протокола событий** добавить новую настройку экспорта **Настройки экспорта в файл**



- 3 В открывшемся окне **Настройка экспорта в файл** задать формат файла (в примере: .xls), колонки (**Окно настройки экспорта**), папку и шаблон имени для создаваемого файла экспорта, признаки и описание (в примере: "Настройки экспорта Протокола событий в xls-файл")



- 4 Разместить на мнемосхеме объемный прямоугольник с текстом «Экспорт Протокола событий» и настроить его реакцию **Скрипт** на нажатие левой кнопки мыши



- 5 Отредактировать текст скрипта

### ВНИМАНИЕ!

Проверьте, что на вкладке Пространства имён добавлены:

- `Krug.ObjectModel.AlarmServiceInterfaces`
- `Krug.ObjectModel.AlarmService.Filtration`

- 6 Сохранить изменения и запустить проект на выполнение

- 7 Для экспорта Протокола событий нажать на кнопку **Экспорт Протокола событий**



```

1 // Создание массива фильтров.
2 IEventNotificationFilter[] filters = new IEventNotificationFilter[1];
3
4 // Создание фильтра Серьёзность
5 //(типы фильтруемых значений серьёзности от 1 до 1000)
6 // и добавление в массив фильтров.
7 SeverityFilterCondition severityFilterCondition = new SeverityFilterCondition();
8 severityFilterCondition.MinLevel = 1; // минимальный уровень серьёзности
9 severityFilterCondition.MaxLevel = 1000; // максимальный уровень серьёзности
10 filters[0] = severityFilterCondition;
11
12 // Время, до которого будет осуществляться выборка
13 DateTime endTime = new DateTime(2018, 01, 01, 15, 0, 0, DateTimeKind.Local);
14
15 //вызов функции ExportEvents должен быть обернут в try... catch
16 try
17 {
18 // Экспортировать 5 событий до времени endTime с использованием сохранённой
19 // настройки экспорта "Настройка экспорта в файл" и применением фильтров filters
20 AlarmServiceHelper.ExportEvents(Client, "Настройка экспорта в файл", endTime, 5, filters);
21 // вывести сообщение об успешном экспорте
22 MessageBox.Show("Экспорт выполнен");
23 }
24 catch (Exception ex)
25 {
26 // вывести сообщение в случае неудачи
27 MessageBox.Show(ex.Message);
28 }
29

```

В результате сформируется файл **Настройки экспорта в файл.xls**

	A	B	C	D	E	F
1	Время	Сообщение	Категория события	Серьёзность	Источник	Качество
2	14.12.2017 10:44:27.045	ПеременнаяДавление=5. Верхняя предупредительная граница	Информационный	750	Давление	Хорошее
3	14.12.2017 10:44:14.336	Сквитировано Давление. Значение 15	Квитирование	1000	Давление	Хорошее
4	14.12.2017 10:41:52.677	ВАГ Давление, значение: 15	Настройка	1000	Давление	Плохое
5	07.12.2017 15:37:41.586	ВАГ Давление, значение: 15	Настройка	1000	Давление	Хорошее
6	07.12.2017 15:37:25.656	ПеременнаяДавление= 15. Нижняя аварийная граница	Диагностика	1000	Давление	Хорошее

## ВНИМАНИЕ!

Данный скрипт можно вызывать с помощью функции реакции или запускать по расписанию

#### 4.1.7 Чтение протокола событий из скрипта

Для чтения событий из скрипта используются специальные методы класса **AlarmServiceHelper**. Для того чтобы методы этого класса стали доступны, в редакторе скриптов во вкладке **Пространства имен** добавьте ссылку на **Krug.ObjectModel.AlarmServiceInterfaces**.

**Чтение событий за заданный промежуток времени с учетом их количества и фильтров**

```
IEventNotification[] ReadEvents (object client, DateTime startTime, DateTime endTime,  
int count, IEventNotificationFilter[] filters);
```

где

**client** – объект, возвращающий сервис IAlarmManager  
**startTime** – начальное время  
**endTime** – конечное время  
**count** – максимальное количество запрашиваемых данных (0 – все данные)

**Чтение событий за заданный промежуток времени**

```
IEventNotification[] ReadEvents (object client, DateTime startTime, DateTime endTime);
```

где

**client** – объект, возвращающий сервис IAlarmManager  
**startTime** – начальное время  
**endTime** – конечное время

## 4.2 Сигнализация

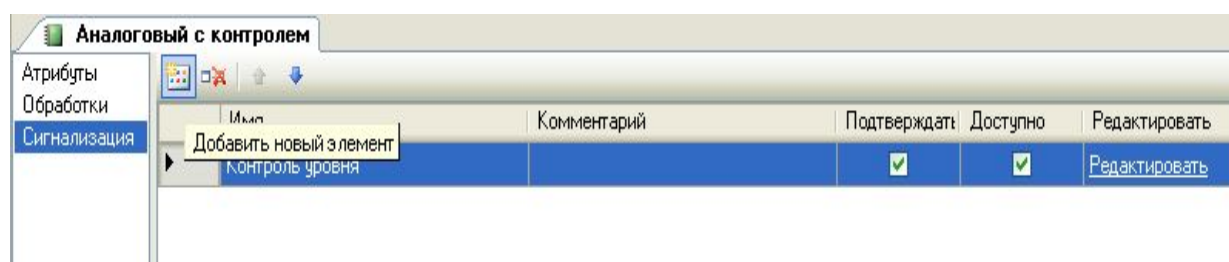
**Сигнализация (тревога, alarm)** – аварийное состояние, т.е. специальный случай какого-либо состояния, требующего повышенного внимания.

В проекте DataRate работа с аварийными состояниями основана на использовании тегов, тип которых включает определенную сигнализацию.

**Как назначить сигнализацию типу тега**

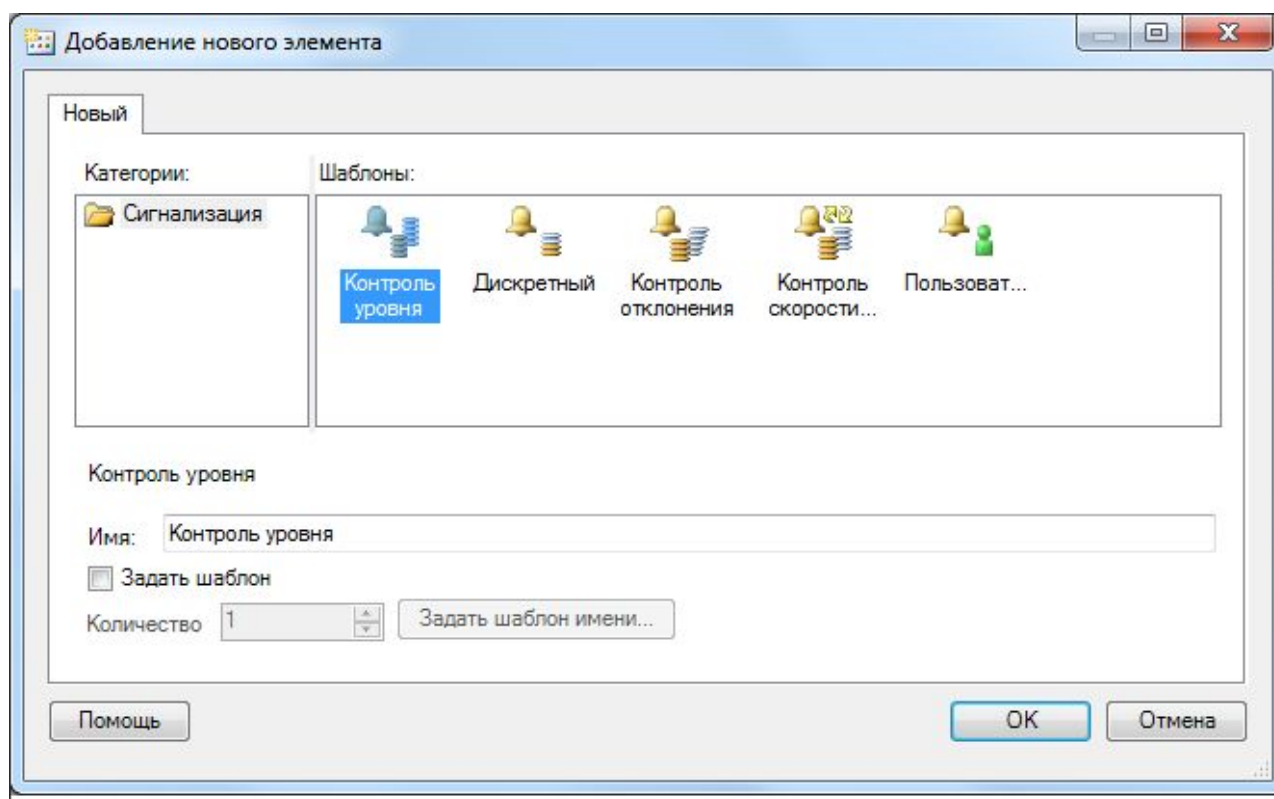
Для добавления сигнализации к типу тега следует:

1. Выбрать элемент **Сигнализация** в списке редактора атрибутов тега (Библиотека типов тегов)



2. Нажать на кнопку 

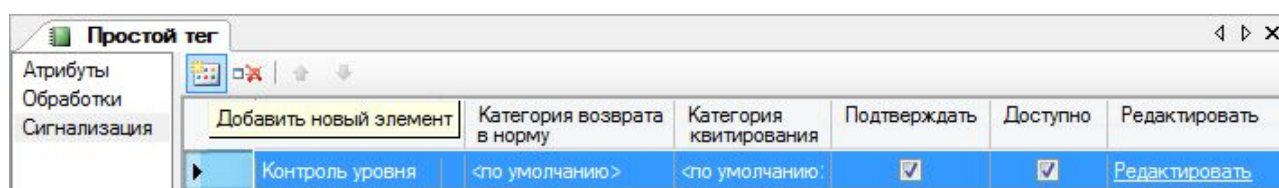
- 3 В появившемся окне **Добавление нового элемента** выбрать нужную сигнализацию.



### Свойства сигнализации

Каждая Сигнализация имеет следующие свойства и поля:

- **Имя** – уникальное имя сигнализации в пределах типа тега
- **Комментарий** – пояснение к сигнализации
- **Сообщение возврата в норму** – сообщение о событии возврата в норму
- **Категория возврата в норму** – указание категории события при возврате в норму
- **Сообщение квитирования** – сообщение о событии квитирования
- **Категория квитирования** – указание категории события при квитировании сигнализации
- **Источник** – источник сигнализации (указанный атрибут тега или полное имя тега, если атрибут не выбран).
- **Подтверждать** – указание, надо ли квитировать данную сигнализацию
- **Доступно** – определение, является ли данная сигнализация доступной
- **Редактировать** – определение сигнальных подсостояний и алгоритмов их наступления



**Сообщения о квитировании и о переходе в норму** можно задать двумя способами:

- через Словарь событий Протокола событий
- непосредственно текстом в окне настроек сигнализации.

Для задания сообщения из словаря событий необходимо ввести через запятую номер события с аргументами в фигурных скобках: 1, {ИмяАтр1}, {ИмяАтр2}.

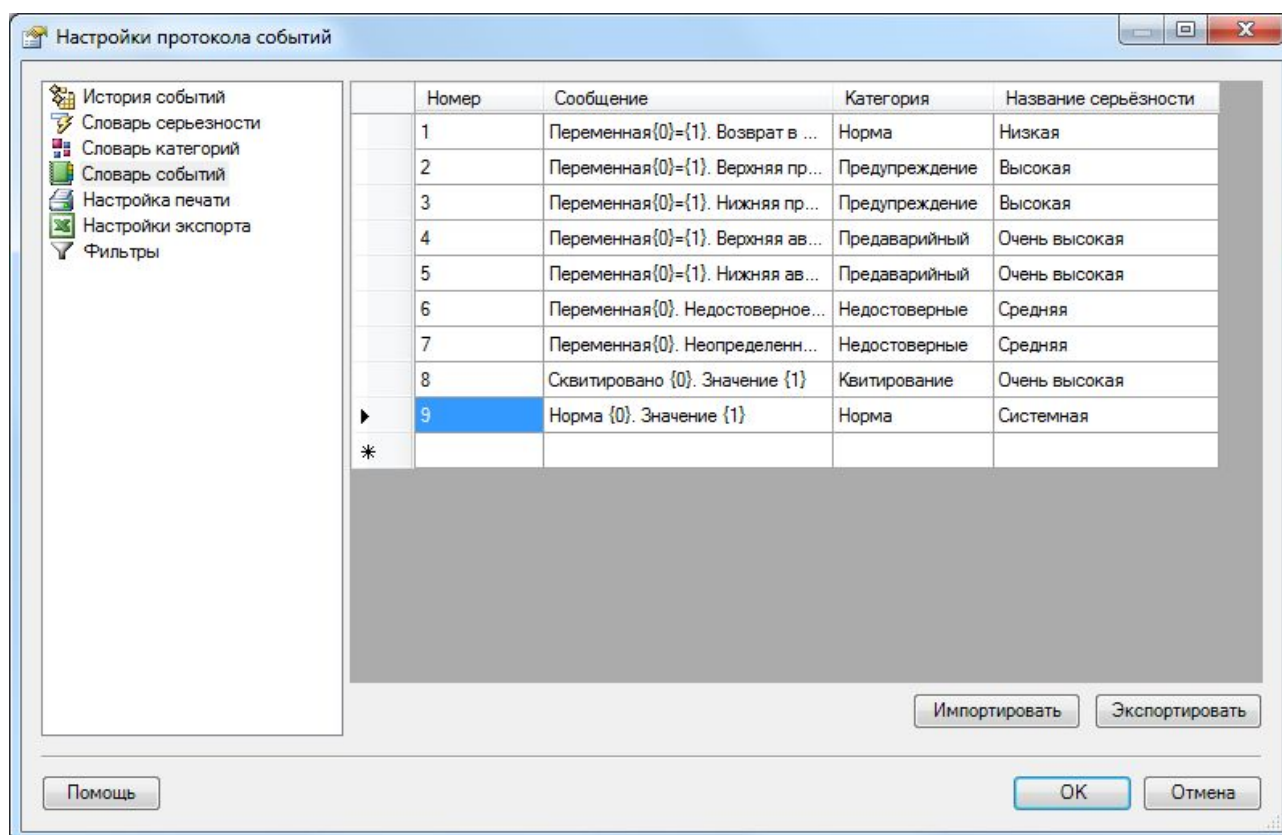
### ВНИМАНИЕ!

Обязательно первым должно быть задано число, далее через запятую аргументы.

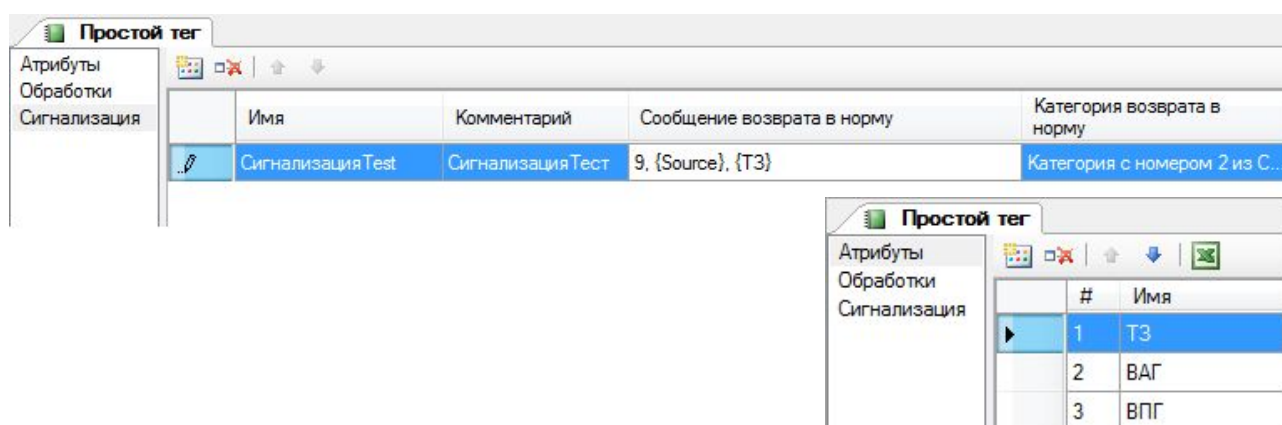
При задании по любой другой схеме текст будет принят за обычную строку и будет помещен в протокол в том виде, как его описали при настройке.

### Пример задания сообщения возврата в норму

- 1 Создаем в словаре событий новое событие с номером 9



- 2 Во вкладке **Сигнализация тега** в поле **Сообщение возврата в норму** описываем сообщение возврата в норму: 9,{Source}, {T3}, и нажимаем на клавишу Ввод (Enter). Атрибуты тега надо писать точно!!





- 3 В результате в поле **Сообщение возврата в норму** будет сформирован следующий текст

Имя	Комментарий	Сообщение возврата в норму	Категория возврата в норму
Сигнализация Test	Сигнализация Test	Событие с номером 3 из Словаря событий с аргументами: Source, T3	Категория с номером 2 из С...

- 4 В протоколе событий при работе системы сообщение возврата в норму будет содержать текст из Словаря сообщений – **Норма** и **Значение** (смотри выше скриншот настройки протокола событий), а также значения атрибутов тега, в примере: значение **Source** – **Давление, T3 – 2**

21.11.2017 14:42:16	Норма Давление. Значение 2	Давление	Хорошее
21.11.2017 14:42:05	ВАГ Давление, значение: 10	Давление	Плохое

Также можно задать произвольный текст сообщения с указанием аргументов в фигурных скобках.

Содержимое поля **Источник** при исполнении попадает в колонку **Источник** протокола событий. В поле **Источник** можно выбрать только атрибут тега, из которого будет взято значение

### Редактирование сигнализации

Для редактирования сигнализации необходимо нажать на надпись **Редактировать**.

#	Имя	Серьезность	Категория	Описание	Доступно	Выражение
1	ВерхняяПредаварийнаяГраница	1000	<по умолч...	Текущее значение превысило порог ...	✓	
2	ВерхняяПредупредительнаяГраница	800	<по умолч...	Текущее значение превысило порог ...	✓	
3	НижняяПредупредительнаяГраница	800	<по умолч...	Текущее значение опустилось ниже ...	✓	
4	НижняяПредаварийнаяГраница	1000	<по умолч...	Текущее значение опустилось ниже ...	✓	

#	Имя	Редактировать
1	Обработка сигнализации	Редактировать

### Характеристики подсостояния

- **Имя** – имя подсостояния, уникальное в пределах сигнализации
- **Серьезность** – серьезность любых уведомлений о событиях, сгенерированных от имени этого подсостояния
- **Категория** – категория события (способ идентификации в протоколе событий)
- **Описание** – текстовая строка, которая будет включена в сообщении о событии, сгенерированного от имени данного подсостояния

- **Доступно** – свойство, определяющее является ли данное подсостояние сигнализации доступным
- **Выражение** – выражение, которое определяет положение, представленное данным подсостоянием.

#### Настройка подсостояния:

**Серьезность** подсостояния может быть задана:

- значением серьезности – от **1** до **1000**
- номером серьезности из Словаря серьезности. Для этого нужно набрать сочетание символов "**№**" + **номер серьезности** из словаря (например, "№3"), при этом в ячейке соответствующего подсостояния появится надпись "**Серьезность с номером <номер> из Словаря серьезности**"
- названием серьезности из словаря серьезности, например "**Высокая**".

**Категория** подсостояния может быть задана:

- строкой названия категории по умолчанию "<по умолчанию>", при этом невозможно настроить цвета сообщений о событиях, они будут стандартными – черные символы текста на белом фоне.
- номером категории события из словаря. Для этого нужно ввести цифрой номер категории из Словаря категорий событий (например, "3"), при этом в ячейке соответствующего подсостояния появится надпись "**Категория с номером <номер> из Словаря категорий**"
- названием категории из словаря категорий событий, например "Предупреждение".

**Описание** события подсостояния может быть задано:

- текстовой строкой, определяющей сообщение о данном подсостоянии
- номером события из Словаря событий. Для этого нужно ввести цифрой номер события из Словаря событий (например "3"), при этом в ячейке соответствующего подсостояния появится надпись "**Событие с номером <номер> из Словаря событий**". Если событие задано из словаря событий, то настройки **Серьезность** и **Категории** событий игнорируются и соответствующие данные берутся из Словаря событий.

#### Характеристики обработки сигнализации

- **Имя** – имя обработки, уникальное в пределах сигнализации

Список обработок сигнализации представляет собой список алгоритмов, по которым рассчитывается состояние сигнализации.

Щелчок на ссылке **Редактировать** в строке обработки сигнализации открывает окно **Редактор скриптов** для редактирования алгоритма обработки.

#### 4.2.1 Установка сигнального подсостояния

Установка сигнального подсостояния производится вызовом функции **SetActiveSubCondition** объекта **Condition**.

Существует несколько вариантов вызова функции **SetActiveSubCondition**, отличающиеся набором необходимых параметров.

#### Вызов функции установки сигнального подсостояния:

- с указанием имени сигнального подсостояния  
`Condition.SetActiveSubCondition(subConditionName);`  
 где **subConditionName** – имя подсостояния сигнализации (текстовая строка).



- с указанием имени сигнального подсостояния и использованием комментария к сигнализации в качестве сообщения в протоколе событий  
`Condition.SetActiveSubCondition(subConditionName, useCommentOfAlarm);`  
где  
**subConditionName** – имя подсостояния сигнализации (текстовая строка);  
**useCommentOfAlarm** – флаг использования комментария к сигнализации в качестве сообщения (булево значение). Если в качестве параметра указать **true**, то в качестве текста сообщения будет использован комментарий к сигнализации. Если указать **false**, то текст в сообщении сформируется как при вызове функции без параметра **useCommentOfAlarm**.
- с указанием имени сигнального подсостояния, источника события и использованием комментария к сигнализации в качестве сообщения в протоколе событий  
`Condition.SetActiveSubCondition( subConditionName, messageSourceId, useCommentOfAlarm);`  
где  
**subConditionName** – имя подсостояния сигнализации (текстовая строка);  
**messageSourceId** – источник события (текстовая строка);  
**useCommentOfAlarm** – флаг использования комментария к сигнализации в качестве сообщения (булево значение). Если в качестве параметра указать **true**, то в качестве текста сообщения будет использован комментарий к сигнализации. Если указать **false**, то текст в сообщении сформируется как при вызове функции без параметра **useCommentOfAlarm**.
- с использованием произвольного текстового сообщения в качестве сообщения в протоколе событий  
`Condition.SetActiveSubCondition(subConditionName, message);`  
где  
**subConditionName** – имя подсостояния сигнализации (текстовая строка);  
**message** – произвольное текстовое сообщение.
- с указанием источника событий и произвольного текстового сообщения в качестве сообщения в протоколе событий  
`Condition.SetActiveSubCondition(subConditionName, messageSourceId, message);`  
где  
**subConditionName** – имя подсостояния сигнализации (текстовая строка);  
**messageSourceId** – источник события (текстовая строка);  
**message** – произвольное текстовое сообщение (текстовая строка).
- с использованием сообщений из словаря событий элемента «Протокол событий»  
`Condition.SetActiveSubCondition(subConditionName, messageId, messageParameters);`  
где  
**subConditionName** – имя подсостояния сигнализации (текстовая строка);  
**messageId** – номер события в словаре событий (числовое значение);  
**messageParameters** – необязательные параметры текстовой строки события (объекты произвольного типа перечисленные через запятую).

- с указанием источника событий сообщения из словаря событий элемента «Протокол событий»:

```
Condition.SetActiveSubCondition(subConditionName, messageSourceId,
messageId, messageParameters);
```

где

**subConditionName** – имя подсостояния сигнализации (текстовая строка);  
**messageSourceId** – источник события (текстовая строка);  
**messageId** – номер события в словаре событий (числовое значение);  
**messageParameters** – необязательные параметры текстовой строки события (объекты произвольного типа перечисленные через запятую).

### ВНИМАНИЕ!!!

Если при установке сигнального подсостояния используется сообщение из словаря событий, то нужно обратить внимание на текст сообщения, т.к. сообщение может требовать передачи ему дополнительных параметров.

Например, для регистрации сообщения **Переменная{0}={1}. Возврат в норму** требуется два дополнительных параметра. Следовательно, правильный способ вызова функции SetActiveSubCondition будет следующий:

```
Condition.SetActiveSubCondition(null,1,"Параметр1","Параметр2");
```

#### 4.2.2 Сигнализация контроля уровня

Шаблон сигнализации **Сигнализация контроля уровня** обеспечивает сравнение текущего значения атрибута тега (ТЗ) с одним или несколькими заданными пределами. Если значение атрибута тега превышает предел, то устанавливается сигнализация.

**Сигнализация контроля уровня** представляет сигнализацию со следующими подсостояниями:

- Нижняя предаварийная граница (НАГ) – минимальный нижний предел (LoLo)
- Нижняя предупредительная граница (НПГ) – нижний предел (Lo)
- Верхняя предупредительная граница (ВПГ) – верхний предел (Hi)
- Верхняя предаварийная граница (ВАГ) – максимальный верхний предел (HiHi)

**Скрипт обработки сигнального состояния:**

```
double КонтролируемоеЗначение = ТЗ;
if (ВАГ <= КонтролируемоеЗначение) {
    Condition.SetActiveSubCondition(ВерхняяПредаварийнаяГраница);
}
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {
    Condition.SetActiveSubCondition(ВерхняяПредупредительнаяГраница);
}
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {
    Condition.SetActiveSubCondition(НижняяПредупредительнаяГраница);
}
else if (КонтролируемоеЗначение <= НАГ) {
    Condition.SetActiveSubCondition(НижняяПредаварийнаяГраница);
}
else {
    Condition.SetActiveSubCondition(null);
}
```

Скрипт обработки сигнального состояния с использованием в качестве сообщения в протоколе событий комментария к сигнализации

```
double КонтролируемоеЗначение = ТЗ;  
if (ВАГ <= КонтролируемоеЗначение) {  
    Condition.SetActiveSubCondition(ВерхняяПредаварийнаяГраница, true);  
}  
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {  
    Condition.SetActiveSubCondition(ВерхняяПредупредительнаяГраница, true);  
}  
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {  
    Condition.SetActiveSubCondition(НижняяПредупредительнаяГраница, true);  
}  
else if (КонтролируемоеЗначение <= НАГ) {  
    Condition.SetActiveSubCondition(НижняяПредаварийнаяГраница, true);  
}  
Else {  
    Condition.SetActiveSubCondition(null, true);  
}
```

Скрипт обработки сигнального состояния с указанием источника события

```
double КонтролируемоеЗначение = ТЗ;  
if (ВАГ <= КонтролируемоеЗначение) {  
    Condition.SetActiveSubCondition(  
        ВерхняяПредаварийнаяГраница, "Объект1.Простой тег", false);  
}  
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {  
    Condition.SetActiveSubCondition(  
        ВерхняяПредупредительнаяГраница, "Объект1.Простой тег", false);  
}  
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {  
    Condition.SetActiveSubCondition(  
        НижняяПредупредительнаяГраница, "Объект1.Простой тег", false);  
}  
else if (КонтролируемоеЗначение <= НАГ) {  
    Condition.SetActiveSubCondition(  
        НижняяПредаварийнаяГраница, "Объект1.Простой тег", false);  
}  
else {  
    Condition.SetActiveSubCondition(null, "Объект1.Простой тег", false);  
}
```

Скрипт обработки сигнального состояния с использованием в качестве сообщения в протоколе событий комментария к сигнализации и указанием источника события

```
double КонтролируемоеЗначение = ТЗ;  
if (ВАГ <= КонтролируемоеЗначение)  
{
```

```

Condition.SetActiveSubCondition(
    ВерхняяПредаварийнаяГраница, "Объект1.Простой тег", true);
}
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ)
{
    Condition.SetActiveSubCondition(
        ВерхняяПредупредительнаяГраница, "Объект1.Простой тег", true);
}
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ)
{
    Condition.SetActiveSubCondition(
        НижняяПредупредительнаяГраница, "Объект1.Простой тег", true);
}
else if (КонтролируемоеЗначение <= НАГ)
{
    Condition.SetActiveSubCondition(
        НижняяПредаварийнаяГраница, "Объект1.Простой тег", true);
}
else
{
    Condition.SetActiveSubCondition(null, "Объект1.Простой тег", true);
}

```

**Скрипт обработки сигнального состояния с использованием в качестве сообщения в протоколе событий произвольного текстового сообщения**

```

double КонтролируемоеЗначение = ТЗ;
if (ВАГ <= КонтролируемоеЗначение) {
    Condition.SetActiveSubCondition(
        ВерхняяПредаварийнаяГраница,
        "Достигнута верхняя предаварийная граница");
}
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {
    Condition.SetActiveSubCondition(
        ВерхняяПредупредительнаяГраница,
        "Достигнута верхняя предупредительная граница");
}
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {
    Condition.SetActiveSubCondition(
        НижняяПредупредительнаяГраница,
        "Достигнута нижняя предупредительная граница");
}
else if (КонтролируемоеЗначение <= НАГ) {
    Condition.SetActiveSubCondition(
        НижняяПредаварийнаяГраница,
        "Достигнута нижняя предаварийная граница");
}

```

```
else {  
    Condition.SetActiveSubCondition(null, "Возврат в нормальное состояние");  
}
```

Скрипт обработки сигнального состояния с указанием источника события и произвольного текстового сообщения

```
double КонтролируемоеЗначение = ТЗ;  
if (ВАГ <= КонтролируемоеЗначение) {  
    Condition.SetActiveSubCondition(  
        ВерхняяПредаварийнаяГраница,  
        "Объект1.Простой тег", "Достигнута верхняя предаварийная граница");  
}  
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {  
    Condition.SetActiveSubCondition(  
        ВерхняяПредупредительнаяГраница,  
        "Объект1.Простой тег", "Достигнута верхняя предупредительная граница");  
}  
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {  
    Condition.SetActiveSubCondition(  
        НижняяПредупредительнаяГраница,  
        "Объект1.Простой тег", "Достигнута нижняя предупредительная граница");  
}  
else if (КонтролируемоеЗначение <= НАГ) {  
    Condition.SetActiveSubCondition(  
        НижняяПредаварийнаяГраница,  
        "Объект1.Простой тег", "Достигнута нижняя предаварийная граница");  
}  
else {  
    Condition.SetActiveSubCondition(  
        null, "Объект1.Простой тег", "Возврат в нормальное состояние");  
}
```

Скрипт обработки сигнального состояния с использованием сообщений из словаря событий элемента "Протокол событий"

```
double КонтролируемоеЗначение = ТЗ;  
if (ВАГ <= КонтролируемоеЗначение)  
{  
    Condition.SetActiveSubCondition(ВерхняяПредаварийнаяГраница, 1);  
}  
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {  
    Condition.SetActiveSubCondition(ВерхняяПредупредительнаяГраница, 2);  
}  
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {  
    Condition.SetActiveSubCondition(НижняяПредупредительнаяГраница, 3);  
}  
else if (КонтролируемоеЗначение <= НАГ) {
```

```

    Condition.SetActiveSubCondition(НижняяПредаварийнаяГраница, 4);
}
else {
    Condition.SetActiveSubCondition(null, 5);
}

```

**Скрипт обработки сигнального состояния с указанием источника события и сообщения из словаря событий элемента "Протокол событий"**

```

double КонтролируемоеЗначение = ТЗ;
if (ВАГ <= КонтролируемоеЗначение) {
    Condition.SetActiveSubCondition(
        ВерхняяПредаварийнаяГраница, "Объект1.Простой тег", 1);
}
else if (ВПГ <= КонтролируемоеЗначение && КонтролируемоеЗначение < ВАГ) {
    Condition.SetActiveSubCondition(
        ВерхняяПредупредительнаяГраница, "Объект1.Простой тег", 2);
}
else if (НАГ < КонтролируемоеЗначение && КонтролируемоеЗначение <= НПГ) {
    Condition.SetActiveSubCondition(
        НижняяПредупредительнаяГраница, "Объект1.Простой тег", 3);
}
else if (КонтролируемоеЗначение <= НАГ) {
    Condition.SetActiveSubCondition(
        НижняяПредаварийнаяГраница, "Объект1.Простой тег", 4);
}
else {
    Condition.SetActiveSubCondition(null, "Объект1.Простой тег", 5);
}

```

#### 4.2.3 Дискретная сигнализация

Шаблон сигнализации **Дискретная сигнализация** указывает, какому значению атрибута тега соответствует сигнализация.

Дискретная сигнализация используется для атрибутов тегов с двумя возможными значениями – истина или ложь и включает одно подсостояние – **Сигнальное**.

В скрипте обработки используются: **ДЗ** – дискретное значение, **СЗ** – сигнальное значение.

**Скрипт обработки сигнального состояния:**

```

if (ДЗ == СЗ) {
    Condition.SetActiveSubCondition(Сигнальное);
}
else {
    Condition.SetActiveSubCondition(null);
}

```



### 4.2.4 Сигнализация отклонения

Шаблон сигнализации **Сигнализация отклонения** обеспечивает вычисление абсолютной величины разности текущего значения тега и целевого значения (**ЦЗ**).

Абсолютная величина разности сравнивается с одним или несколькими пределами, выраженными в процентах от диапазона значений тега.

Сигнализация отклонения представляет сигнализацию с двумя подсостояниями: **Малое отклонение (МО)**, **Значительное отклонение (ЗО)**.

Имеется возможность индивидуально задавать значения и приоритеты для предела малого и значительного отклонения, а также указывать для каждого предела, требуется ли его использовать.

**Скрипт обработки сигнального состояния:**

```
double КонтролируемоеЗначение = ТЗ;
double НижнийПределМО = ЦЗ - Диапазон*МО / 100;
double ВерхнийПределМО = ЦЗ + Диапазон*МО / 100;
double НижнийПределЗО = ЦЗ - Диапазон*ЗО / 100;
double ВерхнийПределЗО = ЦЗ + Диапазон*ЗО / 100;

if ((ВерхнийПределЗО > КонтролируемоеЗначение &&
    КонтролируемоеЗначение >= ВерхнийПределМО) ||
    (НижнийПределЗО < КонтролируемоеЗначение &&
    КонтролируемоеЗначение <= НижнийПределМО)) {
    Condition.SetActiveSubCondition(МалоеОтклонение);
}
else if (ВерхнийПределЗО <= КонтролируемоеЗначение ||
    КонтролируемоеЗначение <= НижнийПределЗО) {
    Condition.SetActiveSubCondition(ЗначительноеОтклонение);
}
else {
    Condition.SetActiveSubCondition(null);
}
```

### 4.2.5 Пользовательская сигнализация

Шаблон сигнализации **Пользовательская сигнализация** служит для создания сигнализации для конкретных требований пользователя.

Для создания сигнализации необходимо добавить атрибуты тега и написать скрипт по обработке сигнального состояния.

### 4.2.6 Сигнализация скорости изменения

Шаблон сигнализации **Сигнализация скорости изменения** вычисляет скорость изменения значения атрибута тега за определенный промежуток времени.

Сигнализация скорости изменения представляет сигнализацию с двумя подсостояниями: **Медленное изменение (СМИ)**, **Быстрое изменение (СБИ)**.

**Скрипт обработки сигнального состояния:**

```
if (AttributeName != "ТЗ") return;
System.TimeSpan ВременнойПромежуток = NewValue.LastChangingDateTime –
```

```

OldValue.LastChangingDateTime;

double КоличествоЕдиницВремени = 0;
// ЕВ – единица измерения времени
if (ЕВ == "Часы") {
    КоличествоЕдиницВремени = ВременнойПромежуток.TotalHours;
}
else if (ЕВ == "Минуты") {
    КоличествоЕдиницВремени = ВременнойПромежуток.TotalMinutes;
}
else if (ЕВ == "Секунды") {
    КоличествоЕдиницВремени = ВременнойПромежуток.TotalSeconds;
}
else if (ЕВ == "Миллисекунды") {
    КоличествоЕдиницВремени = ВременнойПромежуток.TotalMilliseconds;
}
else {
    КоличествоЕдиницВремени = ВременнойПромежуток.TotalSeconds;
}
double РеальнаяСкоростьИзменения = Math.Abs(NewValue.AsDouble –
OldValue.AsDouble)/КоличествоЕдиницВремени;
if (СМИ <= РеальнаяСкоростьИзменения && РеальнаяСкоростьИзменения < СБИ) {
    Condition.SetActiveSubCondition(МедленноеИзменение);
}
else if (СБИ <= РеальнаяСкоростьИзменения) {
    Condition.SetActiveSubCondition(БыстроеИзменение);
}
else {
    Condition.SetActiveSubCondition(null);
}

```

#### 4.2.7 Квитирование состояния

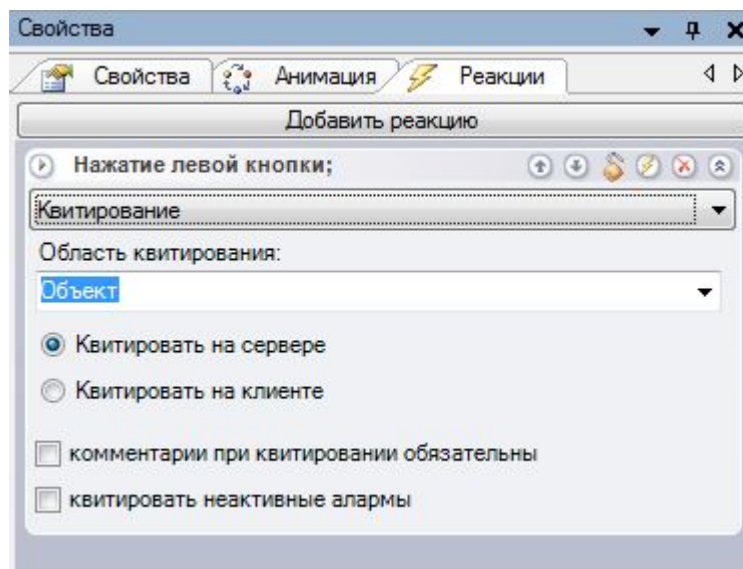
**Квитированием** называется подтверждение состояния.

Квитирование может иметь четыре области видимости:

- **Объект** – квитируются активные сигнальные состояния объекта
- **Мнемосхема** – квитируются активные сигнальные состояния, которые привязаны к свойствам графических примитивов мнемосхемы
- **Рабочий стол** – квитируются все мнемосхемы на рабочем столе.
- **Примитив** – квитируются активные сигнальные состояния, свойства которых привязаны к аргументам примитива.

### Как назначить квитирование

Для задания квитирования необходимо в свойствах графического примитива на вкладке **Реакция** на любую реакцию выставить тип реакции **Квитирование** и выбрать область видимости.



На уровне реакции поддерживается ограничение доступа.

Если установлен признак **Комментарии при квитировании обязательны** (поставлена галочка), квитировать состояние можно только после ввода комментария и указания пользователя, квитировавшего состояние.

Если установлен признак **Квитировать неактивные алармы** (поставлена галочка), то все неактивные алармы, сохраненные в Журнале тревог, будут сквитированы,

Квитирование может происходить как на сервере, так и на клиенте. Возможны следующие варианты алгоритма визуализации:

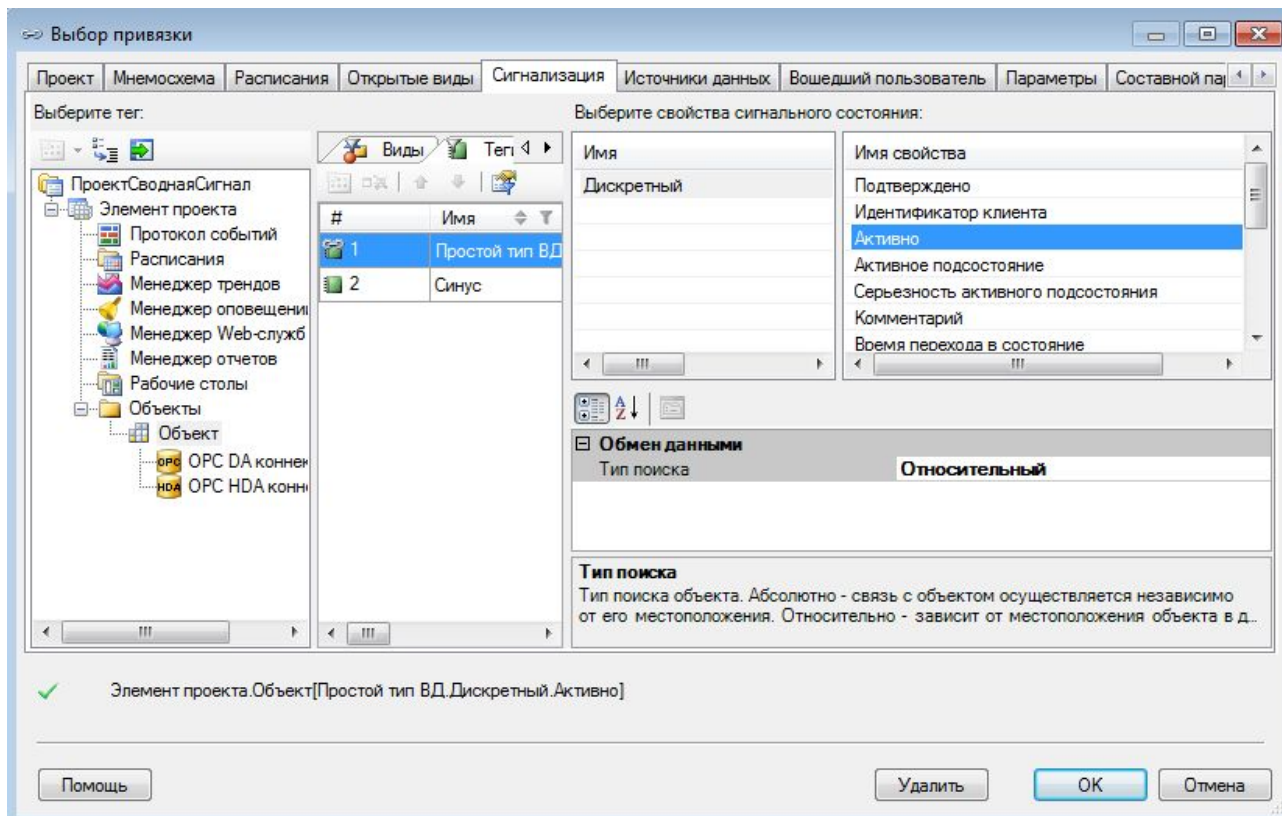
- **Серверное квитирование** происходит на сервере. Все клиенты, которым разрешена визуализация сквитированных сигнализаций, получают соответствующую визуализацию. При этом регистрируется событие с идентификатором клиента, который осуществил квитирование.
- **Клиентское квитирование** происходит только на том клиенте, который сквитировал сигнализацию. Визуализация сигнализации данного клиента отображает это изменение. Для других клиентов никаких изменений визуализации не происходит. При этом регистрируется событие о локальном квитировании сигнализации с идентификатором клиента, который осуществил квитирование.
- **Квитирование сигнализации пользователем**, вошедшим на сервер, всегда приводит к серверному квитированию не зависимо от настройки алгоритма квитирования.

#### 4.2.8 Сводное сигнальное состояние. Привязка к сигнализации

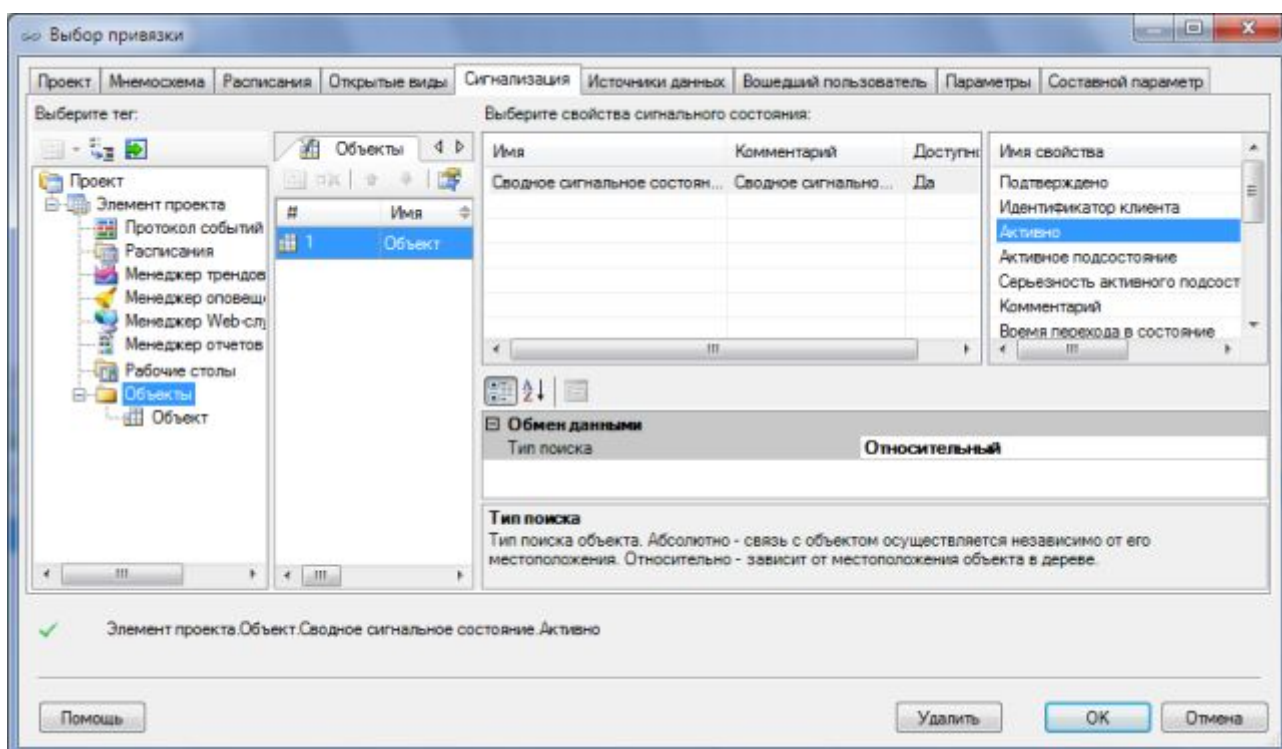
**Сводное сигнальное состояние** — это обобщенная сигнализация для объекта или вида объекта.

Окно **Выбор привязки** позволяет создать связь с одним из свойств **Сводного сигнального состояния**:

- тега



- объекта или мнемосхемы.

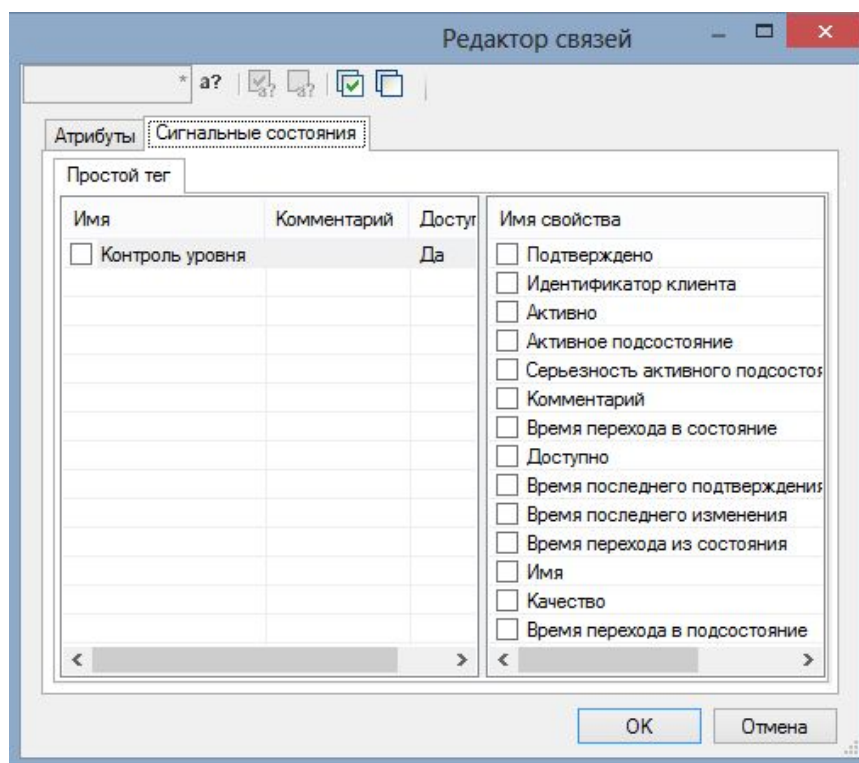


### ВНИМАНИЕ!!!

Осуществить привязку можно с помощью операции Drag&Drop.

Для этого:

- 1 Перетащите тег на вид Мнемосхема или в окно Аргументы
- 2 В открывшемся окне Редактор связей перейдите на вкладку Сигнальные состояния и выберите необходимые свойства.



### 4.2.9 Доступ к сигнализации

Сигнализация поддерживает два действия, доступ к которым можно ограничить:

- **Квитирование сигнализации.** Доступ к квитированию сигнализации может быть настроен на уровнях папки и объекта
- **Визуализация сигнализации.** Доступ к визуализации сигнализации настраивается на уровнях папки, объекта, тега, отдельной сигнализации.

### ВНИМАНИЕ!!!

Если для роли Пользователя визуализация сигнализации запрещена, а квитирование разрешено, то сквитировать эту сигнализацию Пользователь не сможет.

Настройка доступа к квитированию никак не влияет на доступ к визуализации.



## Свойства сигнализации, сводного сигнального состояния объекта

Имя	Группа событий	Значение по умолчанию	Описание
Имя	Информация о сигнализации		Имя сигнализации/сводного сигнального состояния. Задаётся в настройках сигнализации. Для сводного сигнального состояния формируется из полного имени объекта
Доступно		true	Доступность сигнализации: true – сигнализация работает, false – не работает. Задаётся в настройках сигнализации. Если значение равно false, то остальные свойства (кроме имени) заполняться не будут.
Качество		0	Качество основных элементов данных, на которых основано данное состояние
Время последнего изменения		01.01.0001 4:00:00	Время последнего изменения в свойствах сигнального состояния
Активно	Сигнальное состояние	false	Свойство, характеризующее активность сигнализации. Показывает, находится ли объект в сигнальном состоянии или нет. Значение: true — контролируемое значение находится в сигнальном диапазоне, false — контролируемое значение вернулось в норму. Для сводного сигнального состояния свойство имеет значение true, если имеется хотя бы одна активная сигнализация, иначе — false.
Время перехода в состояние		01.01.0001 4:00:00	Время перехода в сигнальное состояние. Значение: время, в которое сигнализация стала активной. У сводного сигнального состояния в это свойство попадает соответствующее время первой сигнализации, которая стала активной (значение свойства меняется, когда все сигнализации неактивны, и одна из них перешла в активное состояние).
Время перехода из состояния		01.01.0001 4:00:00	Время возврата в состояние нормы. Значение: время, в которое сигнализация стала неактивной. У сводного сигнального состояния в это свойство попадает соответствующее время последней из сигнализаций, которая стала неактивной (значение свойства меняется, когда все сигнализации, кроме одной, неактивны, и затем эта сигнализация также переходит в неактивное состояние).

## ОБЪЕКТЫ И СЕРВИСЫ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Имя	Группа событий	Значение по умолчанию	Описание
Активное подсостояние	Сигнальное подсостояние		Имя активного подсостояния. Значение: имя текущего активного подсостояния сигнализации. У сводного сигнального состояния в это свойство попадает значение одного из сигнальных подсостояний сигнализаций с учётом их серьёзности. Приоритетом обладает наиболее серьёзное подсостояние. Если серьёзность одинаковая, то это первая из сработавших сигнализаций.
Время перехода в подсостояние		01.01.0001 4:00:00	Время последнего входа в подсостояние. У сводного состояния в это свойство попадает значение соответствующего свойства сигнализации, значение свойства "Активное подсостояние" которой попало в сводное сигнальное состояние.
Серьёзность активного подсостояния		0	Серьёзность активного подсостояния (от 0 до 1000). Задаётся в настройках сигнализации. У сводного состояния в это свойство попадает значение соответствующего свойства сигнализации, значение свойства "Активное подсостояние" которой попало в сводное сигнальное состояние.
Подтверждено	Квитирование сигнализации	true	True – при квитировании, False – при смене подсостояния. У сводного сигнального состояния это свойство принимает значение True, если сквитированы все активные сигнализации, иначе (если имеется хотя бы одна не квитированная сигнализация) значение False.
Время последнего подтверждения		01.01.0001 4:00:00	Время последнего квитирования. У сводного сигнального состояния в это свойство попадает время квитирования самого сводного сигнального состояния или время квитирования последней из не квитированных сигнализаций, если сигнализации квитируются отдельно.
Идентификатор клиента			Имя пользователя, совершившего квитирование. Имя пользователя ПК, если нет вошедших пользователей проекта. У сводного сигнального состояния в это свойство попадает идентификатор клиента, квитировавшего само сводное сигнальное состояние, или идентификатор клиента последней из не квитированных сигнализаций, если сигнализации квитируются отдельно.
Комментарий			Комментарий, вводимый пользователем при квитировании. У сводного сигнального состояния в это свойство попадает комментарий клиента, квитировавшего само сводное сигнальное состояние, или комментарий клиента последней из не квитированных сигнализаций, если сигнализации квитируются отдельно.

#### 4.2.10 Пример сигнализации в проекте

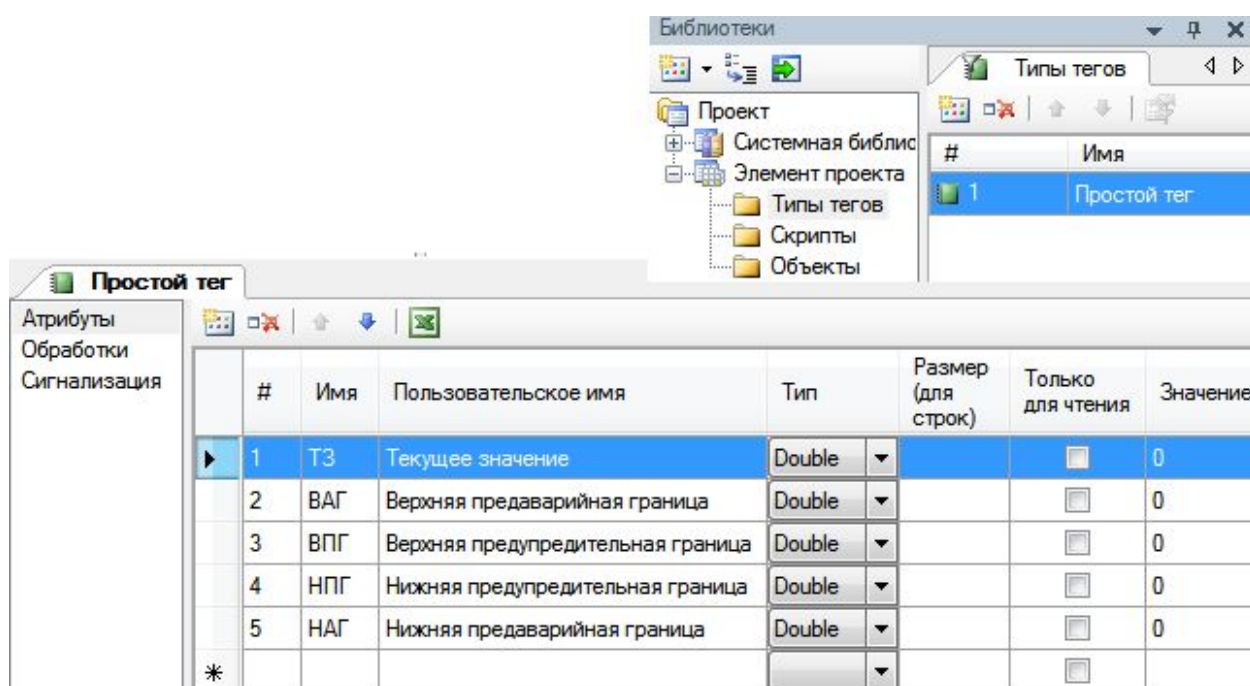
В рассматриваемом примере требуется осуществить мониторинг некоторого параметра с заданными границами сигнализации и квитирование сигнализации при выходе параметра за эти границы.

В примере используется сигнализация **Контроль уровня** и функция реакции **Квитирование**.

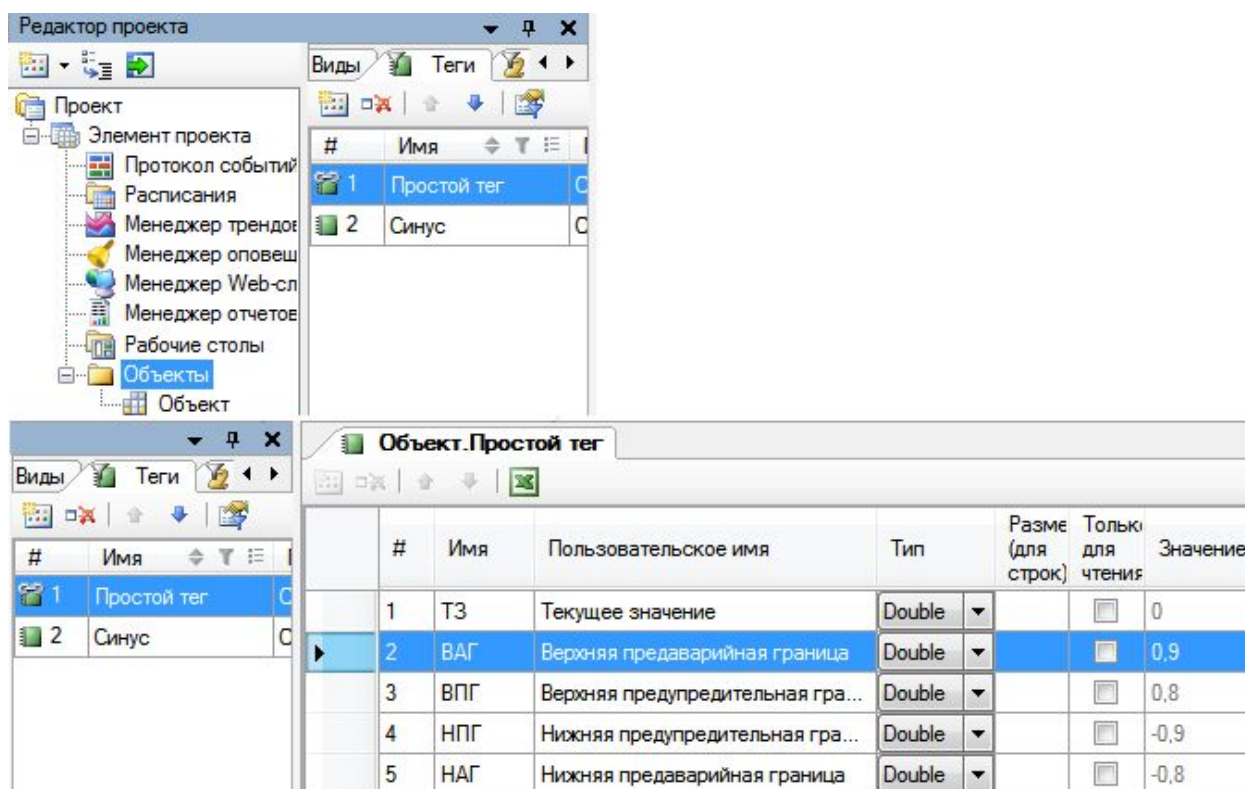
В качестве наблюдаемого параметра в проекте используется атрибут **Value** тега **Синус**.

Для создания соответствующего проекта следует:

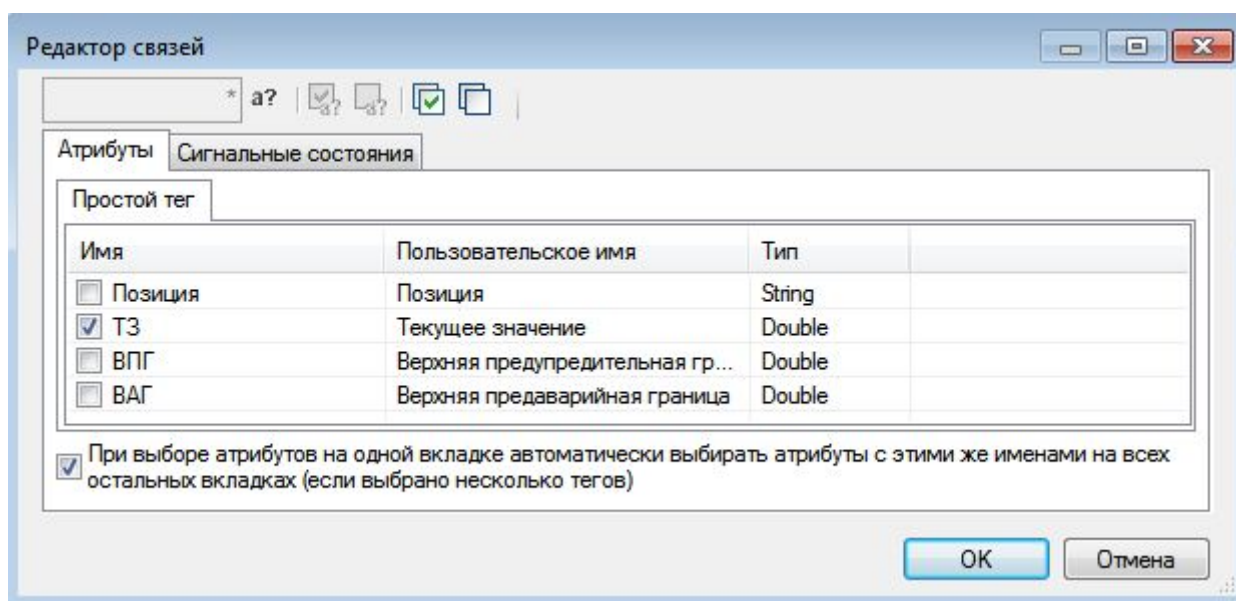
- 1 Создать проект с одним объектом **Объект1**
- 2 Создать в библиотеке тип тега **Простой тег**
- 3 Добавить сигнализацию **Контроль уровня** для типа тега **Простой тег**



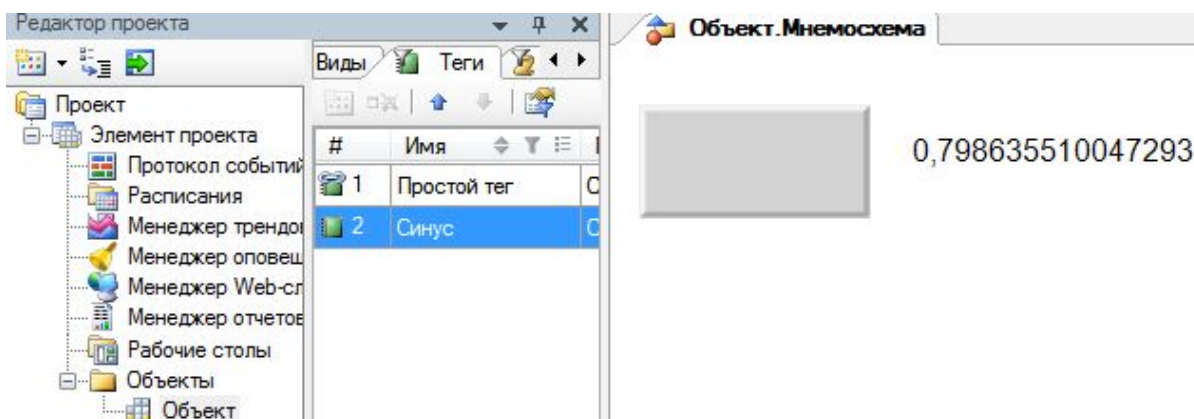
- 4 Создать в объекте тег **Синус**
- 5 Создать в объекте тег **Простой тег**, на основе библиотечного типа **Простой тег** (просто перетащите из библиотеки тип тега **Простой тег** в объект)
- 6 Задать значения границ сигнализации (**НАГ**, **ВАГ**, **НПГ** и **ВПГ**) для тега **Простой тег**. Для этого в редакторе проекта для объекта **Объект1** нужно выбрать вкладку **Теги**, указать **Простой тег** и выбрать в контекстном меню тега пункт **Редактировать** (или просто дважды щелкнуть мышью по наименованию тега)



- 7 Привязать тег **Простой тег** (атрибут **ТЗ**) к тегу **Синус** (атрибут **Value**)
- 8 Создать вид **Мнемосхема1** и поместить на мнемосхему **Объемный прямоугольник**
- 9 Для контроля вывести значение атрибута **ТЗ** тега **Простой тег** на мнемосхему:
  - Выделить в Редакторе проекта тег **Простой тег** и перетащить его на мнемосхему.
  - В появившемся окне Редактор связей на вкладке **Атрибуты/Простой тег** выделить атрибут **ТЗ**. В результате на мнемосхеме создастся примитив **Текст**.

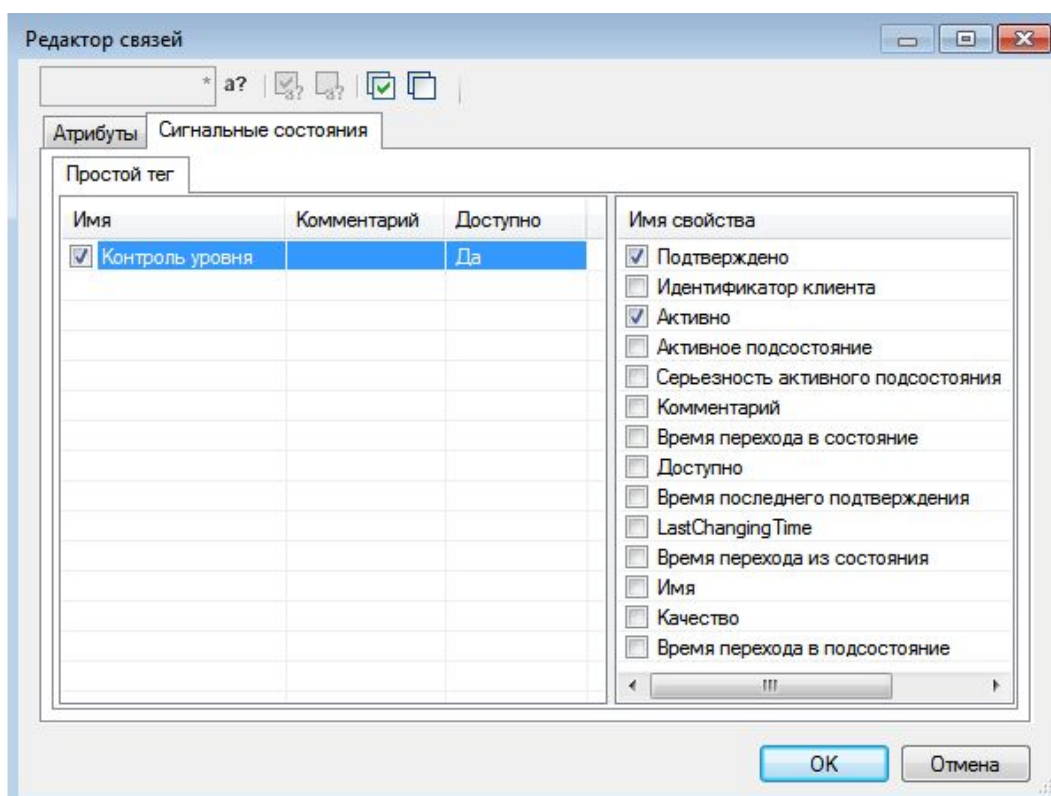


- 10 Протестировать созданный проект. Примитив **Текст** отображает текущее значение атрибута **ТЗ** (значение синуса) тега **Простой тег**.



11 Для отображения смены сигнального состояния задать анимацию **Мигание** для объемного прямоугольника:

- Выделить прямоугольник на мнемосхеме, и перетащить из окна редактора проекта тег **Простой тег** в окно **Аргументы** прямоугольника
- В появившемся окне **Редактор связей** на вкладке **Сигнальные состояния** выделить свойства сигнализации **Подтверждено** и **Активно**.

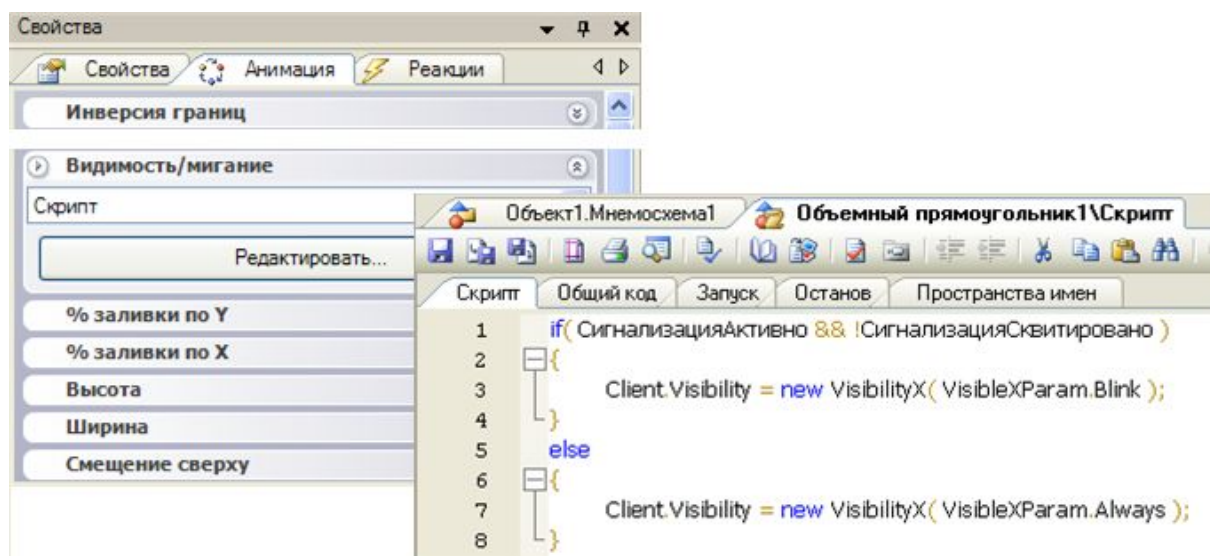


- Задать созданным аргументам **Acked** и **Active** соответствующие наименования в скрипте: **СигнализацияСквтировано** и **СигнализацияАктивно**.

Аргументы - Объемный прямоугольник1				
	Имя в скрипте	Имя	Тип	Привязка
✓	СигнализацияСквтировано	Acked	Boolean	Элемент проекта.Объект1Простой тег.Контроль уровня...
✓	СигнализацияАктивно	Active	Boolean	Элемент проекта.Объект1Простой тег.Контроль уровня...



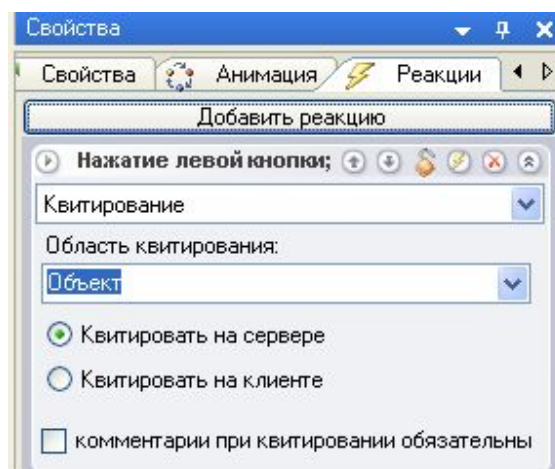
- Настроить анимацию. На вкладке **Анимация** выбрать свойство **Видимость/Мигание** и задать **скрипт** (текст скрипта на рисунке)



Если состояние сигнализации будет активным и не сквитированным, примитив **Объемный прямоугольник** будет мигать. В любом другом случае мигания не будет.

### 12 Настроить квитирование:

- Выделить примитив **Объемный прямоугольник** на мнемосхеме и перейти на вкладку **Реакции** в окне свойств
- Добавить реакцию **Квитирование** на событие **Нажатие левой кнопки мыши**.



### Проект готов!

Сохраните и запустите его на выполнение.

При смене сигнального состояния – выходе значения **T3** тега **Простой тег** за границы сигнализации – прямоугольник начнет мигать. Щелчком левой кнопки мыши на прямоугольнике сигнальное состояние будет сквитировано, и мигание прекратится до следующей смены сигнального состояния.

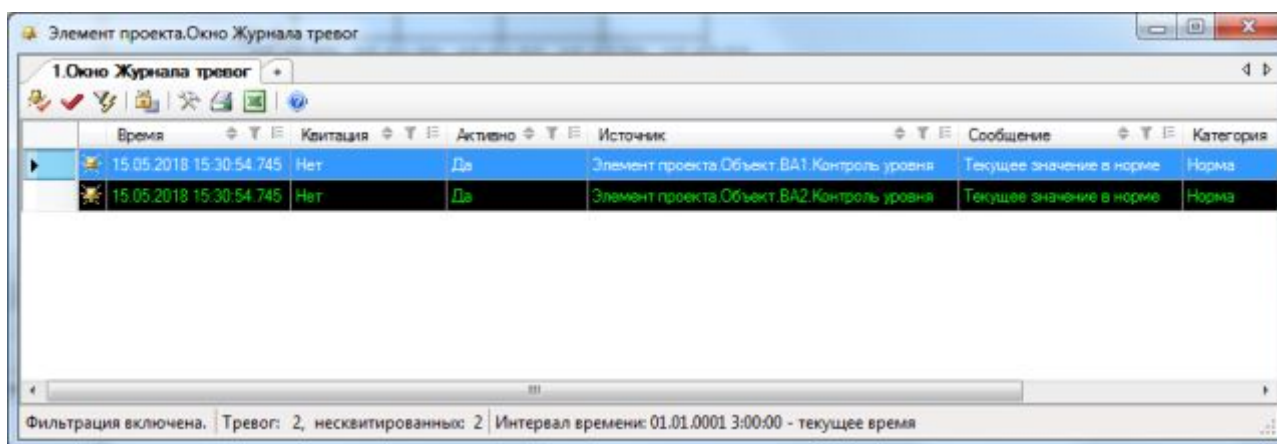
### 4.2.11 Журнал тревог

**Журнал тревог** – это инструмент диспетчера по оценке текущего состояния системы и квитированию актуальных событий.

**Окно Журнала тревог** содержит следующую информацию:

- **метка времени появления тревоги;**
- **метка времени квитирования;**
- **информация об объекте,** к которому относится нарушение (диспетчерское наименование оборудования);

- **описание нарушения** – текст с описанием возникшего нарушения. Формируется автоматически (скриптом) из атрибутов тега параметра, значение которого вызвало нарушение;
- **текущее состояние нарушения (активно/неактивно);**
- **текущее состояние квитирования (квитировано/не квитировано);**
- **категория события;**
- **группа события.**



Элементы управления:

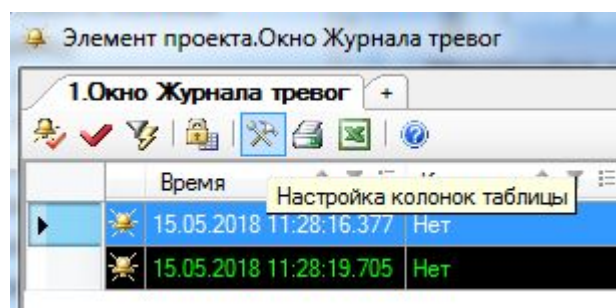
- Квитировать
- Квитировать все видимые тревоги
- Настройка промежутка времени
- "Заморозка" приема тревог
- Настройка колонок таблицы
- Печать событий
- Экспорт в файл
- Помощь

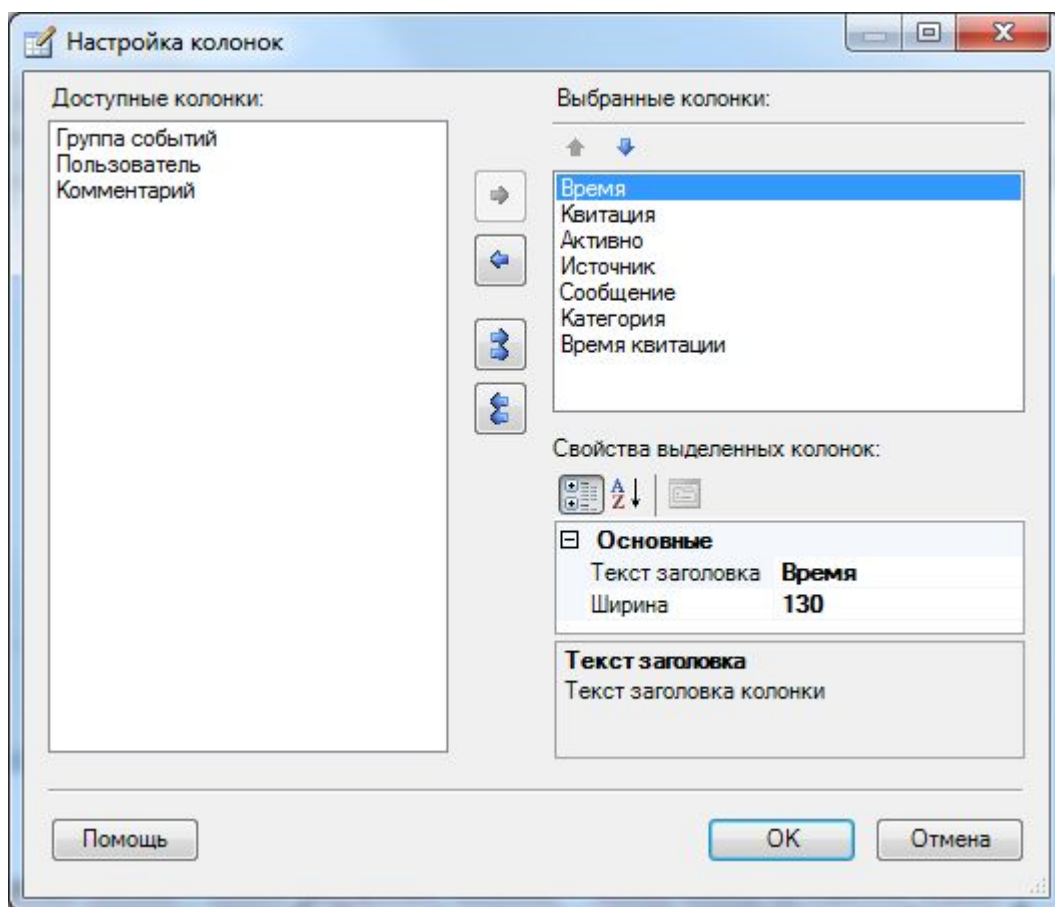
### Настройка колонок журнала тревог

Окно **Настройка колонок** вызывается кнопкой **Настройка колонок таблицы** на панели инструментов окна **Окно Журнала тревог**.

Для **добавления** колонок нужно выделить нужные колонки в списке **Доступные колонки** и нажать на кнопку со стрелкой вправо.

Для **удаления** колонок нужно выделить ненужные колонки в списке **Выбранные колонки** и нажать на кнопку со стрелкой влево.





Стрелки (вверх/вниз) на панели инструментов над списком выбранных колонок позволяют настроить порядок следования колонок.

Свойства **Текст заголовка** и **Ширина**, выделенных колонок, в списке **Выбранные колонки**, можно редактировать в панели **Свойства выделенных колонок**.

Для подтверждения сделанных изменений нужно нажать на кнопку **ОК**, для отмены – на кнопку **Отмена**. Закрытие окна при помощи кнопки в заголовке также отменяет сделанные изменения.

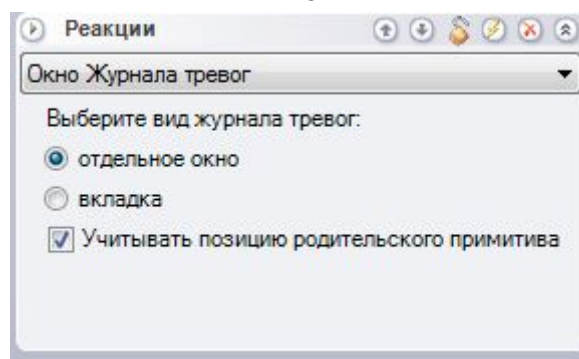
Для вызова справки по выбранной настройке нажмите кнопку **Помощь**.

## Вызов журнала тревог

Вызов Журнала тревог осуществляется с помощью реакции **Окно Журнала тревог**.

Для того чтобы получить доступ к **Журналу тревог** следует:

- 1 В свойствах графического примитива на вкладке **Реакции** на любую реакцию выставить тип реакции **Окно Журнала тревог**
- 2 Выбрать необходимый вид журнала (отдельное окно или вкладка).



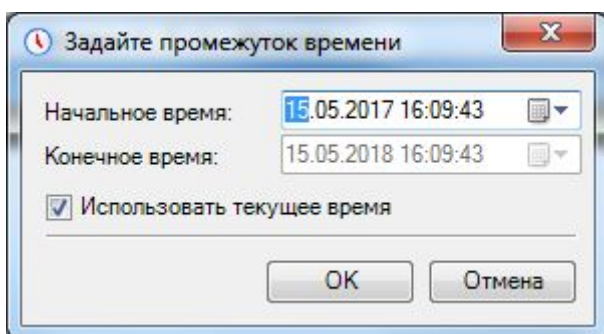
## Задание промежутка времени

**Промежуток времени** – это интервал выборки событий в окно **Журнала тревог**.

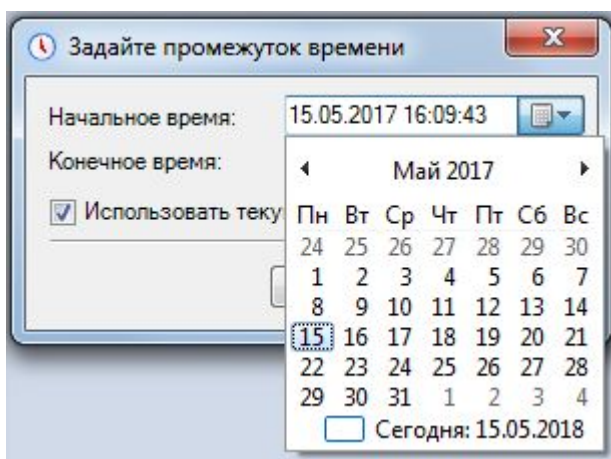
Для настройки промежутка следует:

- 1 В окне **Журнал тревог** нажать на кнопку **Настройка промежутка времени**

- 2 В открывшемся окне **Задайте промежуток времени** нажмите на стрелку в строке **Начальное время**, а затем – **Конечное время**

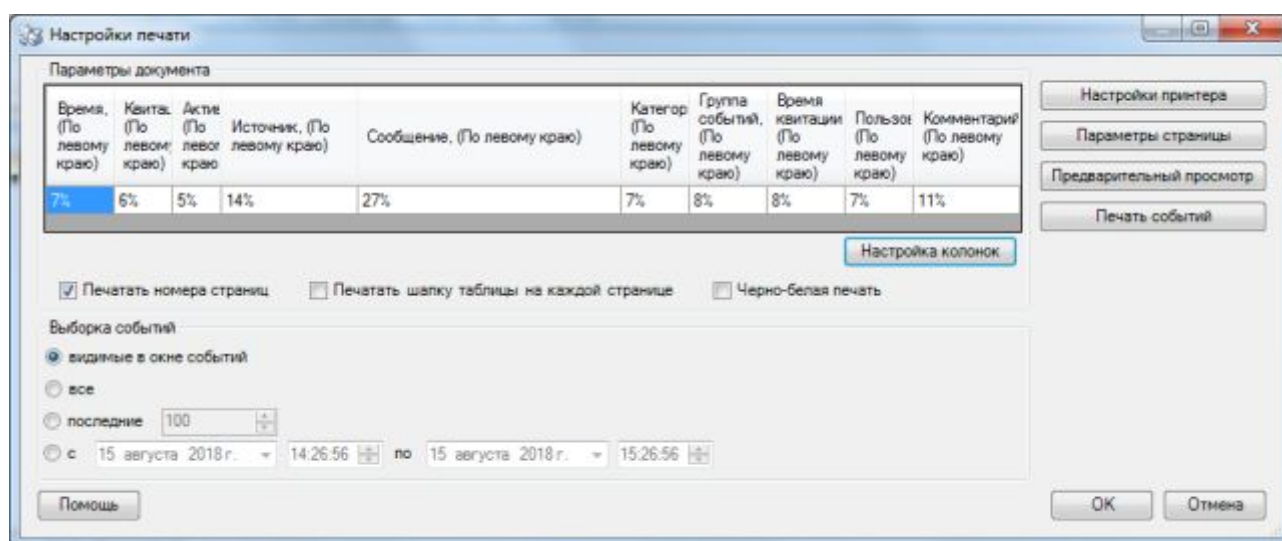


- 3 Выберите начальное и конечное время с помощью выпадающего календаря.



#### 4.2.11.1 Печать журнала тревог

Окно **Настройки печати** вызывается при нажатии на кнопку **Печать событий** на панели инструментов окна Журнала тревог.



#### Параметры документа

Группа настроек **Параметры документа** предназначена для настройки отображения выбранных колонок документа и их параметров.

В заголовке каждой колонки отображается имя колонки и выравнивание.



В ячейке таблицы отображается ширина колонки в процентах. При изменении ширины колонки автоматически пересчитывается ширина всех колонок. Ширину колонки можно изменить путем ввода с клавиатуры значения в ячейку.

Для выбора колонок и задания выравнивания нужно щелкнуть по кнопке **Настройка колонок** и в появившемся окне Настройка колонок для печати выбрать колонки и задать выравнивание текста при печати.

- **Печатать номер страницы** – задает вывод номера страницы при печати.
- **Печатать шапку таблицы на каждой странице** – задает печать шапки таблицы на каждой странице.
- **Черно-белая печать** – задание печати с использованием оттенков серого цвета.

### Выборка событий

Группа настроек **Выборка событий** предназначена для выбора **способа формирования выборки событий** для печати. Можно указать один из четырех способов выборки событий для печати:

- **Видимые в окне событий** – события, видимые в окне событий на момент печати или предварительного просмотра.
- **Все** – все события, зарегистрированные в протоколе событий.
- **Последние [количество]** – последние события протокола событий. Количество событий задается в поле, которое становится доступным при выборе этого способа.
- **С [начальная дата, время] по [конечная дата, время]** – события за заданный промежуток времени. Временной интервал задается в элементах управления выбора даты и времени, которые становятся доступны при выборе этого способа.

**Фильтрация** при выборке событий задается в соответствии с **настройками фильтров** в окне Протокол событий.

### Элементы управления

Для управления настройками печати протокола событий используются следующие кнопки:

- **Настройки принтера** – вызывает стандартное диалоговое окно **Настройки принтера**.
- **Параметры страницы** – вызывает стандартное диалоговое окно **Параметры страницы**.
- **Предварительный просмотр** – вызывает стандартное диалоговое окно **Предварительный просмотр**.
- **Печать событий** – печатает события в соответствии с заданными настройками.
- **ОК** – сохраняет текущие **настройки печати** и закрывает диалоговое окно. При закрытии окна Протокол событий **настройки печати сбрасываются**.
- **Отмена** – отменяет все сделанные изменения и закрывает диалоговое окно.

### Методы экспорта сообщений о тревогах

Для печати протокола событий из скрипта используются специальные методы класса **AlarmServiceHelper**.

Для того чтобы методы этого класса стали доступны, в редакторе скриптов во вкладке **Пространства имен** добавьте ссылку на **Krug.ObjectModel.AlarmServiceInterfaces**

Печать журнала тревог в заданном промежутке времени с использованием принтера, заданного в системе по умолчанию:

**AlarmServiceHelper.PrintAlarms(object client, DateTime startTime, DateTime endTime)**  
где



**client** – клиент скрипта;  
**startTime** – начальное время запрашиваемого интервала;  
**endTime** – конечное время запрашиваемого интервала;

Печать всех тревог за все время с использованием принтера, заданного в системе по умолчанию:

**AlarmServiceHelper.PrintAlarms(object client)**

### Пример

Для иллюстрации использования методов экспорта следует:

- 1 Разместить на мнемосхеме объемный прямоугольник с текстом **Печать** и настроить его реакцию **Скрипт** на нажатие левой кнопки мыши
- 2 Отредактировать текст скрипта

### ВНИМАНИЕ!


Проверьте, что на вкладке **Пространства имён** добавлены:  
**Krug.ObjectModel.AlarmServiceInterfaces**

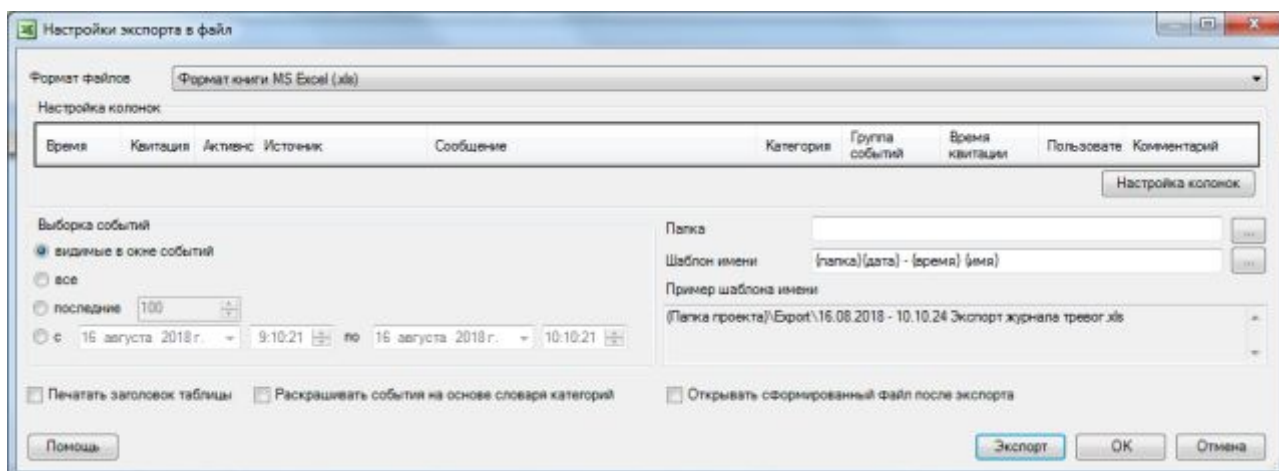
- 3 Сохранить изменения и запустить проект на выполнение
- 4 Для печати журнала тревог нажать на кнопку **Печать**  
 В результате сформируется печатный документ

#### 4.2.11.2 Экспорт журнала тревог

Экспорт **Журнала тревог** позволяет сохранить события **Журнала тревог** в файле формата **xls**, **xml** или **csv**.

Для экспорта следует:

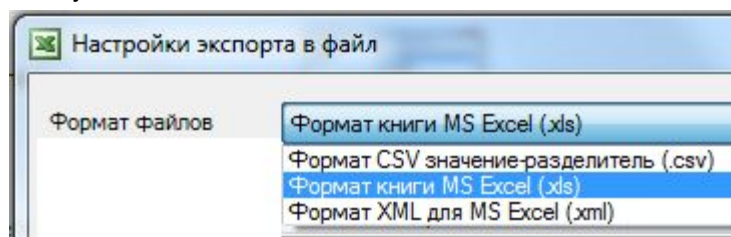
- 1 Нажать на кнопку **Экспорт в файл**  на панели инструментов в Окне Журнала тревог
- 2 В открывшемся окне **Настройки экспорта в файл** задать необходимые параметры



## Выбор формата файла экспорта

Для выбора формата файла экспорта следует:

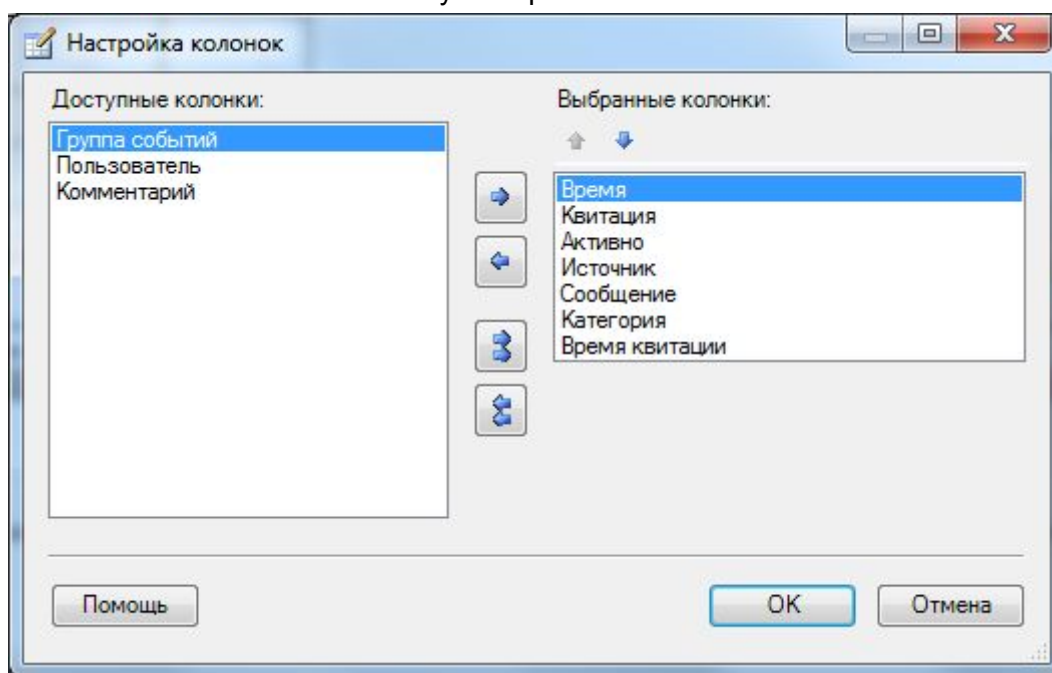
- 1 В поле Формат раскрыть (нажать на стрелку в конце поля) список доступных форматов
- 2 Выбрать в списке требуемый формат файла



## Выбор колонок

Для задания колонок в экспортируемом файле следует:

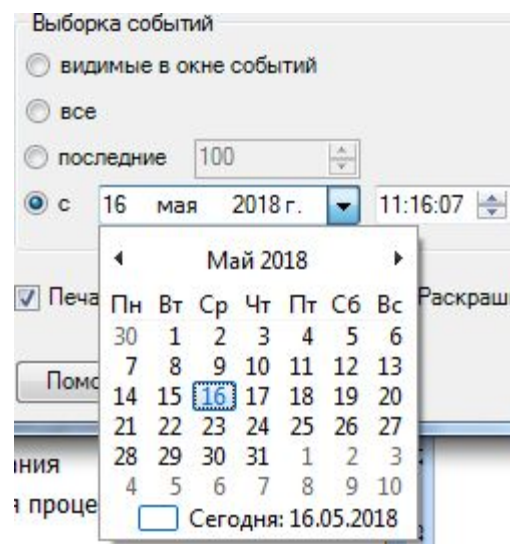
- 1 Нажать на кнопку **Настройка колонок** в окне Настройки экспорта в файл
- 2 Добавить/удалить колонки в список **Выбранные колонки**.  
Для добавления колонок нужно выделить нужные колонки в списке **Доступные колонки** и нажать на кнопку со стрелкой вправо.  
Для удаления колонок нужно выделить ненужные колонки в списке **Выбранные колонки** и нажать на кнопку со стрелкой влево.



## Выбор событий

Для выбора событий, включаемых в файл экспорта, следует:

- 1 В переключателе **Выборка событий** выбрать один из способов выборки: **видимые в окне событий**, **все**, **последние**, **с... по ...**.
- 2 Настроить параметры выборки, если необходимо.  
В случае выборки событий на интервале (выборка **с... по ...**) воспользуйтесь выпадающим календарем.



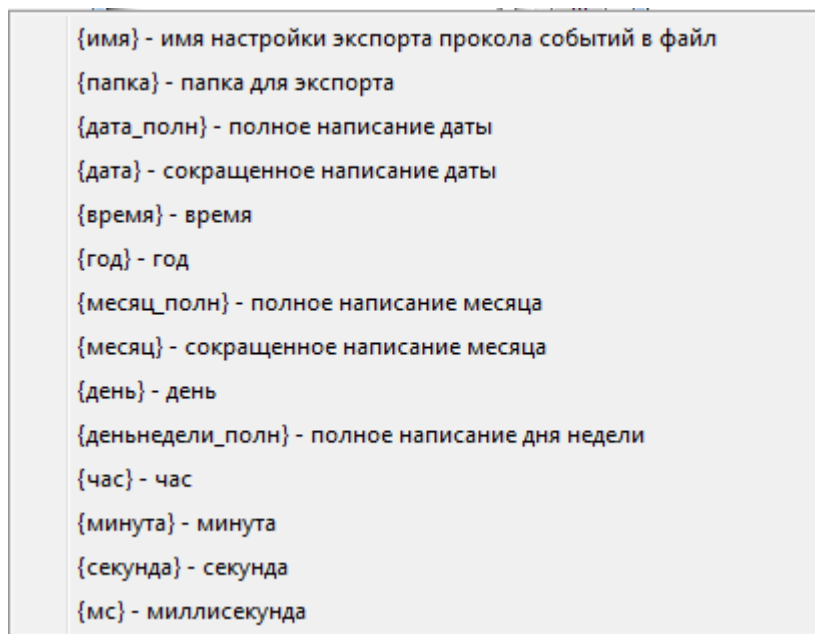
### Выбор папки и настройка шаблона имени файла экспорта

Для выбора папки сохранения файла экспорта следует:

- 1 Нажать на кнопку в строке **Папка**
- 2 В открывшемся стандартном окне **Обзор папок** указать требуемую папку.

Для задания шаблона имени файла экспорта следует:

- 1 Нажать на кнопку в строке **Шаблон имени**
- 2 В открывшемся списке **компонент Шаблона имени** указать необходимый компонент



Пример сохраненного журнала тревог в файле .xls приведен ниже.

	A	B	C	D	E	F	G
	Время	Квитация	Активно	Источник	Сообщение	Категория	Время квитации
2	16.05.2018 15:11:34.063	Нет	Да	Элемент проекта Объект ВА1 Контроль уровня	Текущее значение в норме	Норма	
3	16.05.2018 15:11:34.063	Нет	Да	Элемент проекта Объект ВА2 Контроль уровня	Текущее значение в норме	Норма	
4							
5							
6							

#### 4.2.11.3 Методы экспорта журнала тревог из скрипта

Для экспорта протокола событий из скрипта используются специальные методы класса **AlarmServiceHelper**.

Для того чтобы методы этого класса стали доступны, в редакторе скриптов во вкладке **Пространства имен** добавьте ссылку на **Krug.ObjectModel.AlarmServiceInterfaces**.

### Экспорт тревоги в заданном промежутке времени

`AlarmServiceHelper.ExportAlarms(object client, DateTime startTime, DateTime endTime, string ExportPath, ExportFileFormats format)`

где

**client** – клиент скрипта;  
**startTime** – начальное время запрашиваемого интервала;  
**endTime** – конечное время запрашиваемого интервала;  
**exportPath** – путь к каталогу экспорта  
**format** – формат экспортируемого файла;

### Экспорт всех тревог за все время

`AlarmServiceHelper.ExportAlarms(object client, string ExportPath, ExportFileFormats format)`

где

**client** – клиент скрипта;  
**exportPath** – путь к каталогу экспорта  
**format** – формат экспортируемого файла;

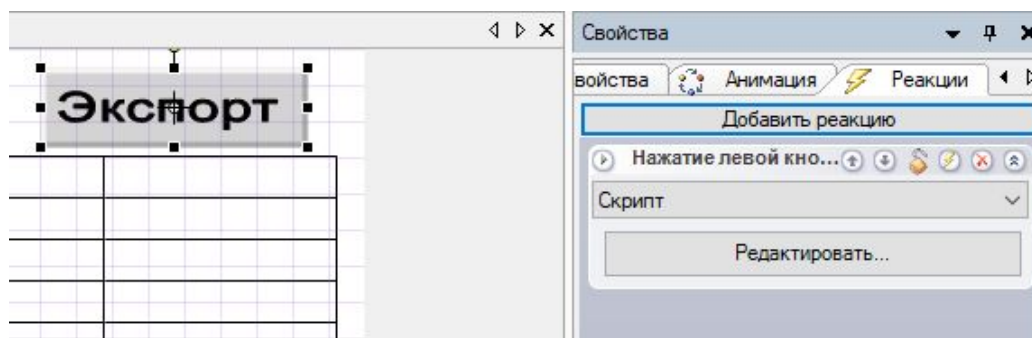
### Экспорт всех тревог в рабочий каталог проекта в формате xls

`AlarmServiceHelper.ExportAlarms(object client)`

### Пример

Для демонстрации работы функций экспорта следует::

- 1 Разместить на мнемосхеме объемный прямоугольник с текстом **Экспорт** и настроим его реакцию **Скрипт** на нажатие левой кнопки мыши



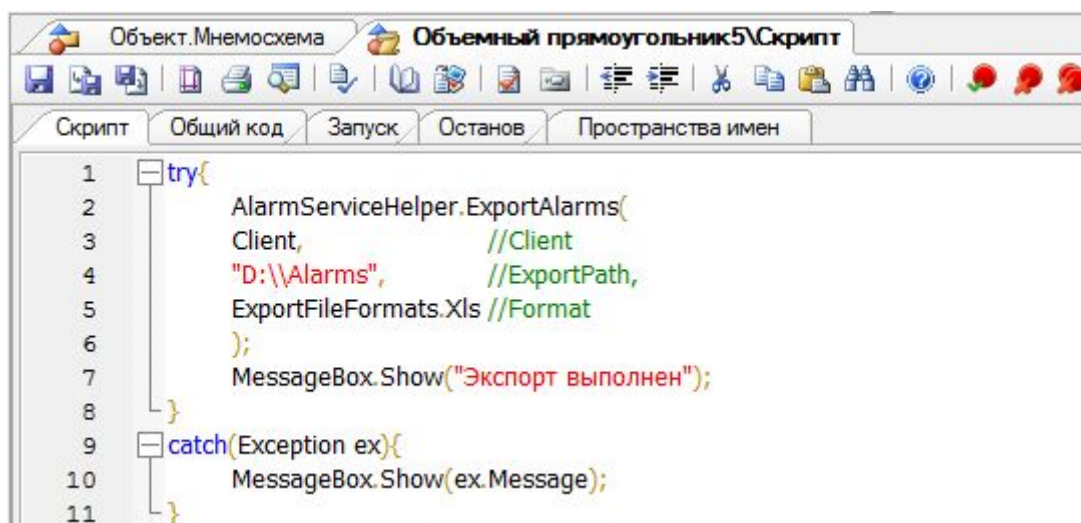
- 2 Отредактировать текст скрипта

#### ВНИМАНИЕ!

Проверьте, что на вкладке Пространства имён добавлено `Krug.ObjectModel.AlarmServiceInterfaces`, а также существует папка `Export` по пути, указанном в скрипте.

- 3 Сохранить изменения и запустить проект на выполнение
- 4 Для экспорта Протокола событий нажать на кнопку **Экспорт**



**ВНИМАНИЕ!**

Данный скрипт можно запускать и по **расписанию**.

- 5 В результате сформируется файл – **d:\Alarms\Export\17.07.2018 - 10.45.40 .xls**. Дата и время в наименовании файла соответствуют моменту нажатия на кнопку **Экспорт**.

1	2	3	4
Время	Время квитации	Источник	Сообщение
04.06.2018 16:29:58.563		Элемент проекта.Объект.Простой тег.Контроль уровня	Переменная авп = 150. Верхняя аварийная граница
04.06.2018 16:29:58.562		Элемент проекта.Объект.Простой тег.Контроль уровня1	Переменная авп = 150. Верхняя аварийная граница
04.06.2018 16:18:31.277		Элемент проекта.Объект.Простой тег.Контроль уровня	Переменная авп = 500. Верхняя аварийная граница
04.06.2018 16:18:31.276		Элемент проекта.Объект.Простой тег.Контроль уровня1	Переменная авп = 500. Верхняя аварийная граница
04.06.2018 16:18:03.587		Элемент проекта.Объект.Простой тег.Контроль уровня	Переменная авп = 15. Нижняя аварийная граница
04.06.2018 16:18:03.585		Элемент проекта.Объект.Простой тег.Контроль уровня1	Переменная авп = 15. Нижняя аварийная граница
04.06.2018 16:15:02.633		Элемент проекта.Объект.Простой тег.Контроль уровня	Переменная авп = 500. Верхняя аварийная граница
04.06.2018 16:15:02.618		Элемент проекта.Объект.Простой тег.Контроль уровня1	Переменная авп = 500. Верхняя аварийная граница
04.06.2018 15:05:35.598		Элемент проекта.Объект.Простой тег.Контроль уровня	Переменная авп = 123. Верхняя аварийная граница
04.06.2018 15:05:35.597		Элемент проекта.Объект.Простой тег.Контроль уровня1	Переменная авп = 123. Верхняя аварийная граница
22.05.2018 11:41:44.755		Элемент проекта.Объект.Простой тег.Контроль уровня	Переменная авп = 150. Верхняя аварийная граница
22.05.2018 11:41:44.754		Элемент проекта.Объект.Простой тег.Контроль уровня1	Переменная авп = 150. Верхняя аварийная граница

#### 4.2.11.4 Методы получения журнала тревог из скрипта

Методы обеспечивают получения из скрипта по проекту, отдельному объекту или тегу объекта следующих данных журнала тревог:

- **всего тревог**
- **всего активных тревог**
- **максимальное значение серьезности активных тревог**

- всего несквитированных тревог
- всего неактивных несквитированных тревог

### По проекту

#### Всего тревог

[GetFullAlarmsCount\(\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetFullAlarmsCount().ToString();
```

#### Всего активных тревог

[GetActiveAlarmsCount\(\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetActiveAlarmsCount().ToString();
```

#### Максимальное значение серьезности активных тревог

[GetMaxSeverityActiveAlarms\(\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetMaxSeverityActiveAlarms().ToString();
```

#### Всего несквитированных тревог

[GetFullUnAckedAlarmsCount\(\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetFullUnAckedAlarmsCount().ToString();
```

#### Всего неактивных несквитированных тревог

[GetUnActiveAlarmsCount\(\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetUnActiveAlarmsCount().ToString();
```

### По объекту

#### Всего активных тревог по объекту

[GetActiveAlarmsCount\(объект\)](#)

Пример использования в скрипте:

```
// Получение браузера объектной модели
```

```
ObjectModelBrowser ob = ObjectModelBrowser.CreateOnClient(Client);
```

```
// Имя текущего объекта
```

```
Trace.WriteLine(ob.Node.FullName);
```

```
Client.Text = ActiveAlarmHelper.GetActiveAlarmsCount(ob.Node).ToString();
```

#### Максимальное значение серьезности активных тревог по объекту

[GetMaxSeverityActiveAlarms\(объект\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetMaxSeverityActiveAlarms(ob.Node).ToString();
```

#### Всего несквитированных тревог по объекту

[GetFullUnAckedAlarmsCount\(объект\)](#)

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetFullUnAckedAlarmsCount(ob.Node).ToString();
```



**Всего неактивных несквитированных тревог по объекту**

`GetUnActiveAlarmsCount(объект)`

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetUnActiveAlarmsCount(ob.Node).ToString();
```

#### По тегу

**Всего активных тревог по тегу объекта**

`GetActiveAlarmsCount(объект, имя_тега)`

Пример использования в скрипте:

```
// Получение браузера объектной модели
ObjectModelBrowser ob = ObjectModelBrowser.CreateOnClient(Client);
// Имя текущего объекта
Trace.WriteLine(ob.Node.FullName);
Client.Text = ActiveAlarmHelper.GetActiveAlarmsCount(ob.Node,
    "Простой тег").ToString();
```

**Максимальное значение серьезности активных тревог по тегу объекта**

`GetMaxSeverityActiveAlarms(объект, имя_тега)`

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetMaxSeverityActiveAlarms(ob.Node,
    "Простой тег").ToString();
```

**Всего несквитированных тревог по тегу объекта**

`GetFullUnAckedAlarmsCount(объект, имя_тега)`

Пример использования в скрипте:

```
Client.Text = ActiveAlarmHelper.GetFullUnAckedAlarmsCount(ob.Node,
    "Простой тег").ToString();
```

**Всего неактивных несквитированных тревог по тегу объекта**

`GetUnActiveAlarmsCount(объект, имя_тега)`

Пример использования в скрипте:

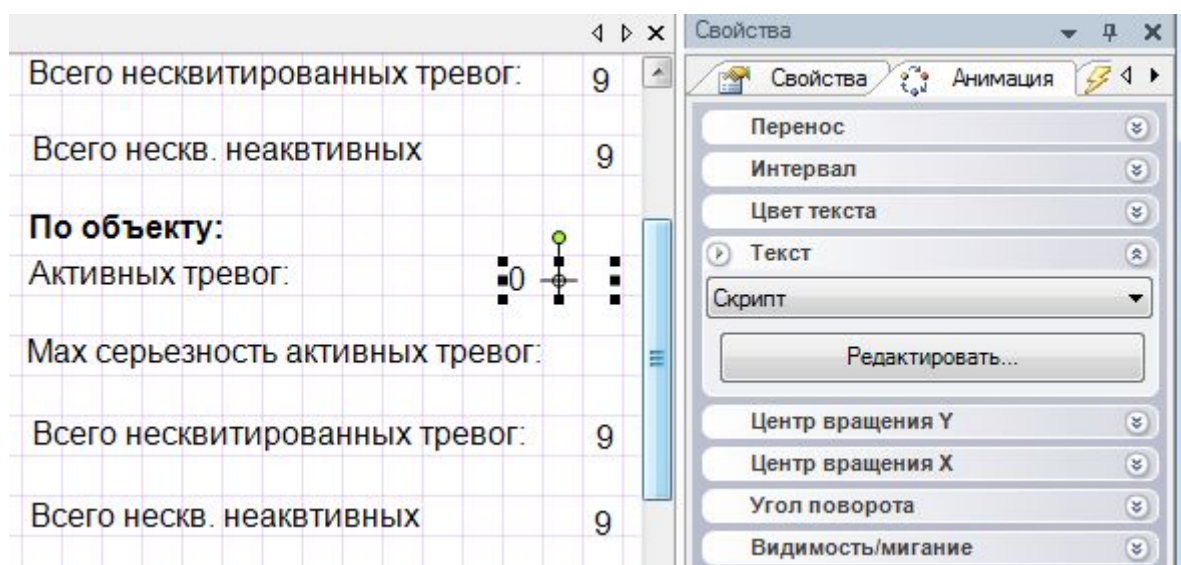
```
Client.Text = ActiveAlarmHelper.GetUnActiveAlarmsCount(ob.Node,
    "Простой тег").ToString();
```

#### Пример использования функций работы с журналом тревог

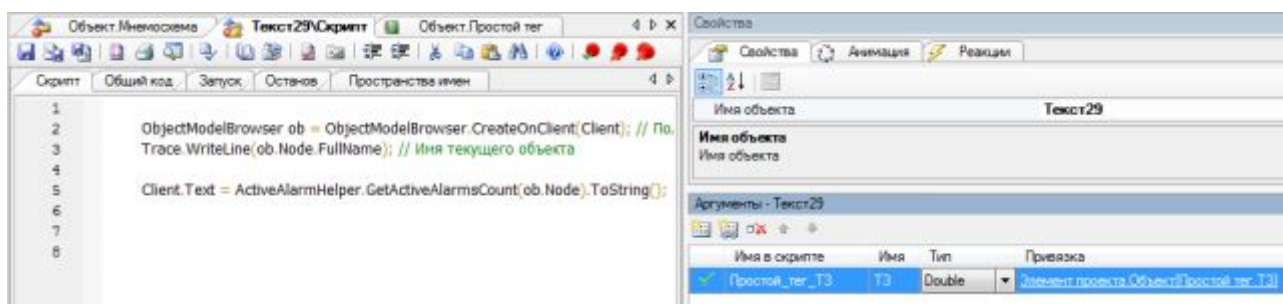
Для отображения на мнемосхеме объекта данных **Журнала тревог** в примере используется анимация **Скрипт** на свойство **Текст** графического примитива **Текст**.

Для отображения данных следует:

- 1 Открыть мнемосхему объекта **Объект** и разместить на ней примитив **Текст**
- 2 В окне свойств примитива открыть закладку **Анимация** и выбрать анимацию **Скрипт** на свойство **Текст**



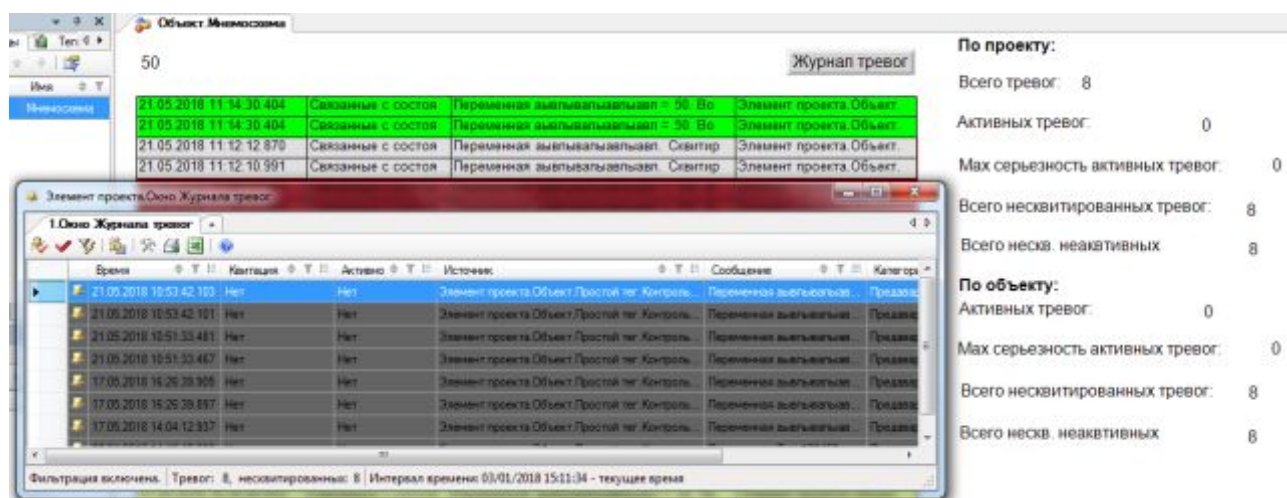
- 3 Нажать на кнопку **Редактировать** и добавить в закладке **Скрипт** код скрипта, содержащий вызов функции `GetActiveAlarmsCount()` – Всего тревог по объекту



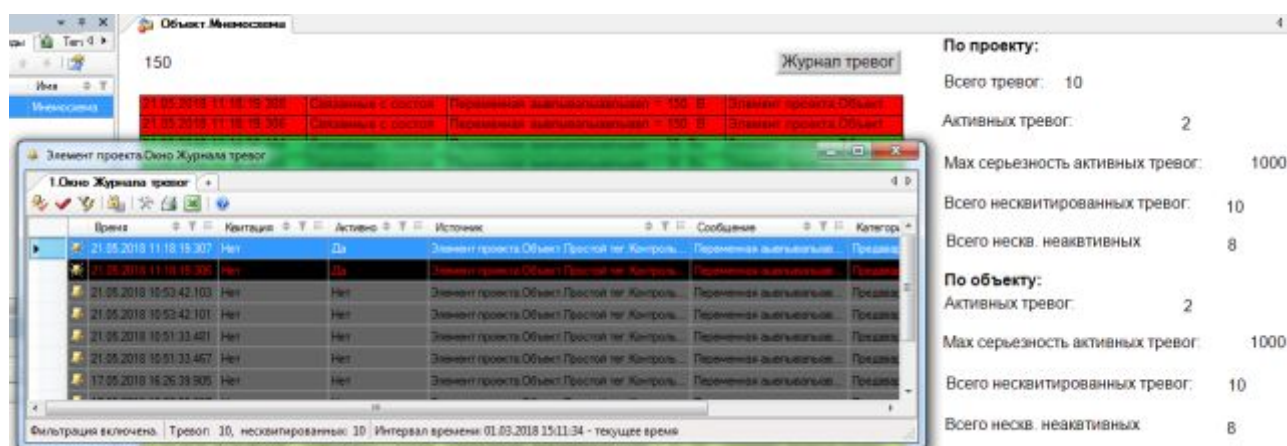
Аргументом скрипта является тег **Простой тег** объекта **Объект**. По изменению текущего значения (**T3**) этого тега будет срабатывать скрипт

	#	Имя	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение
▶	1	T3	Текущее значение	Double		<input type="checkbox"/>	50
	2	ВАГ	Верхняя предаварийная граница	Double		<input type="checkbox"/>	90
	3	ВПГ	Верхняя предупредительная граница	Double		<input type="checkbox"/>	80
	4	НПГ	Нижняя предупредительная граница	Double		<input type="checkbox"/>	30
	5	НАГ	Нижняя предаварийная граница	Double		<input type="checkbox"/>	20
	6	Позиция	Позиция	String	256	<input type="checkbox"/>	авп

- 4 Запустить проект на исполнение. При текущем значении тега **Простой тег**, не выходящего за предупредительные и предаварийные границы (на скриншоте – 50), количество активных тревог – 0.



- 5 Изменить текущее значение так, чтобы оно вышло за предупредительные и предаварийные границы (на скриншоте – **150**). Количество активных тревог отлично от 0 (на скриншоте – **2**)



Общее количество несквитированных тревог определяется значением промежутка времени, настроенным в **Журнале тревог**, и изменением текущего значения.

#### 4.2.12 Окно активной сигнализации

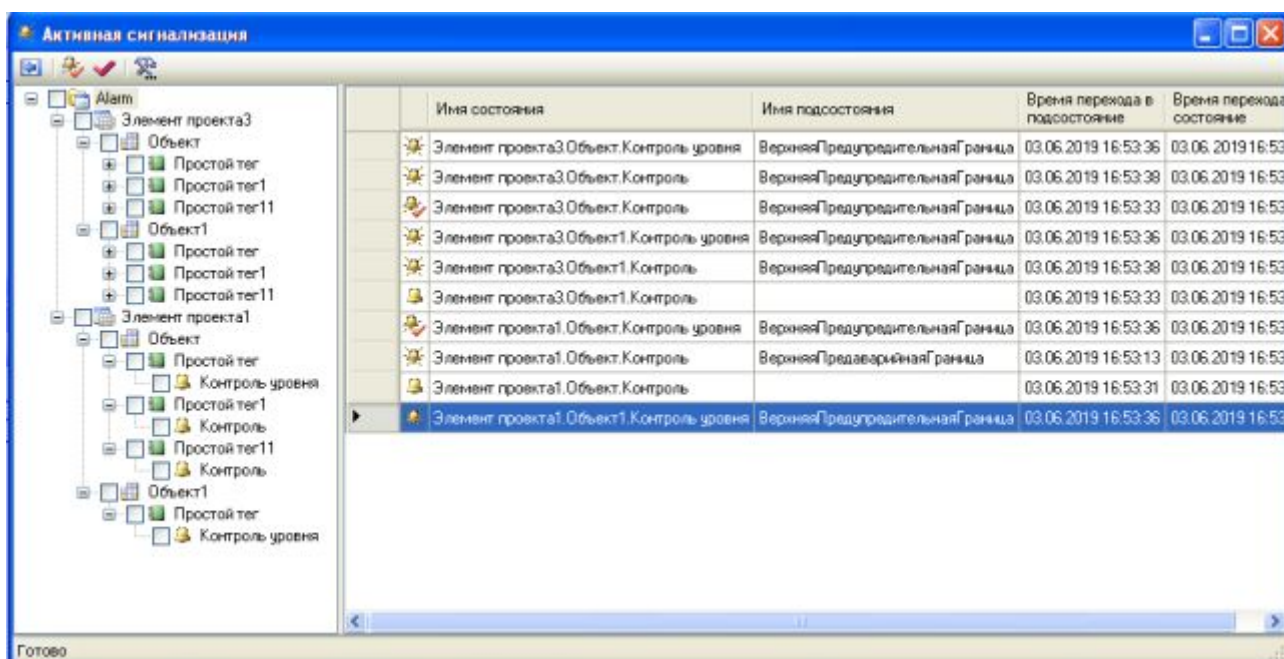
Окно **Активная сигнализация** используется для просмотра сигнализаций проекта.

#### **ВНИМАНИЕ!**

Рекомендуется в версиях 4.2 и выше использовать **Журнал тревог**!

Окно разделено на две части:

- **дерево активных сигнализаций проекта** в левой части
- **список активных сигнализаций** в правой части. В окне отображаются активные в текущий момент сигнализации и неактивные (зафиксированные ранее) сигнализации, которые не квитированы.



**Дерево активных сигнализаций** проекта позволяет пользователю увидеть иерархию объектов, в которых наступило интересующее его состояние (сигнализация, аларм). Ветки в дереве строятся только для сигнализаций, которые отображаются в окне. При выделении элемента в дереве список активных сигнализаций отображает только те сигнализации, которые "принадлежат" элементу по иерархии.

**Список активных сигнализаций** отображает сигнализации, принадлежащие выделенному в данный момент элементу в дереве активных сигнализаций проекта. Колонки списка можно настраивать при помощи окна **Настройка колонок**.

**Панель инструментов** включает следующие команды:

- показать дерево активных сигнализаций проекта
- скрыть дерево активных сигнализаций проекта
- квитировать выбранные в дереве сигнализации
- квитировать все активные сигнализации
- открыть окно **Настройка колонок**

### ВНИМАНИЕ!!!

Окно активной сигнализации показывает реальное состояние сигнализаций всего проекта на сервере **DataRate** без проверки прав доступа на визуализацию сигнализации и игнорируя локальные состояния сигнализации в клиенте и веб-клиенте.

### Квитирование сигнализаций

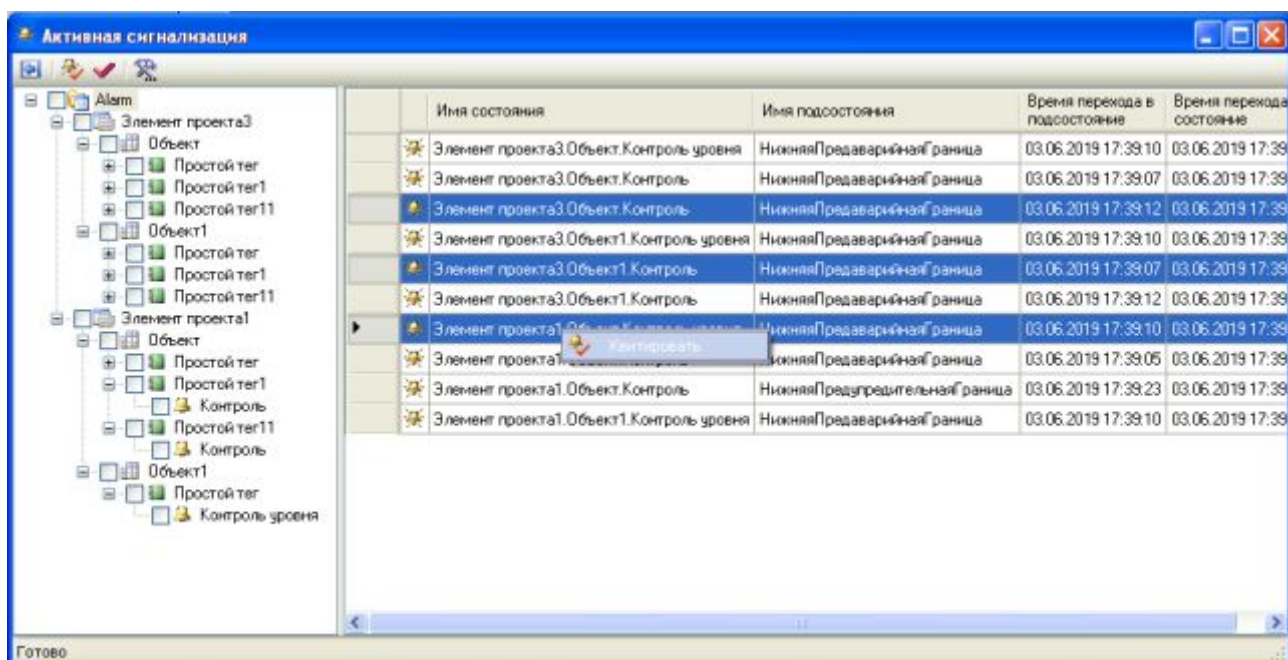
#### ВНИМАНИЕ!!!

Квитирование неактивной сигнализации удаляет ее из списка активных сигнализаций.

Квитирование в окне активных сигнализаций всегда приводит к квитированию на сервере, не зависимо от того, где это окно показывается (на сервере, на клиенте, в веб клиенте)

Существует несколько способов квитирования из окна **Активная сигнализация**:





Командой **Квитировать выбранные в дереве сигнализации**. Для этого следует выбрать с помощью галочек нужные сигнализации в дереве активных сигнализаций (при выборе элемента автоматически выбираются его дочерние элементы) и нажать кнопку на панели инструментов

- Командой **Квитировать все активные сигнализации**. Для этого следует нажать кнопку на панели инструментов
- Из контекстного меню **списка активных реализаций**. Для этого следует выделить в списке требуемые сигнализации, щелкнуть правой клавишей мыши на списке и в появившемся контекстном меню выбрать **Квитировать**.

#### 4.2.12.1 Настройка колонок активной сигнализации

Окно **Настройка колонок** вызывается кнопкой **Настройка колонок** на панели инструментов окна **Активная сигнализация**

**Для добавления колонок** нужно выделить нужные колонки в списке **Доступные колонки** и нажать кнопку со стрелкой вправо.

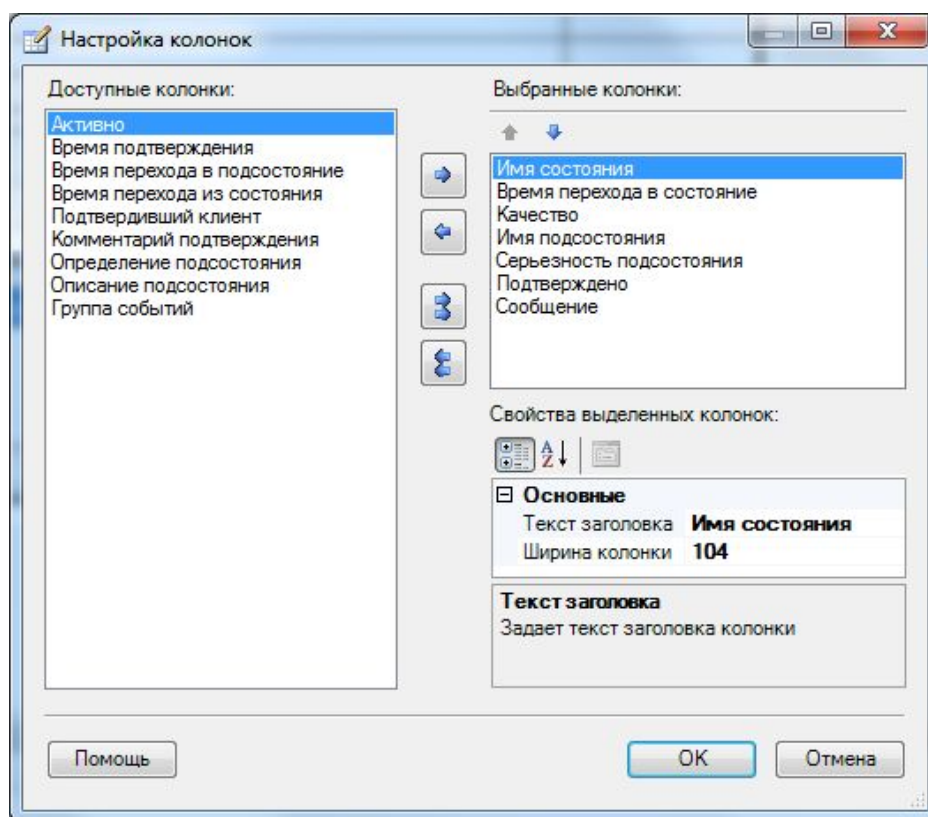
**Для удаления колонок** нужно выделить ненужные колонки в списке **Выбранные колонки** и нажать кнопку со стрелкой влево.

**Стрелки** (вверх/вниз) на панели инструментов над списком выбранных колонок позволяют настроить порядок следования колонок.

**Свойства колонок**, выделенных в списке **Выбранные колонки**, можно редактировать в панели **Свойства выделенных колонок**.

Для подтверждения сделанных изменений нужно нажать кнопку **ОК**, для отмены – кнопку **Отмена**. Закрытие окна при помощи кнопки в заголовке также отменяет сделанные изменения.

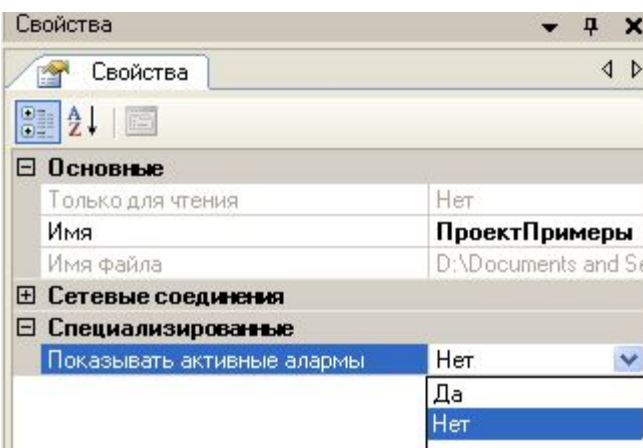
Для вызова справки по выбранной настройке нажмите кнопку **Помощь**.



## 4.2.12.2 Вызов окна активной сигнализации

Окно **Активная сигнализация** может открываться:

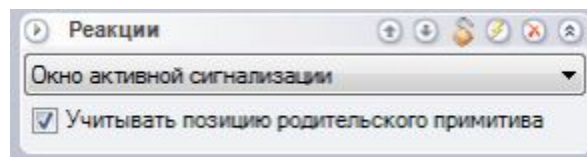
- **Автоматически** при возникновении **новой активной сигнализации**. По умолчанию автоматическое открытие окна активной сигнализации **отключено**. Для включения автоматического показа окна активной сигнализации нужно установить свойство проекта **Показывать активные алармы** в значение **Да**.



- С помощью функции реакции **Окно активной сигнализации**.

Для вызова окна следует:

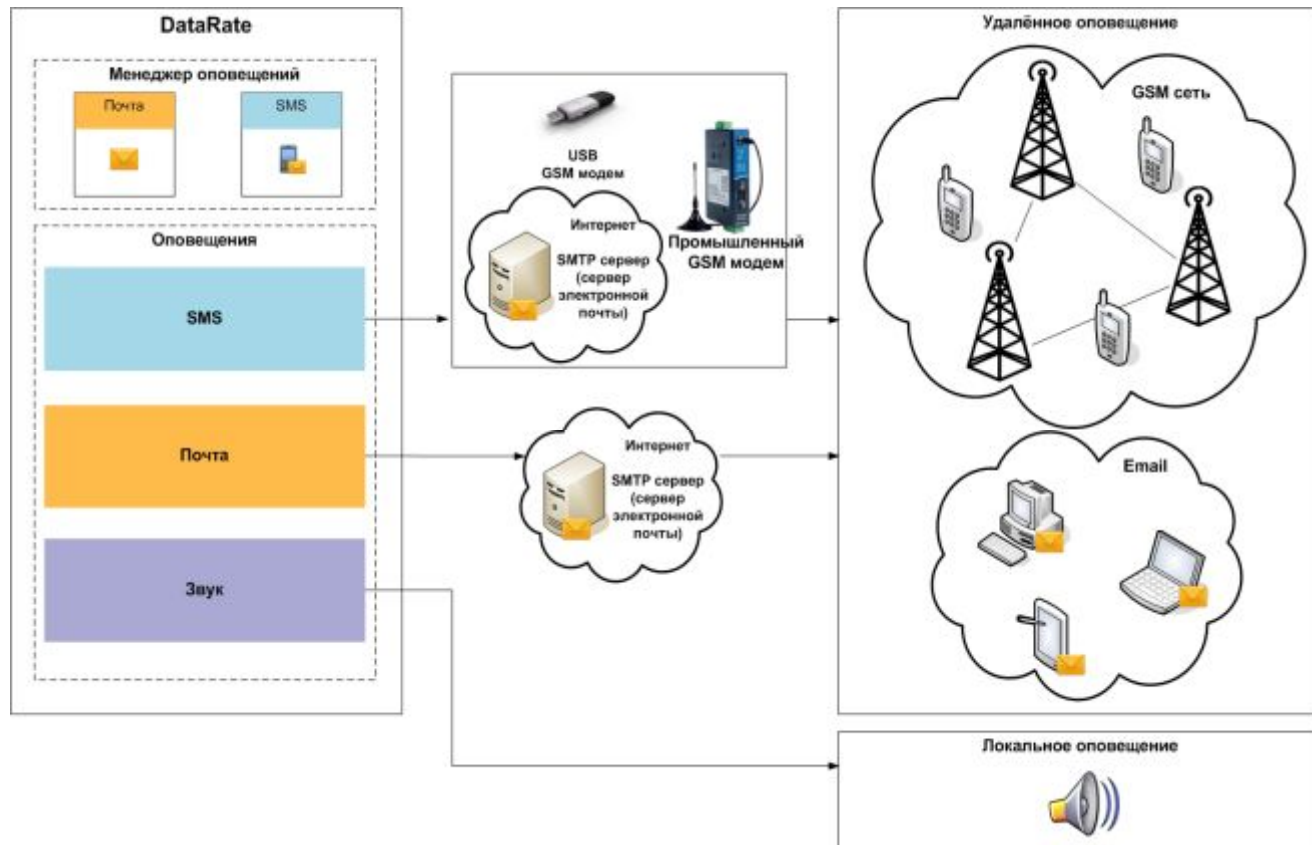
- 1 Создать графический примитив (например, **Прямоугольник**) или выбрать примитив из библиотеки изображений, который будет открывать окно активной сигнализации
- 2 Добавить примитиву реакцию **Окно активной сигнализации** на событие **Нажатие левой кнопки**





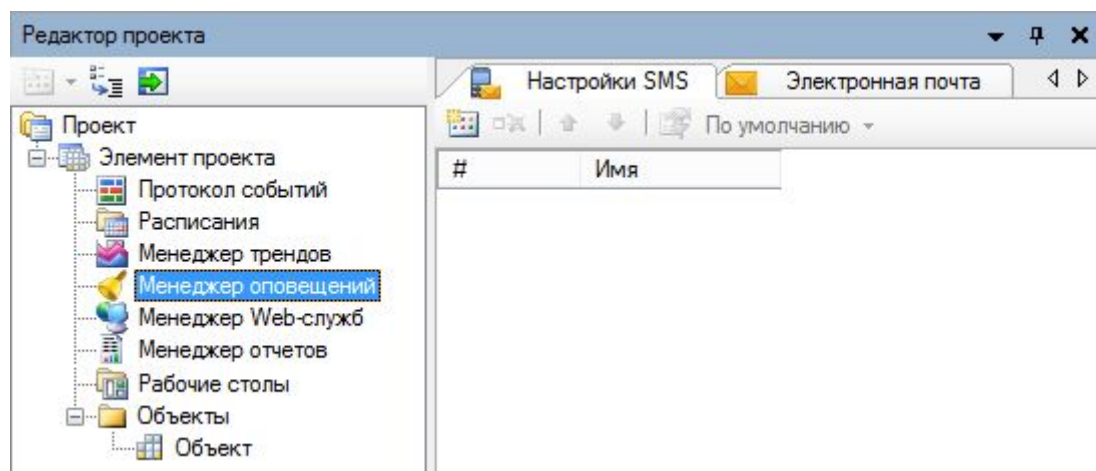
### 4.3 Оповещения

**Оповещение** – это действие, выполняемое для оповещения о достижении определенного состояния объекта с учетом заданного уровня серьезности.



#### 4.3.1 Менеджер оповещений

**Менеджер оповещений** – модуль элемента проекта **DataRate**, предназначенный для управления почтовыми и SMS оповещениями, а также для управления взаимодействием с аппаратурой для отправки SMS.



При инициализации любого почтового или SMS оповещения настроечные параметры берутся из **Менеджера оповещений**. Достаточно один раз настроить параметры оповещений и использовать их много раз в рамках проекта.

### 4.3.1.1 Настройки Менеджера оповещений для почтовых оповещений

**Настройки Менеджера оповещений почтового оповещения** позволяют:

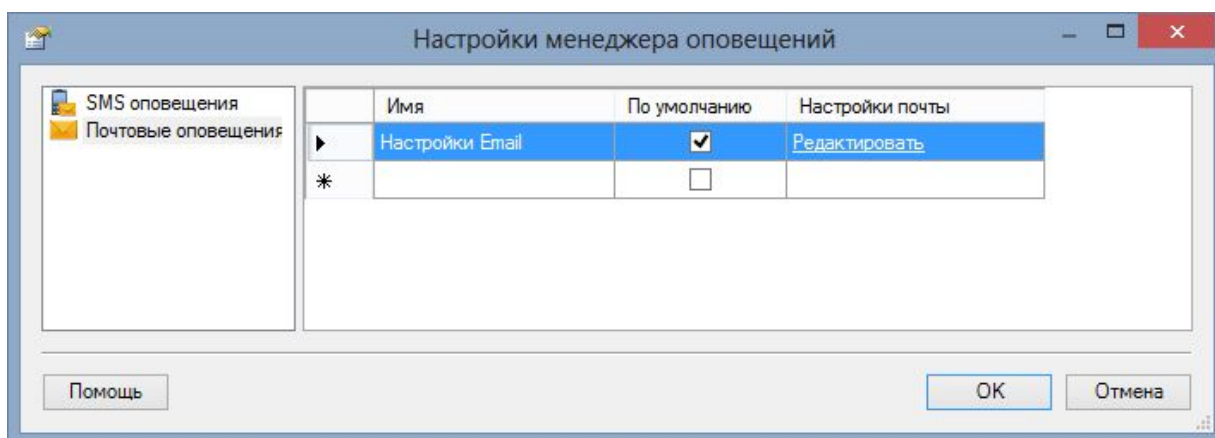
- Задать **список адресатов** (имя, почта)
- Выбрать **настройки SMTP сервера** (имя, порт, адрес отправителя и адрес для ответа, имя и пароль пользователя, домен, использование SSL).

Для формирования настроек следует:

- 1 Открыть окно **Настройки оповещений по почте** одним из следующих способов:

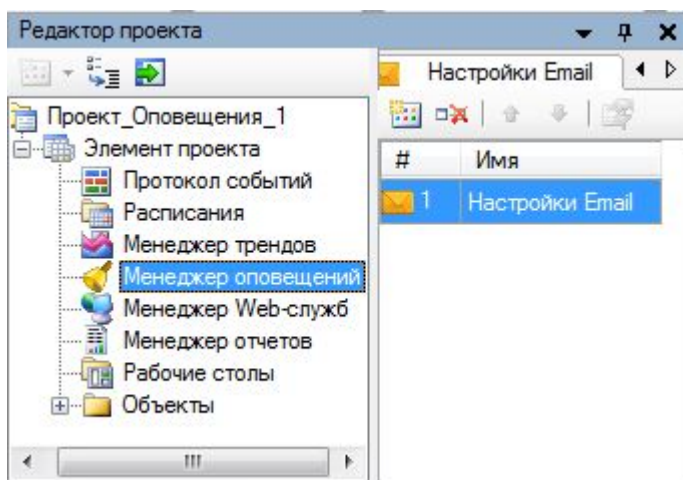
#### Способ 1

- В окне **Редактор проекта** дважды щёлкнуть по узлу **Менеджер оповещений** или в контекстном меню **Менеджера оповещений** выбрать пункт **Редактировать...**
- В открывшемся окне **Настройки менеджера оповещений** выбрать группу **Почтовые оповещения** и щёлкнуть левой клавишей мыши в пустой строке таблицы
- В отмеченной строке указать имя настройки в колонке **Имя** либо установить флаг **По умолчанию**, а затем в колонке **Настройки почты** щёлкнуть по надписи **Редактировать**

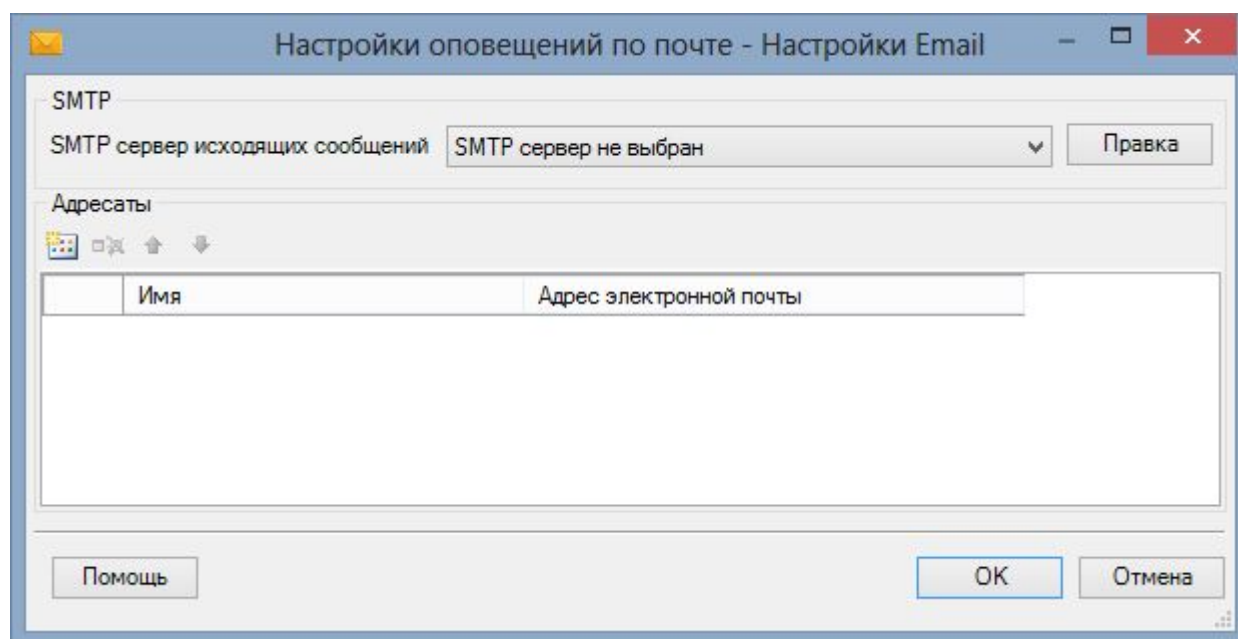



#### Способ 2

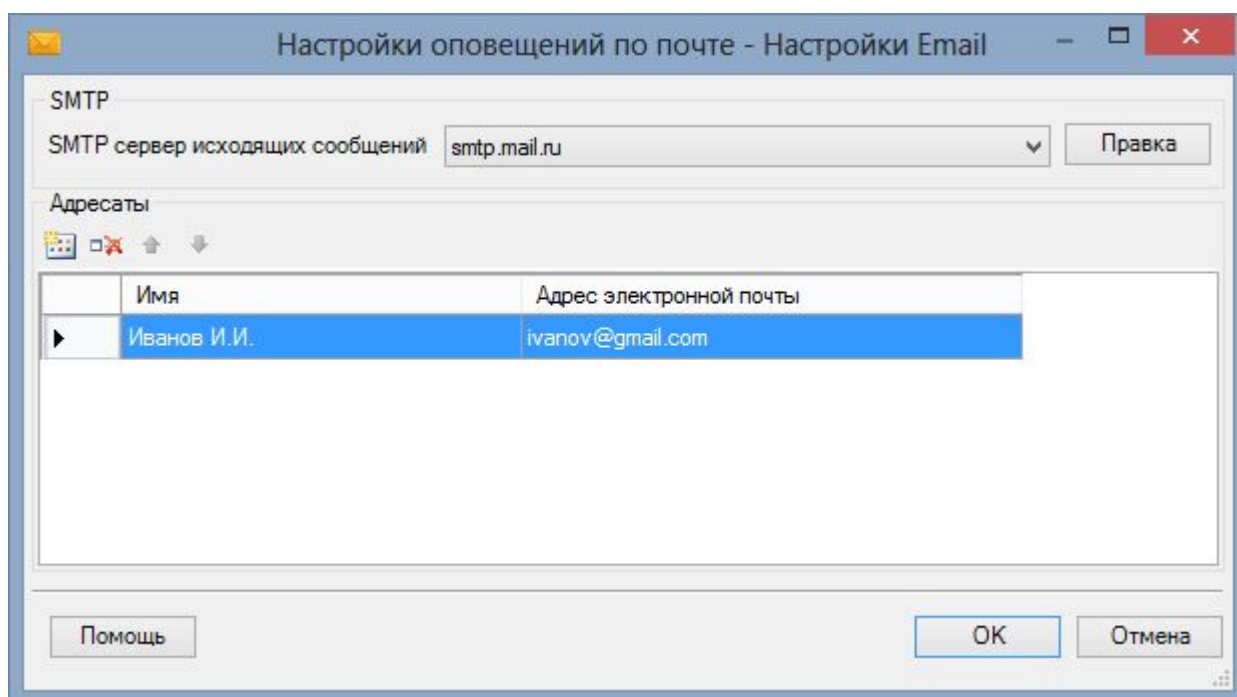
- В окне **Редактор проекта** выбрать узел **Менеджер оповещений**
- Перейти на вкладку **Настройки Email** и щёлкнуть по кнопке **Добавить новый элемент**
- Выполнить двойной щелчок мышью на добавленном элементе



- 2 В открывшемся окне **Настройки оповещений по почте** нажать на кнопку **Правка**
- 3 Создать и настроить **SMTP сервер**



- 4 В поле **SMTP сервер исходящих сообщений** выбрать из списка созданный **SMTP сервер**
- 5 Создать и настроить **Адресата**: в группе **Адресаты** нажать на кнопку **Добавить адресата**  и задать настройки **Имя**, **Адрес электронной почты**



- 6 Нажать на кнопку **OK**

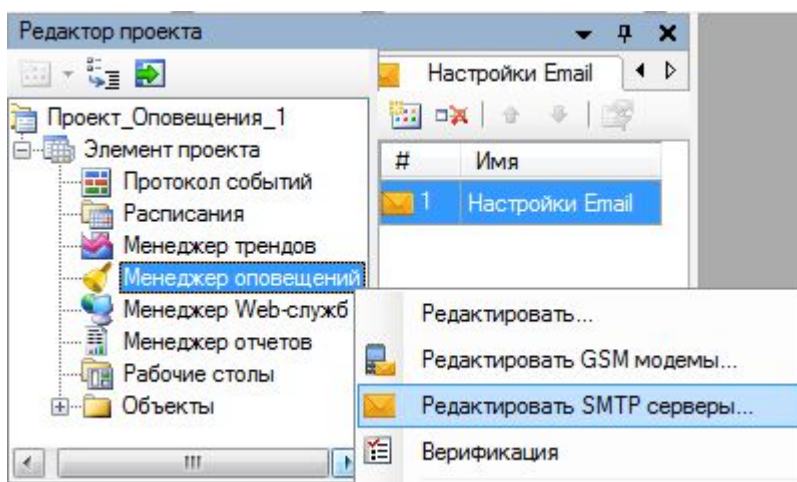
### **ВНИМАНИЕ!**

Фирма не несёт ответственности за отправку и доставку сообщений. Будьте внимательны при вводе адреса электронной почты.


### Создание и настройка SMTP сервера

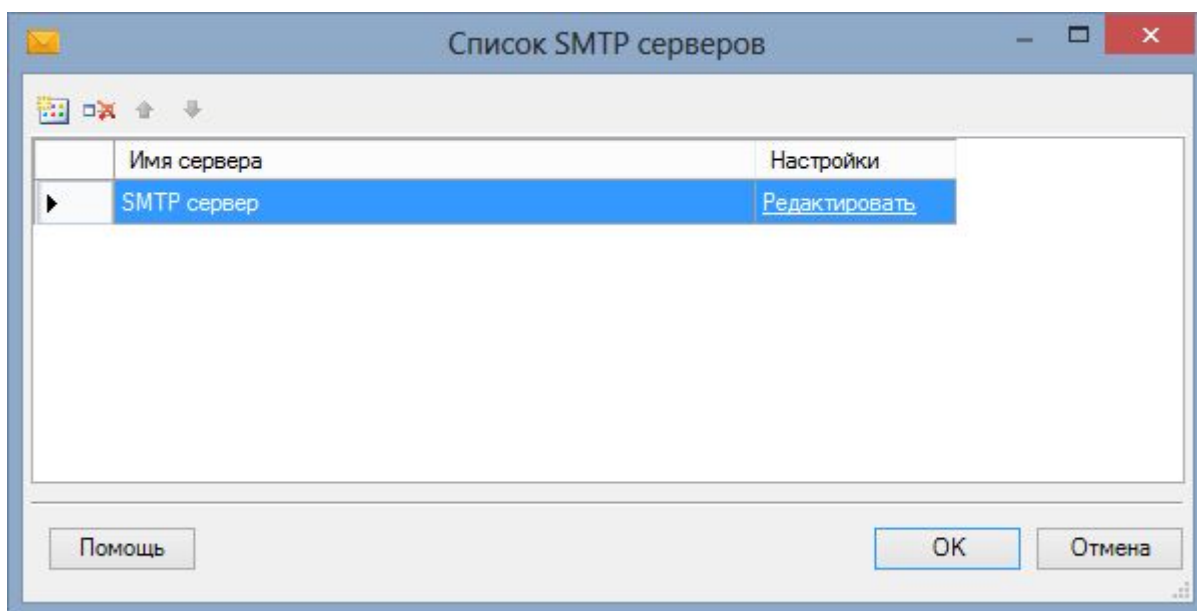
Для создания и настройки **SMTP сервера** следует:

- 1 Открыть окно **Список SMTP серверов** одним из следующих способов:
  - В контекстном меню **Менеджера оповещений** выбрать пункт **Редактировать SMTP серверы...**

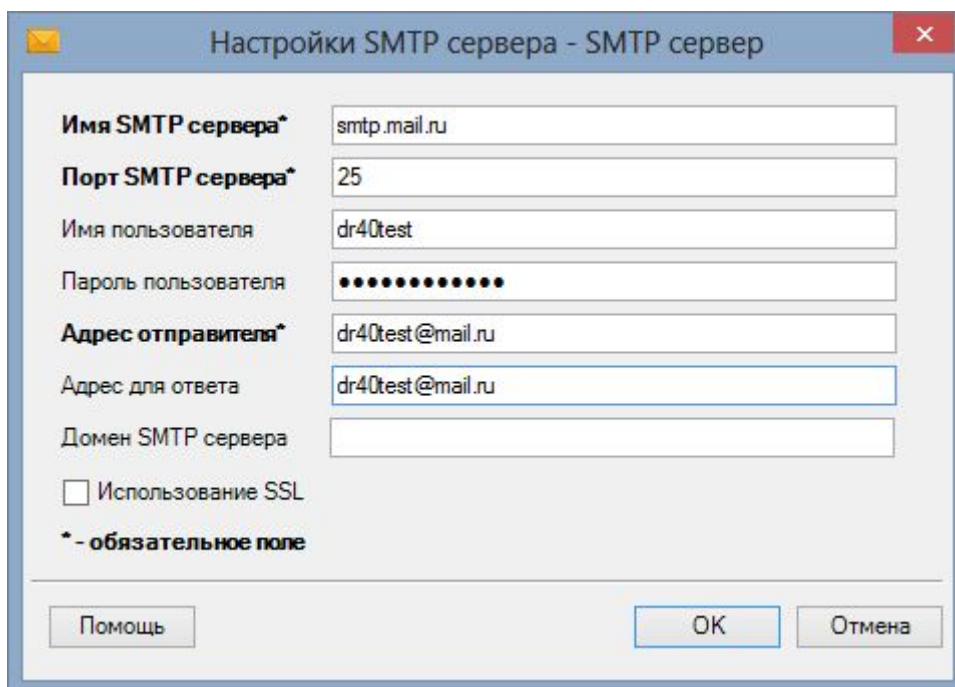


или

- Открыть окно **Настройки оповещений по почте** и нажать на кнопку **Правка**
- 2 В открывшемся окне **Список SMTP серверов** нажать на кнопку **Добавить SMTP сервер** 
  - 3 Задать имя сервера и в колонке **Настройки** щелкнуть по надписи **Редактировать**



- 4 В открывшемся окне **Настройки SMTP сервера** задать основные и вспомогательные настройки:
  - **Имя SMTP сервера** и **Порт SMTP сервера** (по умолчанию - 25)
  - **Имя пользователя** и **Пароль пользователя** SMTP сервера
  - **Адрес отправителя**
  - **Адрес для ответа** (дополнительно)



- **Домен SMTP сервера** – домен пользователя, от лица которого будут отправляться сообщения
- 5 Установить флаг **Использовать SSL**, чтобы использовать защищённое соединение с SMTP сервером
  - 6 Нажать на кнопку **ОК**.

**ВНИМАНИЕ!**

Убедитесь, что указанный порт SMTP сервера доступен с компьютера, на котором запущена **DataRate** Среда Исполнения. Для этого выполните команду (должна быть установлена/включена соответствующая служба):

*telnet <имя сервера>.<домен> <порт>*

Например: *telnet smtp.mail.ru 25*

Наиболее распространенные почтовые сервисы (Яндекс.Почта, Mail.ru, Gmail и другие) в своих инструкциях рекомендуют указывать порт 465 для работы с использованием защищенного соединения по SSL. Встроенный почтовый клиент реализует режим явного SSL (Explicit SSL), при этом соединение устанавливается через порт 25 (или 587) в незашифрованном виде, а затем переключается в защищенный режим. При попытке связаться через порт 465 будет получена ошибка превышения интервала ожидания и письмо не будет отправлено.

**ВНИМАНИЕ!**

Если выбрана опция "Использовать SSL", но почтовый сервер не поддерживает SSL соединение, то письмо не будет отправлено. Об отсутствии поддержки SSL можно узнать, запустив проект на исполнение в Среде Разработки **DataRate**, при этом в окне Отладка на соответствующей вкладке будет отображено описание ошибки.



### 4.3.1.2 Настройки Менеджера оповещений для SMS оповещений

**Настройки Менеджера оповещений SMS оповещения** позволяют:

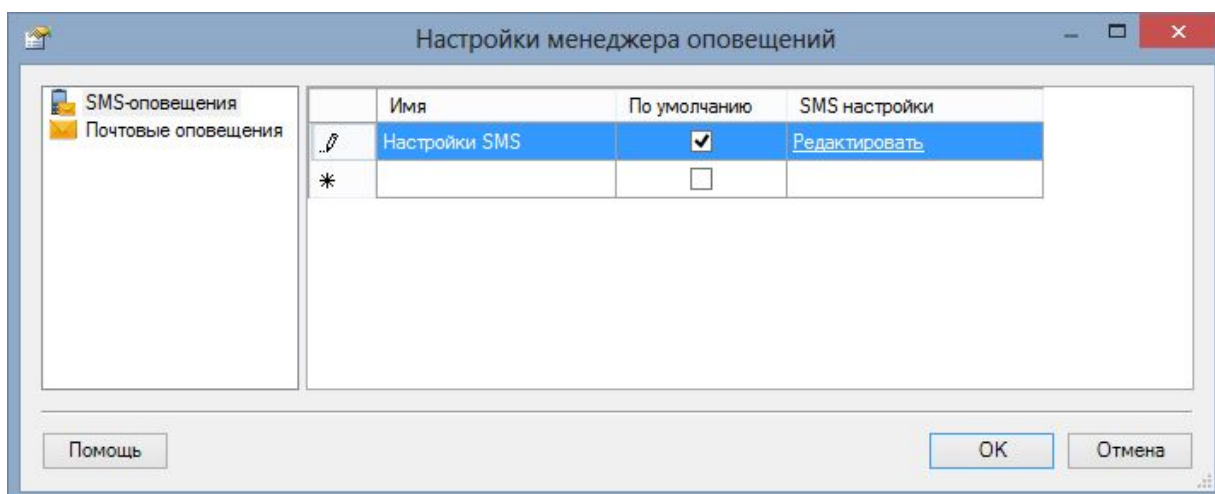
- Задать **список адресатов** (имя, номер телефона, почта)
- Выбрать **настройки модема** (имя, описание, порт, скорость обмена, таймаут, USSD-запрос для проверки баланса) и настройки SMTP сервера.

Для формирования настроек следует:

- 1 Открыть окно **Настройки оповещений по почте** одним из следующих способов:

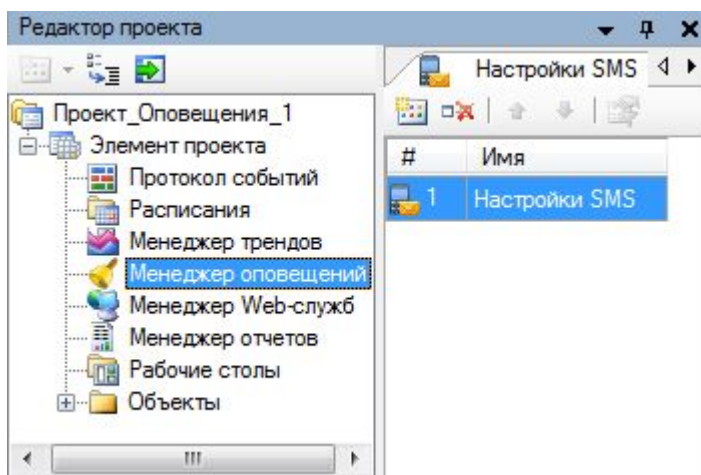
#### Способ 1

- В окне **Редактор проекта** дважды щёлкнуть по узлу **Менеджер оповещений** или в контекстном меню **Менеджера оповещений** выбрать пункт **Редактировать...**
- В открывшемся окне **Настройки менеджера оповещений** выбрать группу **SMS оповещения** и щелкнуть левой клавишей мыши в пустой строке таблицы
- В отмеченной строке указать имя настройки в колонке **Имя** либо установить флаг **По умолчанию**, а затем в колонке **SMS настройки** щелкнуть по надписи **Редактировать**




#### Способ 2

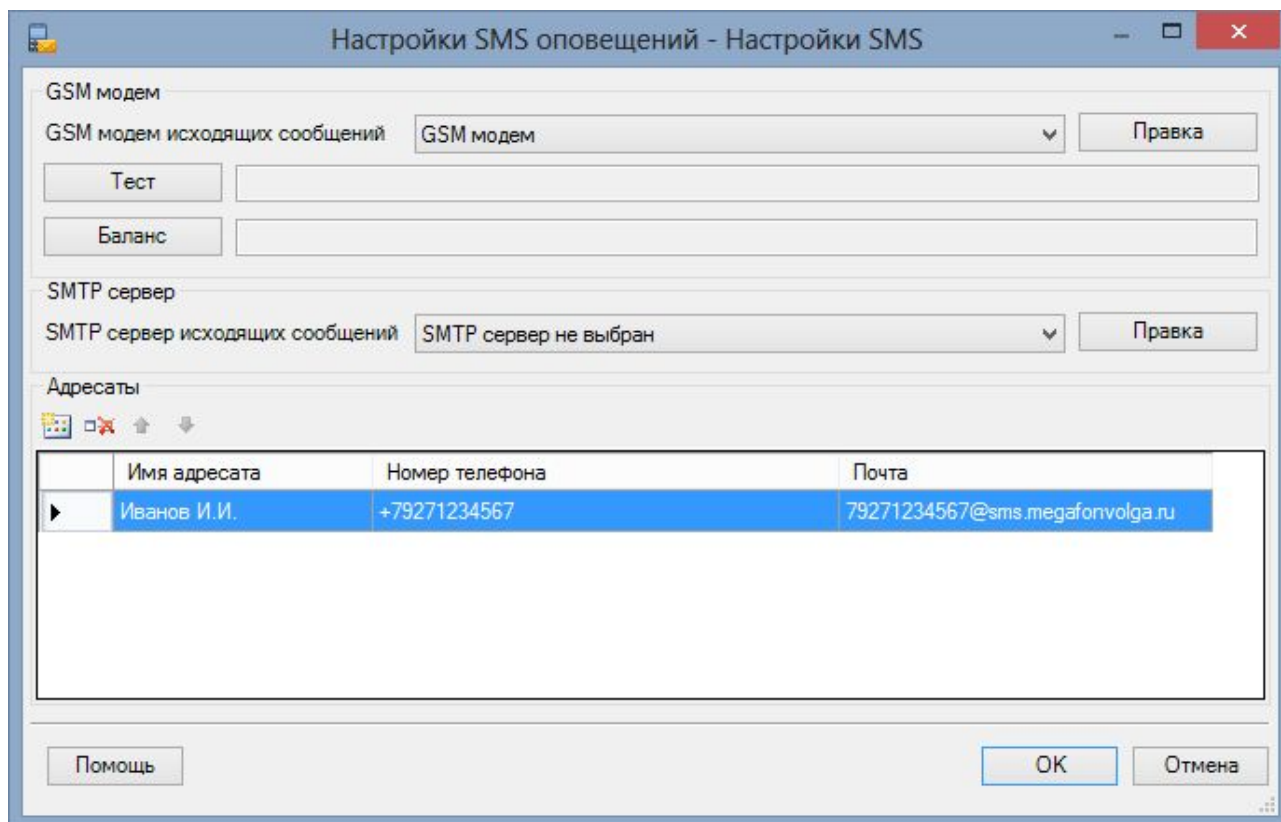
- В окне **Редактор проекта** выбрать узел **Менеджер оповещений**
- Перейти на вкладку **Настройки SMS** и щелкнуть по кнопке **Добавить новый элемент**
- Выполнить двойной щелчок мышью на добавленном элементе



- 2 В открывшемся окне **Настройки SMS оповещений** в панели **GSM модем** нажать на кнопку **Правка**



- 3 Создать и настроить **GSM модем**
- 4 В поле **GSM модем исходящих сообщений** выбрать из списка созданный **GSM модем**
- 5 Создать и настроить **Адресата**: в группе **Адресаты** нажать на кнопку **Добавить адресата**  и задать настройки **Имя**, **Номер телефона** в формате **+79xxxxxxxxx**  
Дополнительные настройки – **Почта** в формате **79xxxxxxxxx@<домен>** :  
 Мегафон: **79xxxxxxxxx@sms.megafonvolga.ru** (Поволжье)  
           **79xxxxxxxxx@sms.megafonmoscow.ru** (Московский регион)  
           **79xxxxxxxxx@sms.megafonnw.ru** (Северо-западный регион)  
           и другие  
 Beeline: **79xxxxxxxxx@sms.beemail.ru**  
 MTC: **79xxxxxxxxx@sms.mts.ru**



Настройки SMS оповещений - Настройки SMS

GSM модем

GSM модем исходящих сообщений: GSM модем [Правка]

Тест

Баланс

SMTP сервер

SMTP сервер исходящих сообщений: SMTP сервер не выбран [Правка]

Адресаты

Имя адресата	Номер телефона	Почта
Иванов И.И.	+79271234567	79271234567@sms.megafonvolga.ru

Помощь [ОК] [Отмена]

- 6 Для получения информации о состоянии GSM модема (подключен/не подключен) нажать на кнопку **Тест**
  - 7 Для получения информации о текущем балансе нажать на кнопку **Баланс**
- ВНИМАНИЕ!**  
 Функционирование механизма получения информации о текущем балансе зависит от особенностей выбранного оператора мобильной связи, тарифного плана (предоплатный, постоплатный) и аппаратных особенностей устройства.
- 8 Нажать на кнопку **ОК**

### ВНИМАНИЕ!

Фирма не несёт ответственности за отправку и доставку сообщений. Будьте внимательны при вводе номера адресата.

Узнайте подробную информацию о подключении услуги у Вашего оператора мобильной связи.

Отправлять SMS можно на любой Email адрес, при этом текст сообщения будет автоматически разбиваться на сообщения по 70 символов кириллицей или по 160 символов латиницей.

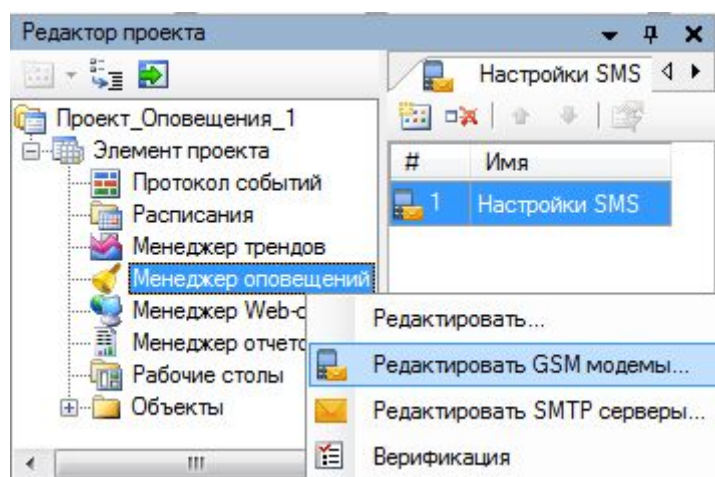
### Создание и настройка GSM модема

**GSM модем** – устройство, позволяющее осуществлять обмен данными в сетях мобильной связи **GSM 900/1800**.


**USB GSM модем** – устройство, позволяющее осуществлять обмен данными в сетях мобильной связи **GSM 900/1800** через **USB** интерфейс

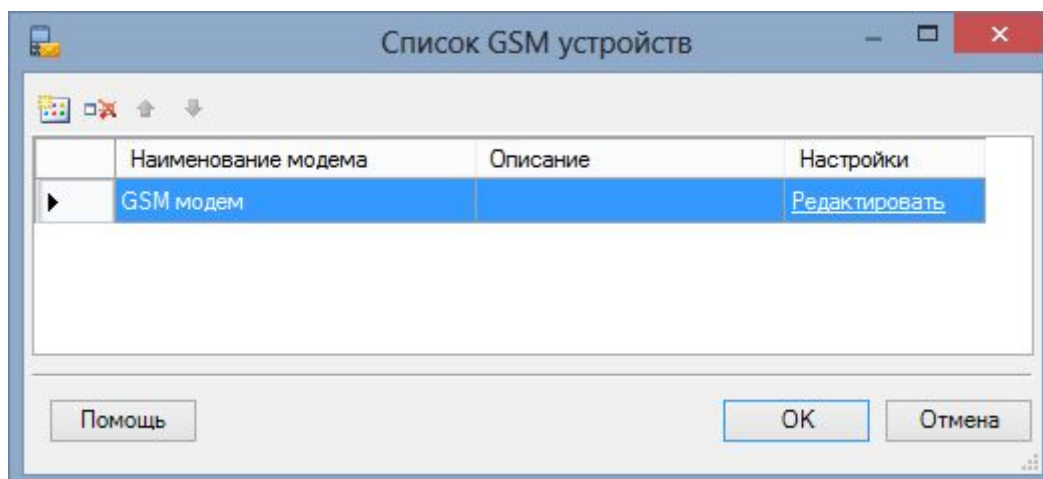
Для создания и настройки **GSM модема** следует:

- 1 Открыть окно **Список GSM устройств** одним из следующих способов:
  - В контекстном меню **Менеджера оповещений** выбрать пункт **Редактировать GSM модемы...**



или

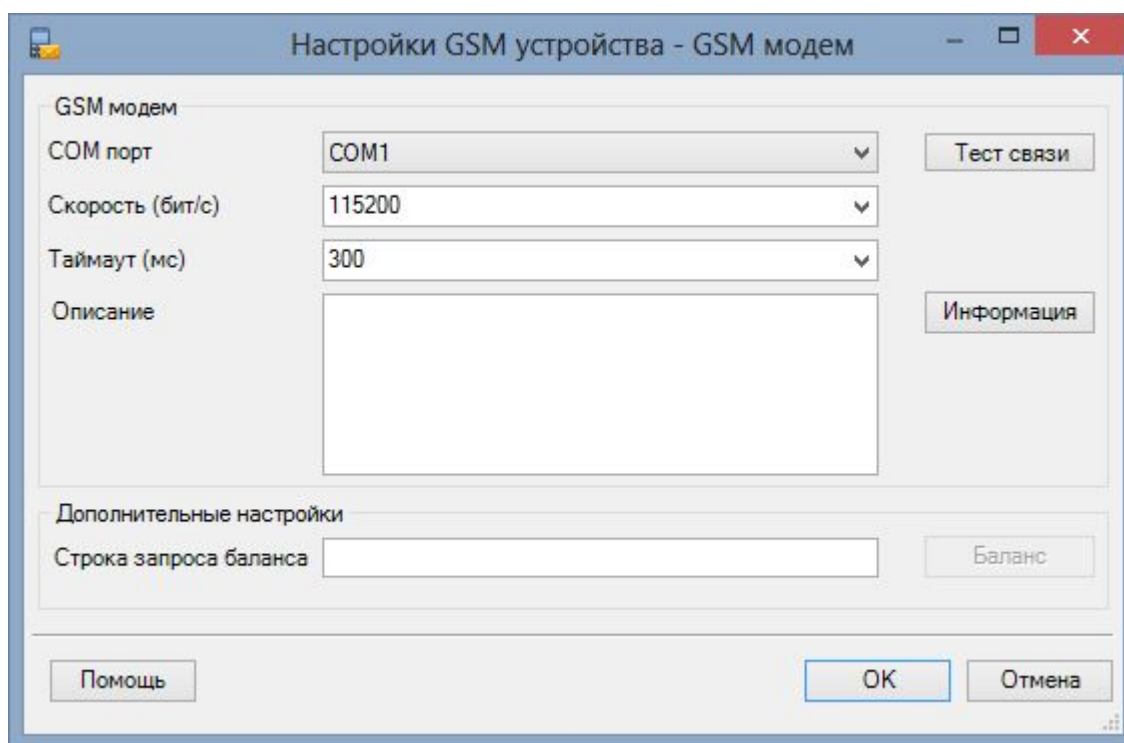
- Открыть окно **Настройки SMS оповещений** и нажать на кнопку **Правка**
- 2 В открывшемся окне **Список GSM устройств** нажать на кнопку **Добавить модем** 
  - 3 Задать имя модема и в колонке **Настройки** щелкнуть по надписи **Редактировать**



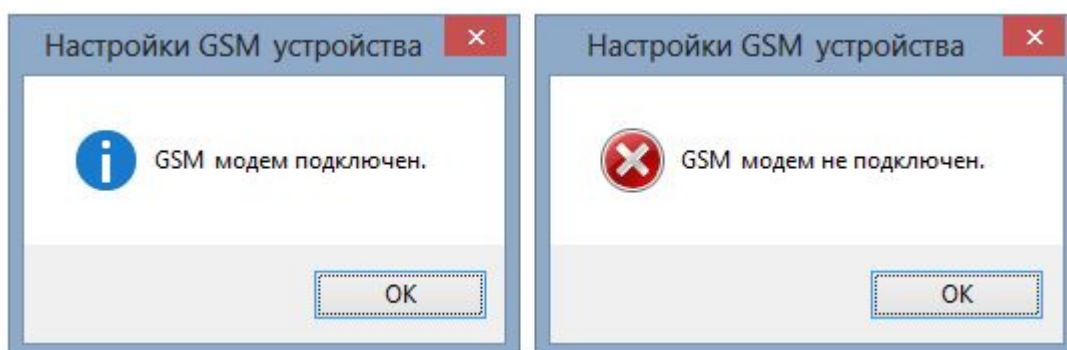
- 4 В открывшемся окне **Настройки GSM устройства** задать настройки **GSM модема**:
- Выбрать из списка **COM порт**, к которому подключен GSM модем

**ВНИМАНИЕ!**

**GSM модем или телефон, используемый в качестве GSM модема, может опознаваться на нескольких портах. При указании неверного порта SMS будут не отправлены. Будьте внимательны при выборе значения данной настройки.**



- Выбрать значение **Скорость (бит/с)** из списка: **2400, 4800, 9600, 19200, 38400, 57600, 115200**
- Выбрать значение **Таймаут (мс)** из списка: **150, 300, 600, 1200, 1500, 1800, 2000**
- Нажать на кнопку **Тест связи**, чтобы получить информацию о состоянии **GSM модема**

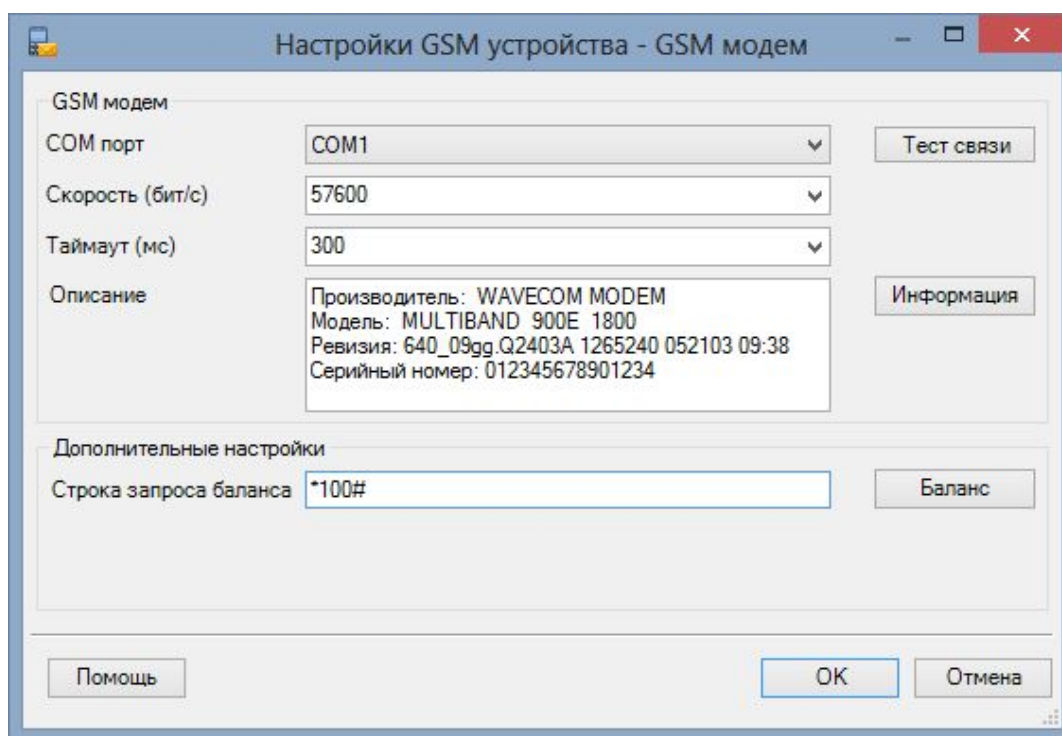


- Нажать на кнопку **Информация**, чтобы отобразить в поле **Описание** сведения о **GSM модема**
- Дополнительно ввести в поле **Строка запроса баланса** USSD-запрос и нажать на кнопку **Баланс** для проверки правильности введенного USSD-запроса и получения информации о текущем балансе

### **ВНИМАНИЕ!**

Отправка сообщения возможна только при наличии средств на счёте, поэтому своевременно проверяйте баланс.

- 5 Нажать **ОК** в окне **Настройки GSM устройства**
- 6 Нажать **ОК** в окне **Список GSM устройств**.




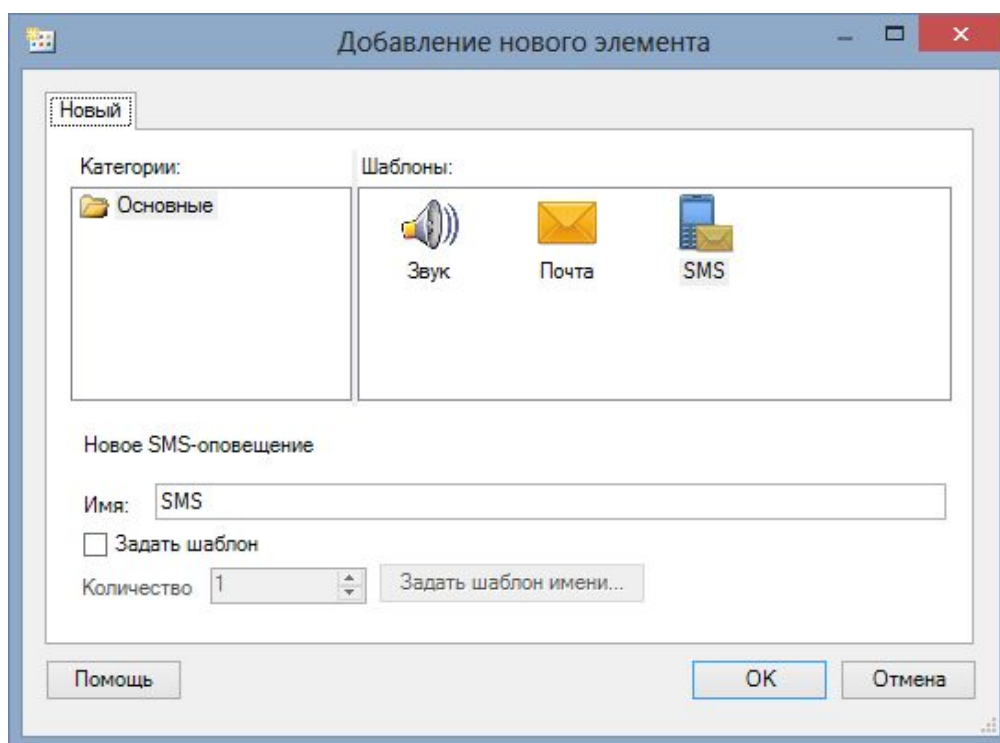
### **ВНИМАНИЕ!**

Работа некоторых моделей GSM устройств возможна только при наличии запущенного программного обеспечения, поставляемого вместе с данным GSM устройством.

### 4.3.2 Создание, привязка аргумента и срабатывание оповещений

Для создания оповещения следует:

- 1 В окне **Редактор проекта** создать объект
- 2 В контекстном меню созданного объекта выбрать пункт **Оповещение** или перейти на вкладку **Оповещения** и щелкнуть на кнопке **Добавить новый элемент** 
- 3 В появившемся окне **Добавление нового элемента** выбрать категорию **Основные** и шаблон оповещения: **Звук**, **Почта** или **SMS**
- 4 Задать **Имя** и нажать на кнопку **ОК**.  
Если необходимо создать несколько оповещений следует установить признак **Задать шаблон**, указать **Количество** и задать шаблон имени.



При необходимости оповещение можно отключить, задав значение **Нет** свойства **Разрешено**.

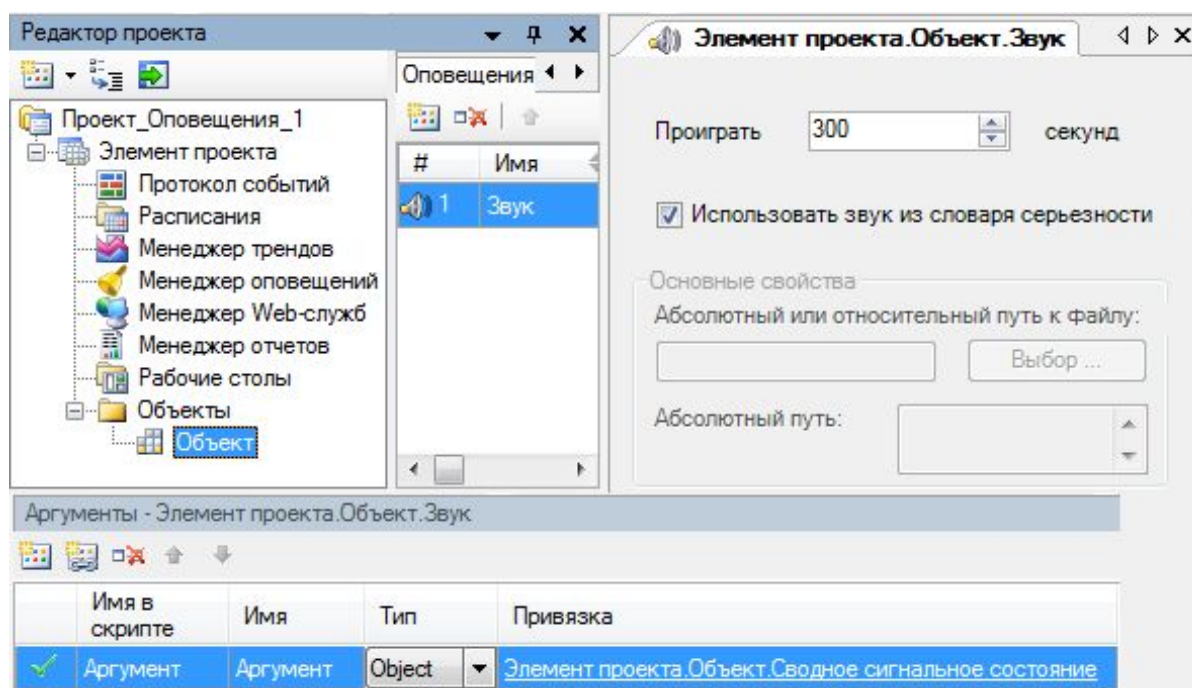
#### Настройка привязки аргумента оповещения

Аргумент оповещения должен быть привязан:

- к **сводному сигнальному состоянию** (или его свойству) любого **объекта**
- или к **сводному сигнальному состоянию** мнемосхемы
- или к **сигнальному состоянию** (или его свойству) определенного **тега** объекта.

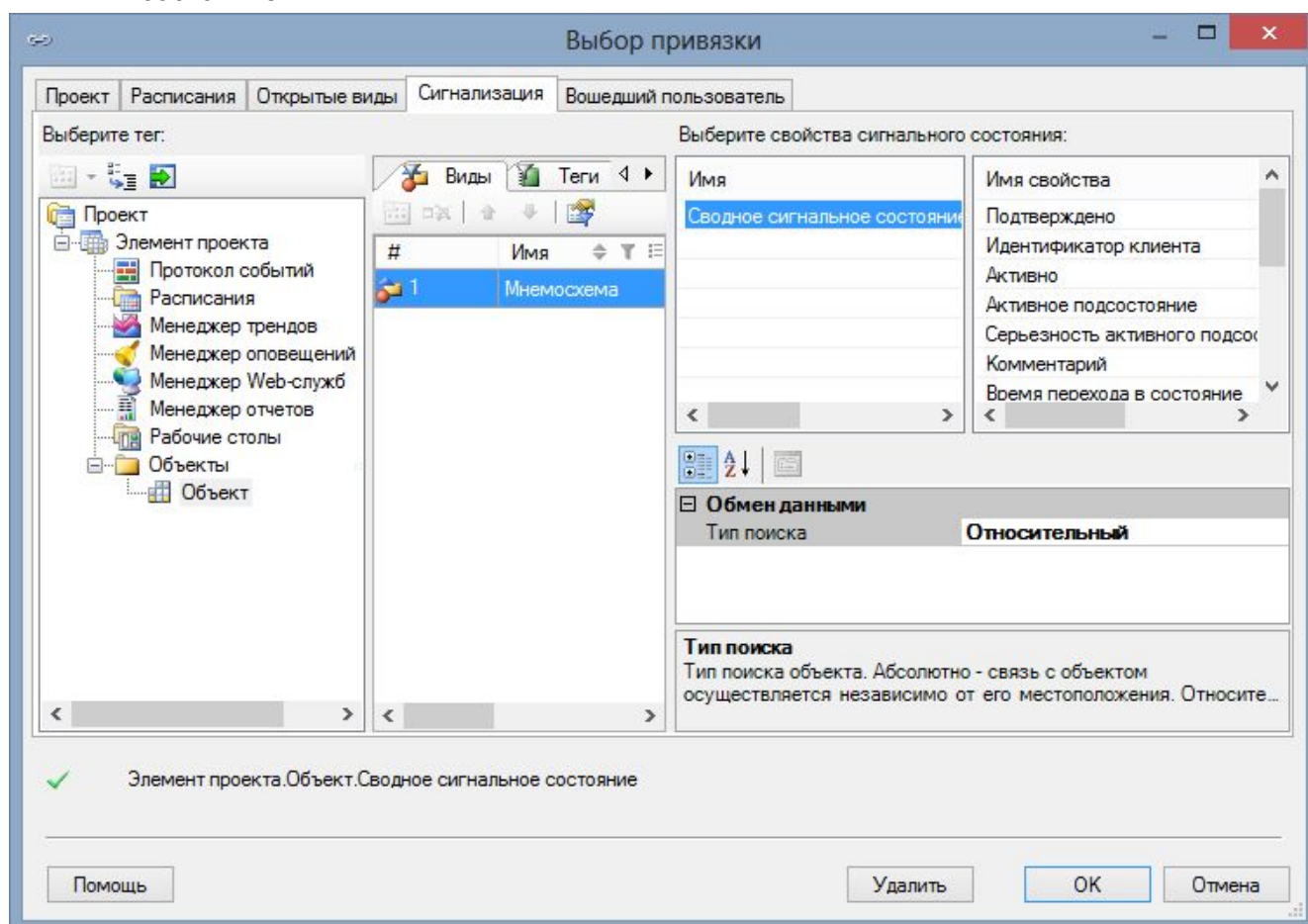
По умолчанию оповещение привязывается к **Сводному сигнальному состоянию** объекта, в котором оно создано.





Сменить привязку можно в окне **Аргументы**:

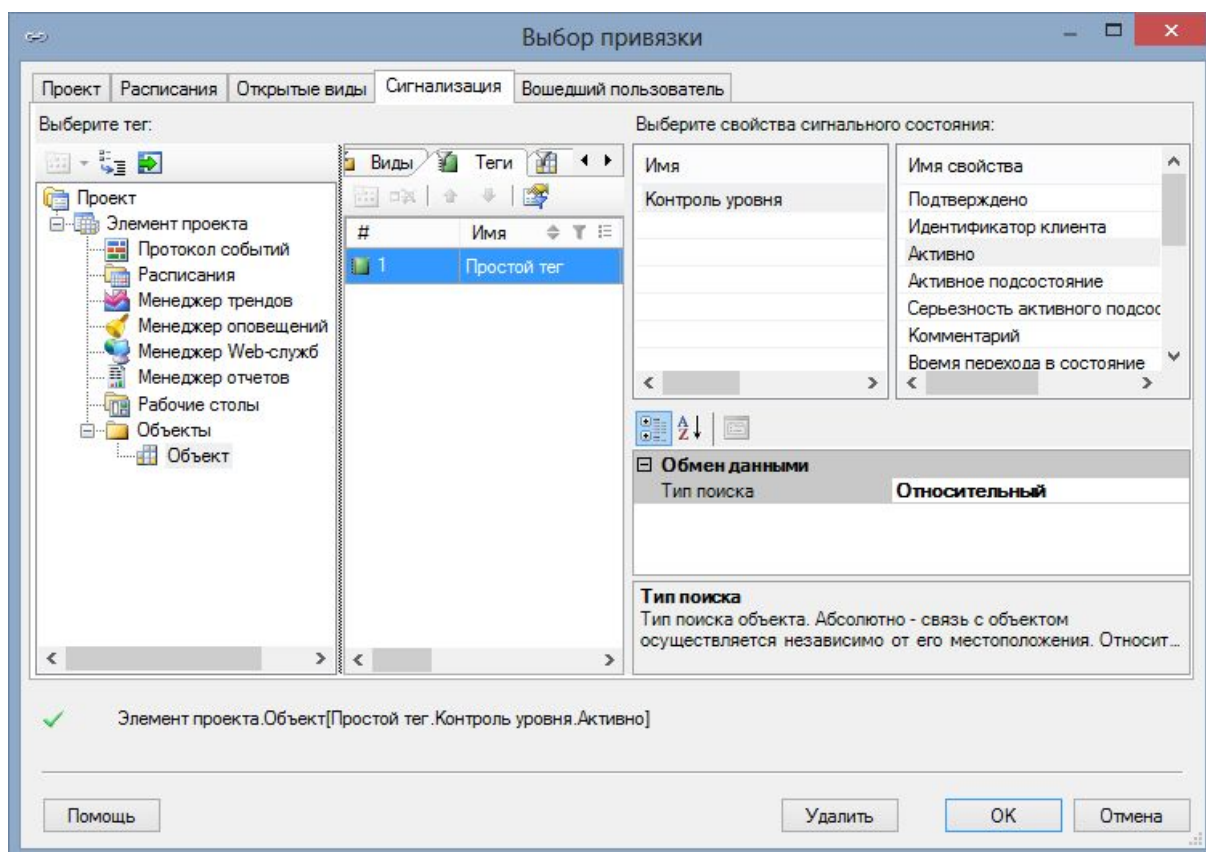
- 1 Щелкнуть по ссылке в поле **Привязка** в окне **Аргументы** оповещения
- 2 В открывшемся окне **Выбор привязки** выбрать необходимое сигнальное состояние



Для привязки к свойству сигнального состояния определенного тега следует:

- 1 Перейти на вкладку **Теги**
- 2 Выбрать тег и свойство сигнального состояния.

Осуществить привязку можно с помощью операции **Drag&Drop** – перетащить тег в окно **Аргументы** оповещения, в открывшемся окне **Выбор привязки** перейти на вкладку **Сигнализация** и выбрать необходимые свойства.



### Срабатывание оповещений

При работе с оповещениями необходимо учитывать, что если привязка осуществлена к сигнальному состоянию (сводному или состоянию тега), оповещение будет срабатывать при изменении любого из его (сигнального состояния) свойств.

Если привязка осуществлена к свойству сигнального состояния, то оповещение будет срабатывать при изменении выбранного свойства.

## 4.3.3 Звуковое оповещение

Для настройки оповещения **Звук** необходимо задать звуковой файл (\*.wav) и время, которое он будет проигрываться.

Поле **Проиграть** задает время, которое будет проигрываться файл, если состояние не квитировано. Звуковой файл будет проигран в случае смены сводного подсостояния при условии, что оно активно и не квитировано.

Если установлен признак **Использовать звук из словаря серьезности**, то при смене сводного состояния будет проигран звук, соответствующий уровню серьезности сводного состояния.

Если указан путь к файлу, он будет проигрываться при любом текущем уровне серьезности сводного состояния.

Возможно два варианта задания пути к файлу:

- **Относительный.** Файл добавляется в каталог со стандартными звуками **SoundLibrary**, после чего достаточно задать только имя нужного файла
- **Абсолютный.** Файл размещается в любом месте на диске, после чего необходимо указать полный путь к файлу.

Пример:

	Путь	Расположение файла
<b>Относительный</b>	damage.wav	C:\ProgramData\EnergoKrug\DataRate <версия>\SoundLibrary\damage.wav
<b>Абсолютный</b>	D:\warning.wav	D:\warning.wav

При возникновении события с некоторой серьезностью для оповещения используется звуковой файл, заданный в колонке **Звук** у записи с ближайшим сверху значением серьезности. Например, если в словаре задан звук для записей с серьезностями 200 и 300, то при возникновении события с серьезностью 250 будет выбрана запись с серьезностью 300 для воспроизведения звука.

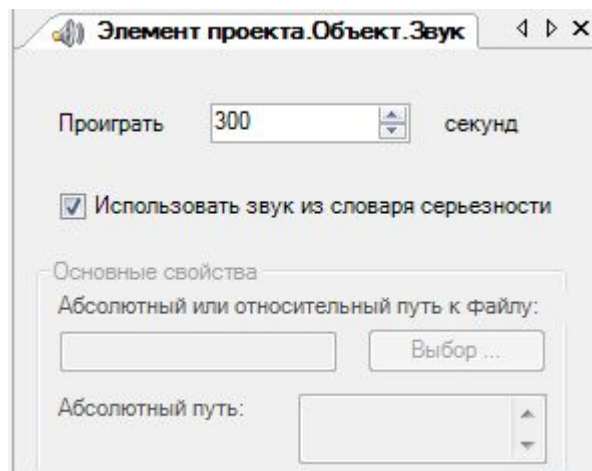
Словарь серьезности		Возникшее событие	
Уровень серьезности	Звук	Уровень серьезности	Проигрываемый звук
200	<b>Звук1.wav</b>	250	<b>Звук2.wav</b>
300	<b>Звук2.wav</b>		

Чтобы звук не проигрывался, имя файла из колонки **Звук** нужно удалить. Для этого можно нажать на ссылку с именем файла и в появившемся диалоговом окне нажать кнопку **Отмена** или выделить ячейку с именем файла и нажать на клавиатуре клавишу **Delete**.

Оповещение может быть отключено установкой свойства **Разрешено** в значение **Нет**.

### ВНИМАНИЕ!!!

Для корректной работы необходимо настроить колонку **Звук** в Словаре серьезности в Протоколе событий проекта.



#### 4.3.4 Почтовое оповещение

**Почтовое оповещение** – это элемент объектной модели **DataRate**, позволяющий информировать с помощью email о возникновении событий в графическом проекте.

**Почтовое оповещение** может быть реализовано с помощью отправки **Email** через SMTP сервер.

Отправка сообщений осуществляется при изменении сводного состояния объекта, сигнального состояния тега в соответствии с серьёзностью и категорией инициирующего события. Подсистема почтовых оповещений формирует очередь сообщений на основе существующих принципов **Протокола событий**: категорий, серьёзности и т.д.

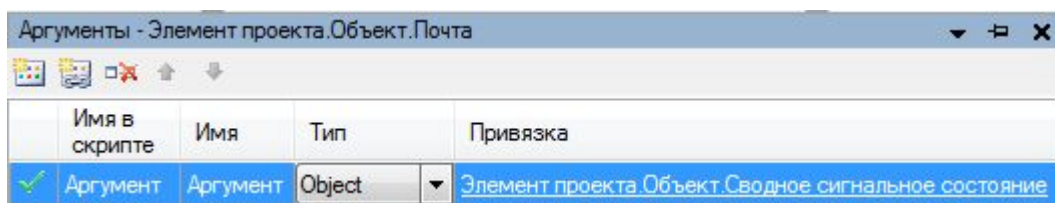
Вы можете создать множество почтовых оповещений для одного объекта, указав для каждого текст оповещения и способ отправки. Количество получателей почтовых оповещений может быть разным, поэтому существует возможность указать несколько настроек почты из **Менеджера оповещений**.

Текст сообщения формируется на основе информации о сигнальном состоянии, по которому срабатывает данное оповещение. Вы можете полностью сформировать текст сообщения или выбрать его из предложенных шаблонов.

Для создания оповещения следует создать объект оповещения, выбрав в категории **Основные** шаблон оповещения **Почта**, и задать его **Имя** (по умолчанию – **Почта**).

#### Настройка почтового оповещения

- 1 При создании объекта оповещения по умолчанию в окне **Аргументы** автоматически создаётся **Аргумент**, привязанный к **Сводному сигнальному состоянию** объекта, в котором создан объект **Почта**. При необходимости Вы можете привязать **Аргумент** к **Сводному сигнальному состоянию** (или его свойству) другого объекта или **Сигнальному состоянию** (или его свойству) тега. Для этого щёлкните по ссылке в колонке **Привязка**



- 2 Для выбора настроек в поле редактирования в закладке **Элемент проекта.Объект. Почта** нажать на кнопку **Выбрать настройки**

- 3 В появившемся окне **Выбор настроек Email** выбрать необходимые настройки и нажать на кнопку **ОК**

- 4 Задать настройки группы **Сообщение** – **Тема**, **Имя письма**, **Тело письма** – вручную или воспользоваться предложенными шаблонами. Для этого нажмите соответствующую кнопку **Шаблон**. Подстановка, выбранная из открывшегося списка, будет вставлена в то место письма, где расположен курсор



- 5 Установить флаг **Добавить сообщение об отправке в Протокол событий** для создания соответствующей записи о результате отправки сообщения в **Протоколе событий**
- 6 В окне **Свойства** установить значение поля **Оповестить о выходе из активного состояния** в **Да**, если необходимо

Основные	
Имя	Почта
Библиотечный	Нет
Разрешено	Да
Оповестить о выходе из активного состояния	Да

- 7 Сохранить проект

### ВНИМАНИЕ!!!

После настройки оповещения рекомендуется провести верификацию объекта (владельца оповещения) для выявления и исправления всех несоответствий

### Методы для отправки email с помощью скрипта

Для отправки оповещения по **Email** в скрипте используется класс **EmailNotificationHelper**.

#### Очистка очереди электронных сообщений

**void ClearEmailMessages (object client);**

где **client** – Клиент.

#### Создание и возвращение настроек почтового сервера

**ISMTPServerSettings CreateSmtpSettings (string fromAddress, string replyAddress, string smtpLogin, string smtpPassword, string smtpServerName, int port, string domain, bool useSSL);**

где **fromAddress** – адрес отправителя  
**replyAddress** – адрес для ответа  
**smtpLogin** – имя зарегистрированного на сервере пользователя  
**smtpPassword** – пароль пользователя  
**smtpServerName** – имя сервера/адрес  
**port** – порт сервера  
**domain** – домен  
**useSSL** – признак использования защищенного соединения.

Метод возвращает настройки почтового сервера.

#### Добавление почтового сообщения в менеджер передачи почты с созданными в скрипте настройками почтового сервера

**void AddEmailMessage (object client, ISMTPServerSettings smtpServer, bool addEvent, string to, string subject, string body, int priority, bool isBodyHtml);**

где **client** – Клиент  
**smtpServer** – настройки SMTP сервера  
**addEvent** – признак добавления сообщения в протокол событий  
**to** – адрес получателя  
**subject** – тема письма  
**body** – тело письма

**priority** – приоритет (0-высокий, 1-низкий, 2-нормальный)  
**isBodyHtml** – HTML формат тела письма.

**Добавление почтового сообщения в менеджер передачи почты с указанными настройками SMTP с указанием адреса получателя**

**void AddEmailMessage(object client, string smtpServerName, bool addEvent, string to, string subject, string body, int priority, bool isBodyHtml);**

где **client** – Клиент  
**smtpServer** – наименование настроек SMTP сервера  
**addEvent** – признак добавления сообщения в протокол событий  
**to** – адрес получателя  
**subject** – тема письма  
**body** – тело письма  
**priority** – приоритет (0-высокий, 1-низкий, 2-нормальный)  
**isBodyHtml** – HTML формат тела письма.

**Добавление почтового сообщения в сервис передачи почты с указанными настройками Email**

**void AddEmailMessage(object client, string settingsName, bool addEvent, string subject, string body, int priority, bool isBodyHtml);**

где **client** – Клиент  
**settingsName** – наименование настроек Email Менеджера оповещений  
**addEvent** – признак добавление сообщения в протокол событий  
**subject** – тема письма  
**body** – тело письма  
**priority** – приоритет (0-высокий, 1-низкий, 2-нормальный)  
**isBodyHtml** – HTML формат тела письма.

### 4.3.5 SMS оповещение

**SMS оповещение** – это элемент объектной модели **DataRate**, позволяющий информировать с помощью технологии SMS о возникновении событий в графическом проекте.

**SMS оповещения** предоставляют возможность получать на мобильные устройства сообщения с информацией о работе системы.

**SMS оповещение** может быть реализовано с помощью:

- Отправки **SMS** через GSM модем (USB GSM модем, промышленный GSM модем, GSM мобильного телефона), подключенный по COM/USB порту к компьютеру, на котором функционирует Среда Исполнения **DataRate**
- Отправки **Email** через SMTP сервер провайдера мобильной связи.

Отправка сообщений осуществляется при изменении сводного состояния объекта, сигнального состояния тега в соответствии с серьезностью и категорией инициирующего события. Подсистема SMS оповещений формирует очередь сообщений на основе существующих принципов **Протокола событий**: категорий, серьезности и т.д.

Вы можете создать множество SMS оповещений для одного объекта, указав для каждого текст оповещения и способ отправки. Количество получателей SMS оповещений может

быть разным, поэтому существует возможность задать в **Менеджере оповещений** несколько настроек SMS.

Текст сообщения формируется на основе информации о сигнальном состоянии, по которому срабатывает данное оповещение. Вы можете полностью сформировать текст сообщения или выбрать его из предложенных шаблонов.

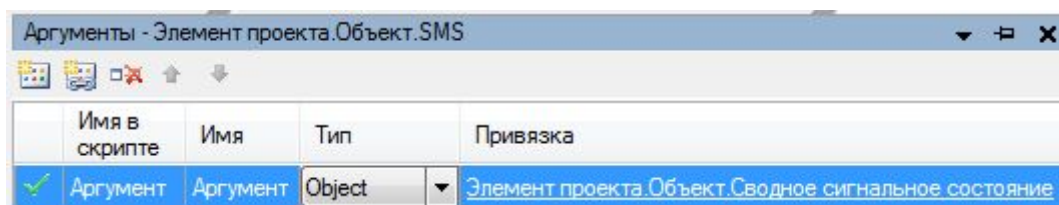
### ВНИМАНИЕ!!!

**Советуем обратиться к Вашему оператору мобильной связи для получения подробной информации об имеющихся корпоративных тарифах.**

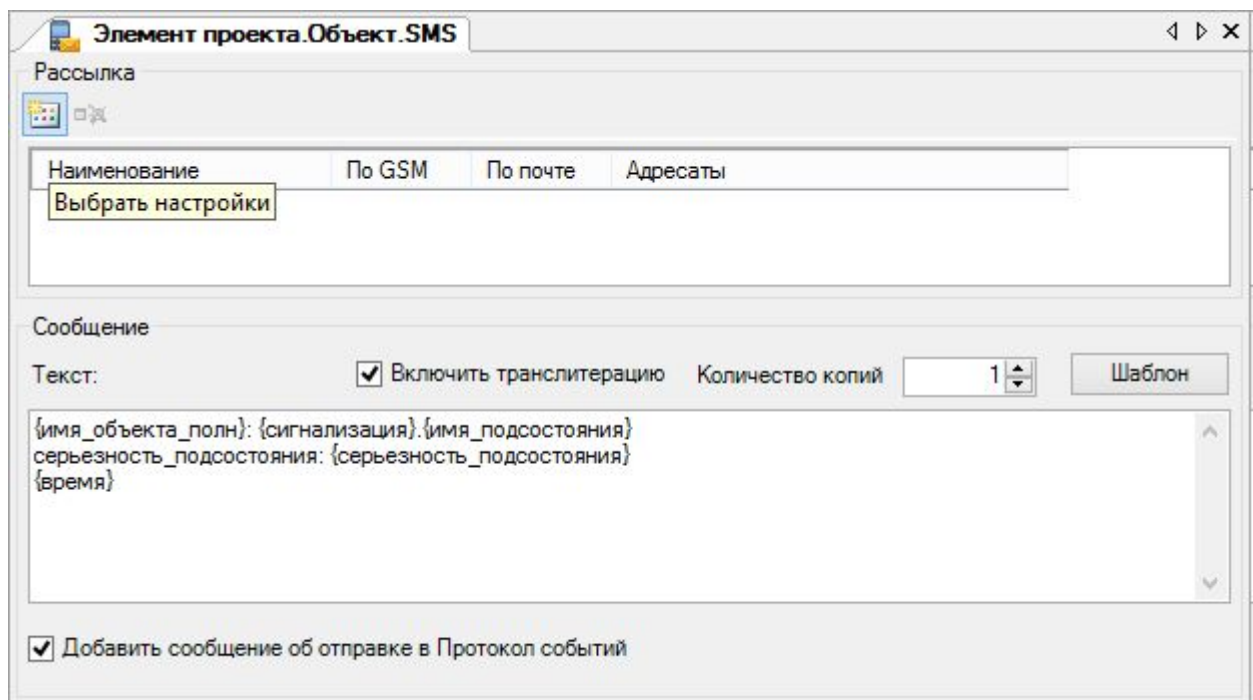
Для создания SMS оповещения следует создать объект оповещения, выбрав в категории **Основные** шаблон оповещения **SMS**, и задать его **Имя** (по умолчанию – **SMS**).

### Настройка SMS оповещения

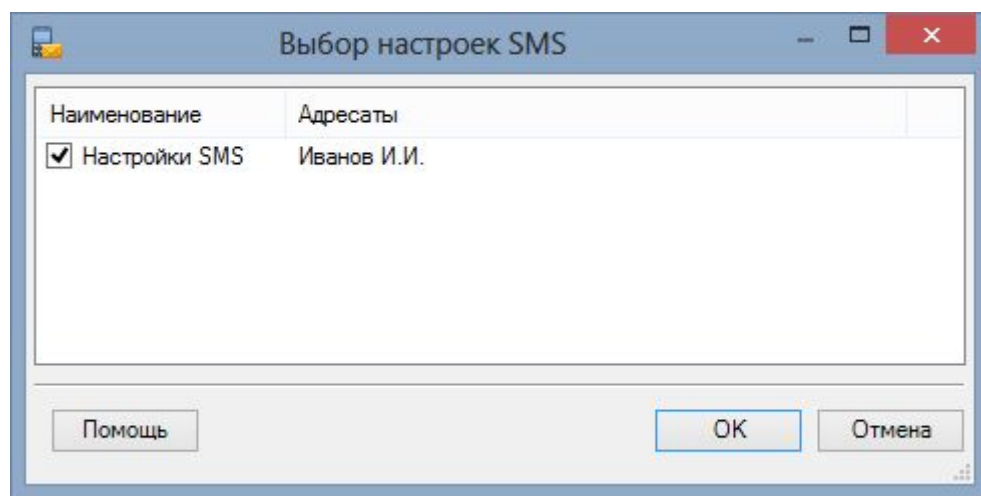
- 1 При создании объекта оповещения по умолчанию в окне **Аргументы** автоматически создаётся **Аргумент**, привязанный к **Сводному сигнальному состоянию** объекта, в котором создан объект **SMS**. При необходимости Вы можете привязать **Аргумент** к **Сводному сигнальному состоянию** (или его свойству) другого объекта или **Сигнальному состоянию** (или его свойству) тега. Для этого щёлкните по ссылке в колонке **Привязка**



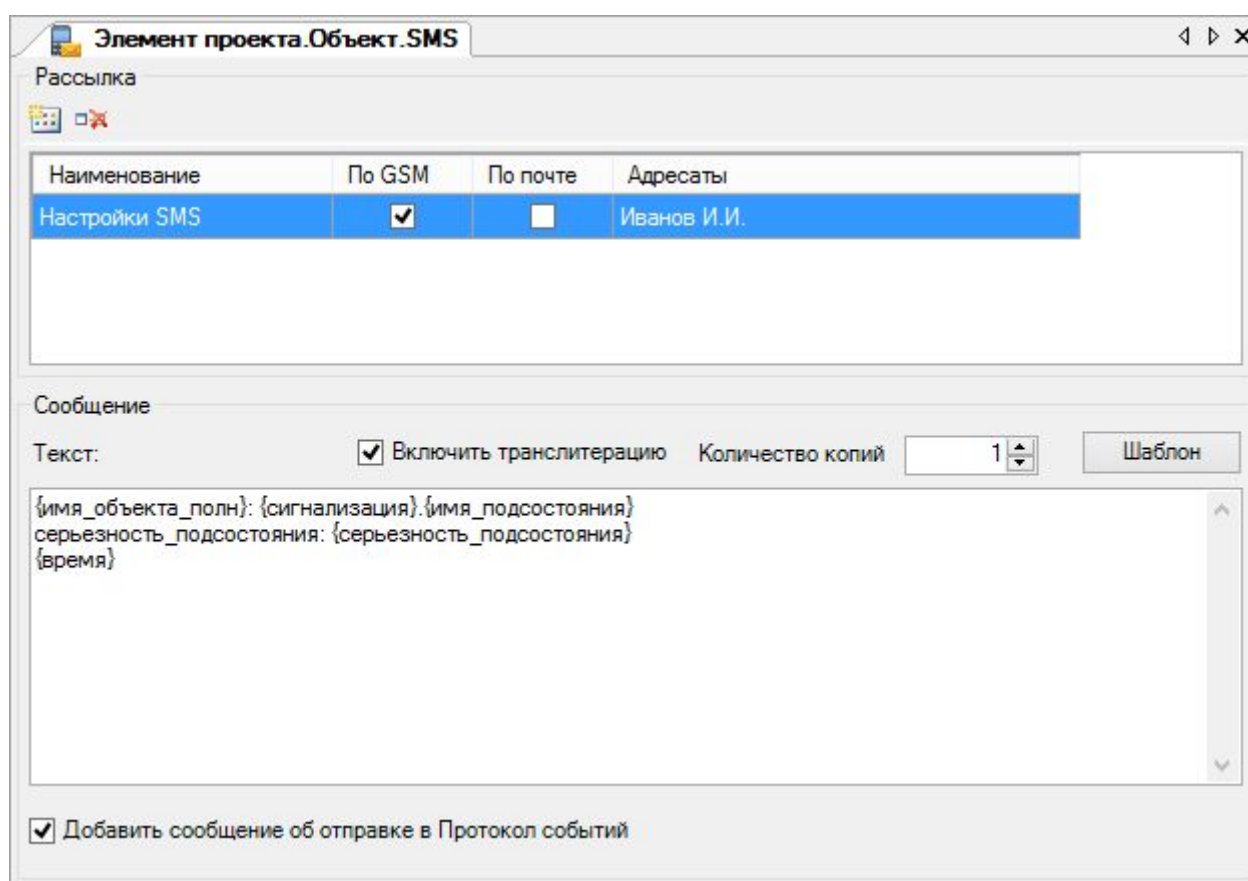
- 2 Для выбора настроек в поле редактирования в закладке **Элемент проекта.Объект. SMS** нажать на кнопку **Выбрать настройки**



- 3 В появившемся окне **Выбор настроек SMS** выбрать необходимые настройки и нажать на кнопку **OK**



- 4 Задать настройки группы **Рассылка**. Установите флаг **По GSM** для отправления SMS через GSM модем или флаг **По почте** для отправления SMS через email на SMTP сервер провайдера связи.



- 5 Задать настройки группы **Сообщение**. Установите флаг **Включить транслитерацию** для отправления сообщения латинскими символами. Использование данной опции позволит уместить больше информации в одном SMS.

**ВНИМАНИЕ**

SMS имеет ограничение по длине в 160 латинских символов, либо 70 символов кириллицей.

Для сегментированного (состоящего из нескольких частей) SMS максимальное количество символов составляет:

Сегментов	Количество символов	
	Кириллица	Латиница
1	70	160
2	134	306
3	201	459
4	268	612

Предусмотрена возможность дублировать сообщения, для этого задайте необходимое **Количество копий**. Использование данной опции может увеличить шансы доставки сообщения.

**ВНИМАНИЕ!**

SMS оповещение не является гарантированным способом своевременного получения актуальной информации.

Фирма не несёт ответственности за отправку и доставку сообщений.

Сформируйте **Текст** сообщения вручную или воспользуйтесь предложенными шаблонами. Для этого нажмите кнопку **Шаблон**. В частности, имеется возможность использования значений аргументов, ссылающихся на атрибуты тега.

**Шаблон**

```

{имя_объекта} - имя объекта
{имя_объекта_полн} - полное имя объекта
{тег} - имя тега
{сигнализация} - имя сигнализации
{доступность} - значение свойства "доступность" (как 0 или 1)
{подтверждено} - значение свойства "подтверждено" (как 0 или 1)
{активность} - значение свойства "активность" (как 0 или 1)
{комментарий} - значение свойства "комментарий"
{время_последней_активации} - время последней активации
{время_последнего_подтверждения} - время последнего подтверждения
{время_последней_деактивации} - время последней деактивации события
{качество} - значение свойства "качество"
{имя_подсостояния} - название активного подсостояния
{доступность_подсостояния} - доступность активного подсостояния
{серьезность_подсостояния} - серьезность активного подсостояния
{описание_подсостояния} - описание активного подсостояния
{формулировка_подсостояния} - формулировка активного подсостояния
{время} - время
Значения аргументов

```



- 6 Установить флаг **Добавить сообщение об отправке в Протокол событий** для создания соответствующей записи о результате отправки сообщения в **Протоколе событий**
- 7 В окне **Свойства** установить значение поля **Оповестить о выходе из активного состояния** в **Да**, если необходимо

Основные	
Имя	SMS
Библиотечный	Нет
Разрешено	Да
Оповестить о выходе из активного состояния	Да

- 8 Сохранить проект.

### ВНИМАНИЕ!!!

После настройки оповещения рекомендуется провести верификацию объекта (владельца оповещения) для выявления и исправления всех несоответствий

### Пример отправки SMS оповещения с помощью скрипта

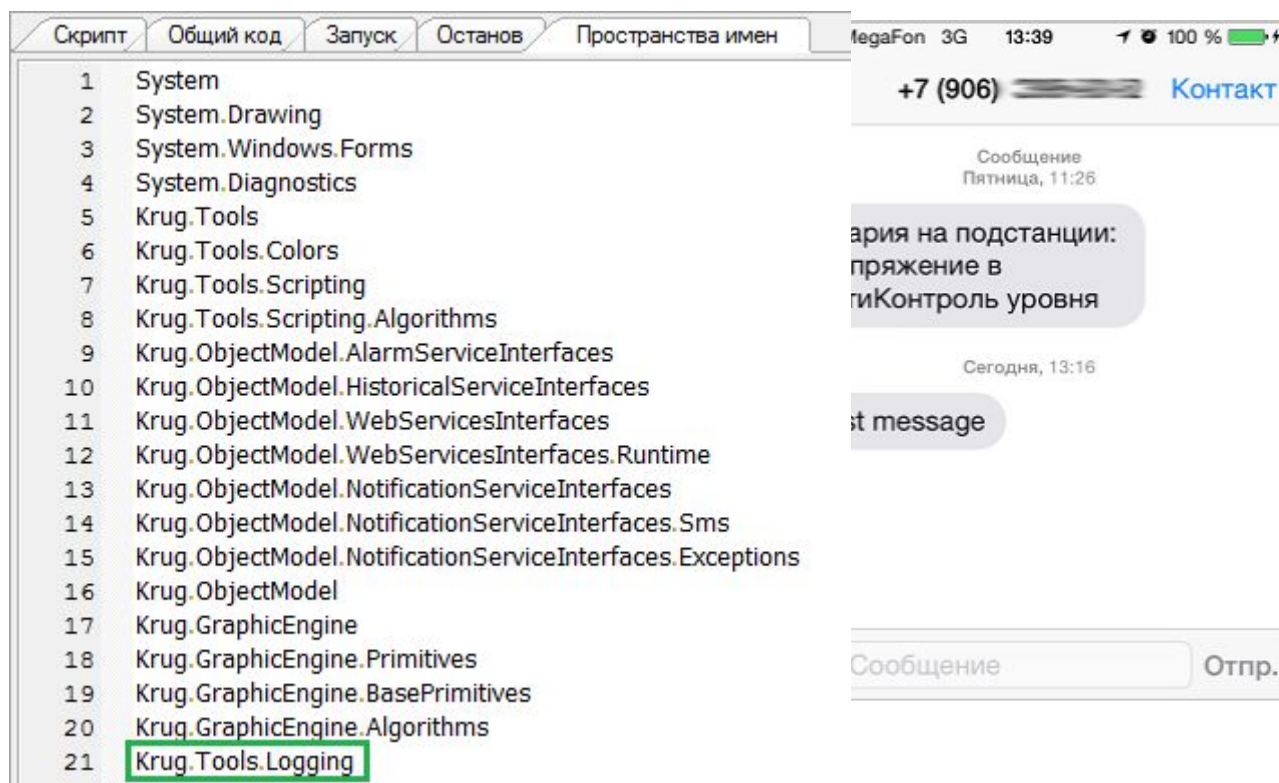
Для примера отправки SMS оповещения создадим мнемосхему с графическим примитивом, реакция которого на нажатие левой клавиши мыши запускает скрип отправки оповещения.


Для отправки SMS оповещения с помощью скрипта следует:

- 1 Создать **GSM модем**, указав имя **TestModem**
- 2 Создать объект с видом **Мнемосхема** и добавить на **Мнемосхему** любой графический примитив
- 3 Выделить созданный примитив и в окне **Свойства** перейдите на вкладку **Реакции**
- 4 Добавить реакцию **Скрипт** на событие **Нажатие левой кнопки** и нажать на кнопку **Редактировать...**
- 5 Отредактировать скрипт:
  - Перейти на вкладку **Пространства имен** и добавить **Krug.Tools.Logging**
  - Перейти на вкладку **Скрипт** и добавить следующий код:

```
try
{
    SmsNotificationHelper.SendMessage(Client, "TestModem",
        "+79271234567", "Test message", true, true, 1);
}
catch (NotificationException x)
{
    Log.TraceError(NotificationServiceLog.Trace, x, null, true);
}
```

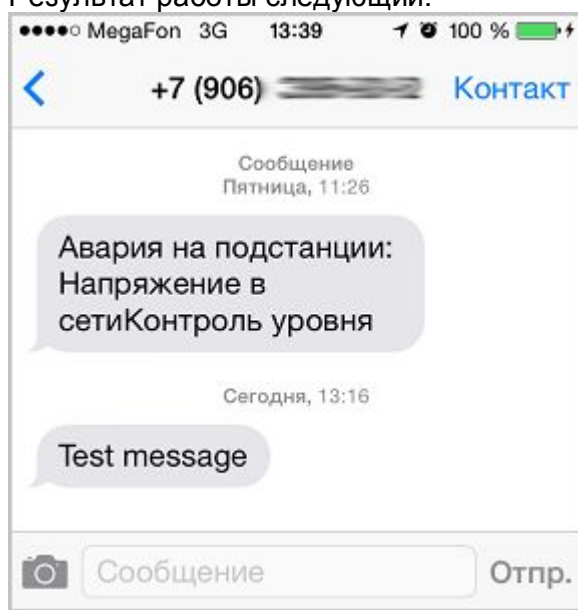
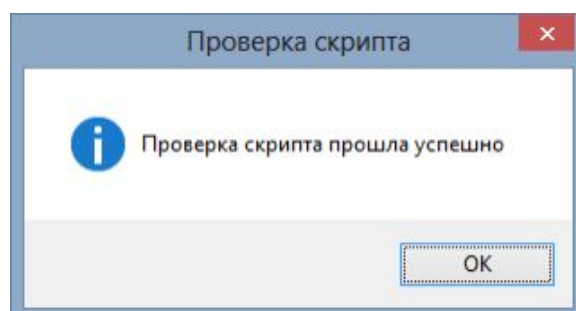
Подробное описание методов для отправки SMS с помощью скрипта приведено далее.



Нажать на кнопку **Проверить скрипт** , чтобы убедиться, что скрипт написан корректно

6 Сохранить проект

7 Запустить проект на исполнение, открыть мнемосхему и щелкнуть левой кнопкой мыши по созданному примитиву. Результат работы следующий:



Запись в окне **Протокол событий**:

Время	Качество	Серьезность	Название серьезности	Сообщение
28.07.2017 17:20:42	Хорошее	0	0	SMS оповещение: 'SMS' отправлено по протоколу GSM.
28.07.2017 17:20:40	Плохое	500	Системная	Текущее значение в норме
28.07.2017 17:20:20	Хорошее	0	0	SMS оповещение: 'SMS' отправлено по протоколу GSM.
28.07.2017 17:20:15	Хорошее	1000	Очень высокая	Текущее значение превысило порог верхней предостерегающей границы
28.07.2017 17:20:09	Хорошее	0	0	SMS оповещение: 'SMS' отправлено по протоколу GSM.
28.07.2017 17:20:05	Хорошее	800	800	Текущее значение превысило порог верхней предостерегающей границы
28.07.2017 17:19:13	Хорошее	0	0	SMS оповещение: 'SMS' отправлено по протоколу GSM.
28.07.2017 17:19:11	Плохое	500	Системная	Текущее значение в норме
28.07.2017 17:18:47	Плохое	400	Высокая	Повышенное значение

Фильтрация выключена. Событий: 9 | Интервал времени: 01.01.0001 4:00:00 - текущее время

## Методы для отправки SMS с помощью скрипта

Для отправки **SMS** используется класс **SmsNotificationHelper**.

### Проверка подключения указанного GSM модема

**bool** CheckGsmModem(**object** client, **string** modemName);

где **client** — Клиент  
**modemName** — наименование модема

Возвращает **true**, если модем подключен, иначе — **false**.

### Запрос баланса указанного GSM модема

**string** GetBalance(**object** client, **string** modemName);

где **client** — Клиент  
**modemName** — наименование модема

Возвращает строку баланса, переданную оператором сотовой связи.

## Методы отправки SMS с настройками, указанными по умолчанию

### Отправка SMS с указанным текстом.

**void** SendMessage(**object** client, **string** messageText);

где **client** — Клиент  
**messageText** — текст сообщения

### Отправка SMS с указанным текстом и опцией транслитерации

**void** SendMessage(**object** client, **string** messageText, **bool** useTransliteration);

где **client** — Клиент  
**messageText** — текст сообщения  
**useTransliteration** — признак транслитерации

Отправка SMS с указанным текстом и опцией транслитерации в указанном количестве копий

```
void SendMessage(object client, string messageText, bool useTransliteration,
                 int copyCount);
```

где client – Клиент  
messageText – текст сообщения  
useTransliteration – признак транслитерации  
copyCount – количество копий сообщения

Отправка SMS с указанным текстом и опциями транслитерации и добавления события в Протокол событий

```
void SendMessage(object client, string messageText, bool useTransliteration,
                 bool addEventAfterSent);
```

где client – Клиент  
messageText – текст сообщения  
useTransliteration – признак транслитерации  
addEventAfterSent – признак добавления сообщения в Протокол событий

Отправка SMS с указанным текстом, опциями транслитерации и добавления события в Протокол событий в указанном количестве копий.

```
void SendMessage(object client, string messageText, bool useTransliteration,
                 bool addEventAfterSent, int copyCount);
```

где client – Клиент  
messageText – текст сообщения  
useTransliteration – признак транслитерации  
addEventAfterSent – признак добавления сообщения в Протокол событий  
copyCount – количество копий сообщения

#### Методы отправки SMS с указанными настройками

Отправка SMS с указанным текстом

```
void SendMessage(object client, string settingsName, string messageText);
```

где client – Клиент  
settingsName – имя настройки SMS Менеджера оповещений  
messageText – текст сообщения

Отправка SMS с указанным текстом и опцией транслитерации

```
void SendMessage(object client, string settingsName, string messageText,
                 bool useTransliteration);
```

где client – Клиент  
settingsName – имя настройки SMS Менеджера оповещений  
messageText – текст сообщения  
useTransliteration – признак транслитерации

**Отправка SMS с указанным текстом и опцией транслитерации в указанном количестве копий**

```
void SendMessage(object client, string settingsName, string messageText,  
                bool useTransliteration, int copyCount);
```

где client – Клиент  
settingsName – имя настройки SMS Менеджера оповещений  
messageText – текст сообщения  
useTransliteration – признак транслитерации  
copyCount – количество копий сообщения

**Отправка SMS с указанным текстом и опциями транслитерации и добавления события в Протокол событий**

```
void SendMessage(object client, string settingsName, string messageText,  
                bool useTransliteration, bool addEventAfterSent);
```

где client – Клиент  
settingsName – имя настройки SMS Менеджера оповещений  
messageText – текст сообщения  
useTransliteration – признак транслитерации  
addEventAfterSent – признак добавления сообщения в Протокол событий

**Отправка SMS с указанным текстом, опциями транслитерации и добавления события в Протокол событий в указанном количестве копий**

```
void SendMessage(object client, string settingsName, string messageText,  
                bool useTransliteration, bool addEventAfterSent,  
                int copyCount);
```

где client – Клиент  
settingsName – имя настройки SMS Менеджера оповещений  
messageText – текст сообщения  
useTransliteration – признак транслитерации  
addEventAfterSent – признак добавления сообщения в Протокол событий  
copyCount – количество копий сообщения

### Методы отправки SMS на указанный номер

**Отправка SMS с указанным текстом на указанный номер**

```
void SendMessage(object client, string modemName, string phoneNumber,  
                string messageText);
```

где client – Клиент  
modemName – наименование GSM модема из списка GSM устройств Менеджера оповещений  
phoneNumber – номер телефона адресата  
messageText – текст сообщения



Отправка SMS с указанным текстом и опцией транслитерации на указанный номер

```
void SendMessage(object client, string modemName, string phoneNumber,
                 string messageText, bool useTransliteration);
```

где client – Клиент  
 modemName – наименование GSM модема из списка GSM устройств  
                   Менеджера оповещений  
 phoneNumber – номер телефона адресата  
 messageText – текст сообщения  
 useTransliteration – признак транслитерации

Отправка SMS с указанным текстом и опцией транслитерации в указанном количестве копий на указанный номер

```
void SendMessage(object client, string modemName, string phoneNumber,
                 string messageText, bool useTransliteration, int copyCount);
```

где client – Клиент  
 modemName – наименование GSM модема из списка GSM устройств  
                   Менеджера оповещений  
 phoneNumber – номер телефона адресата  
 messageText – текст сообщения  
 useTransliteration – признак транслитерации  
 copyCount – количество копий сообщения

Отправка SMS с указанным текстом, опциями транслитерации и добавления события в Протокол событий на указанный номер

```
void SendMessage(object client, string modemName, string phoneNumber,
                 string messageText, bool useTransliteration,
                 bool addEventAfterSent);
```

где client – Клиент  
 modemName – наименование GSM модема из списка GSM устройств  
                   Менеджера оповещений  
 phoneNumber – номер телефона адресата  
 messageText – текст сообщения  
 useTransliteration – признак транслитерации  
 addEventAfterSent – признак добавления сообщения в Протокол событий

Отправка SMS с указанным текстом, опциями транслитерации и добавления события в Протокол событий в указанном количестве копий на указанный номер

```
void SendMessage(object client, string modemName, string phoneNumber,
                 string messageText, bool useTransliteration,
                 bool addEventAfterSent, int copyCount);
```

где client – Клиент  
 modemName – наименование GSM модема из списка GSM устройств  
                   Менеджера оповещений  
 phoneNumber – номер телефона адресата  
 messageText – текст сообщения  
 useTransliteration – признак транслитерации  
 addEventAfterSent – признак добавления сообщения в Протокол событий  
 copyCount – количество копий сообщения.

Методы для работы со списком настроек.

**Получение списка имён GSM устройств Менеджера оповещений**

**string[ ] GetModemNames(object client);**

где **client** – Клиент

Возвращает список имён GSM устройств Менеджера оповещений.

**Получение списка имён групп настроек SMS Менеджера оповещений**

**string[ ] GetSettingsNames(object client);**

где **client** – Клиент

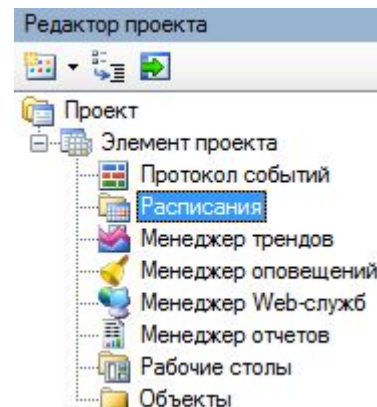
Возвращает список имён групп настроек SMS Менеджера оповещений.

## 5 РАСПИСАНИЯ

**Расписание** – это совокупность правил запуска определенных действий в процессе исполнения проекта.

Для формирования расписаний используется системный объект **Расписания**, который автоматически добавляется при создании элемента проекта.

**DataRate** обеспечивает привязку скриптов к расписаниям. В процессе исполнения проекта скрипт, для которого настроена привязка к расписанию, вызывается всякий раз, когда наступает время активации в соответствии с заданными в расписании параметрами.

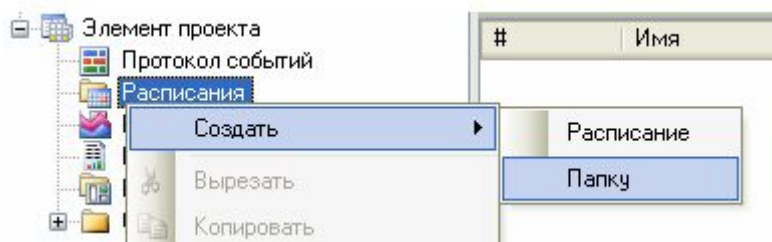


### 5.1 Папки расписаний

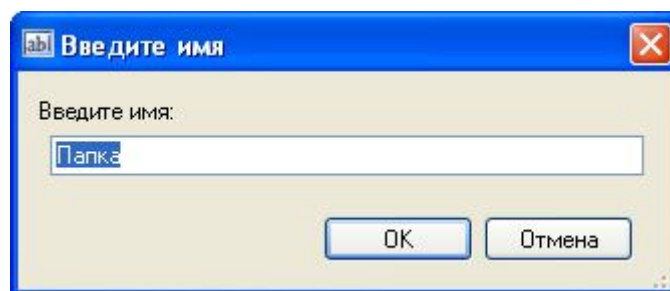
**Папки расписаний** используются для упорядочивания множества расписаний, входящих в элемент проекта.

Для создания папки следует:

1. Выбрать **Создать\Папку** в контекстное меню элемента проекта **Расписания**



2. В появившемся диалоговом окне указать имя новой папки.



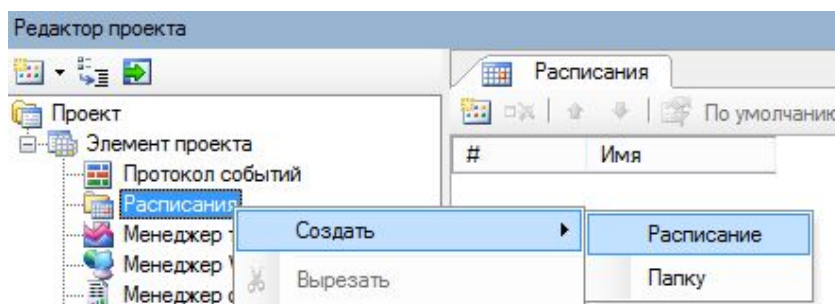
Созданная папка расписаний отображается в дереве элемента проекта, как вложенный элемент узла **Расписания**, а также в списке дочерних элементов в правой части окна редактора проекта, когда курсор в дереве установлен на родительском узле.



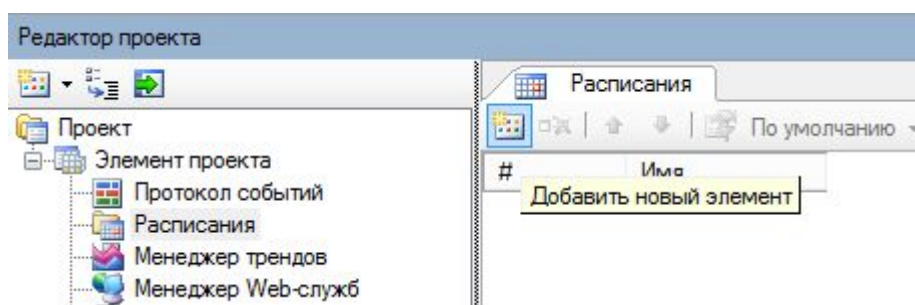
## 5.2 Создание расписания

Для создания расписания следует:

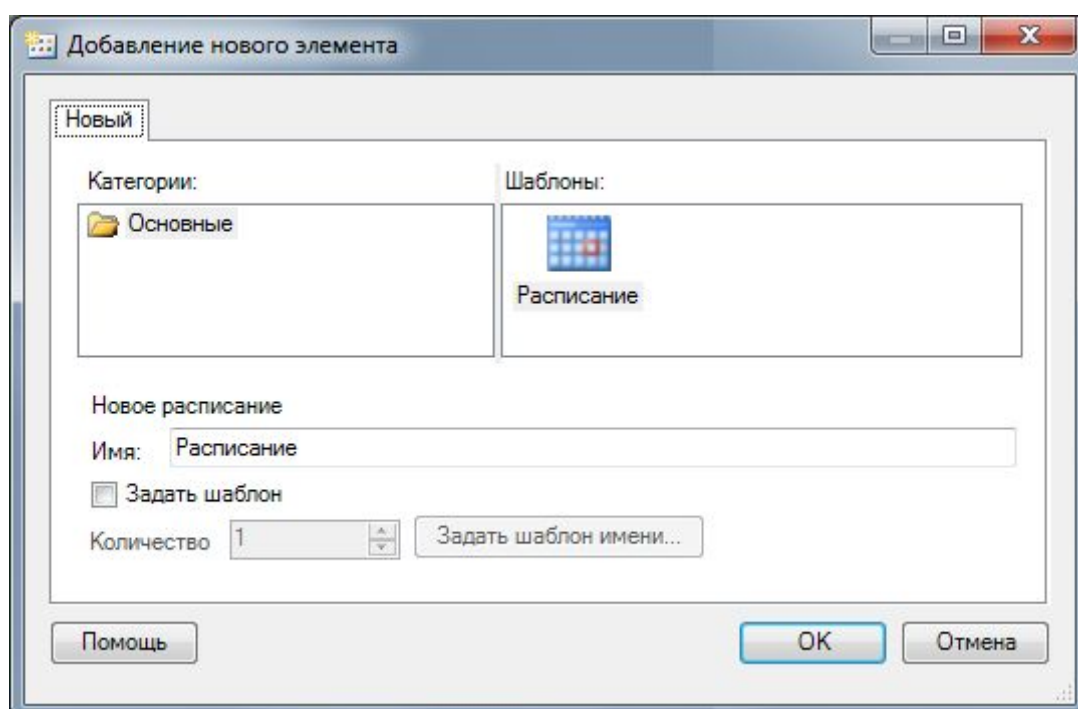
1. Выбрать команду **Создать\Расписание** в контекстном меню элемента проекта **Расписания** или папки этого элемента



или выбрать элемент проекта **Расписания** и нажать на кнопку **Добавить новый элемент**

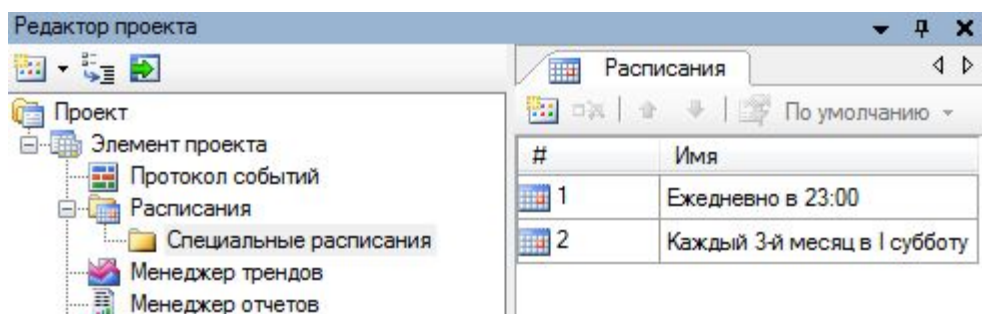


2. В появившемся окне **Добавление нового элемента** задать **Имя** создаваемого расписания. Имя должно быть уникальным среди дочерних элементов узла дерева, в котором это расписание создается
3. Нажать **ОК** и определить параметры создаваемого расписания в окне редактирования расписания.



После завершения редактирования расписание добавляется как дочерний элемент того узла дерева проекта **DataRate**, для которого была вызвана команда **Создать Расписание**.


Поскольку сами расписания не могут содержать дочерних элементов, они не отображаются в дереве проекта. Расписания отображаются в списке дочерних элементов узла дерева в правой части окна редактора проекта, когда курсор в дереве установлен на родительском узле.

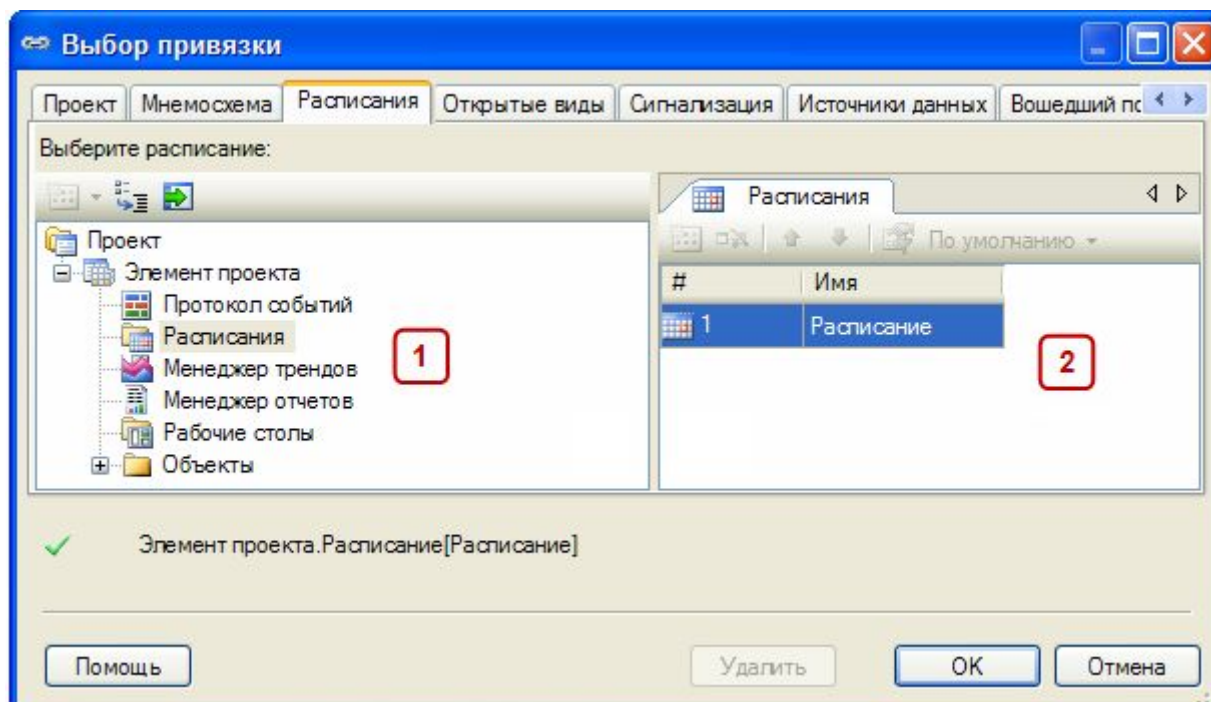


### 5.3 Привязка к расписанию

Для привязки аргумента к расписанию предназначена вкладка **Расписания** окна **Выбор привязки**.

Для привязки аргумента к расписанию следует:

- 1 Вызвать окно **Выбор привязки**. Для этого в окне **Аргументы** (вызов по кнопке  панели инструментов) выделить ячейку в колонке **Привязки** и нажать на кнопку в ячейке
- 2 Выбрать вкладку **Расписания**
- 3 Выделить папку **Расписания** в дереве проекта (1)
- 4 Выбрать нужное расписание в списке дочерних элементов (2)





### 5.4 Редактирование свойств расписания

Создаваемые в **DataRate** расписания можно классифицировать следующим образом:

- Расписания с однократной активацией (**по умолчанию**)
- Расписания с повторяющейся активацией.

Редактирование свойств расписания осуществляется в окне **Расписание**. Его внешний вид и элементы интерфейса Пользователя определяются типом расписания.

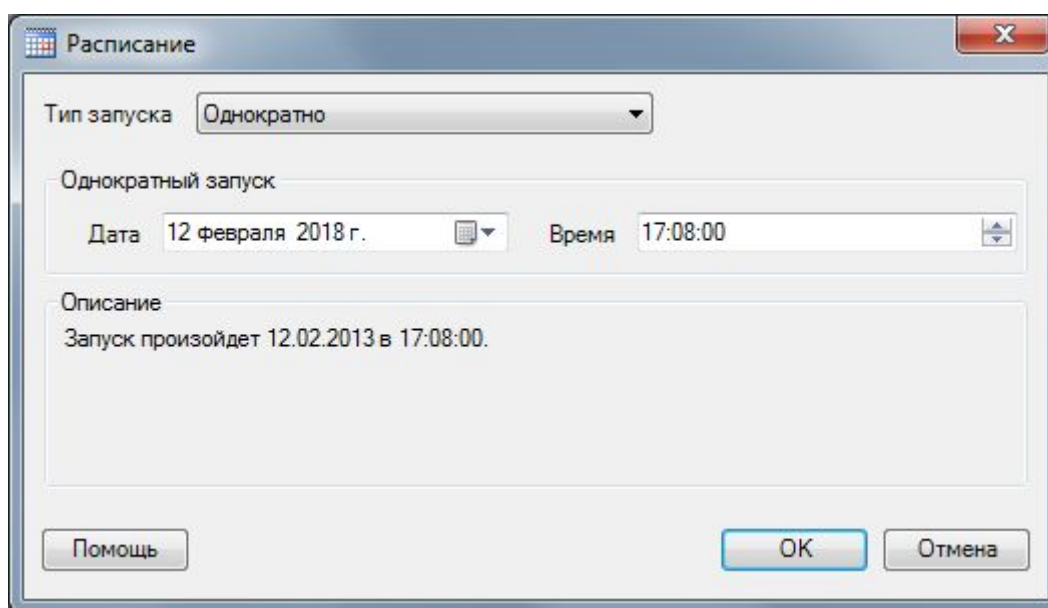
#### 5.4.1 Расписание с однократной активацией

**Расписание с однократной активацией** – это расписание для однократного запуска планируемого действия.

##### Свойства расписания

В окне **Расписание** следует задать:

- **Тип запуска** – **Однократно**
- **Дату** и **Время** однократного запуска планируемого действия



В нижней части окна **Расписание** отображается словесная интерпретация заданных параметров расписания.

#### 5.4.2 Расписание с повторяющейся активацией

**Расписание с повторяющейся активацией** – это расписание для многократного запуска планируемых действий.

Периодичность активации планируемых действий может задаваться в днях, в неделях, в месяцах и в годах.

По способу задания суточной периодичности расписания с повторяющейся активацией можно подразделить на:

- Расписания с однократной активацией в течение суток
- Расписания с повторяющейся активацией в течение суток.

Для расписания с повторяющейся активацией может быть определен срок действия, по истечении которого расписание перестает активироваться.

### Свойства расписания

В окне **Расписание** следует задать тип запуска, параметры периодичности и продолжительности.

**Периодичность** запуска запланированного действия может задаваться в днях, неделях, месяцах или годах. В зависимости от того, какая единица времени выбрана Пользователем, меняется набор параметров периодичности.

**Продолжительность** действия расписания может быть задана датой начала и датой окончания.

#### 5.4.2.1 Задание периодичности

**Периодичность запуска** – свойство **Единица отсчета** – запланированного действия может задаваться в днях, неделях, месяцах или годах.

В зависимости от того, какая единица времени выбрана Пользователем, меняется набор параметров периодичности.

### Периодичность в днях

При задании периодичности в днях секция «Периодичность» имеет следующий вид:

The screenshot shows a form titled 'Периодичность' (Periodicity). It contains a dropdown menu labeled 'Единица отсчета' (Unit of measurement) with 'дни' (days) selected. Below it, there is a label 'Повторять каждый' (Repeat every) followed by a numeric input field containing '1' and a unit label '-й день' (th day).

Свойства расписания:

- **Число дней** между повторяющимися запусками.

### Периодичность в неделях

При выборе недели в качестве единицы периодичности секция **Периодичность** принимает следующий вид:

The screenshot shows the 'Периодичность' (Periodicity) section with 'недели' (weeks) selected in the 'Единица отсчета' (Unit of measurement) dropdown. Below, 'Повторять каждую' (Repeat every) is set to '1' week. There are seven checkboxes for the days of the week: Понедельник (Monday), Среда (Wednesday), Пятница (Friday), Суббота (Saturday), Вторник (Tuesday), Четверг (Thursday), and Воскресенье (Sunday). The 'Воскресенье' checkbox is checked.

Свойства расписания:

- **Число недель** между повторяющимися запусками
- **Дни недели**, по которым будет производиться запуск запланированного задания.

### Периодичность в месяцах

При выборе месяца в качестве единицы периодичности секция **Периодичность** принимает следующий вид:

The screenshot shows the 'Периодичность' (Periodicity) section with 'месяцы' (months) selected in the 'Единица отсчета' (Unit of measurement) dropdown. There are two options for scheduling: 'день месяца' (day of the month) and 'день недели' (day of the week). The 'день месяца' option is selected with a radio button. It is configured as '1' day of the month, 'каждого' (every) '1' month. The 'день недели' option is unselected and shows '1-й' (1st) day, 'понедельник' (Monday), 'каждого' (every) '1' month.

Свойства расписания:

- **Число месяцев** между повторяющимися запусками
- **День месяца** для активизации расписания. Может быть задан как фиксированное **число месяца** или же путем задания **дня недели** и его **порядкового номера** в рамках месяца. В обоих случаях пользователь также определяет интервал в месяцах между датами активизации расписания.

### Периодичность в годах

При выборе года в качестве единицы секция **Периодичность** принимает следующий вид:

**Периодичность**

Единица отсчета: годы

Каждый: 1 год, месяц: январь

☒ день месяца: 1

☐ день недели: 1-й понедельник месяца

Свойства расписания:

- **Число лет** между повторяющимися запусками
- **Месяц** в году, во время которого будет осуществляться активация расписания
- **День** в рамках месяца. Может задаваться как **фиксированное число** или путем указания **дня недели** и его **порядкового номера** в рамках месяца.

#### 5.4.2.2 Периодичность в день запуска

Секция **Периодичность в день запуска** позволяет настраивать время активации расписания в течение суток.

В течение дня, на который попадает активация расписания в соответствии с настройкой периодичности, расписание может быть активировано:

- **Один раз** – для этого требуется указать **фиксированное время запуска**
- **Множество раз** – для этого необходимо определить **периодичность активации в день запуска**.

#### Определение времени однократной активации

В режиме однократной активизации Пользователю предлагается установить время однократной активизации в течение суток.

**Периодичность в день запуска**

☒ Запустить один раз в: 12:00:00 AM

☐ Запускать каждый(ую): 1  с 12:00:00 AM по 11:59:59 PM

Свойства расписания:

- **Запустить один раз в** – время однократного запуска. При таком варианте настройки расписание в день активизации будет активировано только один раз в заданное время.

#### Определение периодичности активации в день запуска

В режиме многократной активизации следует задать параметры, определяющие правило периодической активизации в течение суток.

**Периодичность в день запуска**

☐ Запустить один раз в: 12:00:00 AM

☒ Запускать каждый(ую): 1 час с 12:00:00 AM по 11:59:59 PM

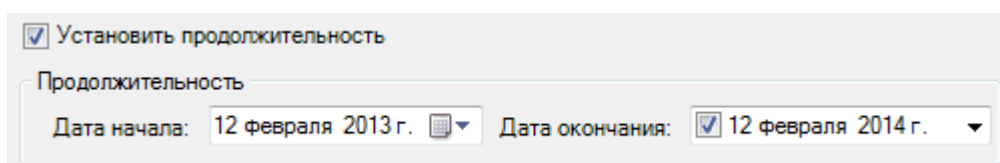
Свойства расписания:

- **Запускать каждый(ую)** – время между повторяющимися запусками. Периодичность, с которой производится повторяющаяся активизация расписания, может быть задана в часах, минутах, секундах или миллисекундах (при использовании миллисекунд Пользователь может задавать значения, кратные 100 миллисекундам)
- **Временной отрезок в течение суток** – временной интервал, в рамках которого будет действовать заданное правило периодической активации.

### 5.4.2.3 Продолжительность действия расписания

**Продолжительность действия расписания** может быть ограничена путем задания даты начала и окончания срока действия данного расписания.

Для задания продолжительности действия расписания необходимо установить галочку в поле **Установить продолжительность**. При установленной галочке в форму редактирования расписания добавляется секция **Продолжительность**.

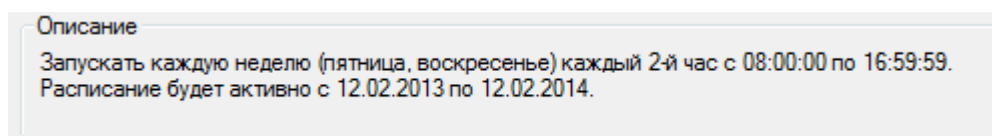


Установка продолжительности действия расписания осуществляется в соответствии со следующими правилами:

- Если галочка **Установить продолжительность** снята, то расписание имеет срок действия, начинающийся с момента запуска проекта на исполнение и заканчивающийся его остановкой
- Если галочка **Установить продолжительность** установлена, а галочка в поле **Дата окончания** – нет, то срок действия расписания начинается с даты, заданной в поле **Дата начала** (или с момента запуска проекта, если запуск произошел уже после установленной даты начала), и продолжается вплоть до остановки проекта
- Если галочка **Установить продолжительность** установлена и также установлена галочка в поле **Дата окончания**, то срок действия расписания начинается с даты, заданной в поле **Дата начала** (или с момента запуска проекта, если запуск произошел уже после установленной даты начала), и продолжается до даты, заданной в поле **Дата окончания** включительно. Если запуск проекта произведен после даты, заданной в поле **Дата окончания**, то расписание в процессе выполнения проекта будет неактивным.

### 5.4.2.4 Словесное описание

В окне **Расписание** в поле **Описание** секции **Отчет** выводится словесная интерпретация заданных параметров расписания:





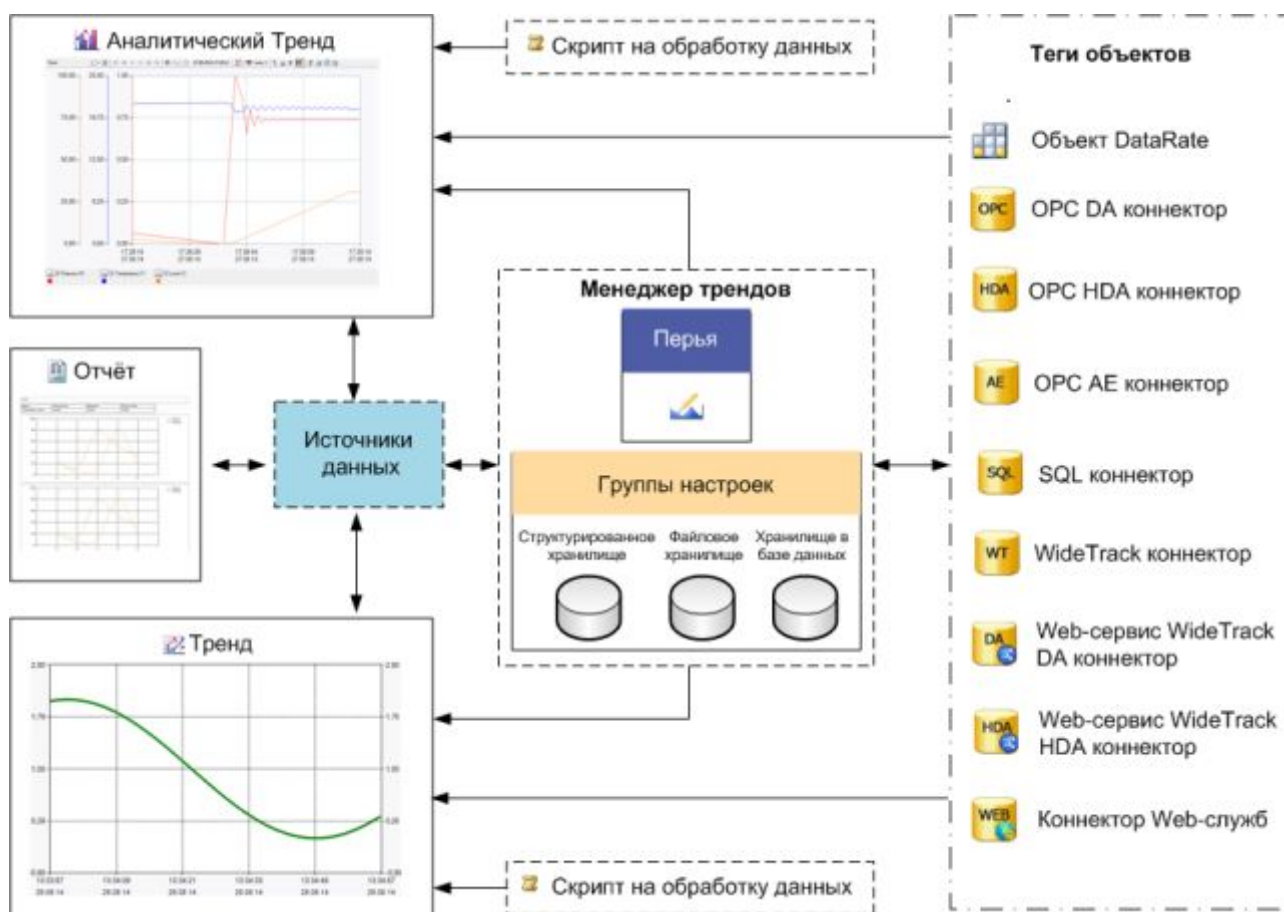
## 6 ИСТОРИЯ ПРОЦЕССА. ТРЕНДЫ

**Тренд** (от англ. Trend - тенденция) – общая направленность изменений показателей любого временного ряда.

**Временной ряд** – это упорядоченная по времени последовательность значений некоторой переменной величины.

Эти понятия интерпретируются в **DataRate** следующим образом:

- **Тренд** – временной ряд атрибута тега
- **Менеджер трендов** – системный сервис (системный объект), осуществляющий формирования истории процесса; входит в состав элемента проекта
- **Оперативный тренд** – история изменения оперативных данных системы; описывается в элементе проекта с помощью пера
- **Перо** – специальный объект, содержащий данные о трендируемом атрибуте тега
- **Исторические тренды** – **архивы трендов**; описываются в элементе проекта с помощью перьев и **Группы настроек истории**; сохраняются в файле с расширением **history**.



Для визуализации трендов в **DataRate** используются графические элементы **Тренд** и **Табличный тренд**, специализированный вид – **Аналитический тренд**.

**Режимный тренд** позволяет одновременно отображать базовое (эталонное) и текущее состояние процесса.

Результаты трендирования можно отображать и в отчетах.

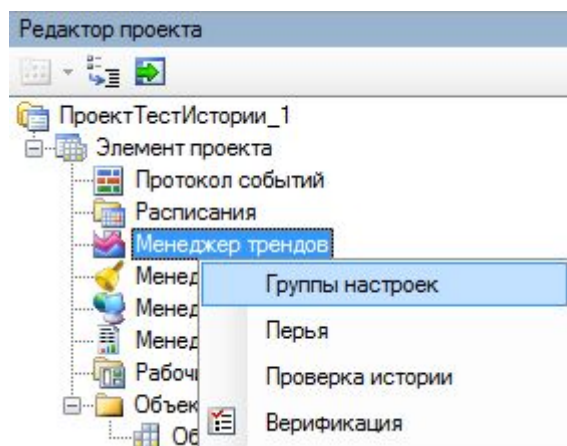
### 6.1 Менеджер трендов

Модуль **Менеджер трендов** предназначен для ведения трендов атрибутов тегов объектов.

Менеджер трендов автоматически добавляется при создании **Элемента проекта**.

Для настройки модуля Менеджер трендов следует:

1. Выбрать в редакторе проекта узел дерева **Проект/Элемент проекта/Менеджер трендов**
2. Вызвать контекстное меню, щелкнув правой клавишей мыши на иконке **Менеджера трендов**
3. Выбрать в контекстном меню необходимый пункт:
  - **Группы настроек** открывает диалоговое окно **Группы настройки истории**, в котором можно добавлять, удалять и редактировать группы настроек хранения истории
  - **Перья** открывает диалоговое окно **Перья**, в котором можно добавлять и удалять перья **Менеджера трендов**
  - **Проверка истории** предназначена для сравнения содержимого хранилища трендов и соответствующего ему резервного хранилища трендов с целью проверки их идентичности или обнаружения различий
  - **Верификация** запускает верификацию данных **Менеджера трендов**.



#### 6.1.1 Группы настроек истории

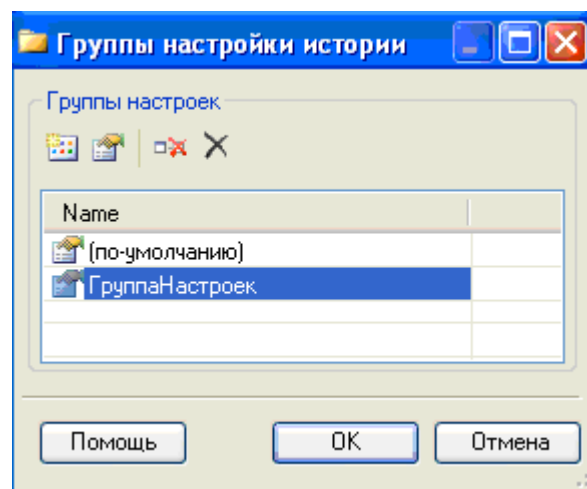
**Группа настроек истории** представляет собой совокупность параметров, управляющих сохранением истории пера **Менеджера трендов**, для которого задана данная группа настроек.

Для редактирования групп настроек истории следует:





1. Вызвать контекстное меню **Менеджера трендов** (двойным щелчком левой клавишей мыши на иконке Менеджера трендов или щелчком правой клавишей мыши)
2. Выбрать пункт **Группы настроек**
3. В появившемся окне **Группы настройки истории** можно добавлять, удалять и редактировать группы настроек истории.

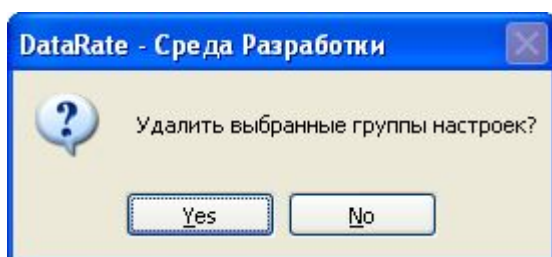
## 6.1.1.1 Редактирование групп настройки истории

Для редактирования групп настройки истории предназначено окно **Группы настройки истории**.

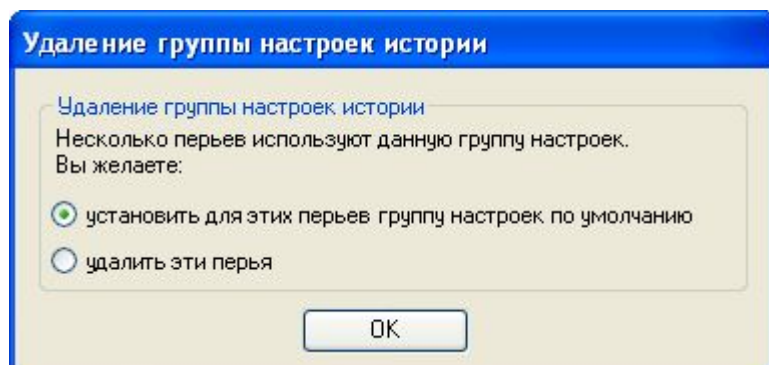


Для управления списком групп настроек истории используйте **кнопки панели инструментов**:

-  – **Добавить группу настроек**, добавляет новую группу настроек и открывает диалоговое окно **Группа настроек** для редактирования этой группы;
-  – **Показать данные группы настроек**, открывает диалоговое окно **Группа настроек** для просмотра и редактирования выделенной группы настроек;
-  – **Удалить выделенные группы настроек**, удаляет выделенные группы настроек, если среди выделенных групп есть группа «**(по умолчанию)**», она не удаляется;
-  – **Удалить все группы настроек**, удаляет все группы настроек, кроме группы «**(по умолчанию)**».



При удалении группы настроек появляется диалоговое окно подтверждения удаления.



Если пользователь подтверждает удаление, то, если удаляемая группа настроек используется перьями **Менеджера трендов**, выводится диалоговое окно **Удаление группы настроек истории**, которое позволяет либо присвоить перьям группу настроек **по умолчанию**, либо **удалить эти перья**.

### 6.1.1.2 Редактирование группы настроек

Для редактирования группы настроек истории следует:

- 1 Выделить группу настроек в окне **Группы настройки истории**
- 2 Нажать на кнопку **Показать данные группы настроек** (или дважды щелкнуть левой клавишей мыши на выделенной группе настроек)
- 3 В появившемся окне **Группа настроек** отредактировать требуемые параметры хранения истории и нажать на кнопку **ОК**.

#### Основные свойства

- **Имя группы** – позволяет задать имя группы настроек.
- **Сохранение истории включено** – определяет, будут ли данные сохраняться на диск
- **Тип хранилища** позволяет выбрать, в каком виде будет храниться данные трендов:
  - **Структурированное хранилище**. Единое хранилище для данных всех перьев. Использование такого типа хранилища целесообразно в проектах с

небольшим количеством перьев. Такое хранилище удобнее при перемещении проекта.

- **Файловое хранилище.** Для каждого пера создается отдельный файл, который помещается в папку для группы настроек. Использование такого типа хранилища целесообразно при количестве перьев свыше 5000 и при появлении новой точки тренда раз в секунду
- **Хранилище в базе данных.** При хранении трендов в базе данных используется подключение к БД и создается хранилище по указанной строке подключения. Хранилище трендов в базе данных является высокоскоростным механизмом работы с трендами. Использование хранилища трендов в БД целесообразно при большом количестве перьев свыше 1000 с большой частотой изменения данных (250мс). БД позволяет хранить большие объемы данных рассчитанные на длительное хранение во внешнем хранилище СУБД.
- **Абсолютный или относительный путь** – путь к файлу с историей позволяет задать абсолютный или относительный путь к файлу, в котором будет храниться история процесса перьев **Менеджера трендов**, для которых задана данная группа настроек
- **Выбрать** – открывает стандартное диалоговое окно выбора файла. Поле **Абсолютный путь** всегда показывает абсолютный путь к файлу.

**Использовать резервное хранилище трендов** (признак **Сохранение истории** включено установлен)

- **Абсолютный или относительный путь** – путь к файлу с резервным хранилищем истории позволяет задать абсолютный или относительный путь к файлу, в котором будет храниться копия истории процесса перьев **Менеджера трендов**, для которых задана данная группа настроек
- **Выбрать** – открывает стандартное диалоговое окно выбора файла. Поле **Абсолютный путь** всегда показывает абсолютный путь к файлу

**Оперативный буфер** (признак **Сохранение истории** включено установлен)

- **Период сохранения на диск** – период сохранения данных на диск в секундах
- **Размер оперативного буфера** – максимальный размер оперативного буфера (хранящегося в оперативной памяти) в точках.

**Ограничения на ведение истории** (признак **Сохранение истории** включено установлен)

- **Ограничение по частоте изменения значения** – минимальный период сохранения событий в миллисекундах
- **Ограничение по размеру файла** – максимальный размер файла хранения истории в мегабайтах для данной группы настроек
- **Ограничение по количеству точек** – максимальное количество точек истории, хранящееся в файле
- **Ограничение по дням** – максимальное количество дней, за которое будет храниться история процесса
- **Удаление сбойных точек** – автоматическое удаление из истории сбойных точек (точки с плохим качеством и «нереальной» меткой времени – 10 лет назад/5 лет вперед от текущего момента времени).

Для включения нужного ограничения необходимо установить соответствующую галочку, при этом поле редактирования этого ограничения становится активным.



### Докачка трендов из внешних источников

Для сохранения тренда из внешних источников (таких как OPC HDA) нужно задать трендирование соответствующего тега (например, OPC HDA tag).

При этом история будет вычитываться в два этапа:

#### 1 Загрузка истории при старте.

Если история содержит какие-либо значения, то при старте проекта **DataRate** загрузка архивов будет происходить на глубину от текущего момента до метки времени последней сохраненной точки.

Если история пуста, то при старте проекта **DataRate** загрузка осуществляется от текущего момента в прошлое на установленную в параметре **Загружать историю за глубину дней**. В этом случае последней меткой времени будет считаться дата и время установленной глубины.

Загрузка может производиться:

- **посуточно** – запрашиваются все точки за кратный суткам интервал, начиная с последней метки времени
- **полностью одним пакетом** – запрашиваются все точки от последней до текущей временной метки
- **пакетами указанного размера** – запрашивается определенное количество точек, начиная с последней метки времени.

#### 2 Докачка истории.

Докачка истории – это опрос архивных значений из внешних источников с периодом, указанным в параметре **Интервал запуска докачки**. Если в качестве значения параметра **Интервал запуска докачки** указан **0**, то менеджер трендов для данной группы настроек продолжит работать в режиме загрузки истории одним из способов, описанных выше (посуточно/полностью одним пакетом/пакетами указанного размера).

#### **ВНИМАНИЕ!!!**

На этапе загрузки истории параметр **Интервал запуска докачки** не действует. Процесс загрузки протекает непрерывно до тех пор, пока интервал времени между текущей меткой и меткой последней сохраненной в тренде точки больше значения **Интервал запуска докачки**.

На всех этапах вычитывания истории алгоритм работы менеджера трендов подразумевает периодическую проверку ошибок, возникших при опросе данных, и преждевременную инициализацию процесса сбора архивов в этом случае. Для указания соответствующего периода используется настройка **Интервал запуска докачки после ошибки**.

Настройка **Пытаться читать историю асинхронно** означает, что если сервер поддерживает асинхронный режим, то он будет использован. Иначе чтение будет идти в синхронном режиме.

#### **ВНИМАНИЕ!!!**

Если используется асинхронный режим и в течение времени **Интервал запуска докачки** ответ не был получен, то текущая операция асинхронного чтения отменяется (принудительно прерывается) и начинается новая

#### 6.1.1.3 Рекомендации по настройке системы трендирования

Подсистема трендирования **DataRate** позволяет сохранять значения атрибутов тегов в виде специальных файлов – **Структурированного** и **Файлового хранилищ**. Структурированное хранилище предназначено для «медленных» трендов, а Файловое – для «быстрых».

Размер хранилища («Размер файла истории») распределяется между всеми перьями **Группы настроек истории**: для структурированного хранилища – это ограничение на размер файла истории (\*.history), для файлового – на размер папки с файлами истории перьев. Если в одной Группе настроек будут перья с разной частотой обновления, то за один и тот же период времени хранилище будет содержать различное количество точек для разных перьев.

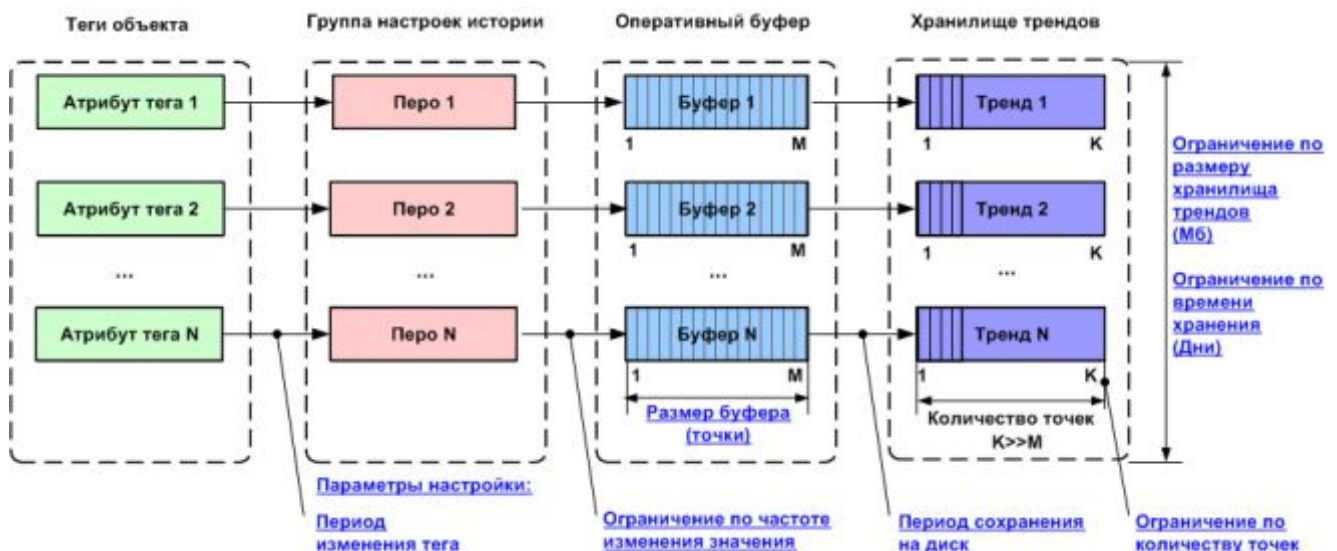
### Размер одного значения изменяемого атрибута тега (точки тренда)

Каждое значение изменяемого технологического параметра для записи в хранилище содержит обязательные атрибуты: значение, метка времени и качество. Так как значение может быть различным по типу (например, целое число, число с плавающей точкой, строка, булево значение), то и размер памяти необходимой для записи одной точки тренда может варьироваться. Для представления качества используется значение типа **Int32** (4 байта), для метки времени – значение типа **Long** (8 байт). Размер памяти для записи одного значения изменяемого атрибута тега типа **Double** (число с плавающей точкой двойной точности) рассчитывается как: **Значение** (8байт) + **Метка времени** (8 байт) + **Качество** (4байт) = 20 байт.

### ВНИМАНИЕ!

При проектировании проекта в Среде Разработки **DataRate** необходимо учитывать размер записи одной точки тренда, который используется для определения необходимого размера хранилища и оперативного буфера.

Тип значения	Размер записи точки тренда (байт)
Byte	13 байт
Int16	14 байт
Int32	16 байт
Int64	20 байт
Single	16 байт
Double	20 байт
Boolean	13 байт
Decimal	28 байт
DateTime	20 байт
String	528 байт
Object	Не может быть заранее определено



### Настройка Оперативного буфера

Настройки «**Размер оперативного буфера (в точках)**» и «**Период сохранения на диск (секунды)**» должны соответствовать данным, которые будут трендироваться **Группой настроек**. Для каждого пера в оперативной памяти формируется свой буфер заданного размера (**Размер оперативного буфера**), который с заданным периодом времени (**Период сохранения на диск**) сохраняется в хранилище.

При использовании **Структурированного хранилища** в один файл будут последовательно записываться данные от разных перьев. Количество файловых операций при этом будет зависеть от количества перьев, размера буфера и периода сохранения. При использовании **Файлового хранилища** необходимо, чтобы не происходило роста очереди на запись данных для каждого отдельного пера.

### Пример

**Группа настроек** содержит **10** перьев, которые обновляются с частотой **0.1 сек** (т.е. 10 раз в секунду). Таким образом, за одну секунду в хранилище должно быть добавлено по 10 точек на каждое перо. Размер оперативного буфера при этом должен быть таким, чтобы за указанный период сохранения не накапливалось точек, больше его размерности. Предположим, что период сохранения **100 сек**. За это время будет получено  $10 \times 100 = 1000$  точек от одного пера. В данном случае размер буфера – **1000 точек**. За 100 сек в хранилище должно сохраниться **10000** точек от всех перьев. Таким образом, если ресурсы компьютера, на котором осуществляется трендирование, не позволят в течение следующих 100 сек записать 10000 точек на диск, то может накапливаться очередь на запись к диску, и в итоге при длительной работе системы часть данных может быть потеряна.

В такой ситуации рекомендуется уменьшить период сохранения и, соответственно, размер оперативного буфера. При настройке этих параметров необходимо максимально снижать нагрузку на вычислительные ресурсы системы и не создавать предпосылки для ее неустойчивой работы, особенно если предполагается использование автоматизированной системы в режиме **24/7**. Вычислительные возможности и скорость обмена с устройствами хранения данных у современных компьютеров могут достаточно сильно отличаться, поэтому в каждом конкретном случае в процессе пуско-наладочных работ следует эмпирическим путем определять размеры оперативного буфера и период сохранения данных на диск.

### Равномерность данных внутри группы настроек

На практике трендируемый тег может включать разнотипные атрибуты с различной частотой их изменения. Например, если тег предназначен для хранения значений напряжения на обмотке электродвигателя, то у него могут быть как атрибуты для текущего значения, так и для аварийных границ по технологическому параметру. При заданной частоте получения технологических значений в тренд чаще будут попадать точки перьев, привязанных к часто изменяющимся текущим значениям (тренд работает по изменению), чем точки перьев, привязанных к технологическим границам, т.к. они изменяются достаточно редко.

Если атрибуты множества таких тегов затрендовать в одну **Группу настроек**, то через некоторый промежуток времени часть перьев, привязанных к атрибутам напряжения, будет содержать гораздо больше точек, чем перья технологических границ.

Аналогичная ситуация может произойти, если в группу настроек поместить перья тегов с разной частотой обновления данных.

И в первом, и во втором случае получаем **неравномерность распределения данных** внутри хранилища. Так как настройки для всех перьев в **Группе настроек** одинаковы, то при большом количестве перьев и частом изменении данных (порядка 0.1 сек) будут возникать промежутки времени, когда нужно записывать большой объем данных, и когда ничего не нужно делать.

Чтобы исключить подобные ситуации, необходимо теги с **одинаковой частотой изменения значений** или обновления данных назначать в **отдельные Группы настроек**.

Если вычислительные ресурсы автоматизированной системы справляется с нагрузкой и предел допустимого дискового пространства не достигим, то неравномерность заполнения хранилища никак не скажется на работе всей системы. Однако анализ данных при этом может вызывать определенные трудности.

### Ограничения на количество точек и размер файла хранилища

**Ограничения на количество точек и размер хранилища** фактически определяют ограничения на количество записываемых данных для каждого пера хранилища.

В случае использования **Файлового хранилища** ограничение на количество точек фактически определяет размер файла для каждого пера:

**Размер файла пера = Количество точек пера × Размер одной точки (байт)**

При этом общий объем дискового пространства хранилища составит:

**Размер хранилища = Количество перьев × Размер файла пера (байт)**

В случае использования **Структурированного хранилища** ограничение на количество точек одного пера фактически определяет размер области файла для каждого пера. Если не указать ограничение на размер файла, но указать ограничение на количество точек, то в результате можно будет примерно рассчитать размер хранилища:

**Размер файла структурированного хранилища = Количество точек пера × Размер одной точки (байт) × Количество перьев**

При одном и том же объеме дисковой памяти исторические данные при использовании **Файлового хранилища** будут сосредоточены во множестве «коротких» файлов, а в случае **Структурированного хранилища** – в одном «длинном». Однако стоит учитывать, что при большом количестве перьев, частом их обновлении и допустимом максимальном количестве точек (2147483647), размер файла **Структурированного хранилища** может оказаться очень большим (несколько Гигабайт). Поэтому следует избегать файлов истории подобного порядка.

### ВНИМАНИЕ!

**Неправильное задание ограничений на количество точек и размер хранилища может привести к неравномерному заполнению хранилища и потере данных.**

### Пример

100 перьев изменяются с периодичностью 100 мс. Разработчик задал ограничение на размер структурированного хранилища – 2Мб. Через некоторый промежуток времени последовательное формирование хранилища привело к тому, что количество точек в начальных перьях составило порядка 1500 точек, а в «последних» – только порядка 600. Точки «последних» перьев просто не поместились, так как был достигнут граничный объем хранилища. Если не ограничивать размер хранилища, но ограничить количество точек (например, 1000

точек), то при этих же данных был получено равномерное заполнение хранилища (в каждом пере 1000 точек) и небольшое превышение заданного ранее размера (потребовалось порядка 2.5 Мб).

Часто применение одного из ограничений не разумно, так как правильное выполнение одного условия может отрицательно отражаться на выполнении других. Необходимо накладывать не только ограничение **на размер структурированного хранилища**, но и **на количество точек** в каждом пере. В этом случае будет сохранен требуемый размер хранилища и будет известно количество точек, а соответственно и интервал времени, за который будут сохранены данные.

### Ограничение на время хранения данных

Группа настроек истории позволяет ограничивать **время хранения данных в хранилище в днях**. Интегральные показатели работы системы обычно хранятся продолжительное время, а оперативно изменяемые параметры работы оборудования часто изменяются, поэтому их стоит размещать в разных **Группах настроек** и накладывать на них свои ограничения по времени хранения. При этом один и тот же технологический параметр может трендироваться с разными условиями, предоставляя конечному пользователю системы информацию в удобном виде.

### Ограничение допустимого свободного дискового пространства

Несмотря на то, что разработчик системы должен сам определять размер хранилищ, учитывая их размещение на дисках, существует ограничение в **500 Мб**, которое препятствует остановке системы при исполнении по причине нехватки дискового пространства. При достижении данного ограничения пользователь в **Протоколе событий** увидит сообщение, предупреждающее о невозможности записи новой порции данных. Таким образом, при проектировании системы необходимо заранее рассчитывать размеры файлов хранилищ исходя из количества данных и времени их хранения.

### Минимизация объемов данных хранилища

#### **ВНИМАНИЕ!**

Чем меньше объем данных, накопленных за продолжительный промежуток времени данных (например, за год), тем меньше размер хранилища, тем быстрее можно получить и обработать накопленные данные.

Есть несколько простых способов уменьшения объема обрабатываемых данных:

- 1 В реальных задачах создавать типы тегов с числовыми атрибутами **минимально допустимой размерности**. Например, для целых чисел можно использовать двухбайтовое, четырехбайтовое и восьмибайтовое целое число
- 2 Трендировать значения конкретного технологического параметра с **минимально допустимым периодом**. Настройка «Использовать ограничение по частоте изменения значения» позволяет сохранять изменяемые значения только в указанные моменты времени. При этом данные, приходящие в перерыв в другое время, обрабатываться не будут. С данной настройкой надо быть предельно внимательным, так как при этом часть данных может не фиксироваться в хранилище и теряться



- 3 В настройках привязки трендируемых тегов **указать апертуру изменяемого значения**. Таким образом, в перо будут попадать только данные, которые соответствуют реальному изменению физической величины
- 4 **Применять настройку по ограничению времени хранения данных**, если данные в Группе настроек истории быстро становятся неактуальными. Это позволит существенно сократить количество неактуальных данных в системе
- 5 **Разделять трендируемые теги по Группам настроек** в соответствии с их интервалами измерения, временем хранения и другим характеристикам.

### Общие рекомендации по настройкам трендирования

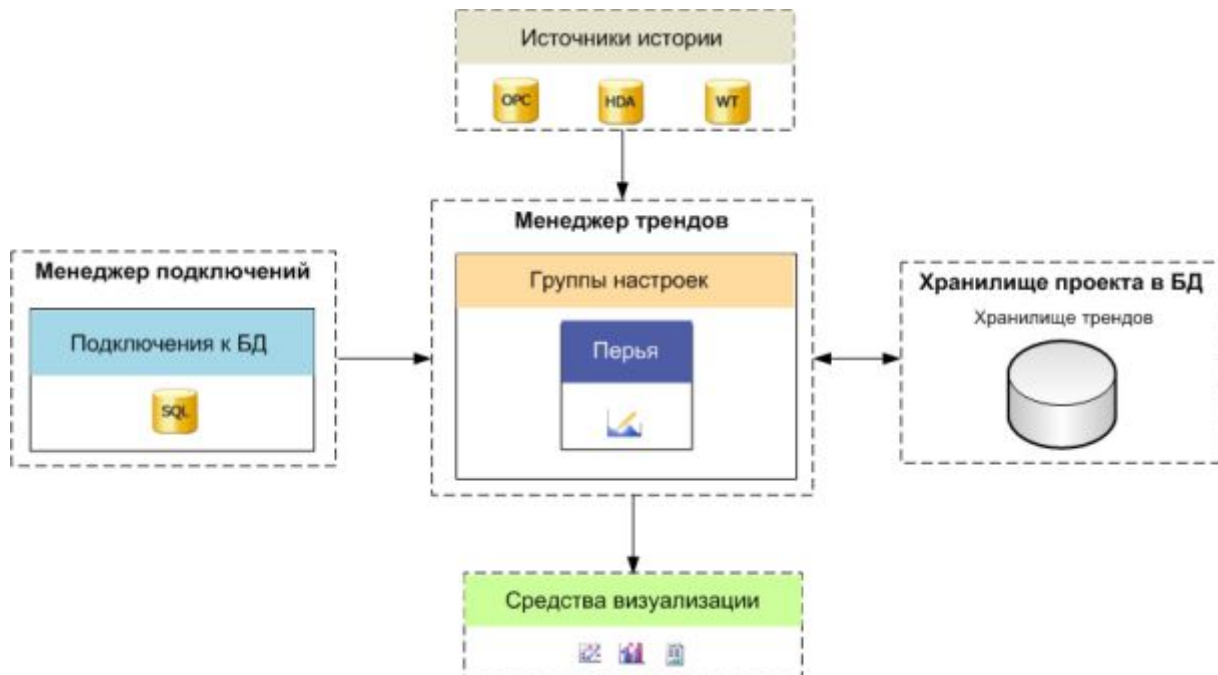
При создании проекта автоматизированной системы следует:

- 1 Сформировать перечень трендируемых технологических параметров
- 2 Определить для каждого технологического параметра период обновления и сохранения данных, срок хранения данных и наиболее частые варианты использования по обращению к истории. На основании этой информации сформировать **Группы настроек** истории и выбрать тип хранилища для каждой группы
- 3 Оценить размеры требуемого дискового пространства и определить ограничения на хранилища истории
- 4 Использовать **SSD-накопители** для хранения трендов.

#### 6.1.1.4 Хранилище трендов в БД

**Хранилище трендов в БД** представляет собой механизм для хранения данных перьев Менеджера трендов в базе данных с выполнением всех стандартных функций по записи/чтению данных истории технологического процесса.

Для использования базы данных в качестве хранилища трендов необходимо создать **Подключение к БД**, которое будет использоваться для связи с БД **Группой настроек истории**.



Для ведения трендов в БД необходимо выбрать тип хранилища "Хранилище в базе данных" и **Подключение**.

При настройке группв настроек истории можно проверить подключение к БД. Группа настроек оперирует готовым подключением и использует его при исполнении для доступа к СУБД. При первом запуске проекта на исполнение происходит создание карты перьев в БД, при последующих запусках происходит проверка перьев в БД и перьев в проекте.

Использование **Группы настроек истории** к хранилищу в БД не отличается от других типов хранилищ

**Группа настроек**

Основные свойства

Имя группы: (по-умолчанию)

☒ Сохранение истории включено

Тип хранилища: Хранилище в базе данных

Подключение: Подключение1 Проверить

Строка соединения с БД:  
Application Name=DataRate;Port=5432;Integrated Security=False;Host=localhost;Database=storage100;Uname=user100;Password=\*\*\*\*\*

Оперативный буфер (в оперативной памяти)

Период сохранения на диск (секунды): 120 Размер оперативного буфера (в точках): 150

Ограничения на ведение истории

☐ Использовать ограничение по частоте изменения значения  
Сохранять значение не чаще чем (мс): 1000

☒ Использовать ограничение по количеству точек  
Максимальное количество точек: 65536

☐ Использовать ограничение по размеру файла  
Размер файла истории (MB): 50

☒ Использовать ограничение по дням  
Максимальное количество дней: 10

☐ Удалять сбойные точки

Докачка трендов из внешних источников

Загружать историю при старте за (дни): 1 Интервал докачки (мин): 5 на резерве (мин): 5

☒ посуточно Интервал запуска докачки после ошибки (мин): 1

☐ полностью одним пакетом

☐ пакетами размером (точки) 1 ☒ Пытаться читать историю асинхронно

Помощь ОК Отмена

**ВНИМАНИЕ!**

Для повышения отказоустойчивости хранилища трендов в базе данных рекомендуем применять физическое дублирование дисков сетевых систем хранения.

Разделяемые диски сетевых систем хранения позволяют избежать избыточности синхронизации данных путём использования только одной копии базы данных. База данных использует единственный дисковый массив, который разделяется между несколькими серверами.

Если основной сервер БД откажет, то подключается резервный сервер и запускает базу данных, что обеспечивает быстрое переключение без потери данных.

### 6.1.2 Проверка истории

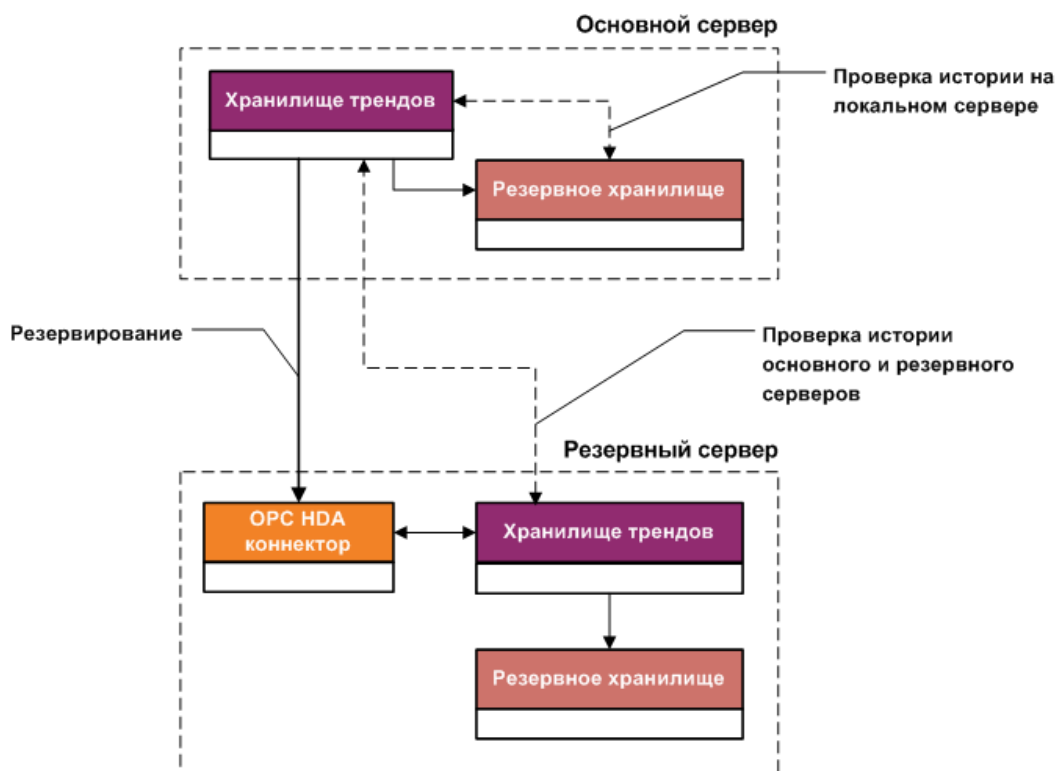
**Проверка истории** – это процедура сравнения содержимого хранилищ трендов с целью проверки их идентичности или обнаружения различий.

#### Общие рекомендации по настройкам трендирования

Резервное хранилище может находиться как на локальном сервере, так и на удаленном резервном сервере.

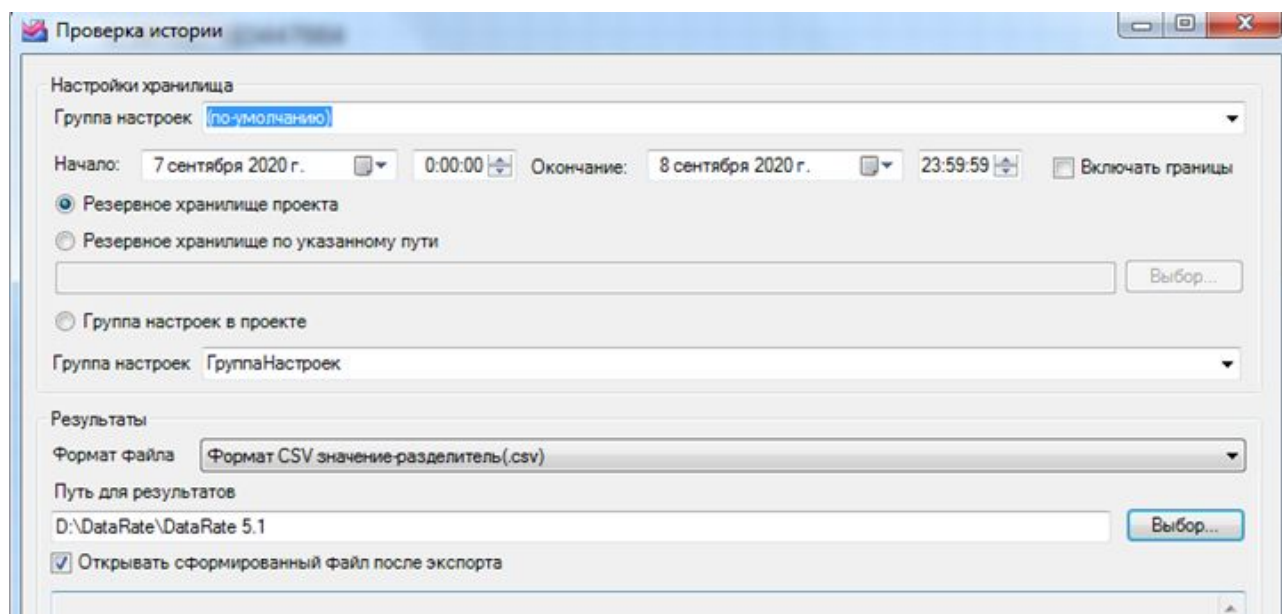
В системах с резервированием проверка истории позволяет контролировать процесс резервирования трендируемых

данных. В этом случае для процедуры проверки в качестве резервного хранилища выбирается хранилище трендов резервного сервера.

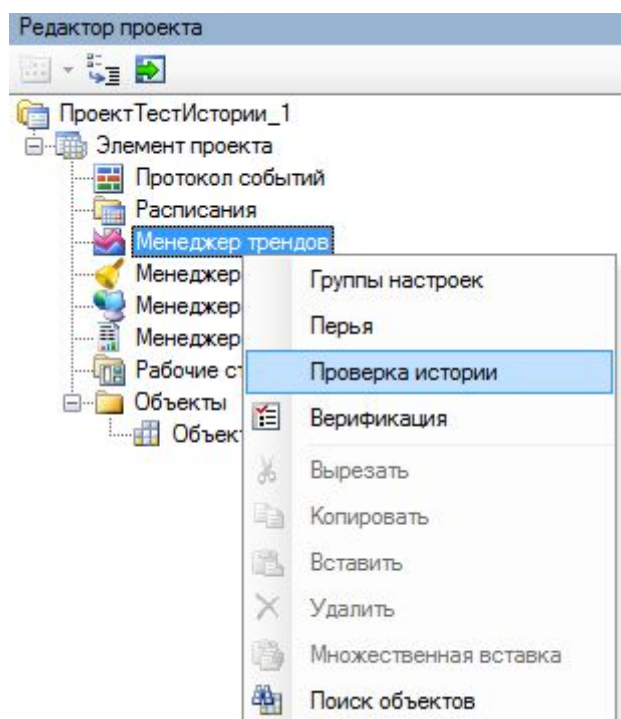


Для выполнения проверки истории следует:

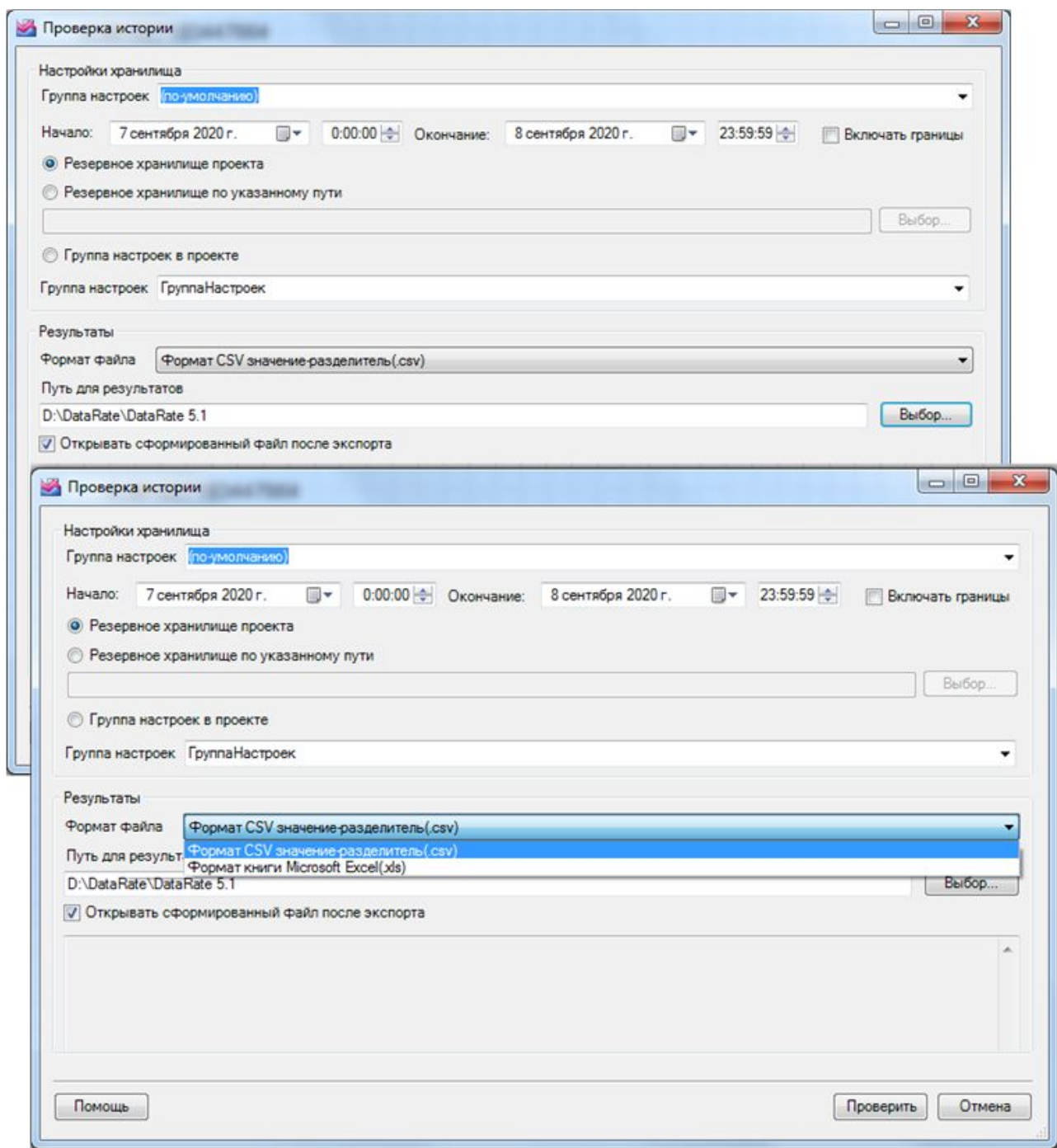
- 1 Настроить сохранение истории. Пример настройки истории локального сервера с использованием резервного хранилища приведен на скриншоте ниже



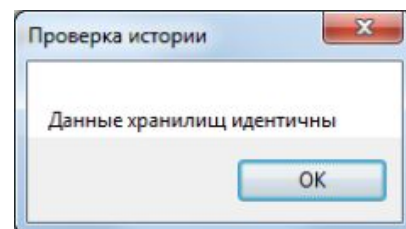
- 2 В контекстном меню **Менеджера трендов** выбрать команду **Проверка истории**



- 3 В открывшемся окне **Проверка истории** задать **Группа настроек**, **Начало** и **Окончание** интервала проверки истории, выбрать **Резервное хранилище проекта** или **Резервное хранилище проекта по указанному пути** (с указанием пути к резервному хранилищу на локальном или резервном сервере), а также **Формат файла** и **Путь для результатов** проверки
- 4 Нажать на кнопку **Проверить**

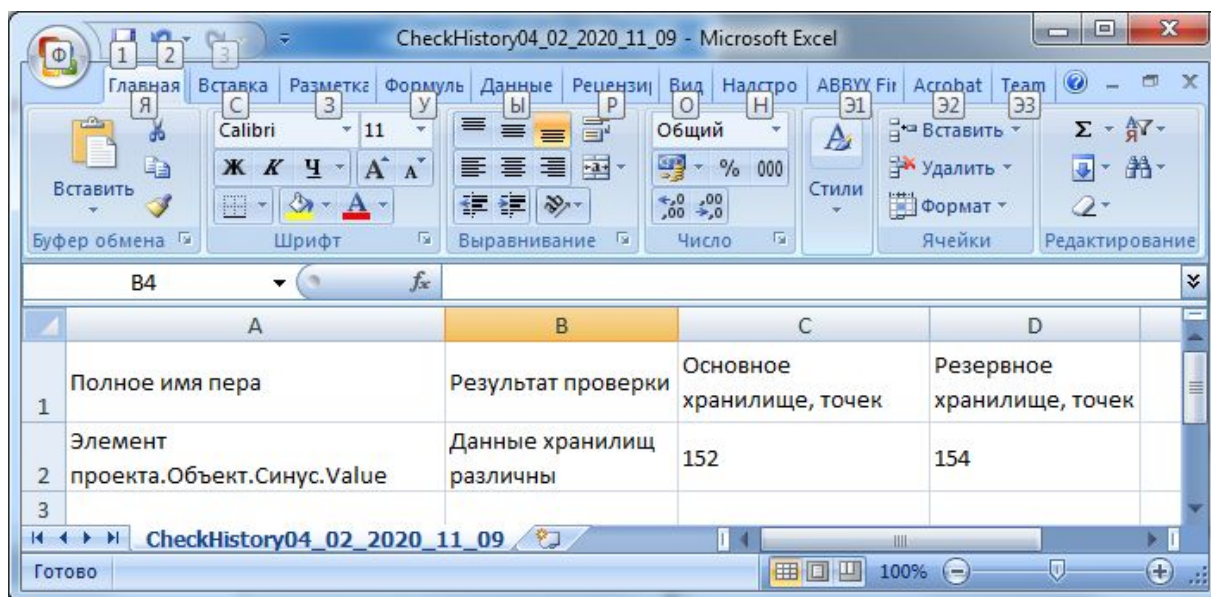
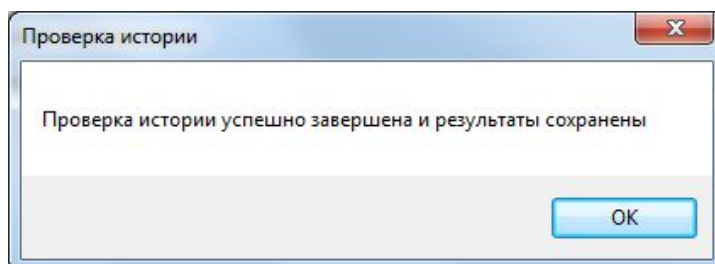


5. Просмотреть результаты проверки. В случае идентичности хранилищ **DataRate** выводит следующее сообщение:





Если обнаружены расхождения в значениях трендов в хранилищах (количество данных различно или при равном количестве метки времени различны), **DataRate** выводит следующее сообщение и формирует соответствующий файл:



	A	B	C	D
1	Полное имя пера	Результат проверки	Основное хранилище, точек	Резервное хранилище, точек
2	Элемент проекта.Объект.Синус.Value	Данные хранилищ различны	152	154

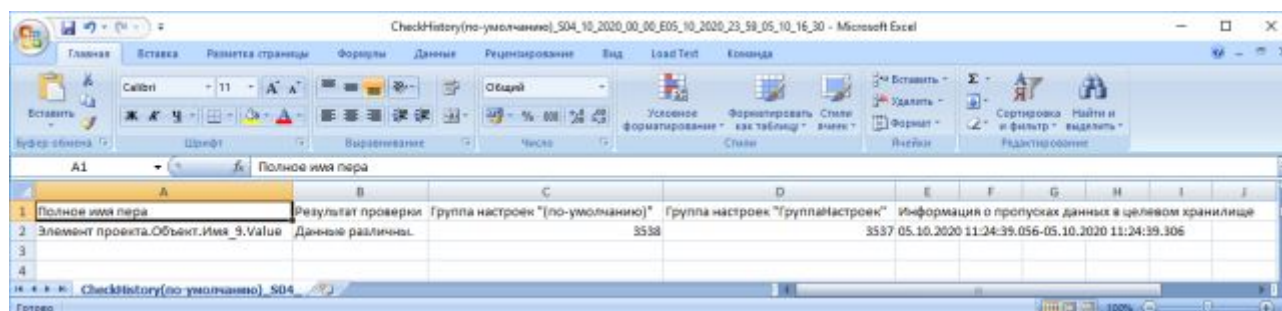
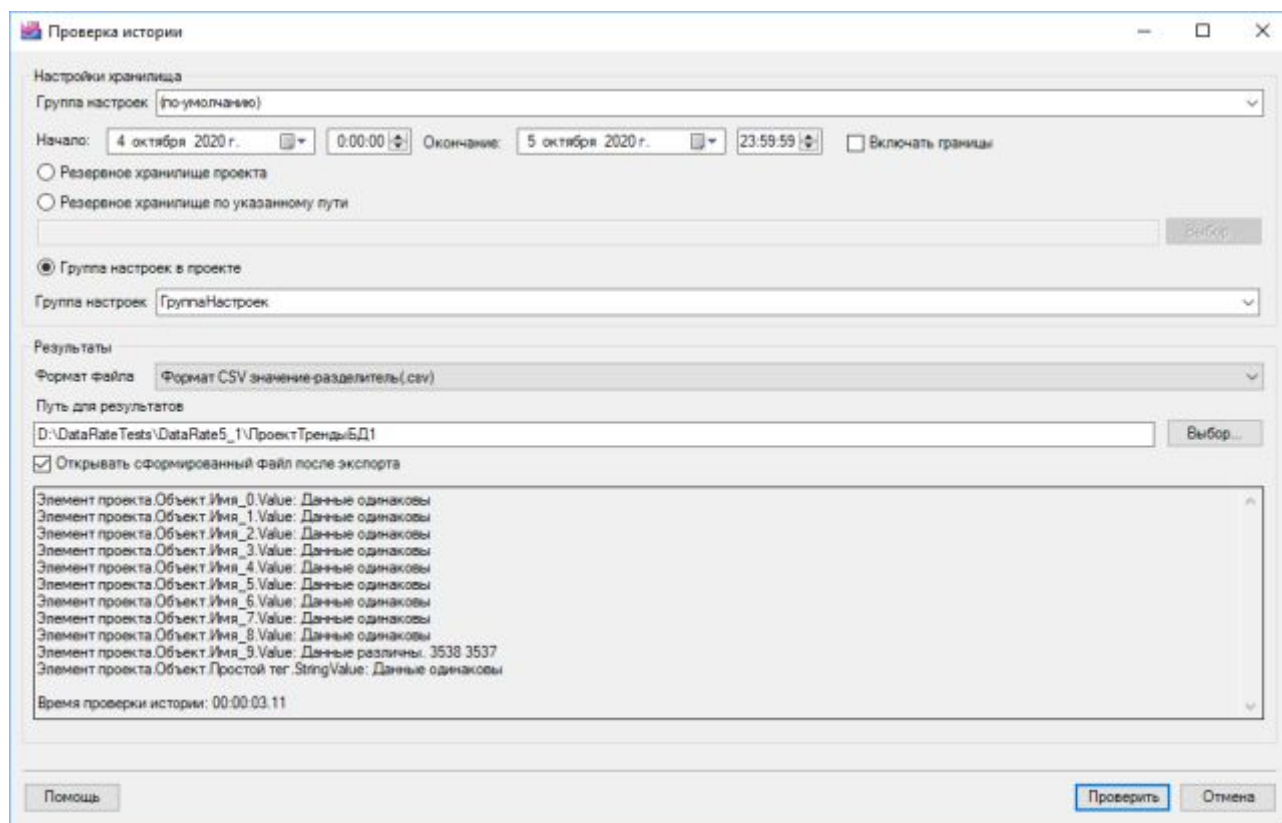
### ПРОВЕРКА ИСТОРИИ ТРЕНДОВ ДЛЯ РАЗНЫХ ГРУПП НАСТРОЕК ИСТОРИИ

Для выполнения проверки истории следует:

- 1 В поле **Группа настроек** раздела **Настройки хранилища** указать/выбрать группу настроек истории с эталонными данными
- 2 Выбрать интервал выборки данных
- 3 Отметить признак **Группа настроек в проекте** и указать/выбрать группу настроек истории с данными для сравнения в поле **Группа настроек**
- 4 Задать формат и путь к файлу для хранения результатов проверки
- 5 Нажать на кнопку **Проверить**:

Результаты проверки отображаются как в окне **Проверка истории**, так и в указанном файле.

Колонка **Информация** о пропусках в целевом хранилище показывает интервалы времени, на которых отсутствуют точки в целевом хранилище, но есть в исходном. Проверка временных меток точек происходит с точностью в 5мс.

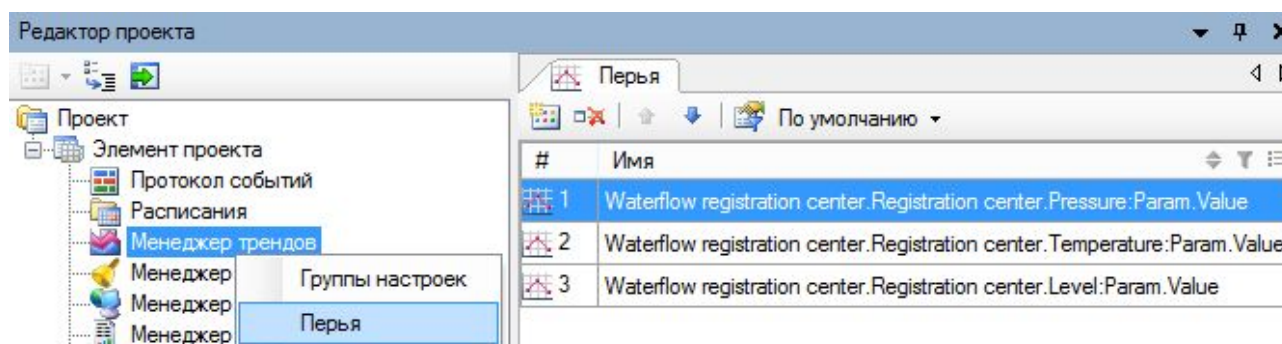


### 6.1.3 Перья Менеджера трендов

**Перо Менеджера трендов** – специальный объект, содержащий данные о трендируемом атрибуте тега.

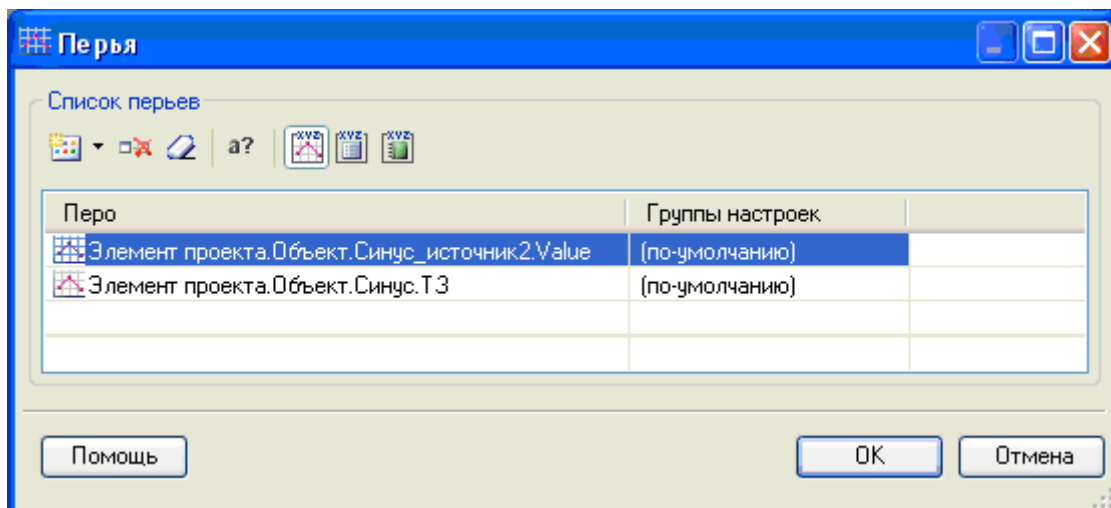
Для просмотра и редактирования списка перьев следует:

- 1 В контекстном меню **Менеджера трендов** выбрать пункт **Перья**
- 2 В появившемся окне **Перья** можно добавлять и удалять перья



### 6.1.3.1 Редактирование перьев Менеджера трендов

Окно **Перья** предназначено для редактирования списка перьев **Менеджера трендов**.




Для управления списком перьев **Менеджера трендов** используйте кнопки панели инструментов:



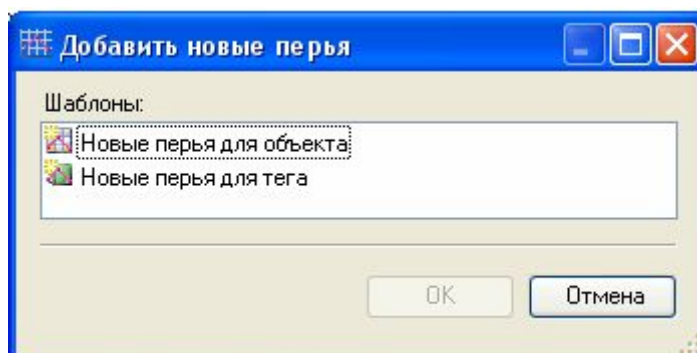
- **Создать перья**, создает одно или несколько перьев
- **Удалить выбранные перья**, удаляет выделенные перья;
- **Очистить историю у выбранных перьев**
- **Выбрать перья по маске**, выделяет перья в соответствии с заданной маской для тегов;
- **Не группировать**, отменяет группировку перьев;
- **Группировать по группам настроек**, группирует перья по группам настроек;
- **Группировать по тегам**, группирует перья по тегам.

### 6.1.3.2 Создание перьев

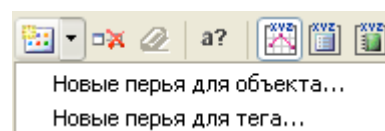
При нажатии на кнопку **Создать перья**  в окне **Перья** открывается окно **Добавить новые перья**

При выборе шаблона **Новые перья для объекта** запускается **Мастер создания перьев для объекта**.

При выборе шаблона **Новые перья для тега** запускается **Мастер создания перьев для тега**.

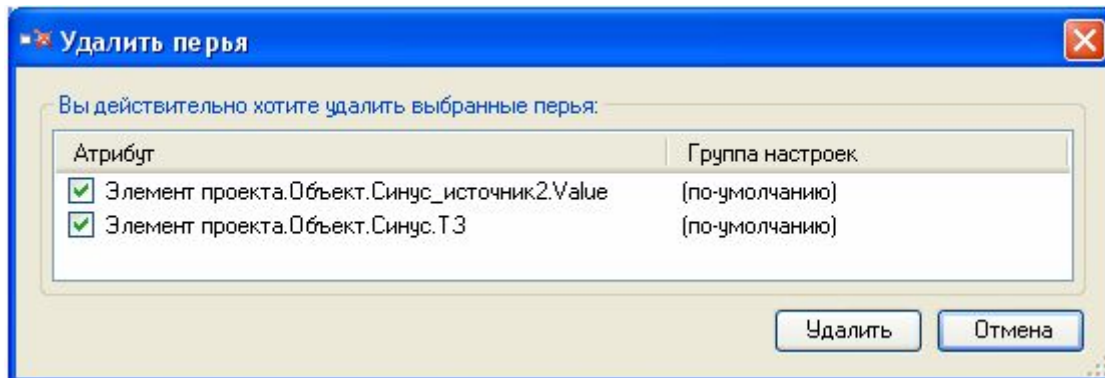


Аналогичные действия доступны в выпадающем меню при нажатии на стрелку справа от кнопки **Создать перья** –



### 6.1.3.3 Удаление перьев

При нажатии кнопки **Удалить выделенные перья** в окне **Перья** открывается окно **Удалить перья**.



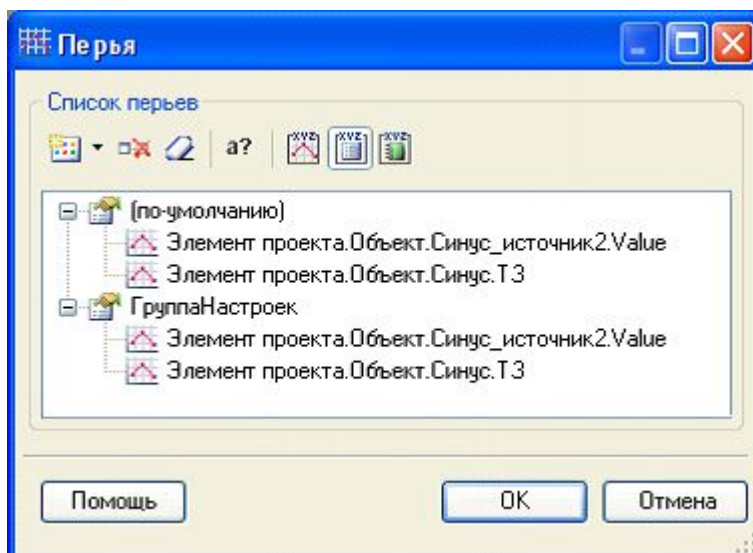
Галочками отмечены перья, которые будут удалены.

Пользователь может:

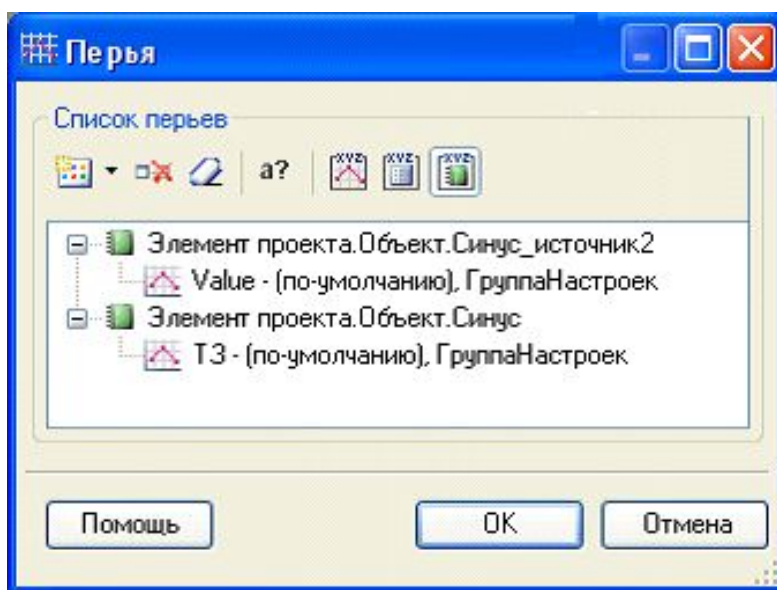
- Подтвердить удаление, нажав кнопку **Удалить**
- Отредактировать удаление, сняв галочки с перьев, которые не нужно удалять
- Отменить удаление, нажав кнопку **Отмена**.

### 6.1.3.4 Группировка перьев

Перья можно группировать по **группам настроек** и по **тегам**.







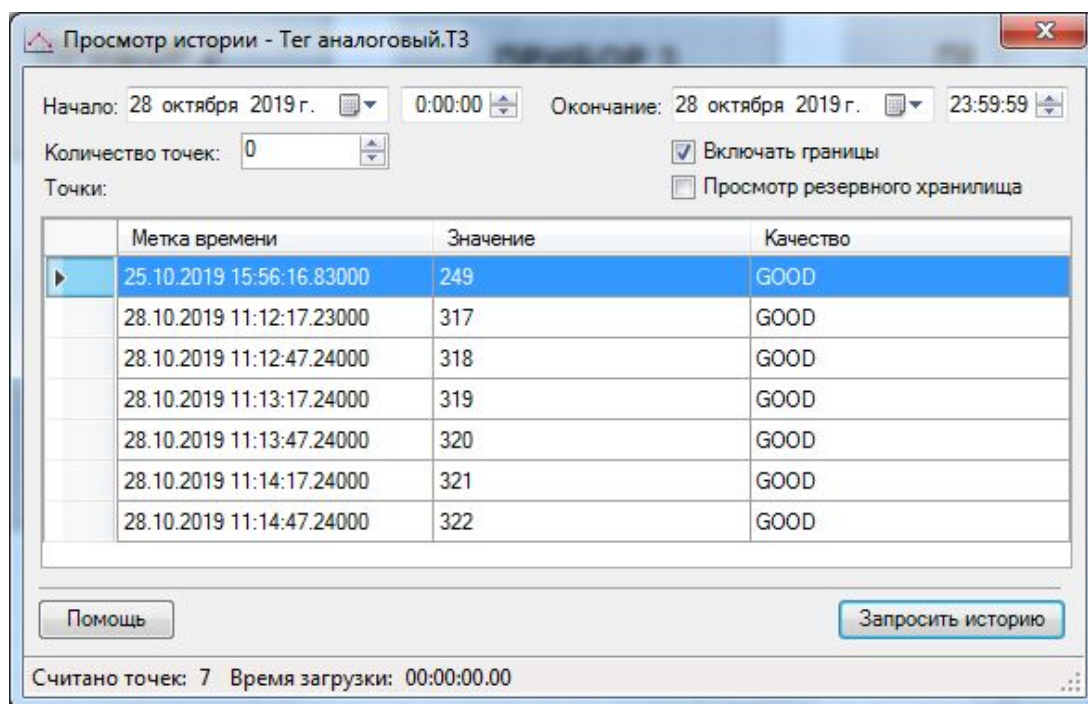
## 6.1.3.5 Просмотр истории пера менеджера трендов

**Менеджер трендов DataRate** обеспечивает предварительный просмотр истории изменения атрибута тега.

Получаемые по запросу данные не сохраняются и предназначены исключительно для предварительного просмотра значений.

Для предварительного просмотра истории следует:

- 1 Выбрать перо **Менеджера трендов**
- 2 Выбрать пункт **Просмотреть историю** в контекстном меню выбранного пера
- 3 В появившемся окне **Просмотр истории** задать параметры запроса истории и нажать кнопку **Запросить историю**

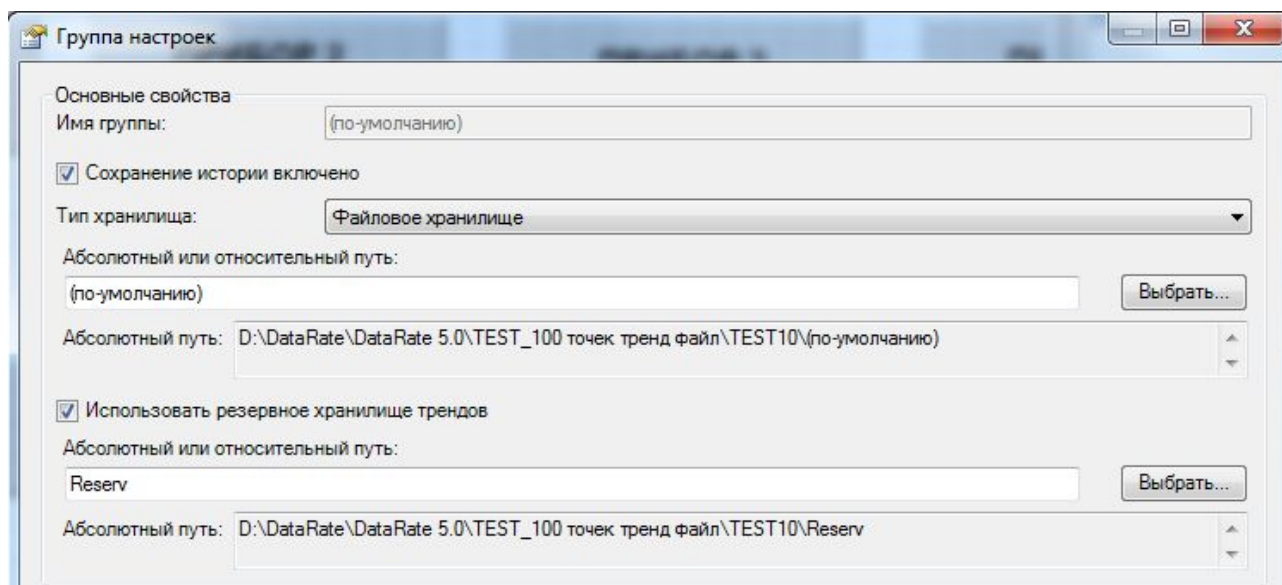




## Просмотр резервного хранилища

Менеджер трендов **DataRate** обеспечивает предварительный просмотр истории изменения атрибута тега не только из основного, но также из резервного хранилища.

Для создания резервного хранилища следует в **Группе настроек** Менеджера трендов установить признак **Использовать резервное хранилище трендов**, задать **Тип хранилища** (**файловое** или **структурное**) и путь (абсолютный или относительный)



Для просмотра резервного хранилища следует:

- 1 Выбрать перо **Менеджера трендов**
- 2 Выбрать пункт **Просмотреть историю** в контекстном меню выбранного пера
- 3 В появившемся окне **Просмотр истории** задать параметры запроса истории
- 4 Установить признак **Просмотр резервного хранилища**
- 5 Нажать кнопку **Запросить историю**

Полученные по запросу исторические данные из основного хранилища отображаются в левом полуокне, данные из резервного хранилища – в правом.

Для просмотра резервного хранилища можно использовать путь, указанный в Группе настройки истории (**Резервное хранилище проекта**), или задать в окне просмотра (**Резервное хранилище по указанному пути**).

## ОБЪЕКТЫ И СЕРВИСЫ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Просмотр истории - Тег аналоговый.T3

Начало: 27 октября 2019 г. 0:00:00 Окончание: 28 октября 2019 г. 23:59:59

Количество точек: 0

Точки:

☐ Включать границы

☒ Просмотр резервного хранилища

☒ Резервное хранилище проекта ☐ Резервное хранилище по указанному пути

Выбор...

Метка времени	Значение	Качество
28.10.2019 11:12:17.23000	317	GOOD
28.10.2019 11:12:47.24000	318	GOOD
28.10.2019 11:13:17.24000	319	GOOD

Точки резервного хранилища:

Метка времени	Значение	Качество
28.10.2019 11:12:17.23000	317	GOOD
28.10.2019 11:12:47.24000	318	GOOD
28.10.2019 11:13:17.24000	319	GOOD

Просмотр истории - Тег аналоговый.T3

Начало: 28 октября 2019 г. 0:00:00 Окончание: 28 октября 2019 г. 23:59:59

Количество точек: 0

Точки:

☐ Включать границы

☒ Просмотр резервного хранилища

☐ Резервное хранилище проекта ☒ Резервное хранилище по указанному пути

D:\DataRate\DataRate 5.0\TEST\_100 точек тренд файл\TEST10\Reserv

Выбор...

Метка времени	Значение	Качество
28.10.2019 11:12:17.23000	317	GOOD
28.10.2019 11:12:47.24000	318	GOOD
28.10.2019 11:13:17.24000	319	GOOD
28.10.2019 11:13:47.24000	320	GOOD
28.10.2019 11:14:17.24000	321	GOOD
28.10.2019 11:14:47.24000	322	GOOD

Точки резервного хранилища:

Метка времени	Значение	Качество
28.10.2019 11:12:17.23000	317	GOOD
28.10.2019 11:12:47.24000	318	GOOD
28.10.2019 11:13:17.24000	319	GOOD
28.10.2019 11:13:47.24000	320	GOOD
28.10.2019 11:14:17.24000	321	GOOD
28.10.2019 11:14:47.24000	322	GOOD

Помощь

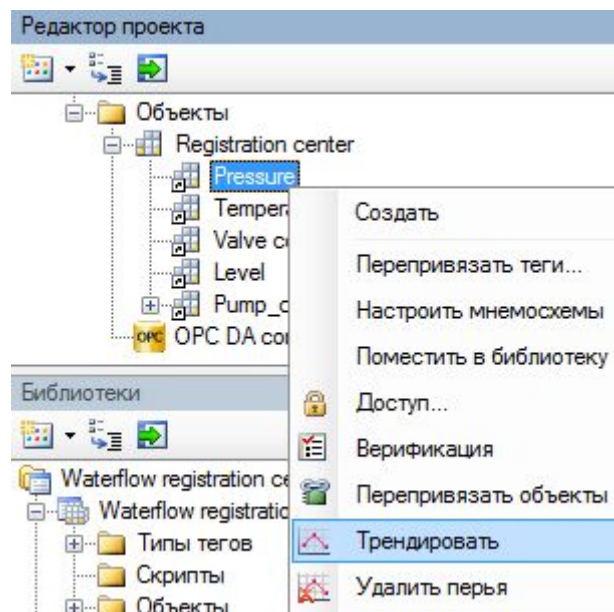
Запросить историю

Считано точек: из основного хранилища 6, из резервного хранилища 6 Время загрузки: основного 00:00:00.00, резервного 00:00:00.00 Данные основного и резервного хранилищ идентичны

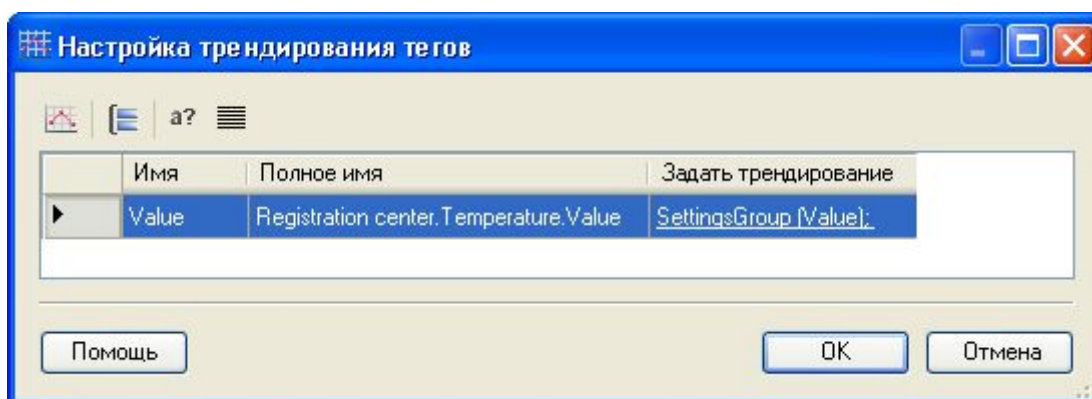
## 6.2 Тренды объектов

Назначить трендирование атрибутам тегов объекта можно двумя способами:

- Создать соответствующее перо в диалоговом окне **Перья Менеджера трендов**, используя **Мастер создания перьев для объекта** или
- Выбрать пункт меню **Трендировать** в контекстном меню объекта.



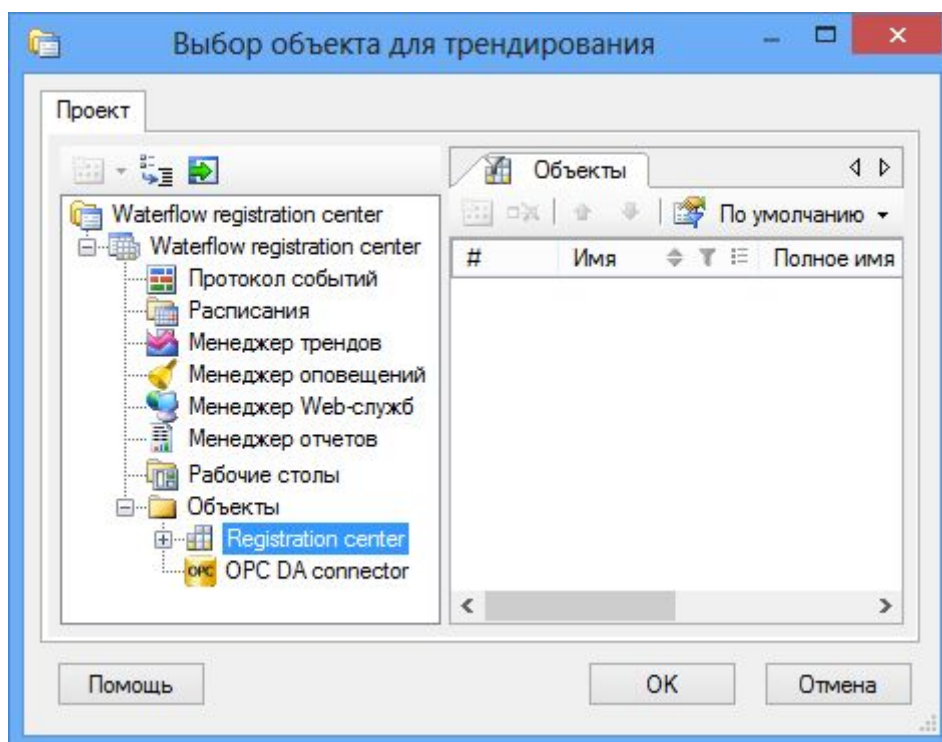
В появившемся окне **Настройка трендирования тегов** можно задать трендирование атрибутам выбранных тегов. При этом для каждого атрибута будет создано **перо** в **Менеджере трендов**.



### 6.2.1 Мастер создания перьев для объекта

Мастер создания перьев для объекта предназначен для создания **перьев Менеджера трендов** и позволяет задать трендирование любых атрибутов для любых тегов объекта.

При запуске мастера (выбрать **Новые перья для объектов** в окне **Добавить новые перья**) открывается окно **Выбор объекта для трендирования**, в котором нужно выбрать объект, для которого создаются перья, и нажать **ОК**.

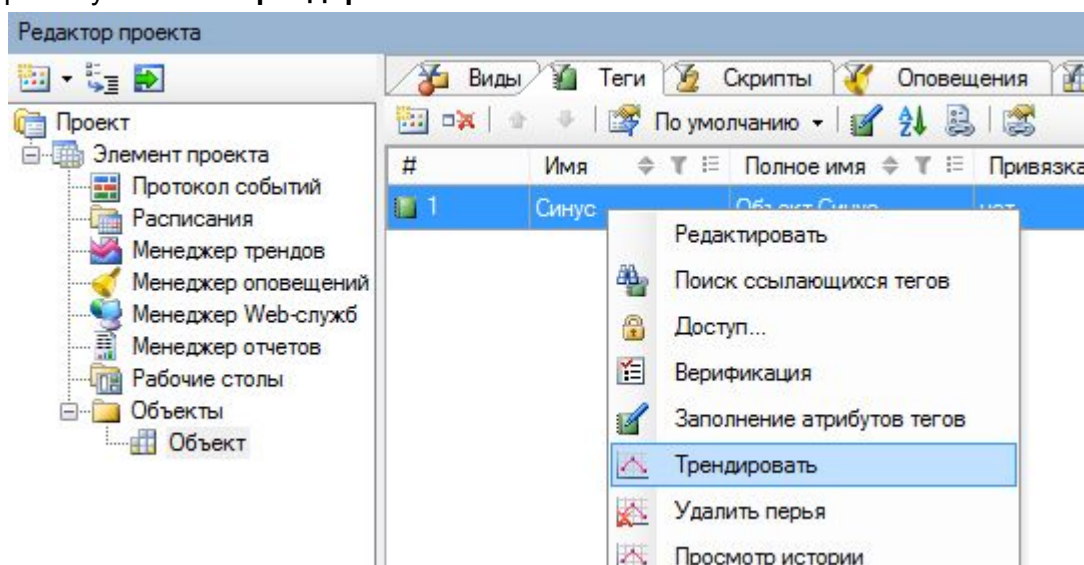


В появившемся после выбора объекта окне **Настройка трендирования тегов** нужно выбрать теги и атрибуты для трендирования. После этого созданные перья добавляются в **Менеджер трендов**.

## 6.3 Тренды тегов

Назначить трендирование атрибутам тега можно двумя способами:

- Создать соответствующее перо в диалоговом окне **Перья Менеджера трендов**, используя **Мастер создания перьев для тега**
- или
- Выбрать пункт меню **Трендировать** в контекстном меню тега.

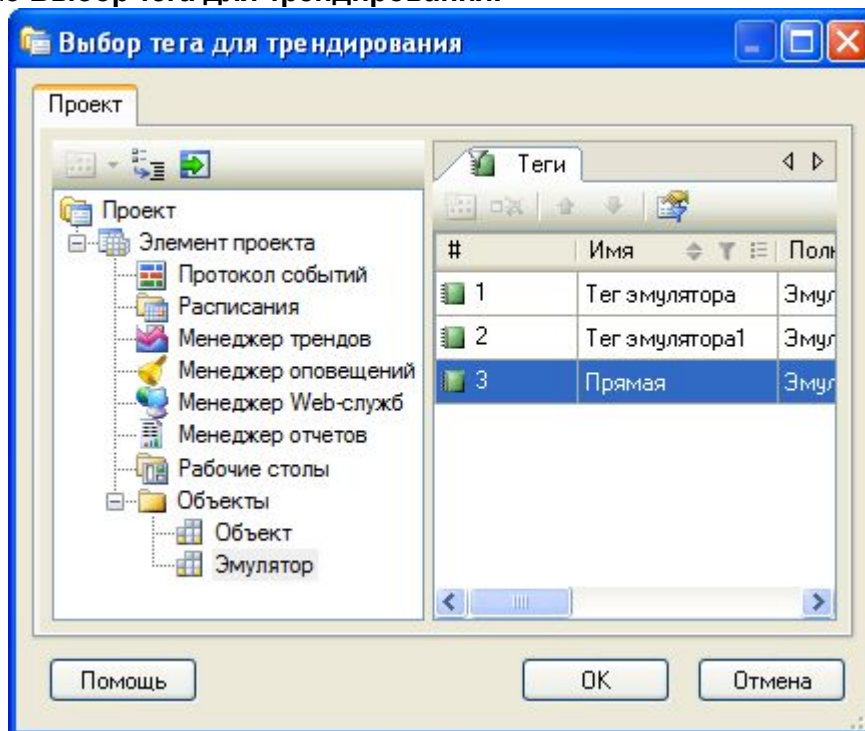




### 6.3.1 Мастер создания перьев для тега

**Мастер создания перьев для тега** предназначен для создания перьев **Менеджера трендов** и позволяет задать трендирование любых атрибутов для выбранного тега объекта.

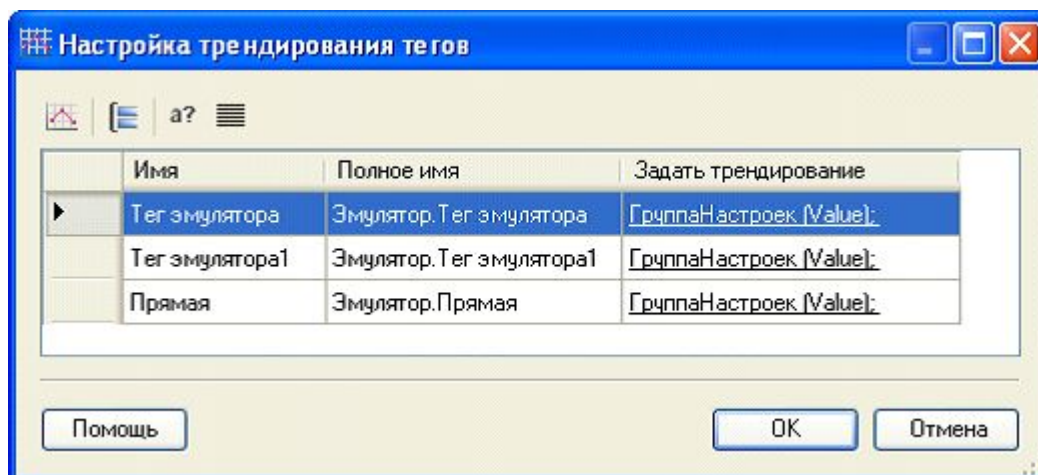
При запуске мастера (выбрать **Новые перья для тега** в окне **Добавить новые перья**) открывается окно **Выбор тега для трендирования**.



В открывшемся после выбора тега объекта окне **Добавить/снять трендирование с атрибутов выбранных тегов** нужно выбрать атрибуты для трендирования. После этого созданные перья добавляются в **Менеджер трендов**.

### 6.3.2 Настройка трендирования тегов

Для настройки трендирования атрибутов тегов выбранного объекта и всех дочерних объектов используется окно **Настройка трендирования тегов**.





---

- **Имя** – имя тега
- **Полное имя** – полное имя тега (полное имя объекта + имя тега)
- **Задать трендирование** – данные о том, трендируется тег или нет. Если ни один атрибут выбранного тега не трендируется, ячейка содержит значение **«Тег не трендируется»**; если у тега есть трендируемые атрибуты – выводится **название группы настроек**, используемых для трендируемых атрибутов, и **имена трендируемых атрибутов**.

Для задания трендирования атрибутам тега следует:

- Щелкнуть левой клавишей мыши по содержимому ячейки в колонке **Задать трендирование**
- В открывшемся окне **Добавить/снять трендирование с атрибутов выделенных тегов** задать трендирование атрибутам выделенного тега.

Для настройки трендирования тегов используйте **кнопки панели инструментов**:

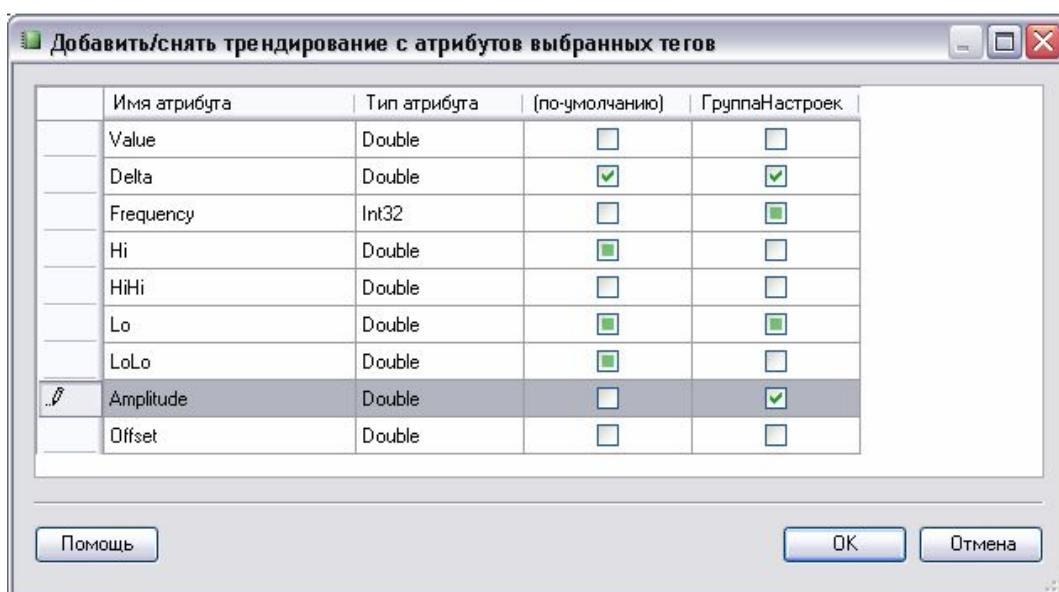


- Назначить/снять трендирование выбранным тегам
- Показать/скрыть теги дочерних объектов. В случае, когда у объекта есть вложенные объекты, можно отобразить теги дочерних объектов
- Выделить теги по имени. Осуществляет выбор тегов по маске
- Выделить все теги

### 6.3.3 Настройка трендирования атрибутов

Для настройки трендирования атрибутов выбранных тегов предназначено окно **Добавить/снять трендирование с атрибутов выбранных тегов**.

Окно содержит таблицу, строки которой содержат все уникальные атрибуты (если у нескольких тегов один и тот же атрибут, он входит в таблицу только один раз) выбранной группы тегов или выбранного единичного тега.



Количество колонок данной таблицы может быть разным, в зависимости от количества групп настроек истории в **Менеджере трендов**.

Каждая **Группа настроек истории** представлена одной колонкой и может содержать в своих ячейках следующие обозначения:

- **Галочка** – означает, что соответствующий атрибут трендируется во всех тегах (в которых он существует) с данной группой настроек истории
- **Пустой квадратик** – означает, что соответствующий атрибут не трендируется во всех тегах (в которых он существует) с данной группой настроек истории
- **Закрашенный квадратик** – означает, что соответствующий атрибут трендируется в некоторых тегах с данной группой настроек истории и используется только для индикации при групповой настройке тегов. Если поменять галочку или пустой квадратик на закрашенный квадратик для некоторого атрибута, никаких изменений не будет. При настройке трендирования одного тега ячейка группы настроек может принимать только два состояния: галочка и пустой квадратик.

#### Как задать/снять трендирование

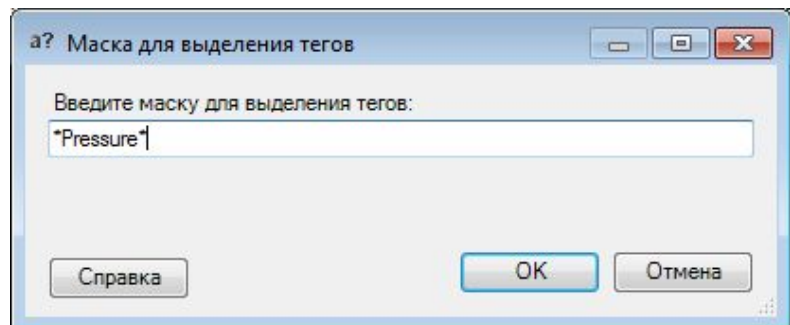
Для задания трендирования соответствующему атрибуту во всех тегах, в которых он существует, нужно поставить галочку в ячейке нужной группы настроек истории.

Для снятия нужно поставить пустой квадратик.

#### 6.3.4 Выбор тегов по маске

**Маска** представляет собой текстовую строку, в которой используются специальные символы:

- \* – заменяет несколько любых символов;
- ? – заменяет один любой символ.



#### Пример

Маска **?Тренд\*** выделит все теги, в имени которых присутствует в начале один любой символ, слово **"Тренд"** и любое количество символов после него. Эта маска подходит, например, для следующих имен тегов: **1Тренд 23 по изменению** или **\_ТрендДанных**.

Выбор тегов по маске осуществляется в окне **Маска для выделения тегов**.

#### 6.3.5 Просмотр истории тега объекта

Для предварительного просмотра истории затрендированного тега следует:

- 1 Выбрать тег объекта
- 2 Выбрать атрибут тега
- 3 Выбрать пункт **Просмотреть историю** в контекстном меню выбранного атрибута
- 4 В появившемся окне **Просмотр истории** задать параметры запроса истории и нажать кнопку **Запросить историю**

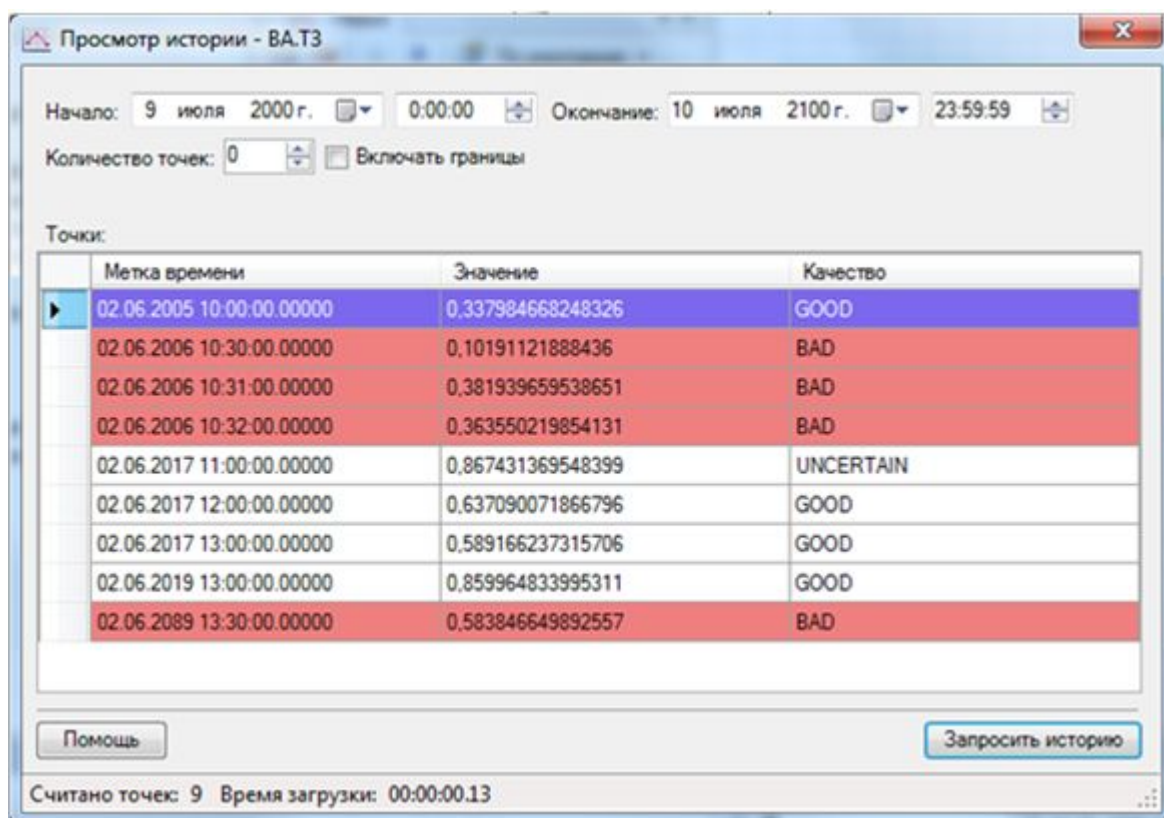
В качестве параметров запроса пользователь должен задать:

- **Начало/Окончание** – начальное и конечное время интервала запроса
- **Количество точек** – количество запрашиваемых на интервале точек. Если задан **0** (ноль), то будут вычитываться все точки
- **Включать границы** – признак включения границ в интервал запроса.

## Отображение точек истории

Каждая точка истории отображаются с тремя атрибутами: **Метка времени**, **Значение**, **Качество**. В зависимости от метки времени и качества точки фон отображения может быть:

- **белый** (отсутствие фона) – корректная точка (корректная метка времени, хорошее качество)
- **розовый** – «сбойная» точка (некорректная метка времени – 10 лет назад/ 5 лет вперед от текущего момента времени, плохое качество)
- **фиолетовый** – «подозрительная» точка (некорректная метка времени, хорошее качество)



## Сравнение данных основного и резервного хранилищ

Просмотр истории позволяет сравнивать данных основного и резервного хранилищ по одному тегу, а также данные одного и того же трендируемого тега в разные группы настроек истории.

Для сравнения можно использовать резервное хранилище проекта, указывать путь к резервному хранилищу на дисковом накопителе или указать наименование другой группы настроек истории.

Просмотр истории - Синус.Value

Начало: 31 августа 2020 г. 0:00:00 Окончание: 1 сентября 2020 г. 23:59:59 Резерв в проекте Другая группа настроек (по умолчанию)

Количество точек: 0 Включать границы Просмотр резервного хранилища

Точки: Резерв по указанному пути Точки резервного хранилища

Метка времени	Значение	Качество
31.08.2020 09:28:34.94256	0	GOOD
31.08.2020 09:28:35.94256	0,0697564737441253	GOOD
31.08.2020 09:28:36.94256	0,139173100960065	GOOD
31.08.2020 09:28:37.94256	0,207911690817759	GOOD
31.08.2020 09:28:38.94256	0,258819045102521	GOOD
31.08.2020 09:28:42.94256	0,529919264233205	GOOD
31.08.2020 09:28:44.19256	0,601815023152048	GOOD
31.08.2020 09:28:45.94256	0,694658370458997	GOOD
31.08.2020 09:28:47.69256	0,777145961456971	GOOD
31.08.2020 09:28:49.19256	0,838670567945424	GOOD
31.08.2020 09:28:49.44256	0,848048096156426	GOOD
31.08.2020 09:28:51.44256	0,913545457642601	GOOD
31.08.2020 09:28:53.19256	0,956304755963035	GOOD
31.08.2020 09:28:57.94256	0,999390827019096	GOOD
31.08.2020 09:28:58.69256	0,996194698091746	GOOD
31.08.2020 09:29:00.19256	0,981627183447664	GOOD
31.08.2020 09:29:03.69256	0,90630778703665	GOOD
31.08.2020 09:29:12.19256	0,515038074910054	GOOD
31.08.2020 09:29:30.19256	-0,656059028990507	GOOD
31.08.2020 09:29:30.94256	-0,694658370458997	GOOD
31.08.2020 09:29:32.69256	-0,777145961456971	GOOD
31.08.2020 09:29:33.69256	-0,819152044288992	GOOD
31.08.2020 09:29:34.69256	-0,857167300702112	GOOD

Метка времени	Значение	Качество
01.09.2020 15:36:10.59081	0	GOOD
01.09.2020 15:36:11.59081	0,0697564737441253	GOOD
01.09.2020 15:36:12.59081	0,139173100960065	GOOD
01.09.2020 15:36:13.59081	0,207911690817759	GOOD
01.09.2020 15:36:14.59081	0,275637355816999	GOOD
01.09.2020 15:36:15.59081	0,342020143325669	GOOD
01.09.2020 15:36:16.59081	0,4067366430758	GOOD
01.09.2020 15:36:17.59081	0,469471562785891	GOOD
01.09.2020 15:36:18.84081	0,544639035015027	GOOD
01.09.2020 15:36:19.84081	0,601815023152048	GOOD
01.09.2020 15:36:20.84081	0,656059028990507	GOOD
01.09.2020 15:36:21.84081	0,707106781186547	GOOD
01.09.2020 15:36:22.84081	0,75470950222772	GOOD
01.09.2020 15:36:23.84081	0,798635510047293	GOOD
01.09.2020 15:36:24.84081	0,838670567945424	GOOD
01.09.2020 15:36:25.84081	0,874619707139396	GOOD
01.09.2020 15:36:26.84081	0,90630778703665	GOOD
01.09.2020 15:36:27.84081	0,933580426497202	GOOD
01.09.2020 15:36:28.84081	0,956304755963035	GOOD
01.09.2020 15:36:29.84081	0,974370064785235	GOOD
01.09.2020 15:36:30.84081	0,987688340595138	GOOD
01.09.2020 15:36:32.09081	0,997564050259824	GOOD
01.09.2020 15:36:33.09081	1	GOOD

Помощь Запросить историю

Считано точек: из основного хранилища 251, из резервного хранилища 117. Время загрузки: основного 00:00:00.20, резервного 00:00:00.18. Данные основного и резервного хранилищ различны

### 6.3.6 Удаление точки из тренда

Удаление точки из тренда возможно в окне просмотра истории.  
Удалить можно любую точку, в том числе и «хорошую».

#### ВНИМАНИЕ!

Можно удалять только по одной точке.

Отменить операцию удаления точки и восстановить удаленную точку нельзя!

Для удаления точки из тренда следует:

- 1 В окне **Просмотр истории** выделить требуемую точку
- 2 В контекстном меню (открывается нажатием правой кнопки мыши) выбрать **Удалить точку из тренда**
- 3 Подтвердить удаление точки из тренда



Просмотр истории - ВА.Т3

Начало: 9 июля 2000 г. 0:00:00

Окончание: 10 июля 2100 г. 23:59:59

Количество точек: 0

☐ Включать границы

Точки:

Метка времени	Значение	Качество
02.06.2005 10:00:00.00000	0,337984668248326	---
02.06.2006 10:30:00.00000	0,10191121888436	BAD
02.06.2006 10:31:00.00000	0,381939659538651	BAD
02.06.2006 10:32:00.00000	0,363550219854131	BAD
02.06.2017 11:00:00.00000	0,867431369548399	UNCERTAIN
02.06.2017 12:00:00.00000	0,637090071866796	GOOD
02.06.2017 13:00:00.00000	0,589166237315706	GOOD
02.06.2019 13:00:00.00000	0,859964833995311	GOOD
02.06.2089 13:30:00.00000	0,583846649892557	BAD

Копировать

Удалить точку из тренда

Просмотр истории - ВА.Т3

Начало: 9 июля 2000 г. 0:00:00

Окончание: 10 июля 2100 г. 23:59:59

Количество точек: 0

☐ Включать границы

Точки:

Метка времени	Значение	Качество
02.06.2006 10:30:00.00000	0,10191121888436	BAD
02.06.2006 10:31:00.00000	0,381939659538651	BAD
02.06.2006 10:32:00.00000	0,363550219854131	BAD
02.06.2017 11:00:00.00000	0,867431369548399	UNCERTAIN
02.06.2017 12:00:00.00000	0,637090071866796	GOOD
02.06.2017 13:00:00.00000	0,589166237315706	GOOD
02.06.2019 13:00:00.00000	0,859964833995311	GOOD
02.06.2089 13:30:00.00000	0,583846649892557	BAD

Помощь

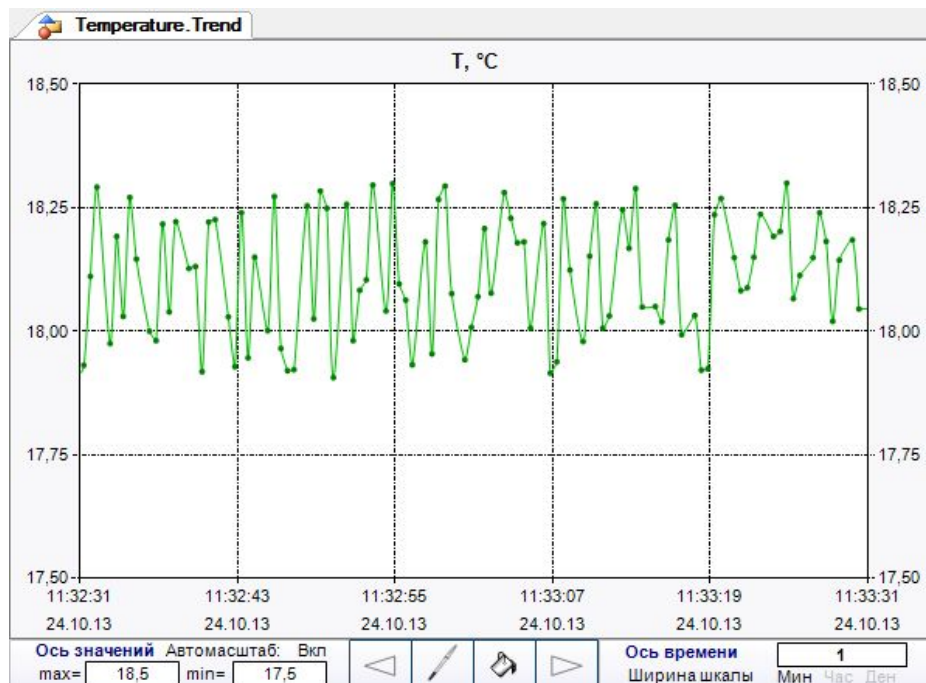
Запросить историю

Считано точек: 8    Время загрузки: 00:00:00.60



## 6.4 Графический элемент Тренд

Графический элемент **Тренд** предназначен для графического отображения изменений атрибутов тегов во времени.



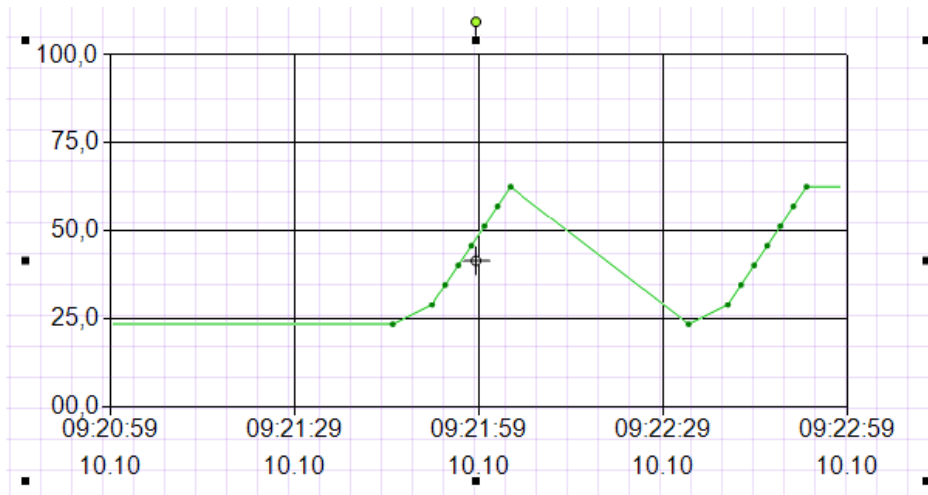
### 6.4.1 Создание графического элемента Тренд

Для создания графического элемента **Тренд** на мнемосхеме следует:

1. Выбрать иконку графического элемента **Тренд** в меню или на панели инструментов



2. Нажать левую клавишу мыши на мнемосхеме и растянуть рамку до нужных размеров графического элемента.



## 6.4.2 Общие настройки

### Положение

Параметры положения графического элемента **Тренд** на мнемосхеме:

- **Смещение слева** на мнемосхеме в пикселах
- **Смещение сверху** на мнемосхеме в пикселах
- **Ширина** в пикселах
- **Высота** в пикселах
- **Угол поворота** относительно центра в градусах
- **Центр вращения X** равен ширине, деленной на позицию центра вращения
- **Центр вращения Y**

Положение	
Смещение слева	64
Смещение сверху	54
Ширина	335
Высота	179
Угол поворота	0
Центр вращения X	0,5
Центр вращения Y	0,5

### Основные свойства

- **Имя объекта** - уникальное имя графического элемента в пределах мнемосхемы или группы
- **Видимость/мигание** - анимация видимости и мигания объекта
- **Период обновления, мс** - период обновления данных перьев тренда в миллисекундах (1секунда = 1000 миллисекунд)
- **Обновлять асинхронно** - если установлено в **Да**, то обновления данных тренда будет происходить асинхронно. Особенностью асинхронного режима является небольшое отставание данных по времени в тренде.

Основные	
Имя объекта	Тренд1
Тип	Тренд
Видимость/мигание	Видим
Период обновления, мс	1000
Обновлять асинхронно	Да

## 6.4.3 Настройка координатной сетки

Настройка координатной сетки графического элемента **Тренд** включает задание параметров отображения времени (**Ось времени**) и значений (**Ось значений**).

### Ось времени

- **Настройка шкалы времени** – способ настройки шкалы времени. При автоматической настройке количество линий сетки и количество меток временной шкалы подбираются автоматически, а шкала времени движется вместе с графиком; при ручной настройке количество линий сетки и количество меток временной шкалы задаются с помощью соответствующих значений
- **Ширина шкалы времени (сек)** – отображаемый интервал времени в секундах
- **Тип единиц шкалы времени** – позволяет выбрать единицу шкалы времени (секунды, минуты, часы, дни)

Ось времени	
Настройка шкалы времени	Ручная
Ширина шкалы времени в единицах	120
Тип единиц шкалы времени	Секунды
Ширина шкалы времени	00:02:00
Смещение, %	0
Количество точек	80
Количество линий сетки.	3
Количество меток временной шкалы	3
Цвет меток	Черный
Шрифт меток	Arial; 12
Ориентация разметки	Односторонняя А
Формат разметки 1	мм:сс
Формат разметки 2	дд/ММ

- **Смещение, %** – смещение положения текущего времени от правой границы тренда в % от ширины шкалы времени

- **Количество точек** – количество опорных точек для построения графика по вычисляемым значениям
- **Количество линий сетки** – количество вертикальных линий сетки поля отображения тренда
- **Количество меток временной шкалы** – количество меток времени, отображаемых на временной шкале
- **Цвет меток** – цвет меток времени на оси времени
- **Шрифт меток** – шрифт вывода данных оси времени
- **Ориентация разметки** – ориентация места вывода шкалы времени

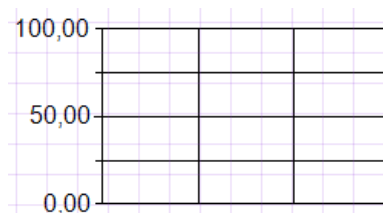
- **Формат разметки 1** – верхняя строка времени
- **Формат разметки 2** – нижняя строка времени

Форматы разметки позволяют вывести время в нужном формате:

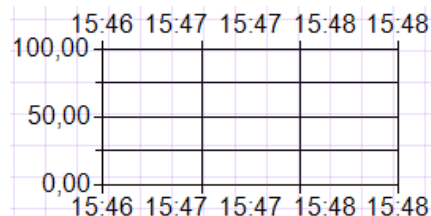
- **ГГГГ** или **ГГ** год в длинном или коротком формате
- **ММ** - месяц в виде цифры
- **ММММ** - месяц в виде названия
- **ДД** - День
- **ЧЧ** - час
- **мм** - минуты
- **сс** - секунды

### Ориентация разметки

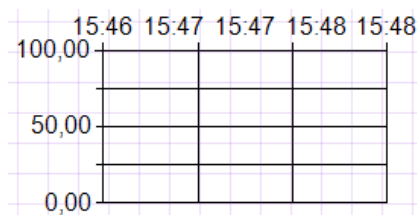
**Без разметки** – отображение времени отсутствует



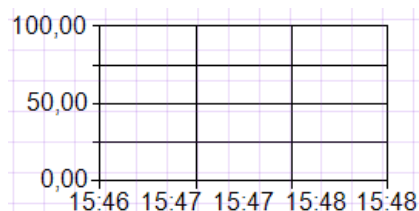
**Двухсторонняя** – шкала времени выводится сверху и снизу



**Односторонняя Б** – шкала времени выводится сверху



**Односторонняя А** – шкала времени выводится снизу.



## Ось значений

<ul style="list-style-type: none"> <li>• <b>Настройка шкалы значений</b> – способ настройки шкалы значений. При автоматической настройке количество линий сетки и количество меток оси значений подбираются автоматически, при ручной - задаются с помощью соответствующих значений</li> <li>• <b>Автомасштабирование границ</b> – включает режим автоматического подбора границ оси значений тренда по максимальному и минимальному значениям отображаемых данных</li> <li>• <b>Верхнее значение</b> – верхнее значение шкалы отображения параметра</li> <li>• <b>Формат меток</b> – формат меток оси значений</li> <li>• <b>Нижнее значение</b> – нижнее значение шкалы отображения параметра</li> <li>• <b>Количество линий сетки</b> – количество горизонтальных линий разлиновки тренда</li> </ul>	<table> <tr> <th colspan="2">Ось значений</th></tr> <tr> <td>Настройка шкалы значений</td><td>Ручная</td></tr> <tr> <td>Автомасштабирование границ</td><td>Да</td></tr> <tr> <td>Верхнее значение</td><td>100</td></tr> <tr> <td>Формат меток</td><td>00.0</td></tr> <tr> <td>Нижнее значение</td><td>0</td></tr> <tr> <td>Количество линий сетки</td><td>3</td></tr> <tr> <td>Количество меток оси значений</td><td>3</td></tr> <tr> <td>Цвет меток</td><td>Черный</td></tr> <tr> <td>Шрифт меток</td><td>Arial; 12</td></tr> <tr> <td>Ориентация разметки</td><td>Односторонняя А</td></tr> </table> <ul style="list-style-type: none"> <li>• <b>Количество меток оси значений</b> – количество промежуточных цифр между началом и концом шкалы</li> <li>• <b>Цвет меток</b> – цвет цифр на оси значений</li> <li>• <b>Шрифт меток</b> – шрифт вывода значений градуировки шкалы</li> <li>• <b>Ориентация разметки</b> – ориентация места вывода шкалы значений</li> </ul>	Ось значений		Настройка шкалы значений	Ручная	Автомасштабирование границ	Да	Верхнее значение	100	Формат меток	00.0	Нижнее значение	0	Количество линий сетки	3	Количество меток оси значений	3	Цвет меток	Черный	Шрифт меток	Arial; 12	Ориентация разметки	Односторонняя А
Ось значений																							
Настройка шкалы значений	Ручная																						
Автомасштабирование границ	Да																						
Верхнее значение	100																						
Формат меток	00.0																						
Нижнее значение	0																						
Количество линий сетки	3																						
Количество меток оси значений	3																						
Цвет меток	Черный																						
Шрифт меток	Arial; 12																						
Ориентация разметки	Односторонняя А																						

## ВНИМАНИЕ!

Формат значений поддерживает следующие символы форматирования:

**0** – означает, что если на месте символа не будет значения, то в позицию символа будет вставлена цифра 0.

Например: формат - **0000.0000**, исходное число **3,034** результат = **0003,0340**

**#** – означает, что если на месте символа не будет значения, то в позицию символа ничего вставлено не будет.

Например: формат - **###0.0000**, исходное число **3,034** результат = **3,0340**

## 6.4.4 Добавление/Удаление перьев

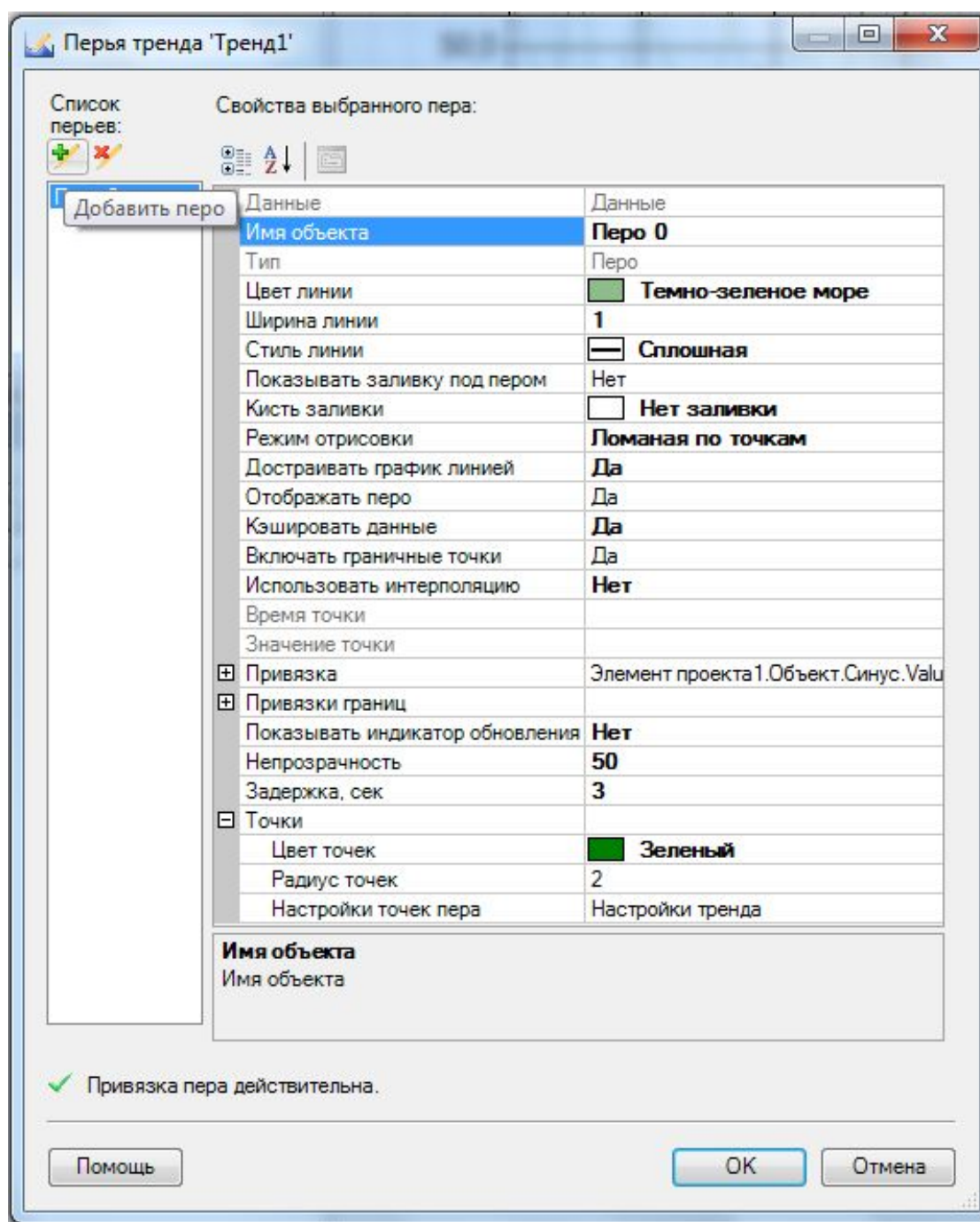
### Добавление перьев

Для добавления перьев в графический элемент **Тренд** следует:

- 1 Щелкнуть на свойстве **Перья** в окне **Свойства** и нажать на кнопку .

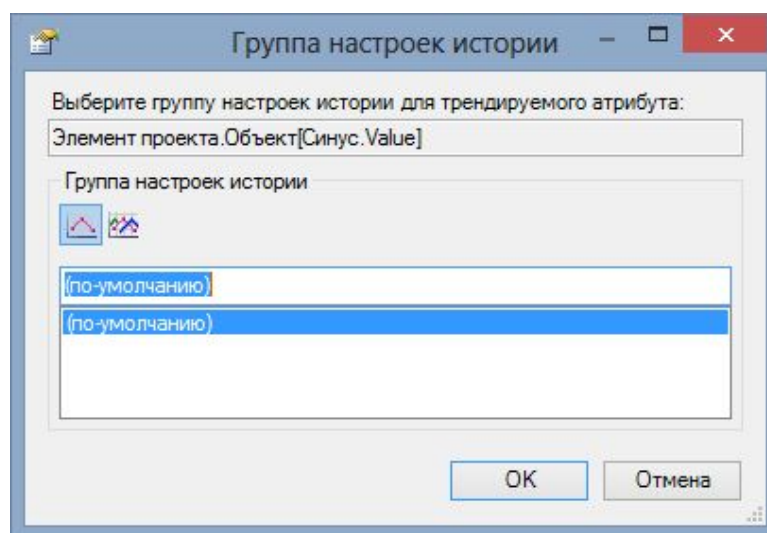
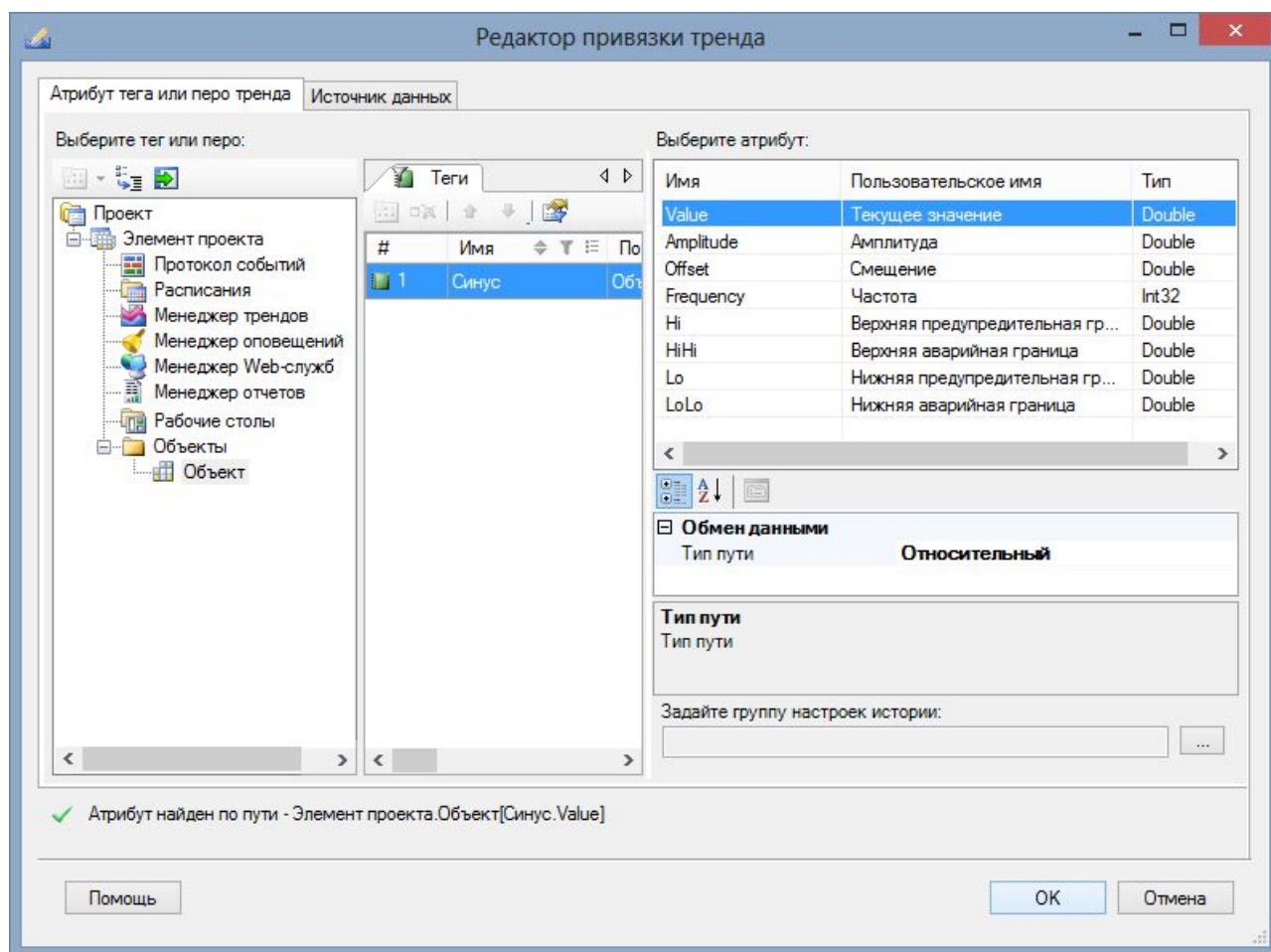


- 2 В появившемся окне **Перья тренда** нажмите кнопку **Добавить перо** .



- 3 В открывшемся окне **Средактор привязки тренда** выберите атрибут тега для трендирования или перо **Менеджера трендов**. Если выбранный атрибут не трендируется в **Менеджере трендов**, то для него будет создан **оперативный тренд** (без создания и сохранения архива на диске)
- 4 Для выбора группы настроек истории нажмите на кнопку (под списком выбора атрибутов). В появившемся окне **Группы настроек истории** укажите соответствующую группу и нажмите **ОК**



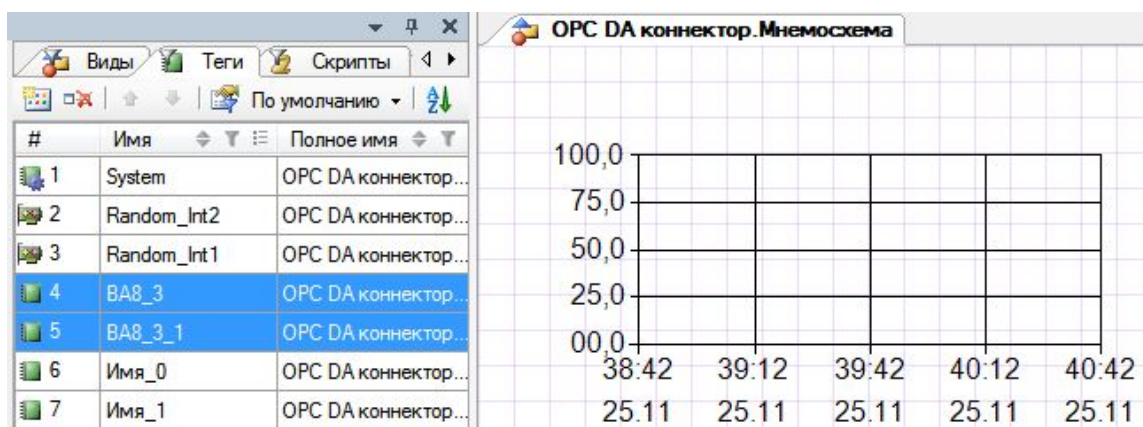


## Добавление перьев с помощью операции drag&drop

Для добавления перьев в графический элемент **Тренд** с помощью операции **drag&drop** («Перетащить и оставить») следует:

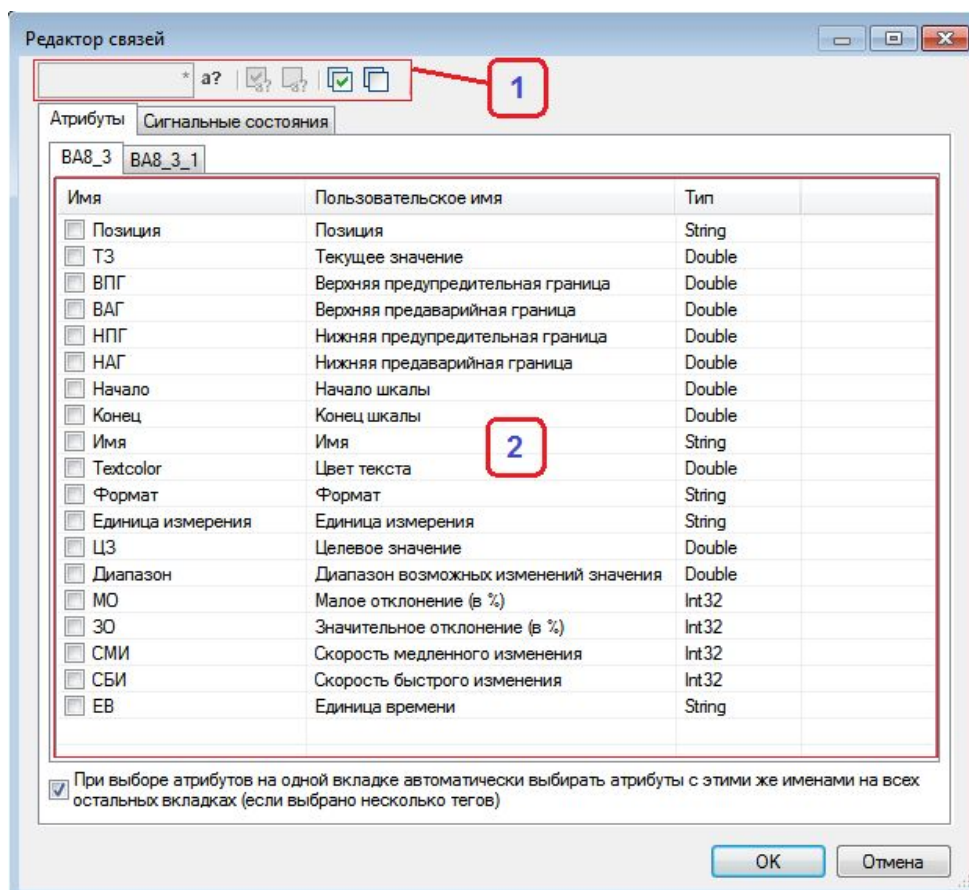
- 1 Отрыть мнемосхему с графическим примитивом **Тренд**
- 2 В левой панели окна Редактор проекта выделить объект, содержащий один или более тегов

- 3 В правой панели окна Редактор проекта перейти на закладку Теги и выделить необходимые теги

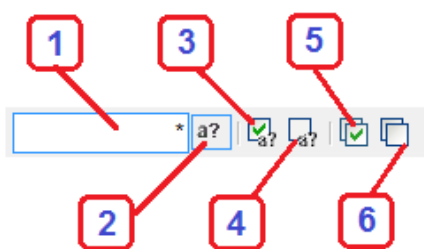


- 4 Переместив указатель в выделенную область, нажать левую кнопку мыши  
 5 Перетащить указатель в область элемента **Тренд** открытой мнемосхемы  
 6 Отпустить левую кнопку мыши

В результате произведенных действий откроется окно **Редактор связей**.

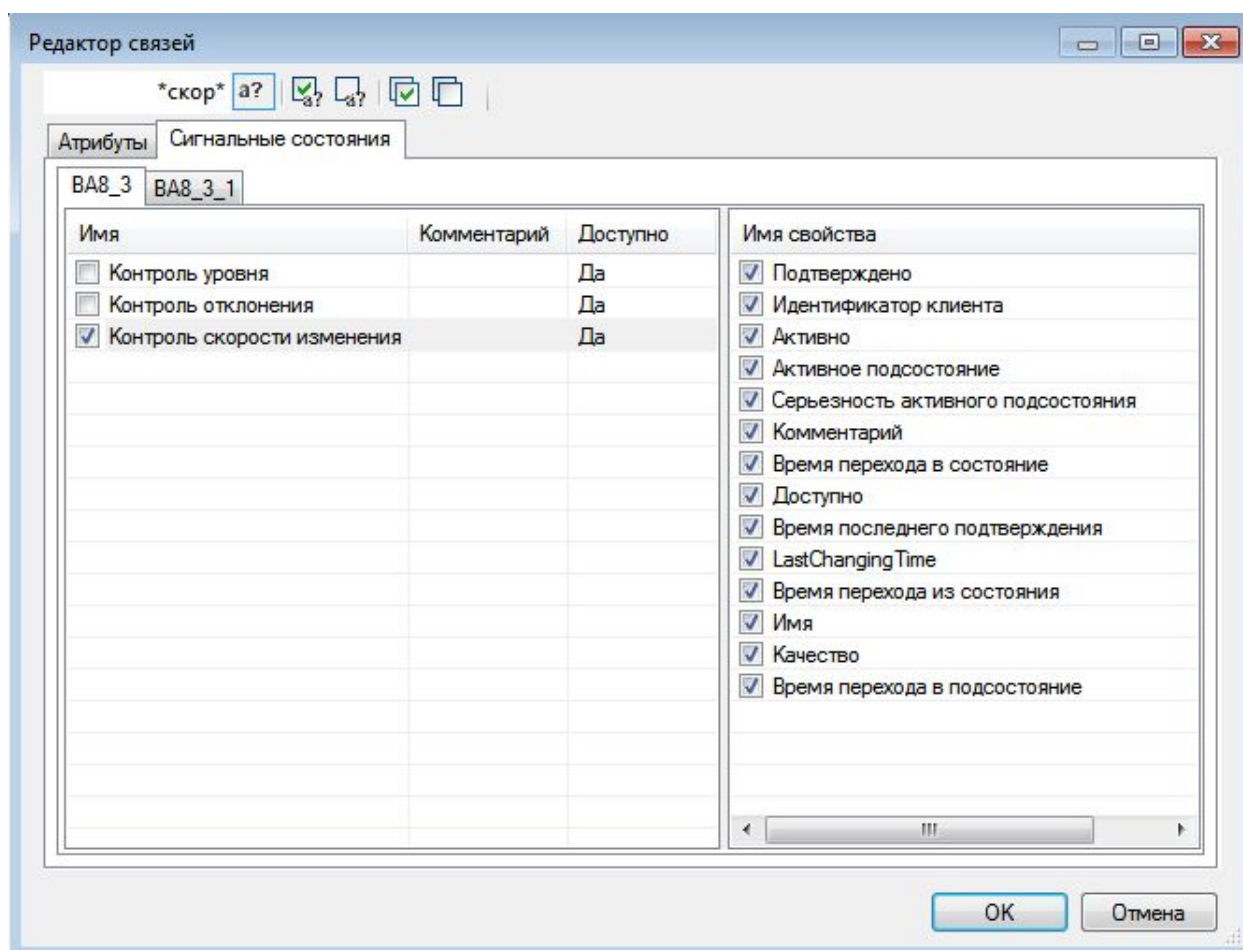


В верхней части окна расположена **Панель Инструментов (1)**. Средняя, рабочая зона окна, представлена многоуровневым страничным навигатором с вкладками **Атрибуты** и **Сигнальные состояния (2)**, которые описывают функциональное назначение области выбора перьев. Сами теги отображаются на вкладках нижнего уровня.



Панель инструментов предназначена для облегчения труда пользователей при наборе перьев тренда и предоставляет следующие сервисные функции по управлению элементами:

- 1 Маскирование путем ручного ввода символов маски в текстовое поле. Разрешается использовать обычные текстовые знаки и метасимволы «?» и «\*», представляющие одиночный знак и группу знаков соответственно



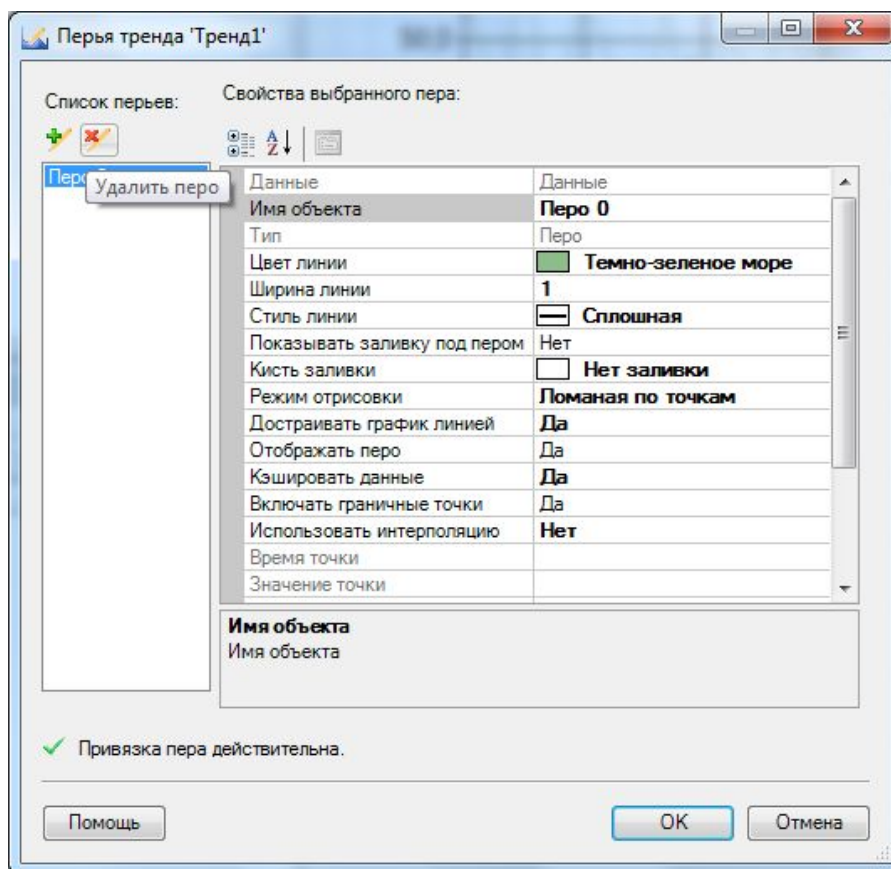
- 2 Включение/выключение маскирования
- 3 Отметка элементов в соответствии с маской
- 4 Снятие отметки с элементов в соответствии с маской
- 5 Отметка всех элементов
- 6 Снятие отметки со всех элементов

В нижней части окна расположены кнопки закрытия диалога. При нажатии на кнопку **ОК** помеченные элементы будут добавлены в список перьев для графического примитива **Тренд**. Нажатие кнопки **Отмена** приведет к отмене операции привязки перьев.

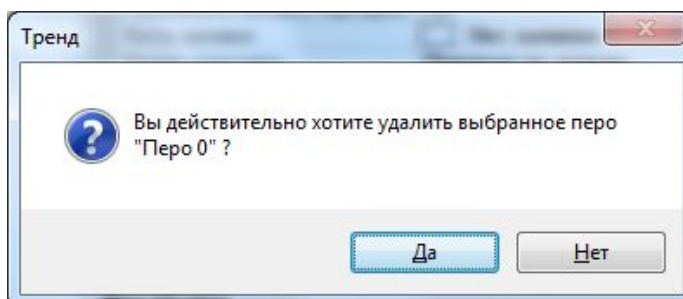
## Удаление перьев

Для удаления пера в окне **Перья тренда** следует:

- 1 Выделить в списке перьев удаляемое перо и нажать кнопку **Удалить перо**



- 2 Подтвердить удаление в окне **Тренд** – нажать на кнопку **Да**. Перо будет удалено из списка перьев элемента Тренд



## 6.4.5 Настройка перьев

Установить параметры отображения перьев тренда позволяет раздел свойств **Настройки перьев**.

### Параметры настройки

- **Отображения перьев** – режим отображения перьев тренда: **Все перья** – групповой тренд, **Одно перо** – одно перо с технологическими границами.
- **Отображать перо** – отображение пера с указанным номером (начинается с 0)
- **Перья** – настройки пера
- **Настройки границ** – настройки границ для режима «Одно перо»
- **Настройки отображения точек** – настройки параметров отображения точек.

<b>Настройки перьев</b>	
Отображение перьев	Все перья
Отображать перо	0
Перья	Перьев всего:
Настройки границ	
Настройки отображения точек	
















## 6.4.5.1 Настройки пера

Для того, чтобы получить доступ к настройкам пера тренда, следует в контекстном меню **Тренда** выбрать пункт меню **Перья** или в окне свойств открыть раздел **Перья** и в списке перьев выбрать перо с нужным номером.

### Параметры настройки

- **Режим отрисовки** – режим отображения тренда
- **Достраивать график линией** – позволяет настроить вид графика пера, когда изменение данных пера происходит медленно. Если значение – **Да**, то при отсутствии данных график будет достраиваться прямой линией параллельной оси времени от последнего значения. Если значение – **Нет**, то график будет строиться при поступлении новых данных
- **Кэшировать данные** – для формирования тренда запрашиваются только обновления данных за последний промежуток времени, а не все данные. Настройка позволяет устранить мерцание тренда при отображении разряженной выборки быстро меняющихся значений и сократить нагрузку на подсистему трендирования. Изменения, произошедшие в истории процесса трендов, не отобразятся на графике до отправки команды пересчета данных.
- **Использовать интерполяцию** – позволяет настроить качество отрисовки тренда. Если интерполяция выключена, прорисовываются все точки тренда, при этом возможны проблемы производительности. Если интерполяция включена, несколько точек интерполируются в одну, отрисовка происходит быстрее, но появляются "скачки" тренда из-за округления.
- **Привязка** – показывает привязку пера тренда к атрибуту тега
- **Конец шкалы** – информация о привязке конца шкалы для пера
- **ВАГ** – информация о привязке верхней предаварийной границы
- **ВПГ** – информация о привязке верхней предупредительной границы
- **НПГ** – информация о привязке нижней предупредительной границы
- **НАГ** – информация о привязке нижней предаварийной границы
- **Начало шкалы** – информация о привязке начала шкалы для пера.

Тип	Перо
Данные	Данные
Имя объекта	<b>Перо 0</b>
Цвет линии	 <b>Черный</b>
Ширина линии	<b>1</b>
Стиль линии	 <b>Сплошная</b>
Показывать заливку под пером	Нет
Кисть заливки	 <b>Нет заливки</b>
Режим отрисовки	<b>Ломаная по точкам</b>
Достраивать график линией	<b>Да</b>
Отображать перо	Да
Кэшировать данные	<b>Да</b>
Использовать интерполяцию	<b>Нет</b>
Время точки	
Значение точки	
 Привязка	Waterflow registration
 <b>Привязки границ</b>	
Показывать индикатор обновлений	<b>Нет</b>
Непрозрачность	<b>50</b>

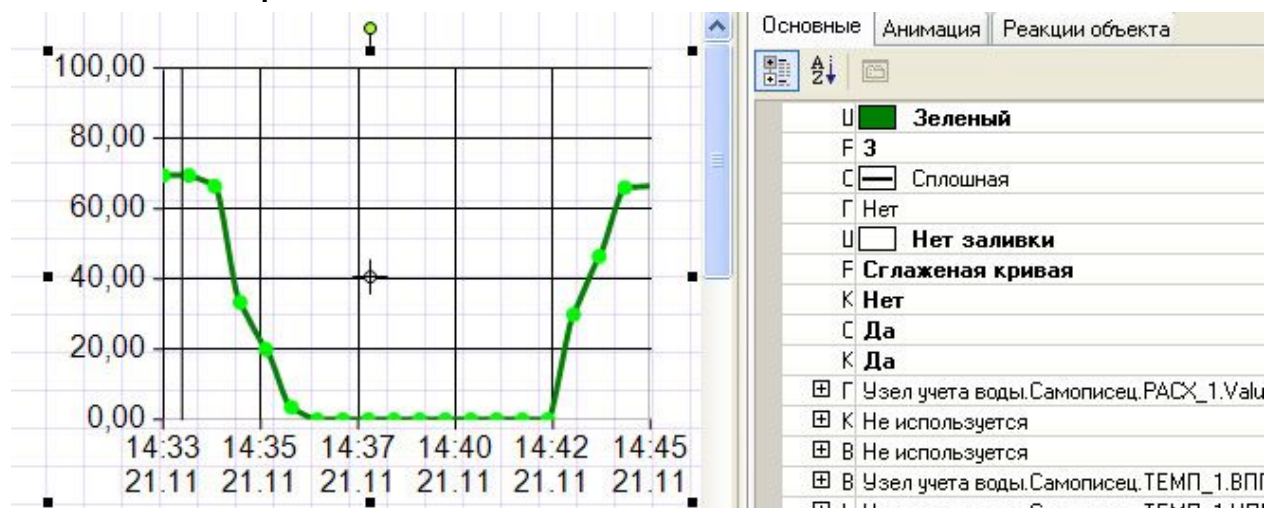
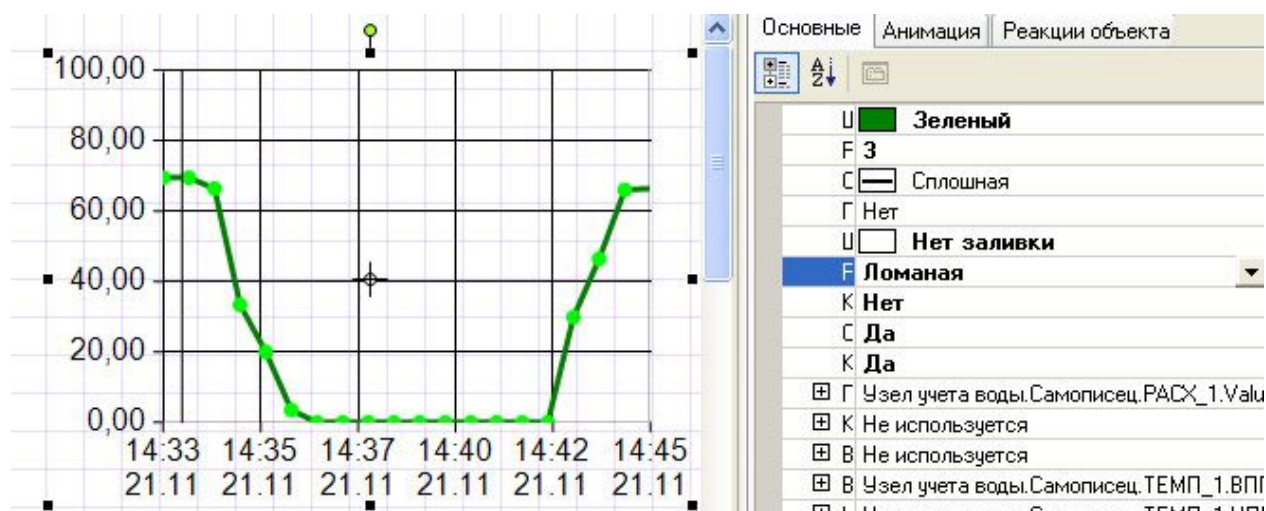
 Привязка	Waterflow registration
Тип	
Привязка	
 Привязки границ	
 <b>Конец шкалы</b>	Не используется
Тип	
Source	
Тип привязки	<b>Не используется</b>
Значение	
 ВАГ	Не используется
 ВПГ	Не используется
 НПГ	Не используется
 НАГ	Не используется
 Начало шкалы	Не используется



## 6.4.5.2 Режимы отрисовки

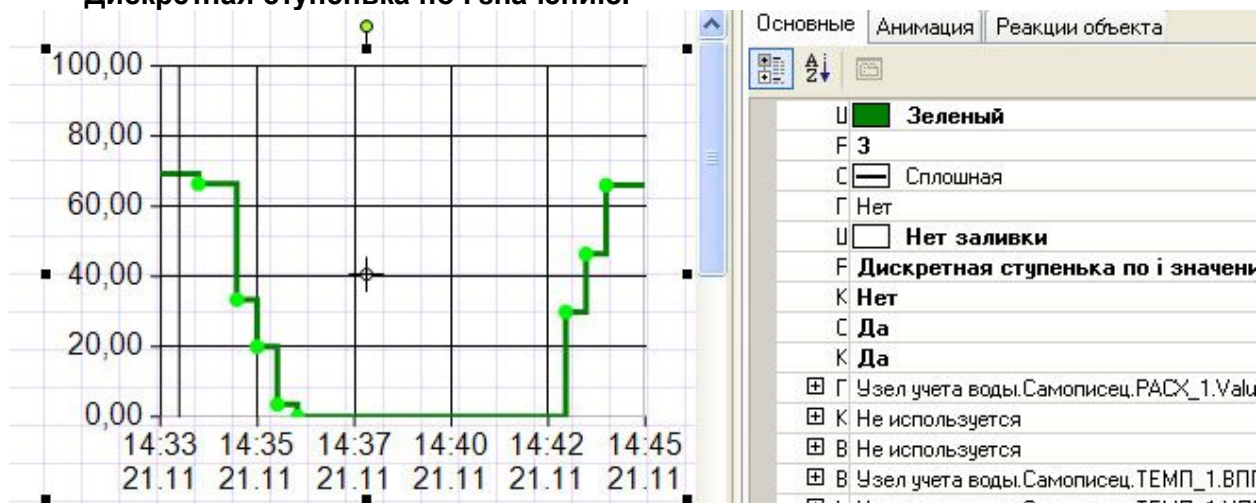
**Интервальный запрос данных – вычисляемые значения**

Отображение тренда строится по вычисленным значениям. В этом случае весь временной интервал делится на заданное число точек, и запрашиваются ближайшие меньшие по времени значения к точкам.

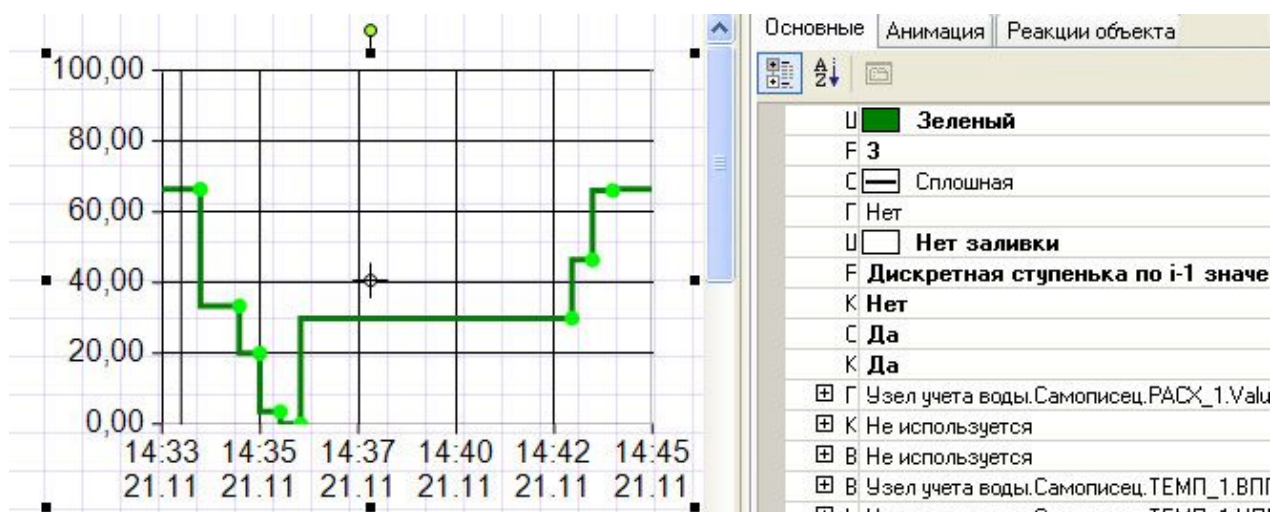
**Сглаженная кривая****Ломаная****Реальные данные**

В этом случае отображение тренда строится по реальным точкам, т.е. каждая точка графика – это изменение трендируемого параметра.

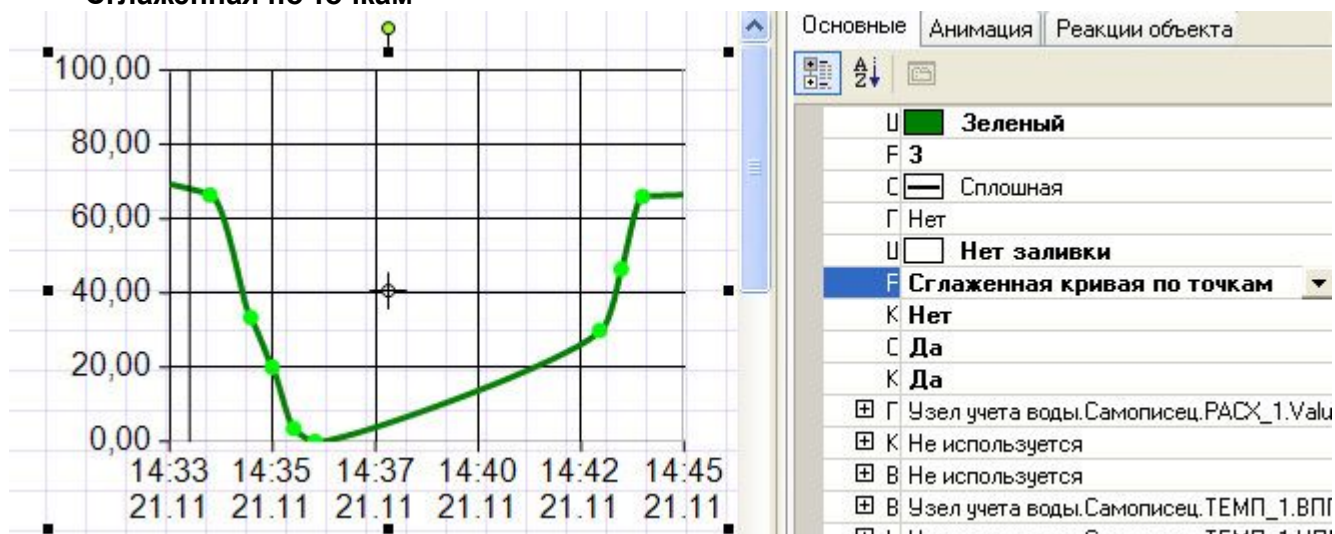
Дискретная ступенька по i значению.



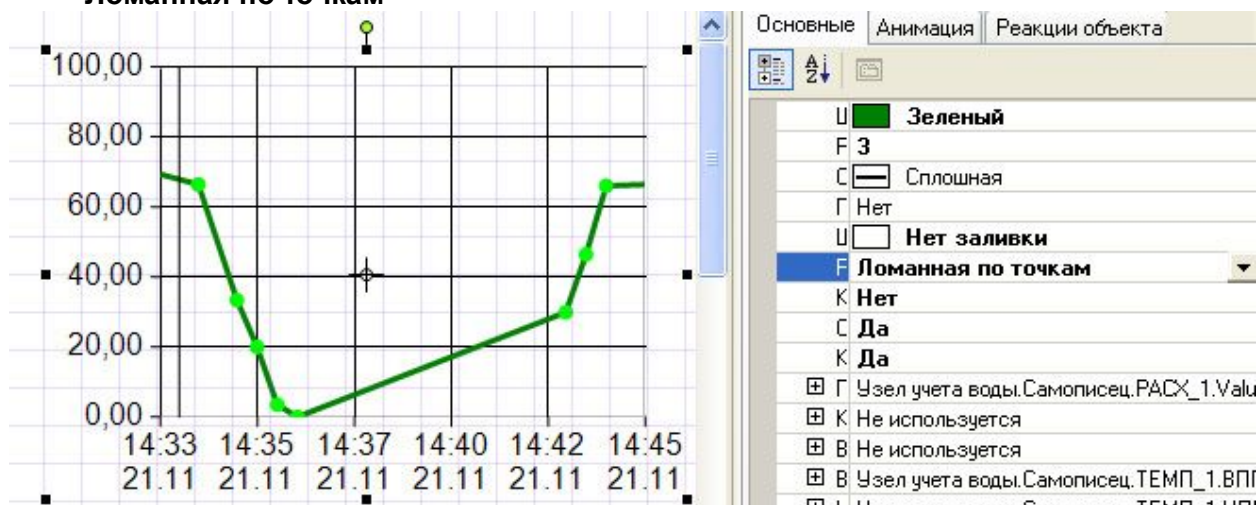
Дискретная ступенька по i-1 значению



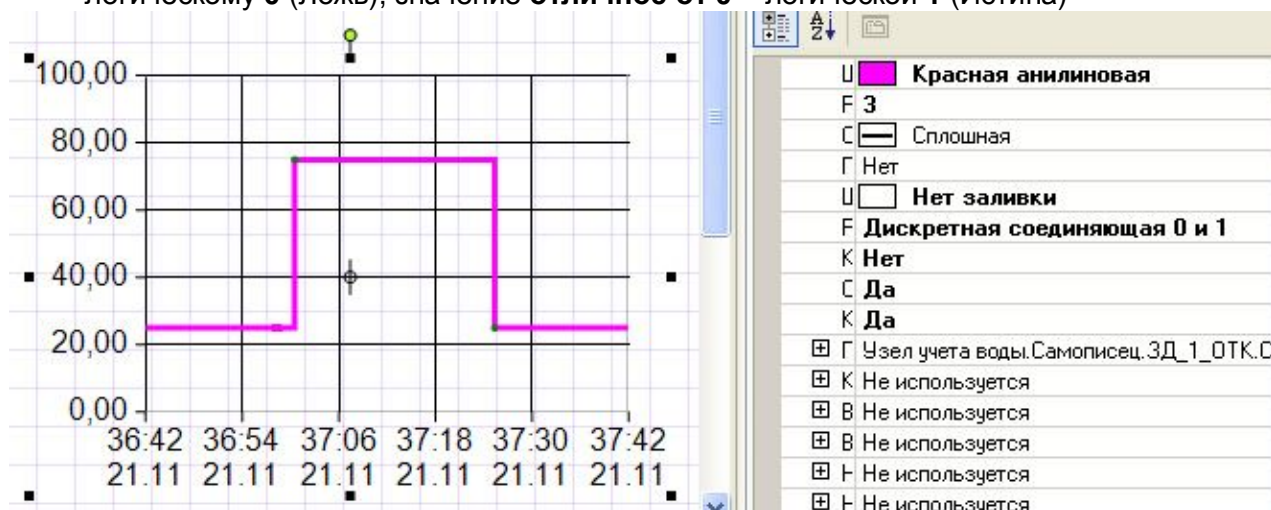
Сглаженная по точкам



Ломанная по точкам

**Дискретная, соединяющая 0 и 1.**

Рассчитана на логические значения. Состояние **0** (Ложь) – 25% шкалы, состояние **не 0** – 75%. Если применяется к нелогическим типам данных, то значение **0** соответствует логическому **0** (Ложь), значение **отличное от 0** – логической **1** (Истина)

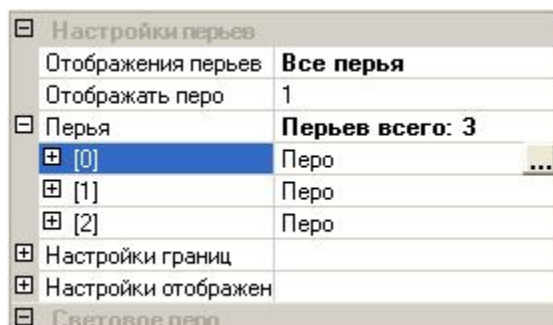


## 6.4.5.3 Редактирование привязки перьев

Редактирование привязок перьев осуществляется в окне **Перья тренда**.

Для вызова данного окна необходимо:

- 1 Выбрать нужное перо в окне свойств и нажать на кнопку  в правой части строки



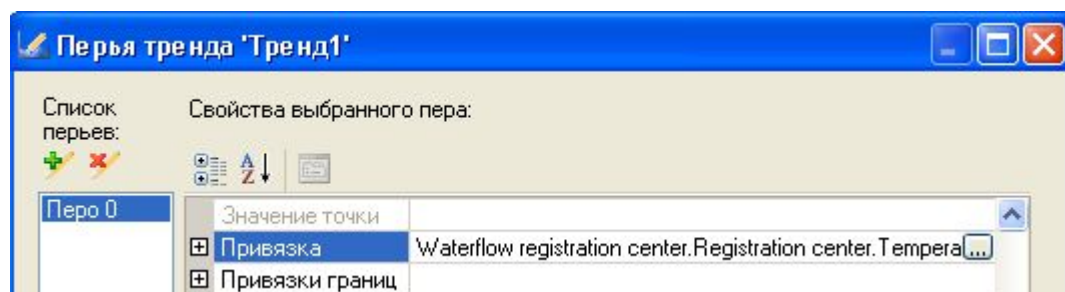
- 2 В появившемся окне **Перья тренда** отредактировать необходимые свойства выделенного пера, например **Привязка**, **Имя группы настроек**, **Привязка границ**.



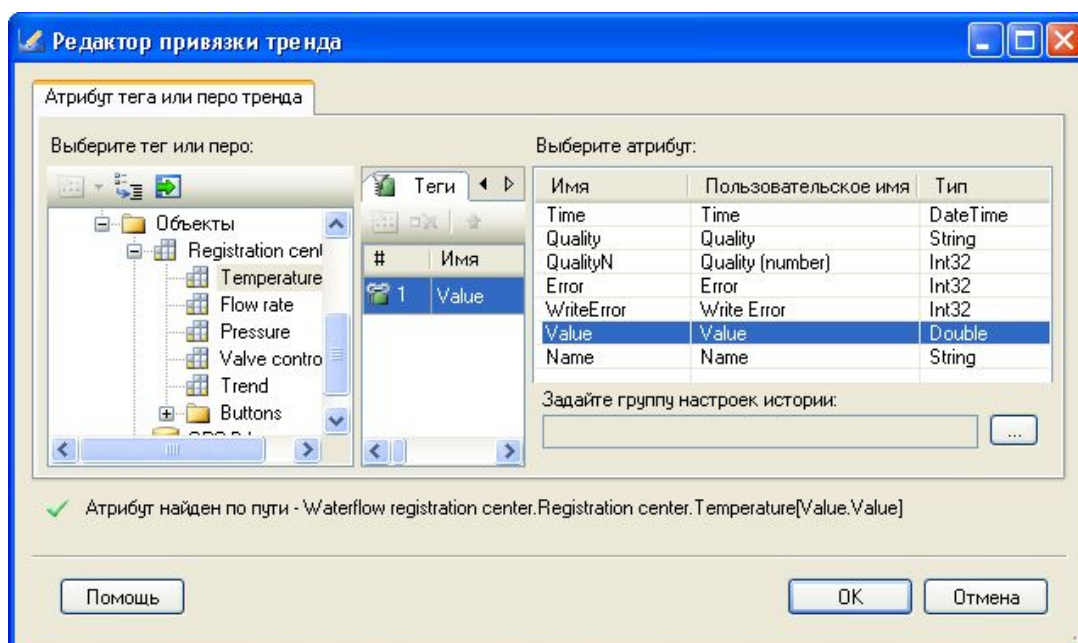
## Изменение привязки пера

Для того чтобы изменить привязку пера, следует:

- 1 В окне **Перья тренда** выделить свойство **Привязка** и нажать появившуюся в строке кнопку **...**

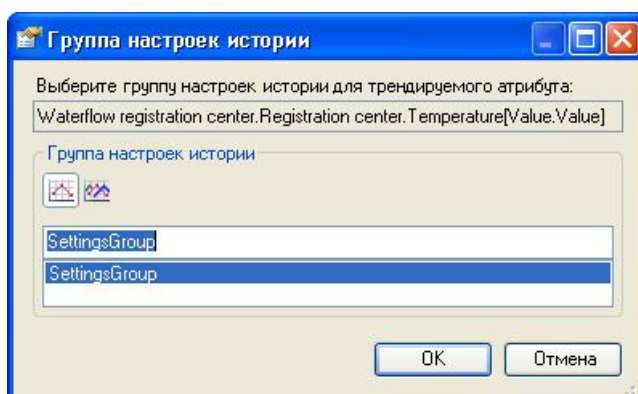


- 2 В открывшемся окне **Редактор привязки тренда** выбрать атрибут тега для трендирования или перо **Менеджера трендов** и нажать **ОК**



Если выбранный атрибут не трендируется в **Менеджере трендов**, то для него будет создан **оперативный тренд** (без создания и сохранения архива).


- 3 В строке **Задайте группу настроек истории** нажать на кнопку **...** и указать в появившемся окне **Группа настроек истории** соответствующую группу и нажать **ОК**.

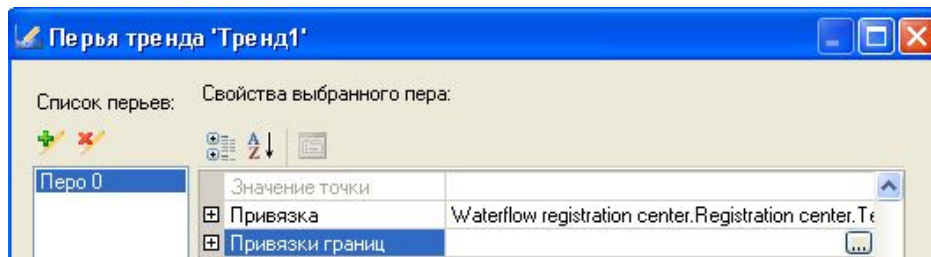


После этого привязка пера будет изменена.

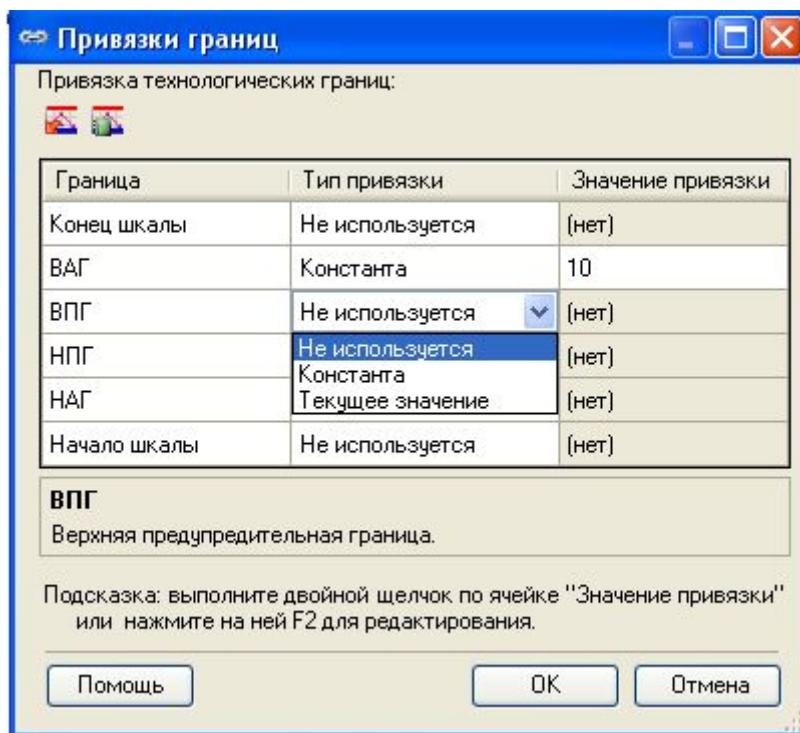
## 6.4.5.4 Привязки границ


Для того, чтобы изменить привязку границ пера, следует:

- 1 Выделить свойство **Привязки границ** и нажать появившуюся в ячейке кнопку 

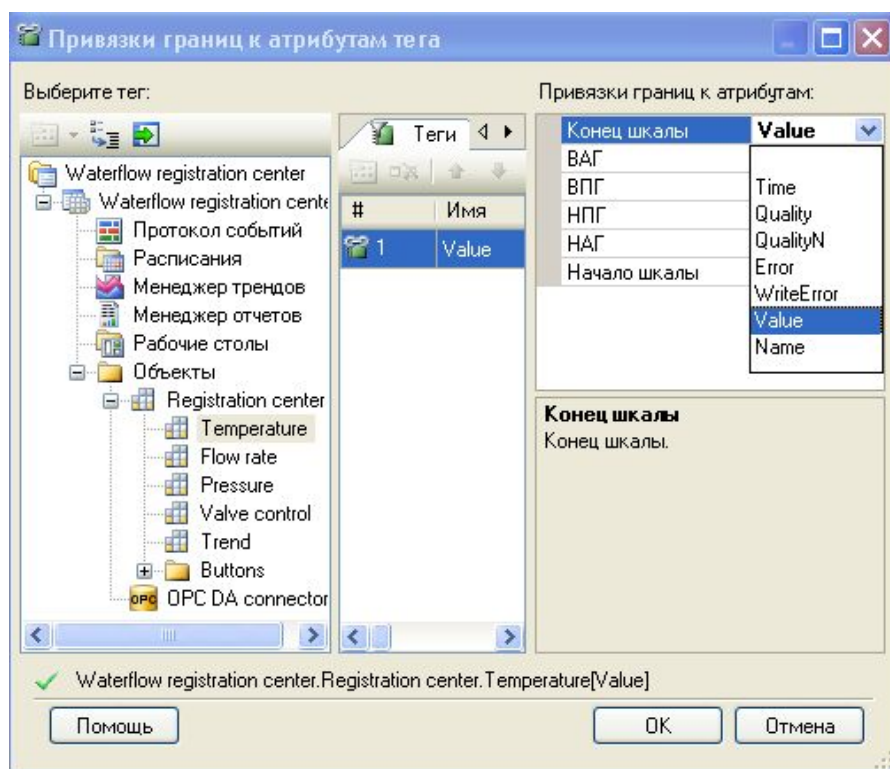


- 2 В открывшемся окне **Привязки границ** задать Тип привязки (Константа, Текущее значение атрибута тега, Не используется) и Значение привязки.



Для задания всех границ сразу от одного тега в окне **Привязки границ** нажать на кнопку  – **Групповая привязка к атрибутам тега**





## 6.4.5.5 Настройки отображения границ

Настройки отображения границ находятся в разделе **Настройки границ** окна **Свойства**

Данные настройки позволят задать отображение границ пера для режима **Одно перо**.

Для каждой границы можно задать:

- **Цвет отображения линии**
- **Размер линии**
- **Стиль линии**

<input checked="" type="checkbox"/> Настройки границ	
<input checked="" type="checkbox"/> Конец шкалы	
Цвет линии	 Синий
Размер линии	1
Стиль линии	 Сплошная
<input checked="" type="checkbox"/> ВАГ	
Цвет линии	 Красный
Размер линии	1
Стиль линии	 Сплошная
<input checked="" type="checkbox"/> ВПГ	
Цвет линии	 Желтый
Размер линии	1
Стиль линии	 Сплошная
<input checked="" type="checkbox"/> НПГ	
Цвет линии	 Желтый
Размер линии	1
Стиль линии	 Сплошная
<input checked="" type="checkbox"/> НАГ	
Цвет линии	 Красный
Размер линии	1
Стиль линии	 Сплошная
<input checked="" type="checkbox"/> Начало шкалы	
Цвет линии	 Синий
Размер линии	1
Стиль линии	 Сплошная

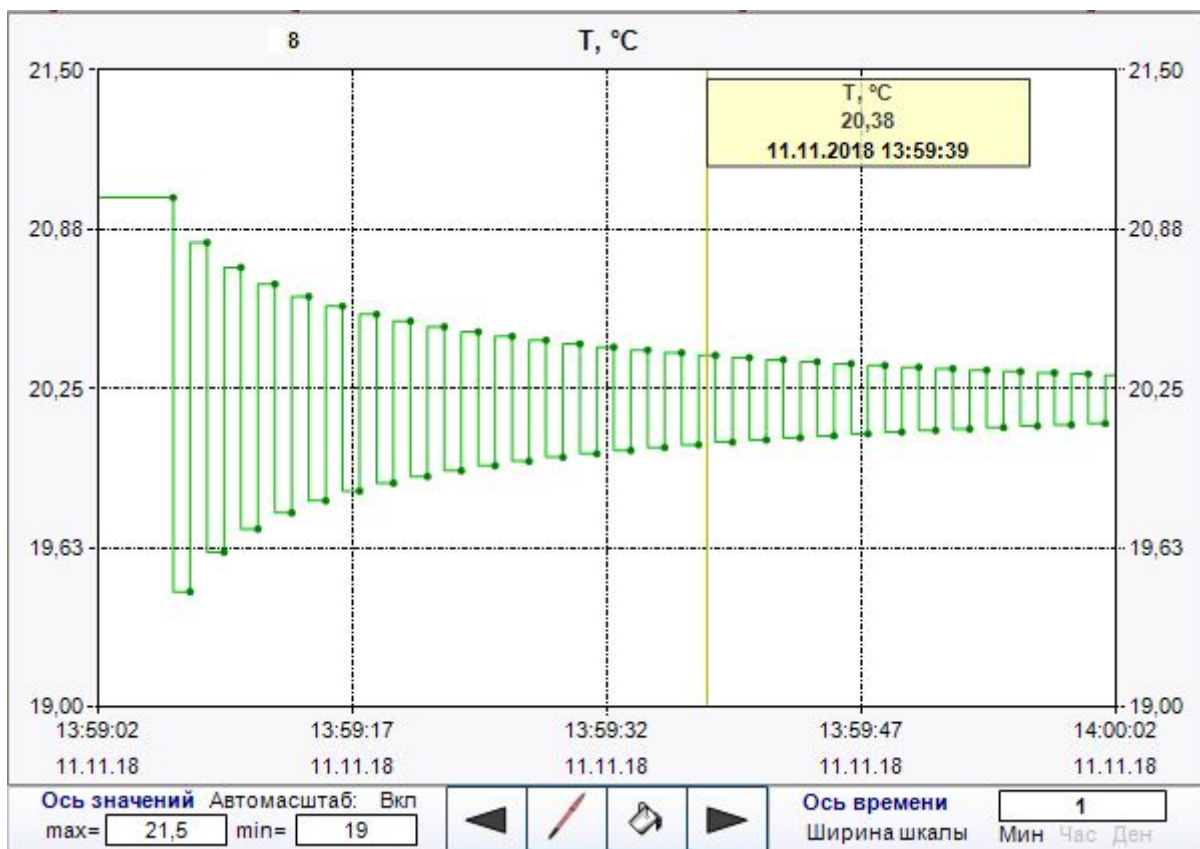
**ВНИМАНИЕ!!!**

В режиме Все перья границы НЕ отображаются.

## 6.4.5.6 Световое перо

**Световое перо** предназначено для получения информации о значениях пера тренда в момент времени, который отмечается курсором на графике тренда.

Световое перо отображается как вертикальная линия, перемещаемая по тренду курсором при нажатии левой клавиши мыши.



## Настройка светового пера

Настройки светового пера находятся в разделе свойств **Световое перо**:

- **Световое перо** – устанавливает режим светового пера (**Да**) или отменяет его (**Нет**)
- **Время отрисовки** – момент времени, от которого (по направлению в прошлое) считываются данные для отображения тренда
- **Время точки** – момент времени, имеющий ближайшее меньшее значение относительно позиции светового пера (только для чтения)
- **Значение точки** – значение пера тренда в момент времени **Время точки** (только для чтения)
- **Позиция пера** – позиция пера в пикселах относительно области тренда (только для чтения)
- **Цвет пера** – цвет линии светового пера
- **Ширина линии** – ширина линии светового пера
- **Стиль линии** – стиль линии светового пера

<input checked="" type="checkbox"/> <b>Световое перо</b>	Нет
Световое перо	Нет
Время отрисовки	09.10.2013 9:49
Время точки	
Значение точки	
Цвет светового пера	<span style="display: inline-block; width: 10px; height: 10px; background-color: black; border: 1px solid black;"></span> Черный
Ширина линии	1
Стиль линии	<span style="display: inline-block; width: 20px; border-bottom: 2px solid black;"></span> Сплошная

### **ВНИМАНИЕ!!!**

Необходимо понимать, что свойства **Время точки** и **Значение точки** для светового пера показывают ближайший момент времени и значение именно этого светового пера.

А для тренда в целом эти свойства отображают ближайшее время и значение того пера, точка которого расположена ближе, чем у остальных перьев, т.е. для одного пера ближайшее время и значение будут совпадать с аналогичными свойствами тренда

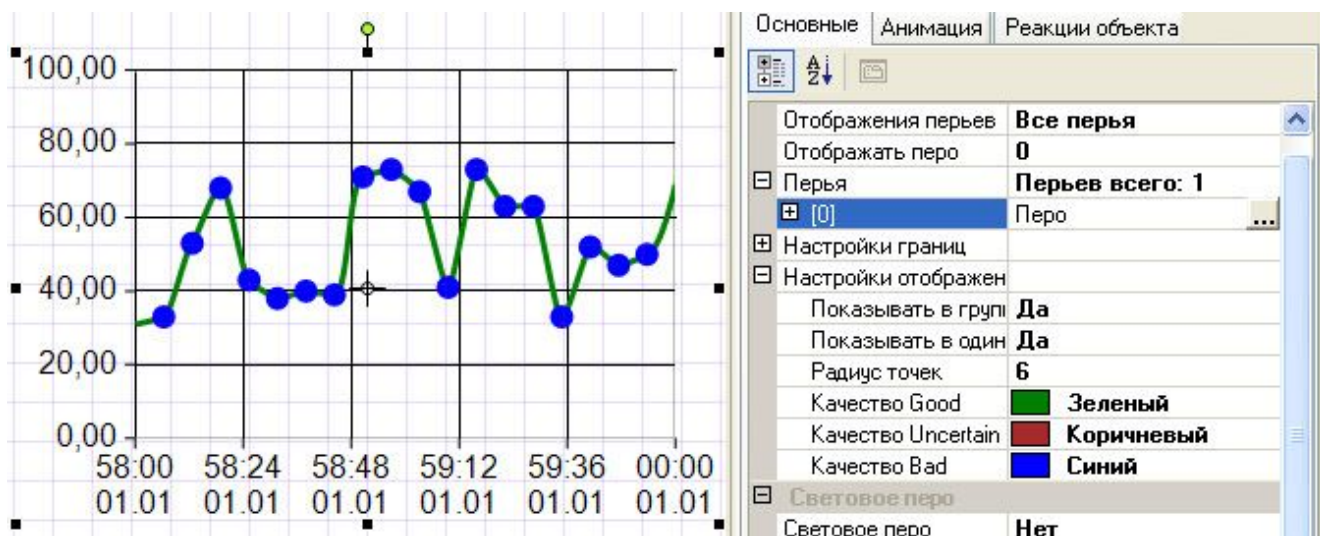
#### 6.4.5.7 Настройки отображения точек

##### **Цвет и качество**

Графический элемент **Тренд** позволяет отображать точки, соответствующие значениям тренда, различными цветами.

Цвет зависит от значения **качества**:

- **Качество Good** – цвет отображения значений хорошего качества
- **Качество Uncertain** – цвет отображения значений неопределенного качества
- **Качество Bad** – цвет отображения значений плохого качества



##### **Параметры отображения**

Кроме настройки цвета Вы можете настроить и другие параметры отображения точек:

- **Показывать в групповом режиме** – настройка показывать или нет точки в групповом режиме
- **Показывать в одиночном режиме** – настройка показывать или нет точки в режиме одного пера
- **Радиус точек** – радиус изображения точки.

#### 6.4.6 Управление графическим элементом Тренд

**DataRate** предоставляет возможность управлять свойствами графического элемента **Тренд** в реальном времени. Для этого необходимо добавить на мнемосхему графические примитивы, с помощью которых можно управлять свойствами пера тренда, и настроить их реакции.

##### Функции управления

- Выбор текущего пера
- Переключение режимов **Одно перо/Все перья**
- Отключение отображения перьев
- Включение и выключение заливки под пером
- Просмотр истории процесса.

##### 6.4.6.1 Выбор текущего пера

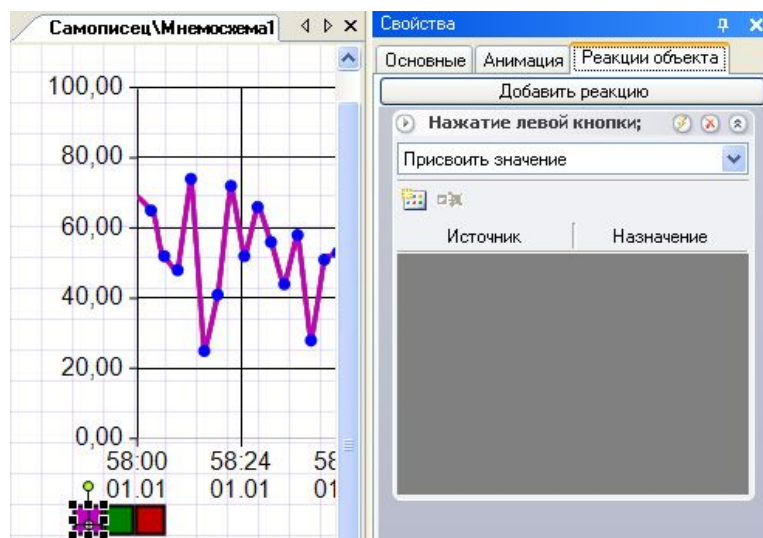
Выбор пера сводится к управлению свойством **Отображать перо** раздела свойств **Настройка перьев** и применяется:

- В режиме **Одно перо** для отображения и просмотра истории процесса по конкретному перу тренда.
- В режиме **Все перья** (режим группового тренда) для отображения значений и просмотра истории этого пера поверх всех перьев тренда.

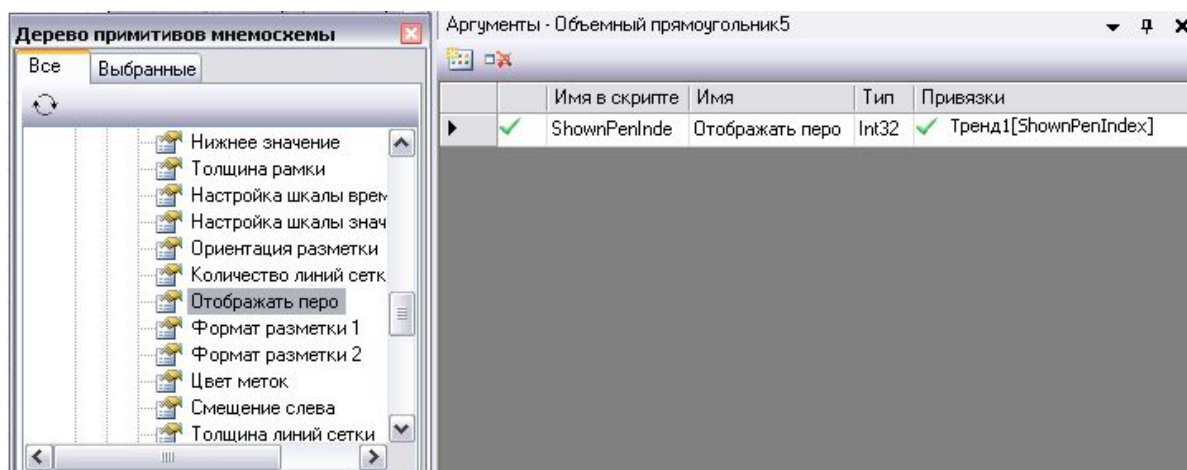
Настройки перьев	
Отображение перьев	Все перья
Отображать перо	0
Перья	Перьев всего:
Настройки границ	

Для управления выбором пера следует:

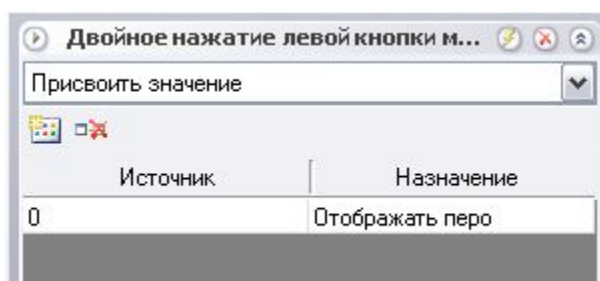
- 1 Создать объект управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**.



- 2 Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки. Перетащить свойство **Отображать перо (ShownPenIndex)** в окно аргументов.



- 3 Выбрать в поле **Назначение** атрибут **Отображать перо**, а в поле **Источник** указать индекс пера в примитиве **Тренд** (в примере номер пера равен 0).



- 4 Готово. Если при выполнении приложения щелкнуть по объекту-кнопка, то будет выбираться заданное перо (как в режиме **Одно перо**, так и в режиме **Все перья**).

## 6.4.6.2 Переключение режимов Одно перо/Все перья

За переключения режимов **Одно перо/Все перья** отвечает свойство **Отображение перьев** раздела **Настройки перьев**.

[-] <b>Настройки перьев</b>	
Отображение перьев	Все перья
Отображать перо	0
[+] <b>Перья</b>	<b>Перьев всего:</b>
[-] <b>Настройки границ</b>	

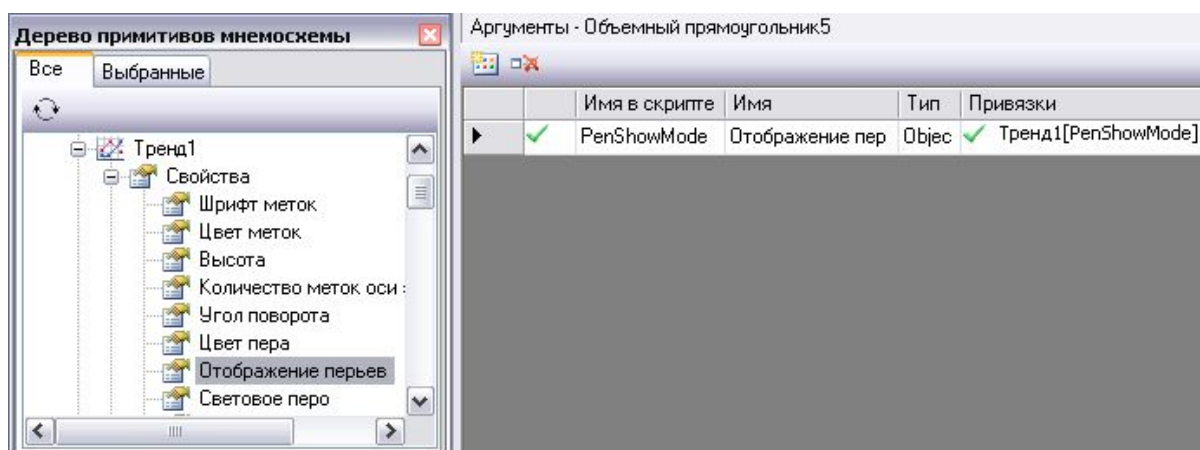
Данное свойство имеет следующие значения:

- Режим **Одно перо**:  
[Krug.ObjectModel.HistoricalService.Primitives.PrimitiveTrend.PenShowMode.SinglePen](#)
- Режим **Все перья**:  
[Krug.ObjectModel.HistoricalService.Primitives.PrimitiveTrend.PenShowMode.AllPens](#).

Для управления переключением режимов следует:

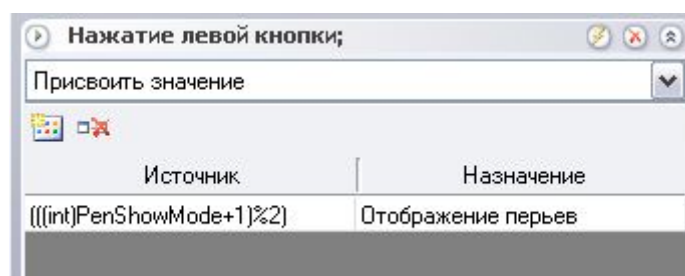
- 1 Создать объект управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
- 2 Вызвать окно **Дерево инемосхемы** и окно **Аргументы** объекта-кнопки
- 3 Перетащить свойство **Отображения перьев (PenShowMode)** в окно аргументов





4. Выбрать в поле **Назначение** атрибут **Отображения перьев**, а в поле **Источник** написать выражение, которое переключает значение свойства **PenShowMode** из **SinglePen** в **AllPens** и обратно:

`(Krug.ObjectModel.HistoricalService.Primitives.PrimitiveTrend.PenShowMode)`  
`((int)PenShowMode+1)%2`



5. Готово. Если при выполнении приложения щелкнуть по объекту-кнопка, то будет осуществляться переключение из режима **Одно перо** в режим **Все перья** (и обратно).

#### 6.4.6.3 Включение заливки и отключение отображения перьев

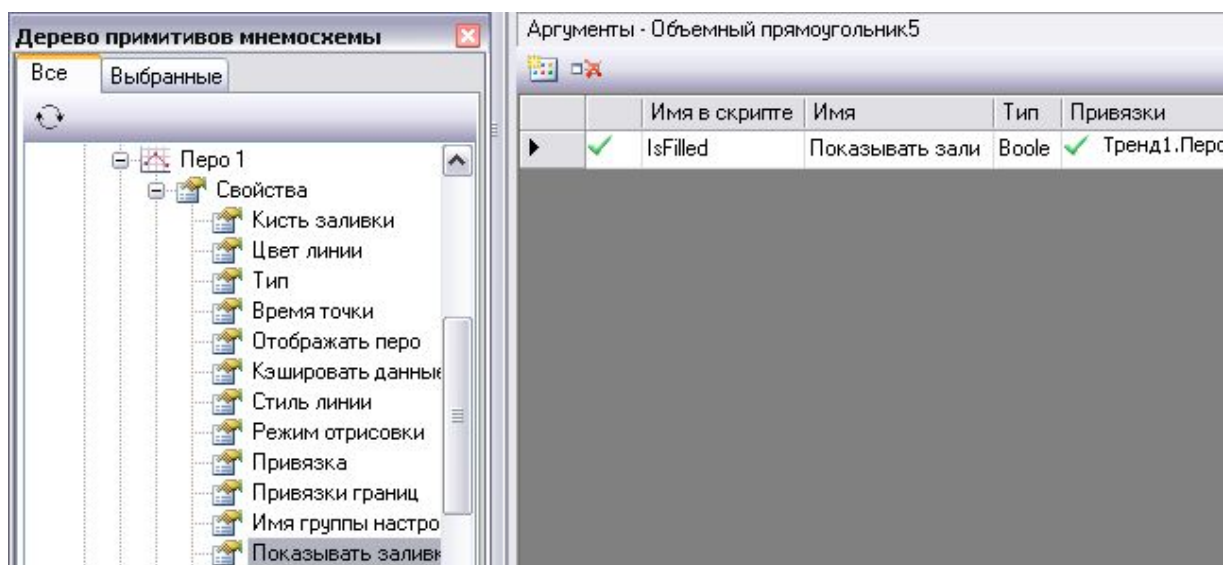
**Выделить значения одного пера** тренда можно с помощью заливки заданным цветом область тренда под этим пером. Управлять заливкой можно с помощью установки свойства **Показывать заливку под пером** (**IsFilled**).

#### **ВНИМАНИЕ!!!**

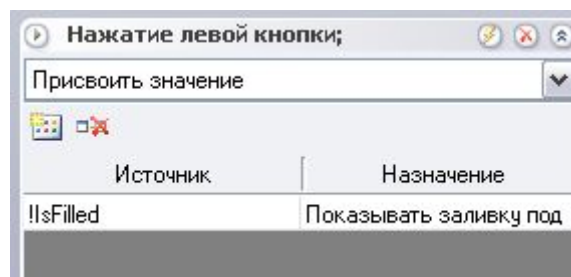
Особенностью доступа к данному свойству является то, что привязку к нему можно организовать только из дерева мнемосхем. При использовании дерева мнемосхем, ориентироваться следует не на номер пера, а на его имя в дереве.

Для управления переключением режимов следует:

1. Создать объект управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
2. Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
3. Перетащить свойство **Показывать заливку под пером** (**IsFilled**) в окно аргументов



4. Выбрать в поле **Назначение** атрибут **Показывать заливку под пером**, а в поле **Источник** написать выражение **!IsFilled** (восклицательный знак перед названием свойства), которое при щелчке по кнопке переключает значение свойства **IsFilled** из значения **Истина** в значение **Ложь** (и обратно).
5. Готово. Если при выполнении приложения щелкнуть по кнопке, то будет либо осуществляться заливка области тренда для выделенного пера, либо тренд будет отображаться без заливки.



## Удаление изображения пера из тренда

Для удаления изображения пера из тренда используется свойство пера **Отображать перо** (**PenVisible**), управление которым аналогично управлению заливкой пера (для отключения используется выражение – **!PenVisible**).

### 6.4.6.4 Просмотр истории

Для управления просмотром значений пера на некотором интервале времени используется режим **Световое перо**.

Управление свойствами тренда сводится к решению трех задач:

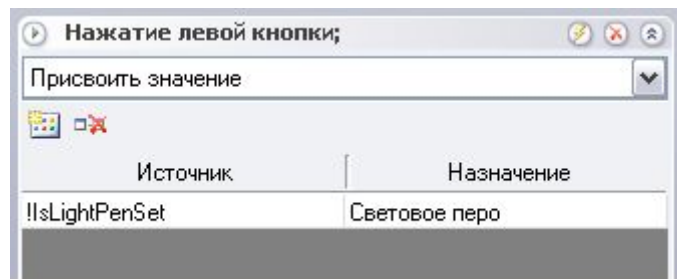
- Перевод графического элемента **Тренд** в режим **Световое перо**
- Управление перемещением по истории пера («в прошлое» и «в будущее»)
- Получение значения пера и соответствующего ему значения момента времени при перемещении по истории пера.

## Перевод графического элемента Тренд в режим Световое перо

Для управления переключением режимов пера (в режим **Световое перо** и обратно) следует:

1. Создать объекту управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
2. Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
3. Перетащить свойство **Световое перо** (**IsLightPenSet**) в окно аргументов

4. Выбрать в поле **Назначение** атрибут **Световое перо**, а в поле **Источник** написать выражение `!!sLightPenSet` (восклицательный знак перед названием свойства), которое при щелчке по кнопке переключает значение свойства **IsLightPenSet** из значения **Истина** в значение **Ложь** (и обратно).

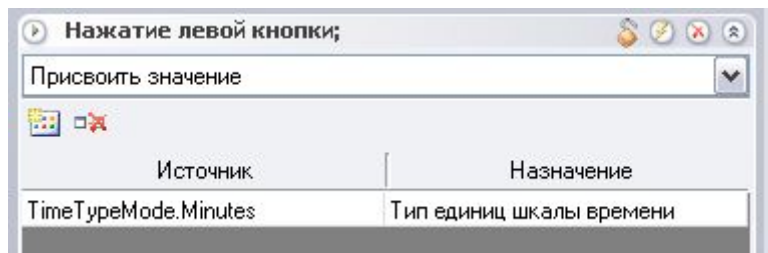


5. Готово. Если при выполнении приложения щелкнуть по кнопке, то будет осуществляться переключение в режим **Световое перо** (и обратно).

### Изменение единиц шкалы времени тренда

Для изменения единиц шкалы времени тренда следует:

1. Создать объект управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
2. Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
3. Перетащить свойство **Тип единиц шкалы времени** в окно аргументов
4. Выбрать в поле **Назначение** атрибут **Тип единиц шкалы времени**, а в поле **Источник** написать выражение для задания требуемых единиц шкалы времени:
  - `TimeTypeMode.Seconds` – в секундах
  - `TimeTypeMode.Minutes` – в минутах
  - `TimeTypeMode.Hours` – в часах
  - `TimeTypeMode.Days` – в днях.
5. Готово. Если при выполнении приложения щелкнуть по объекту-кнопка, то единицы шкалы времени тренда изменятся на заданные.



### Управление перемещением по истории пера

Для управления перемещением по истории пера следует:

1. Создать объекты управления – кнопки (например, прямоугольники) для просмотра истории назад («в прошлое») и вперед («в будущее»). Добавить кнопкам реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
2. Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
3. Перетащить свойства **Ширина шкалы времени** и **Время отрисовки** в окно аргументов каждой кнопки.
 

**Ширина шкалы времени** – интервал времени (структура **TimeSpan**), на котором отображается тренд.

**Время отрисовки** – момент времени, начиная с которого строится график тренда по направлению «в прошлое».

Аргументы - Объемный прямоугольник1

	Имя в скрипте	Имя	Тип	Привязка
✓	DrawingTime	Время отрисовки	DateTime	Табличный тренд1[DrawingTime]
✓	TimeWindowWidth	Ширина шкалы времени	Object	Табличный тренд1[TimeWindowWidth]

4. Выбрать для каждой кнопки в поле **Назначение** атрибут **Время отрисовки**, а в поле **Источник** написать выражение:
- для кнопки просмотра в прошлое на половину шкалы тренда  
`DrawingTime.AddTicks(-((TimeSpan)TimeWindowWidth).Ticks/2)`
  - для кнопки просмотра в будущее на половину шкалы тренда  
`DrawingTime.AddTicks(((TimeSpan)TimeWindowWidth).Ticks/2)`

Нажатие левой кнопки;

Присвоить значение

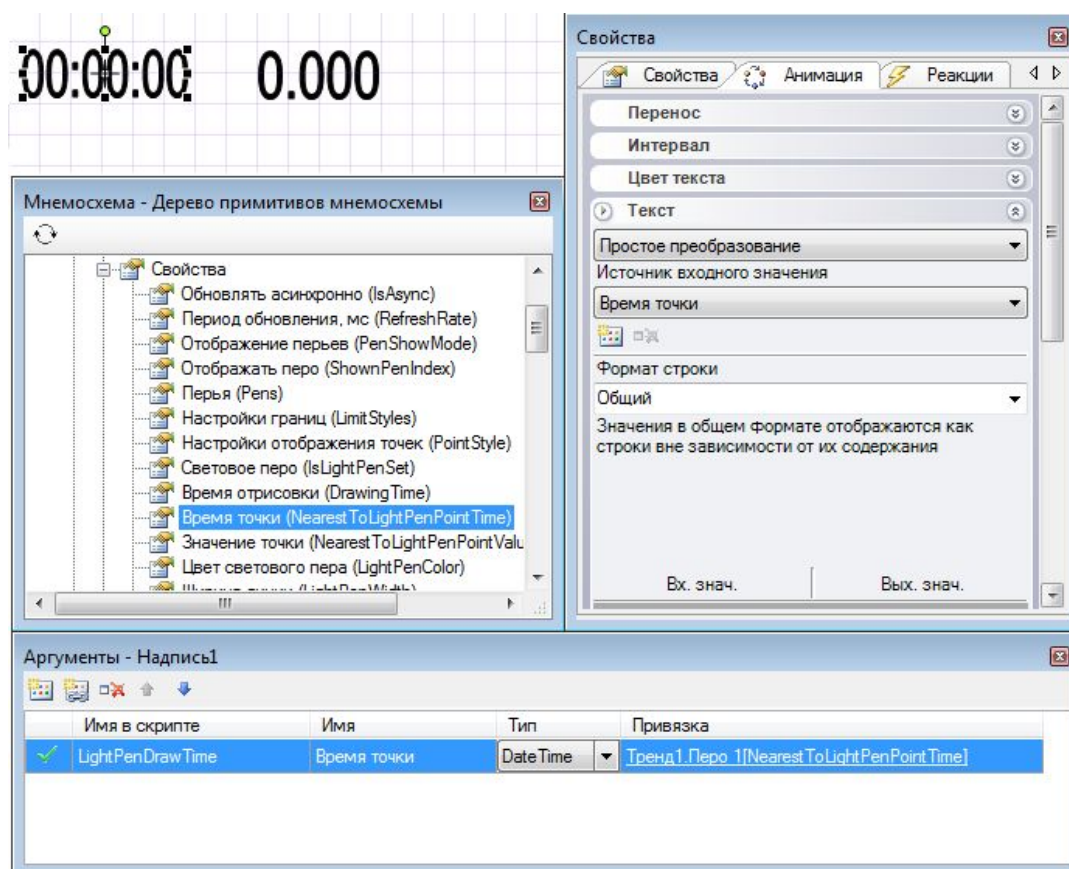
Источник	Назначение
<code>DrawingTime.AddTicks(-((TimeSpan)TimeWindowWidth).Ticks/2)</code>	Время отрисовки

5. Готово. Если при выполнении приложения щелкнуть по объекту-кнопка просмотра «в прошлое», то будет отображаться тренд на интервале времени, "сдвинутым" на половину шкалы тренда назад.  
 При щелчке по объекту-кнопка просмотра «в будущее» будет отображаться тренд на интервале времени, "сдвинутым" на половину шкалы тренда вперед.

### Получение значения пера и соответствующего ему значения момента времени

Используем возможность получить значение пера тренда и значение момента времени, как значения в момент времени, ближайшего меньшего от позиции светового пера.

Для того, чтобы **отобразить значение момента времени в позиции светового пера**, необходимо связать свойство **Время точки** графического элемента **Тренд** с анимацией примитива **Текст**, используемого для отображения значения времени.



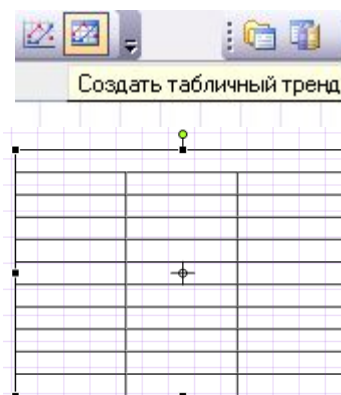
Для того, чтобы **отобразить значение пера в позиции светового пера**, необходимо связать свойство **Значение точки** графического элемента **Тренд** с анимацией примитива **Текст**, используемого для отображения значения пера.

## 6.5 Графический элемент Табличный тренд

**Табличный тренд** представляет собой графический элемент, предназначенный для отображения тренда в виде таблицы.

Для создания графического элемента **Табличный тренд** на мнемосхеме следует:

- 1 Выбрать иконку графического элемента **Табличный тренд** на панели инструментов (или соответствующий пункт в меню)
- 2 Нажать левую клавишу мыши на мнемосхеме и растянуть рамку до нужных размеров графического элемента



Табличный тренд может формироваться двумя способами:

- **Интервальным (интервальный табличный тренд)**, когда метка времени для каждой строки таблицы определяется исходя из заданной ширины шкалы времени и количества точек. Для каждой метки времени отображается заданное в настройках значение пера тренда на интервале между двумя временными точками.
- **По изменению (табличный тренд по изменению)**, когда тренд заполняется значениям по изменению с заданной апертурой, количество которых задается числом строк тренда



## Свойства

Свойства, расположенные в секциях **Основные**, **Положение**, **Таблица**, **Форматирование**, наследуются от графического примитива **Таблица**.

Свойства, расположенные в секциях **Запрос данных**, **Листание**, **Световое перо**, являются специфичными для примитива **Табличный тренд**.

Свойство **Колонки** в секции **Форматирование** представляет список колонок графического элемента **Табличный тренд**. Типы колонок: **Перо** (Описание свойств колонки Перо) и **Время** (Описание свойств колонки Время).

### 6.5.1 Настройка запроса данных

Группа свойств **Запрос данных** позволяет настроить выборку данных табличного тренда.

- **Направление** – направление вывода данных в таблице (от заданного времени в прошлое):
  - **Сверху вниз** – данные выводятся сверху вниз (точка с наибольшим значением временной метки располагается сверху таблицы)
  - **Снизу вверх** – данные выводятся снизу вверх (точка с наибольшим значением временной метки располагается внизу таблицы)
- **Количество точек** – количество точек тренда, равно количеству строк в табличном тренде
- **Нулевое значение** – строковое значение, которое подставляется в ячейку таблицы, если значение точки табличного тренда для этой ячейки отсутствует
- **Тип тренда** – тип формирования выборки данных табличного тренда:
  - **Интервальный**
  - **По изменению**
- **Ширина шкалы времени** - ширина шкалы времени в формате DateTime. Свойство **только для чтения**. Используется в основном в скриптах

Запрос данных	
Направление	Снизу вверх
Количество точек	10
Нулевое значение	Null
Тип тренда	Интервальный
Тип единиц шкалы времени	Секунды
Ширина шкалы времени в единицах	120
Постоянные интервалы	Нет
Ширина шкалы времени	00:02:00

Следующие свойства доступны только для интервального тренда:

- **Тип единиц шкалы времени** – тип единиц шкалы времени (**секунды**, **минуты**, **часы**, **дни**) данных интервального тренда
- **Ширина шкалы времени в единицах** – ширина шкалы времени запроса данных интервального табличного тренда в установленных единицах
- **Постоянные интервалы** – определяет временное значение метки. Если значение **Да**, то при отсутствии точки с заданным временем ищется ближайшая точка и берется ее значение и временное значение интервала, если **Нет**, то берется точка с собственной временной меткой. Свойство доступно для редактирования только для интервального табличного тренда при условии, что есть только одна колонка пера с заданной привязкой. Во всех остальных случаях значение свойства **Да** и оно доступно **только для чтения**.

## Дополнительные свойства

- **Выравнивание рядов данных** – определяет способ выравнивания данных в табличном тренде по изменению, когда количество привязанных колонок перьев больше 1. Выравнивание осуществляется с учетом **Апертуры**. Свойство может принимать следующие значения:

- **По длинному ряду** – определяется самый длинный по времени ряд данных. Временные метки его точек кладутся в основу временных интервалов, по которым осуществляется получение значений точек других рядов данных (как в интервальном табличном тренде).
- **По короткому ряду** – определяется самый короткий по времени ряд данных. Временные метки его точек кладутся в основу временных интервалов, по которым осуществляется получение значений точек других рядов данных (как в интервальном табличном тренде)
- **Апертура** – апертура для выбора точек. Задаёт минимальное изменение значения, при котором точка будет добавлена в табличный тренд по изменению

Тип тренда	По изменению
Выравнивание рядов данных	По короткому ряду
Апертура	0
Если нет точки на интервале	Нулевое значение
Если несколько точек на интервале	Первая на интервале

Следующие свойства доступны для интервального тренда, если свойство **Постоянные интервалы** имеет значение **Да**, и для табличного тренда по изменению, если количество колонок перьев с заданной привязкой больше 1

- **Если нет точки на интервале** – определяет, какое значение подставлять, если на интервале нет точки:
  - **Нулевое значение** – подставляется строковое значение, заданное свойством **Нулевое значение**
  - **Предыдущая точка** – подставляется значение точки с предыдущего интервала
- **Если несколько точек на интервале** – определяет, какое значение подставлять, если на интервале несколько точек:
  - **Первая на интервале** – берется первая точка на интервале
  - **Последняя на интервале** – берется последняя точка на интервале
  - **Среднее** – берется среднее значение всех точек на интервале
  - **Максимум** – берется точка с максимальным значением на интервале
  - **Минимум** – берется точка с минимальным значением на интервале.

Тип тренда	Интервальный
Тип единиц шкалы времени	Секунды
Ширина шкалы времени в единицах	120
Постоянные интервалы	Да
Если нет точки на интервале	Нулевое значение
Если несколько точек на интервале	Первая на интервале

## 6.5.2 Просмотр истории

Для просмотра истории процесса в табличном виде предназначен режим **Листание**.

- **Время отрисовки** – задает, с какого времени будут отображаться данные табличного тренда. Отображение данных осуществляется с заданного времени в прошлое
- **Листание** – значение **Да** задает режим листания, в противном случае – значение **Нет**.

<b>Листание</b>	
Время отрисовки	01.01.2000 15:00
Листание	Нет

### Управление временем отрисовки

Для управления временем отрисовки интервального табличного тренда следует:

- 1 Создать объекты управления – кнопки (например, прямоугольник) для изменения времени отрисовки назад («в прошлое») и вперед («в будущее»). Добавить кнопкам реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
- 2 Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
- 3 Перетащить свойства **Ширина шкалы времени** и **Время отрисовки** в окно аргументов каждой кнопки.

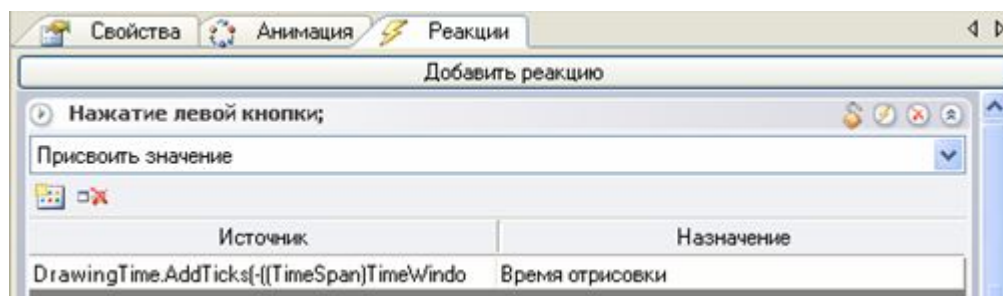
**Ширина шкалы времени** – интервал времени (структура TimeSpan), на котором отображается тренд.

**Время отрисовки** – момент времени, начиная с которого заполняется табличный тренд по направлению «в прошлое».

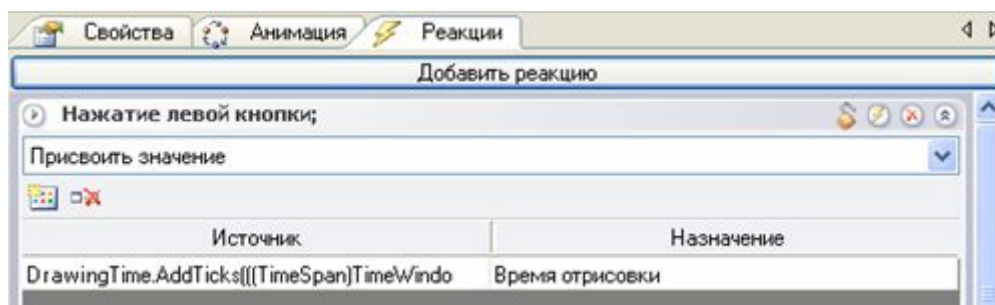
Аргументы - Объемный прямоугольник1			
	Имя в скрипте	Имя	Тип
✓	DrawingTime	Время отрисовки	DateTime
✓	TimeWindowWidth	Ширина шкалы времени	Object

- 4 Выбрать для каждой кнопки в поле **Назначение** атрибут **Время отрисовки**, а в поле **Источник** написать выражение:

- для кнопки просмотра в прошлое на величину интервала табличного тренда `DrawingTime.AddTicks(-((TimeSpan)TimeWindowWidth).Ticks)`



- для кнопки просмотра в будущее на величину интервала табличного тренда `DrawingTime.AddTicks(((TimeSpan)TimeWindowWidth).Ticks)`

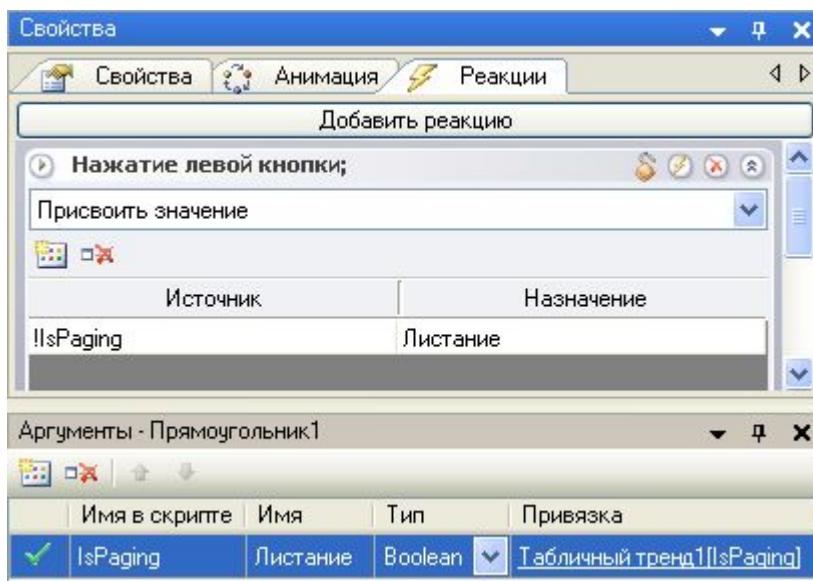


- 5 Готово. Если при выполнении приложения щелкнуть по объекту-кнопка просмотра «в прошлое», то будут отображаться данные на интервале времени, "сдвинутым" на величину интервала табличного тренда назад. При щелчке по объекту-кнопка просмотра «в будущее» будут отображаться данные на интервале времени, "сдвинутым" на величину интервала табличного тренда вперед.

### Управление листанием

Для переключения табличного тренда в режим **листания** и обратно следует:

- 1 Создать объект управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
- 2 Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
- 3 Перетащить свойства **Листание** в окно аргументов
- 4 Выбрать в поле **Назначение** атрибут **Листание**, а в поле **Источник** написать выражение: **!IsPaging** (восклицательный знак перед названием свойства), которое при щелчке по кнопке переключает логическое значение свойства **IsPaging** из значения **Истина** в значение **Ложь** (и обратно).
- 5 Готово. Если при выполнении приложения щелкнуть по кнопке, то будет осуществляться переключение в режим **Листание** (и обратно).



### 6.5.3 Световое перо

**Световое перо** предназначено для подсветки строки табличного тренда, на которой пользователь щелкнул мышью.

#### ВНИМАНИЕ!

Световое перо работает только в режиме **Листание**.

## Свойства

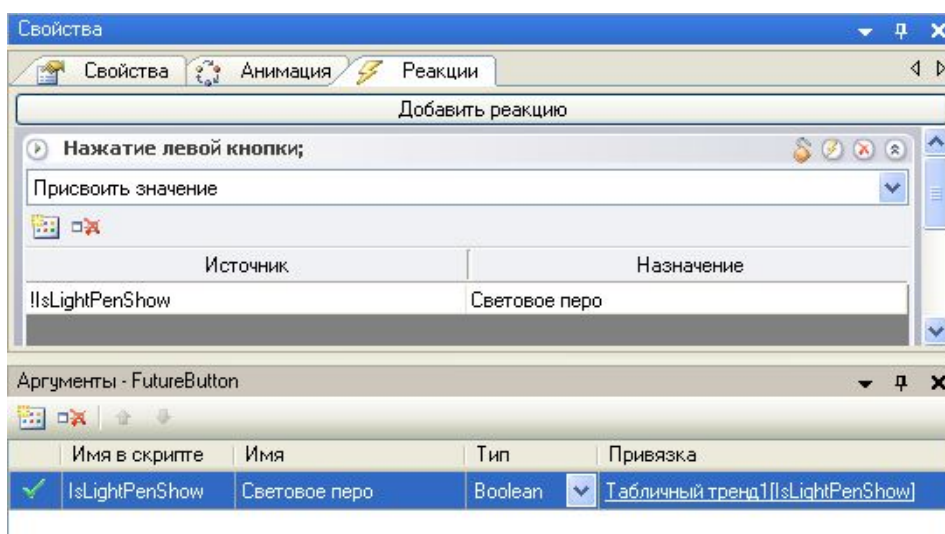
- **Световое перо** – значение **Да** задает режим светового пера табличного тренда, **Нет** – в противном случае
- **Цвет текста** – цвет текста ячеек строки светового пера
- **Цвет фона** – цвет фона ячеек строки светового пера
- **Индекс строки** – индекс строки, на которой находится световое перо. Индексация начинается с 0 с верха табличного тренда. Если ни одна строка не подсвечена, индекс строки равен -1.

☐ <b>Световое перо</b>	
Световое перо	Нет
Цвет фона	<span style="background-color: yellow;"> </span> <b>Желтый</b>
Цвет текста	<span style="background-color: black; color: black;"> </span> <b>Черный</b>
Индекс строки	-1

## Управление световым пером

Для переключения табличного тренда в режим **Световое перо** (и обратно) следует:

- 1 Создать объект управления – кнопку (например, прямоугольник) и добавить ему реакцию **Присвоить значение** на событие **Нажатие левой кнопки мыши**
- 2 Вызвать окно **Дерево мнемосхемы** и окно **Аргументы** объекта-кнопки
- 3 Перетащить свойства **Световое перо** в окно аргументов
- 4 Выбрать в поле **Назначение** атрибут **Световое перо**, а в поле **Источник** написать выражение: **!IsLightPenShow** (восклицательный знак перед названием свойства), которое при щелчке по кнопке переключает логическое значение свойства **IsLightPenShow** из значения **Истина** в значение **Ложь** (и обратно).





- 5 Готово. Если при выполнении приложения щелкнуть по кнопке, то будет осуществляться переключение в режим **Световое перо** (и обратно).

## 6.5.4 Свойства колонки Перо

Колонка **Перо** используется для отображения значений пера элемента табличный тренд.

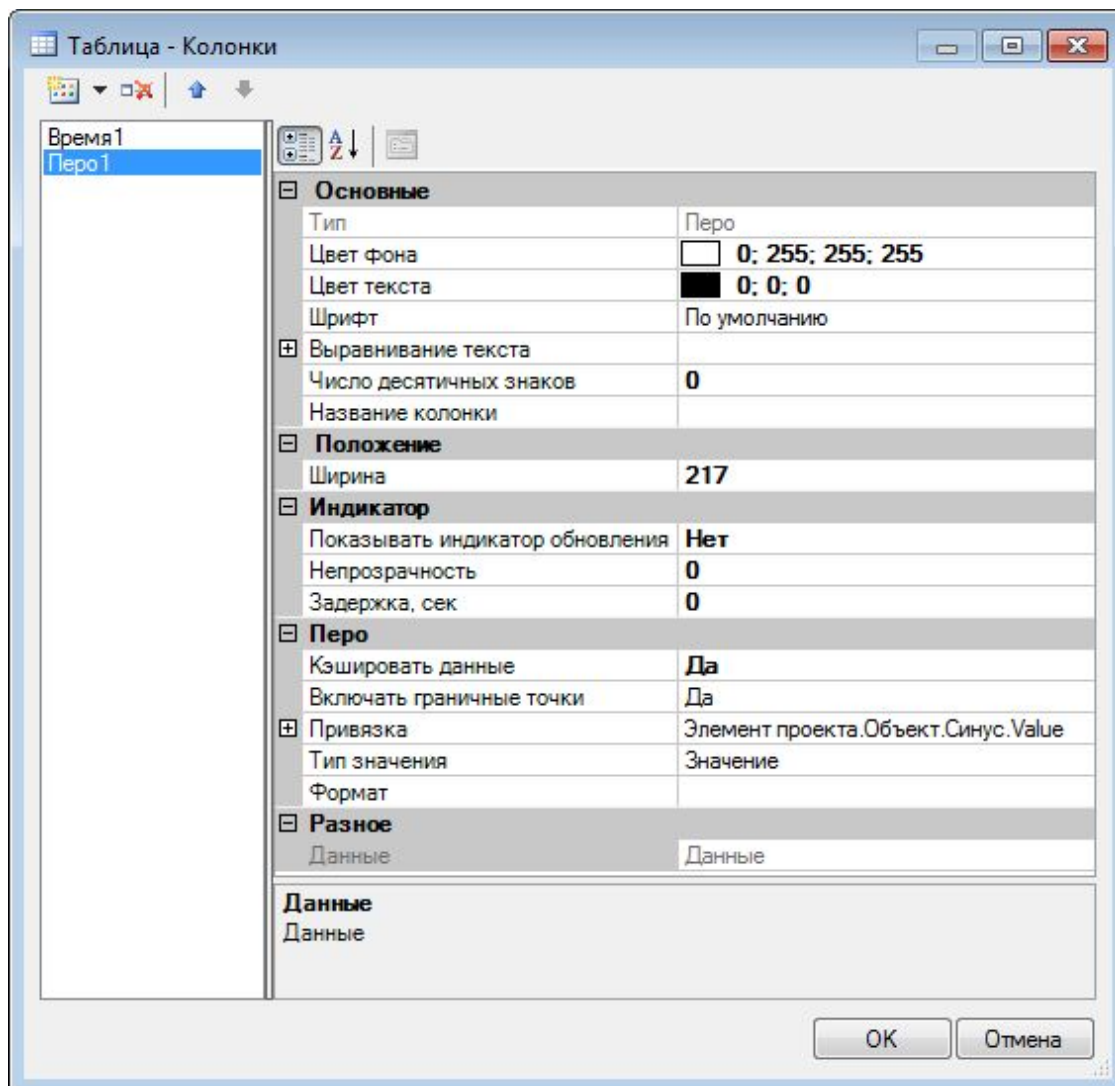
Для привязки и задания свойств пера следует:

- 1 Выделить свойство **Колонки** в окне свойств и нажать на кнопку  в строке

☐ <b>Форматирование</b>	
Заголовки	Количество: 1
<b>Колонки</b>	Количество: 3 
Заголовки колонок	Количество: 0
Подпись	Количество: 0



## 2 В появившемся окне **Таблица-Колонки** задать значения свойств



- **Шрифт, Цвет фона, Цвет текста, Выравнивание текста** – характеристики текста в колонке
- **Показывать индикатор обновления** - при выставлении при загрузке большого количества данных будет отображаться индикатор загрузки.
- **Непрозрачность** - задает уровень непрозрачности индикатора обновления в %.
- **Задержка, сек** - задает время задержки до момента отображения индикатора обновления.
- **Кэшировать данные** – определяет, кэшировать ли данные в тренде. Если значение **Да**, то для формирования тренда запрашиваются только отсутствующие данные, остальные данные хранятся в тренде. Если значение **Нет**, данные каждый раз запрашиваются целиком за тот промежуток времени, который отображается в примитиве **Табличный тренд**
- **Включать граничные точки** - определяет, необходимость отображения граничных точек при запросе данных
- **Привязка** – ссылка на трендируемый атрибут. Настраивается аналогично графическому элементу **Тренд** (Редактирование привязки пера)
- **Тип значения** – задает тип выводимого в колонке значения атрибута (**Время**, **Значение** или **Качество**)

- **Формат** – строка для форматирования значения.

Остальные свойства **колонок Перо** наследует от объекта **Простой столбец**

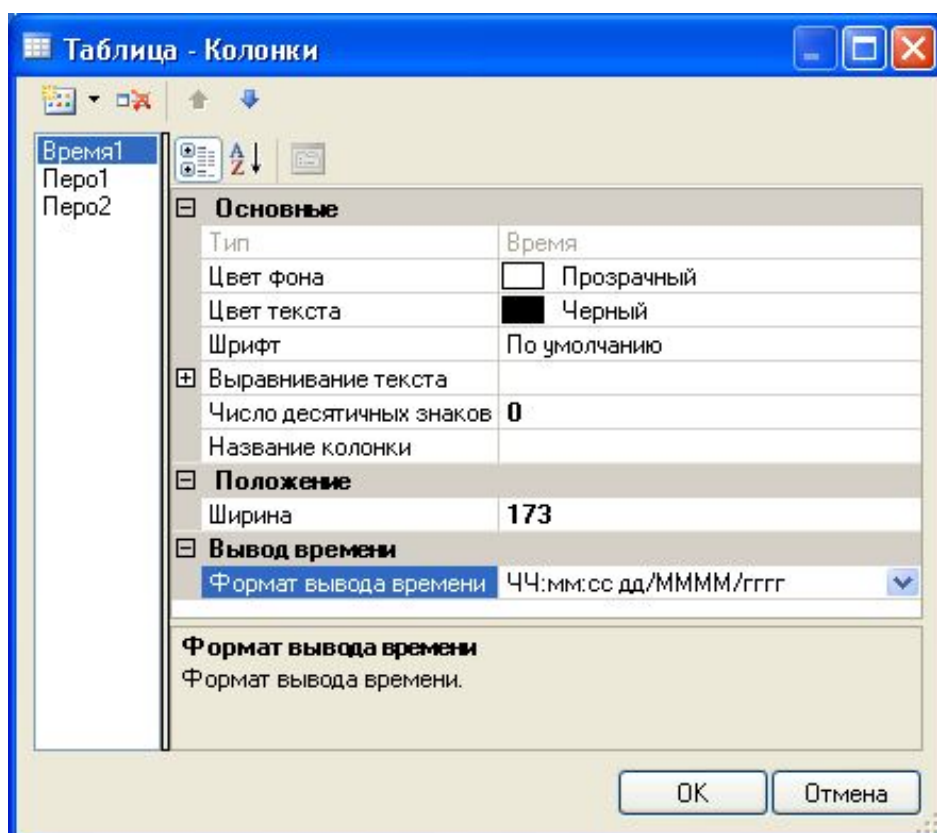
### 6.5.5 Свойства колонки **Время**

Колонка **Время** используется для отображения данных (моментов времени) пера тренда.

#### Свойства

- **Шрифт** – шрифт текста в колонке
- **Цвет фона** – цвет фона колонки
- **Цвет текста** – цвет текста в колонке
- **Выравнивание текста** – выравнивание текста во всех ячейках колонки
- **Формат вывода времени** – формат вывода времени в колонке:
  - **гггг** или **гг** – год в длинном или коротком формате
  - **ММ** – месяц в виде цифры
  - **ММММ** – месяц в виде названия
  - **дд** – день
  - **ЧЧ** – час
  - **мм** – минуты
  - **сс** – секунды.

Остальные свойства колонки **Время** наследует от объекта **Простой столбец**.

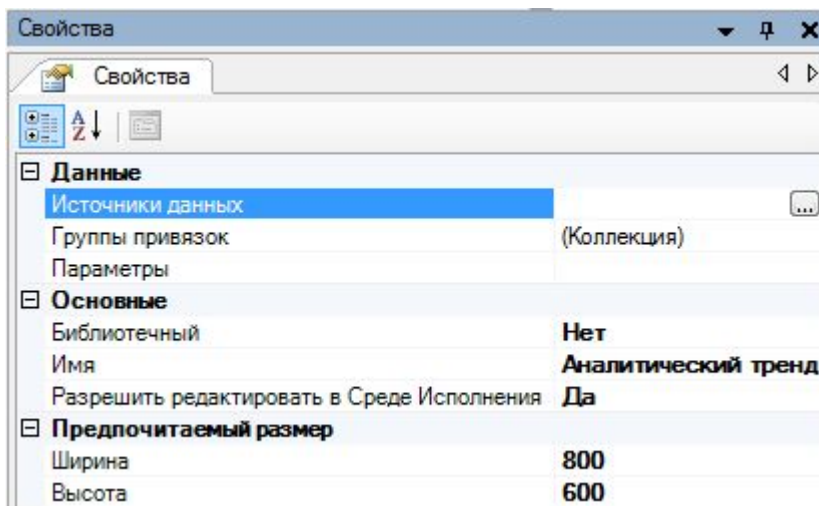




- Синхронизировать перья протокола событий с пером тренда, если у перьев один и тот же источник.

### Свойства вида Аналитический тренд

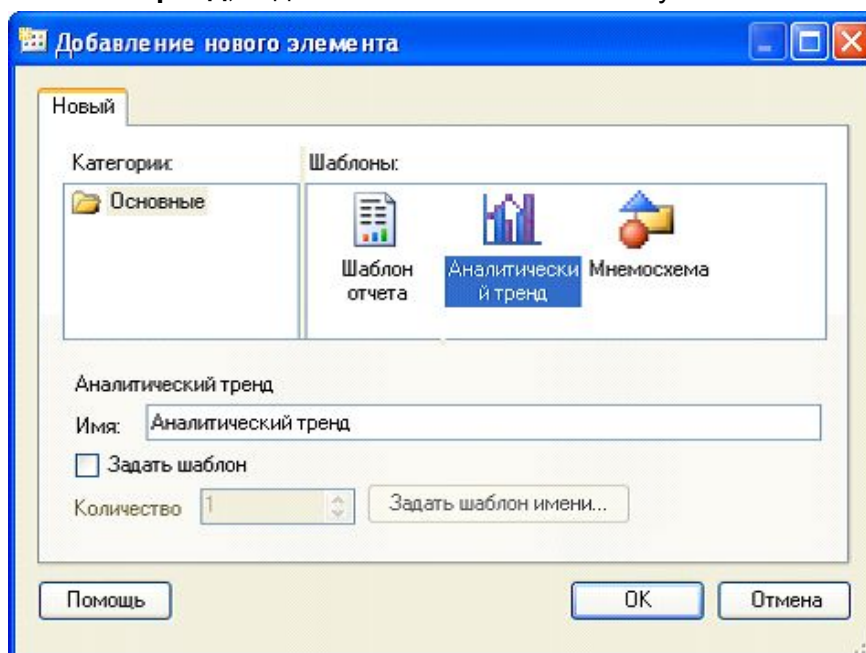
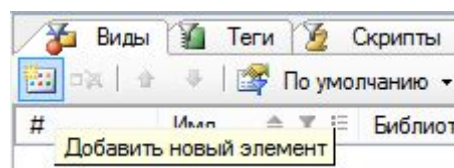
В окне **Свойства** пользователь может задать **Имя** тренда, **Предпочитаемый размер** вида и значение свойства **Разрешить редактировать в Среде исполнения** (Да – позволяет открыть окно настроек в среде исполнения для смены группы настроек), а также задать **Источники данных**, **Параметры** и **Группы привязок**.



### 6.6.1 Создание аналитического тренда

Для создания аналитического тренда следует:

1. Выбрать в контекстном меню объекта **Создать/ Вид** или нажать на кнопку **Добавить новый элемент** на вкладке **Виды** объекта
2. В открывшемся окне **Добавление нового элемента** выбрать шаблон **Аналитический тренд**, задать **Имя** и нажать на кнопку **ОК**.



**Аналитический тренд** будет добавлен в список видов объекта.

Для создания нескольких трендов воспользуйтесь флажком **Задать шаблон**. Введите **Количество** трендов и задайте шаблон имени создаваемых видов. Задать шаблон имени можно нажав на кнопку **Задать шаблон имени**, как описано в разделе Задание шаблона имени.

**3 Для работы с аналитическим трендом необходимо добавить перья тренда и/или перья протокола событий.**

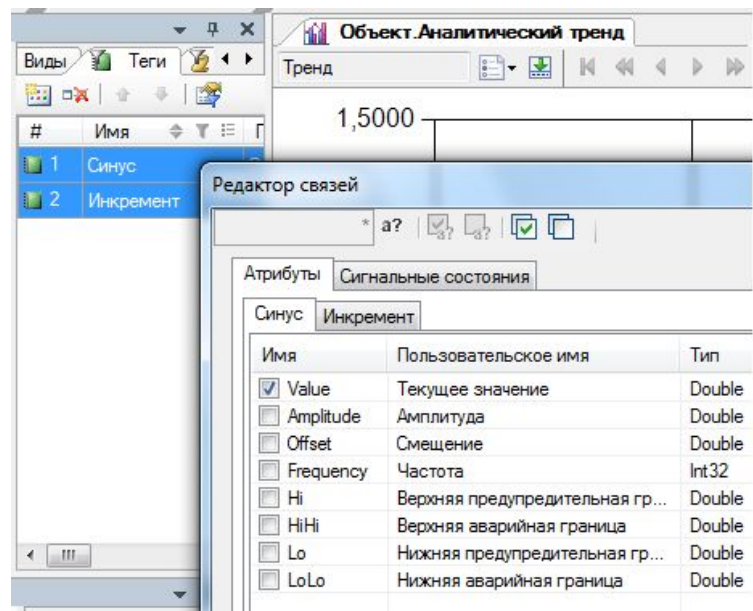
Для добавления перьев тренда можно использовать один из следующих способов.

**Способ 1:**

- В контекстном меню области аналитического тренда выбрать пункт **Свойства** или на панели инструментов нажать на кнопку
- В появившемся окне **Настройки тренда** нажать на кнопку – **добавить перо тренда** и выбрать необходимый тег в открывшемся окне **Редактор привязки тренда** (аналогично добавлению перьев графического элемента Тренд)

**Способ 2:**

- Используя операцию **Drag&Drop**, перетащить в область аналитического тренда выбранные теги объекта
- В появившемся окне **Редактор связей** выбрать атрибуты и свойства сигнальных состояний тегов, которые будут добавлены в качестве перьев **Аналитического тренда**.



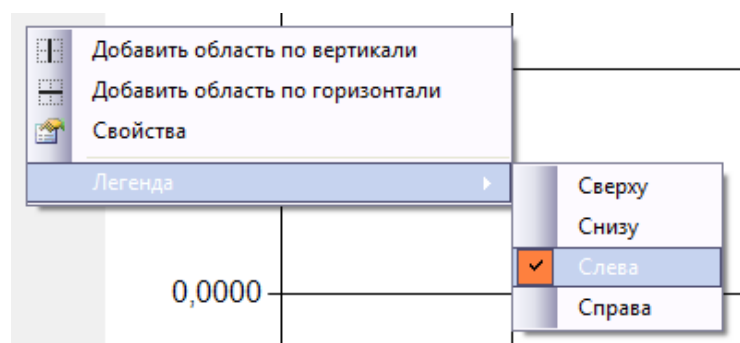
## 6.6.2 Область тренда

**Область тренда** – это рабочее пространство, в котором отображаются элементы Аналитического тренда.

Состав области тренда:

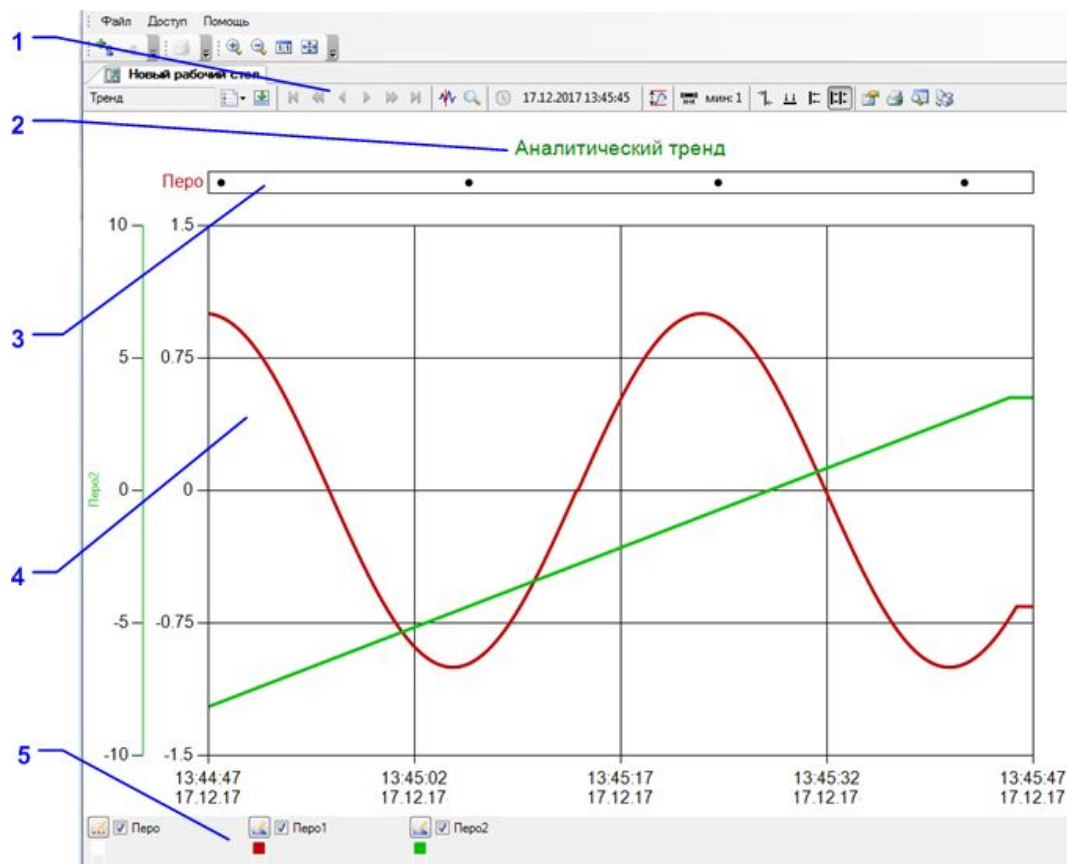
- 1 Панель инструментов** – инструменты для управления трендом
- 2 Заголовок тренда** – константа или текущее значение
- 3 Область перьев протокола событий** – значения перьев протокола событий
- 4 Область перьев тренда** – значения перьев тренда
- 5 Список перьев** – перья, значения которых отображаются в области тренда.

Панель со списком перьев, при необходимости, можно пристыковать к верхнему и боковым краям тренда. Для



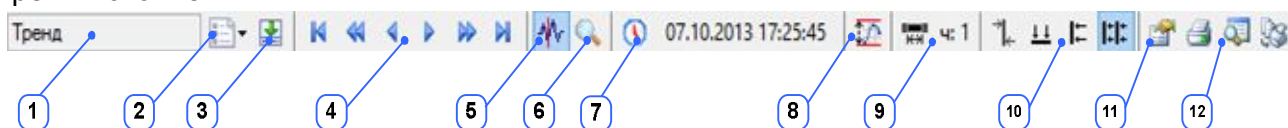


этого в контекстном меню тренда необходимо выбрать пункт **Легенда** и, далее, в раскрывающемся меню указать сторону тренда.



### 6.6.2.1 Панель инструментов

**Панель инструментов** служит для управления отображением Аналитического тренда во время исполнения.



Элементы панели:

- 1 **Имя области тренда**
- 2 **Выбор активной группы настроек**
- 3 **Скрытие/показ панели перьев** – если кнопка нажата, то панель перьев будет скрыта
- 4 **Переход по истории (только в режиме светового пера)**

**Кнопка Описание**



переход к самой первой точке из всех перьев тренда



переход на несколько временных окон назад. Количество временных окон задается в свойстве области




переход на одно временное окно назад




переход на одно временное окно вперед

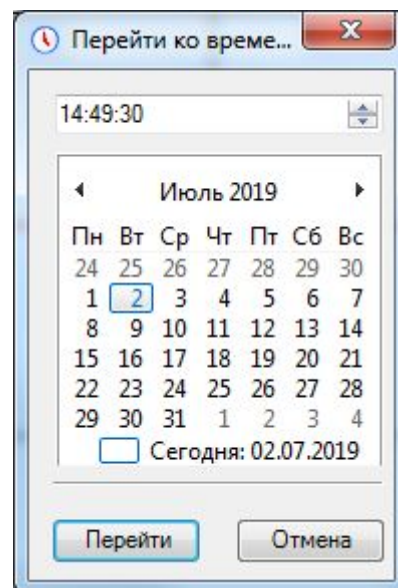


переход на несколько временных окон вперед. Количество временных окон задается в свойстве области

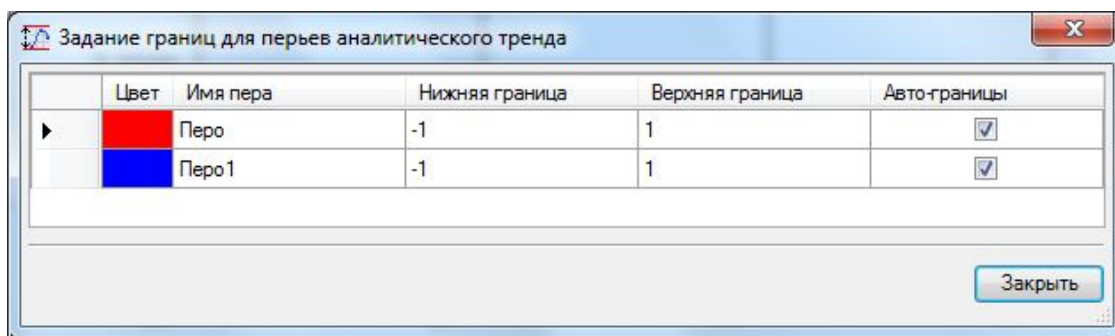
 переход ко времени "Сейчас"

- 5 Включение/выключение светового пера
- 6 Включение/выключение масштабирования
- 7 Вызов окна перехода ко времени (**только в режиме светового пера**) – отображается кнопка

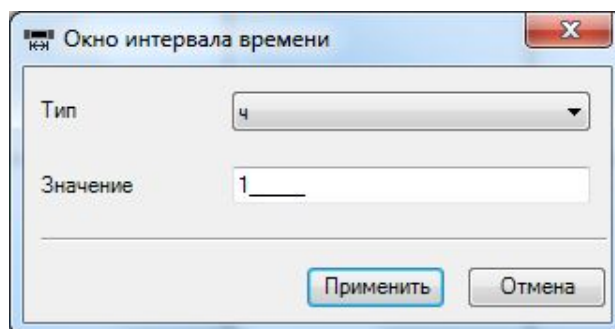
 для вызова окна задания времени перехода и значение времени. При нажатии на кнопку появляется окно **Перейти ко времени**, в котором можно задать требуемое время



- 8 Вызов окна задания границ



- 9 Область задания временного окна – состоит из кнопки вызова диалогового окна для задания временного окна и текстового поля, в котором отображается текущее значение временного окна



- 10 Задание типа синхронизации.

Кнопка Описание



**синхронизировать все** – перья синхронизируются по времени: по наиболее близкому к "сейчас" перу, а также по максимальной и минимальной нижней границе



**синхронизировать по времени** – перья синхронизируются по времени: по наиболее близкому к "сейчас" перу






**синхронизировать границы** – перья синхронизируются по границам, заданным в настройке области



**рассинхронизировать** – перья рассинхронизируются

- 11 Вызов окна свойств Аналитического тренда

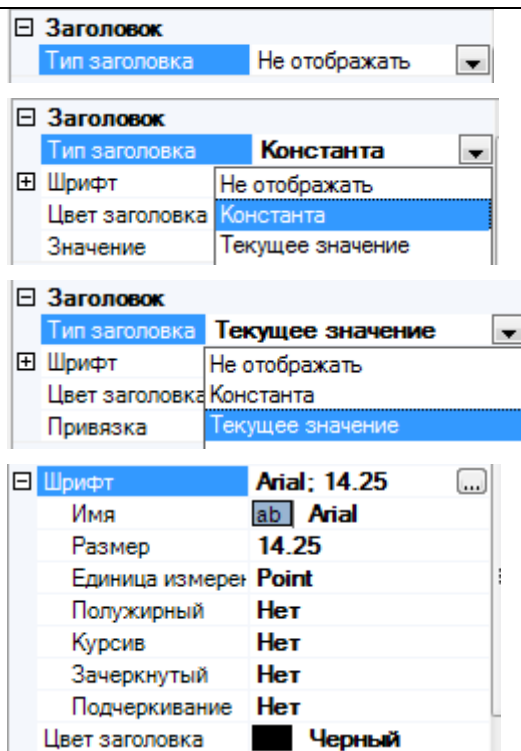
## 12 Печать

Кнопка	Описание
	печать
	предварительный просмотр
	настройка страницы

### 6.6.2.2 Заголовок

#### Свойства

- **Тип заголовка:**
  - **Не отображать** – заголовок отсутствует
  - **Константа** – в заголовке отображается строка, заданная в поле **Значение**
  - **Текущее значение** – в заголовке отображается строковое значение, формируемое на основе ссылки в поле **Привязка**
- **Шрифт и Цвет заголовка** – шрифт и цвет символов заголовка. Выбираются в открывающихся диалоговых окнах.



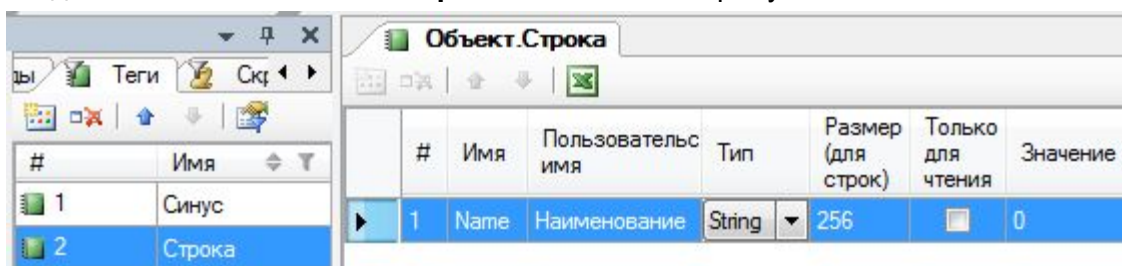
#### Пример формирования заголовка

##### Задача

Требуется отобразить в заголовке тренда текст, за которым следует изменяющееся значение некоторого тега (в примере тег **Синус**).

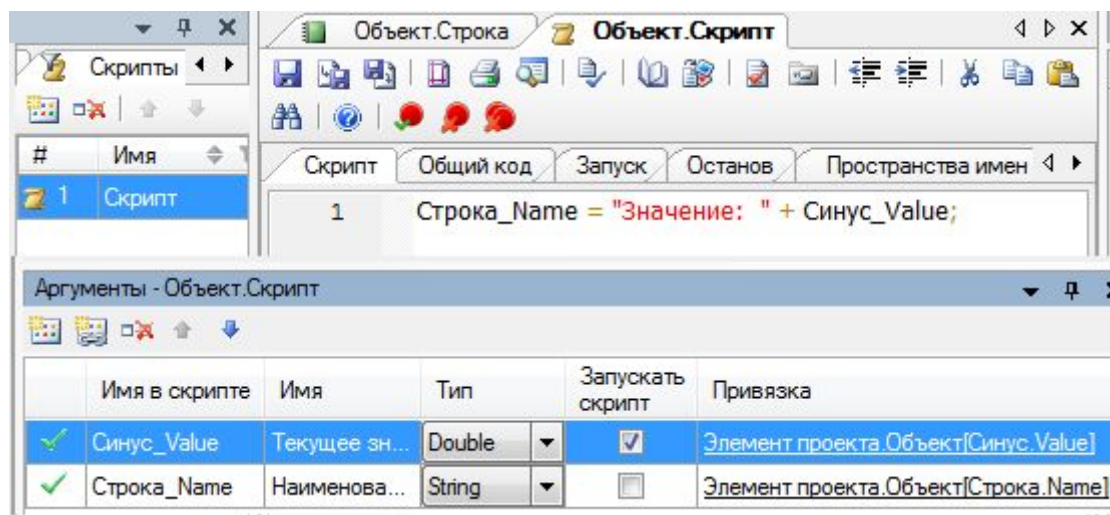
##### Решение

- 1 В проекте в объект **Объект** добавим два тега: **Синус** (стандартный тег **DataRate**) и созданный в библиотеке тег **Строка** с текстовым атрибутом **Name**

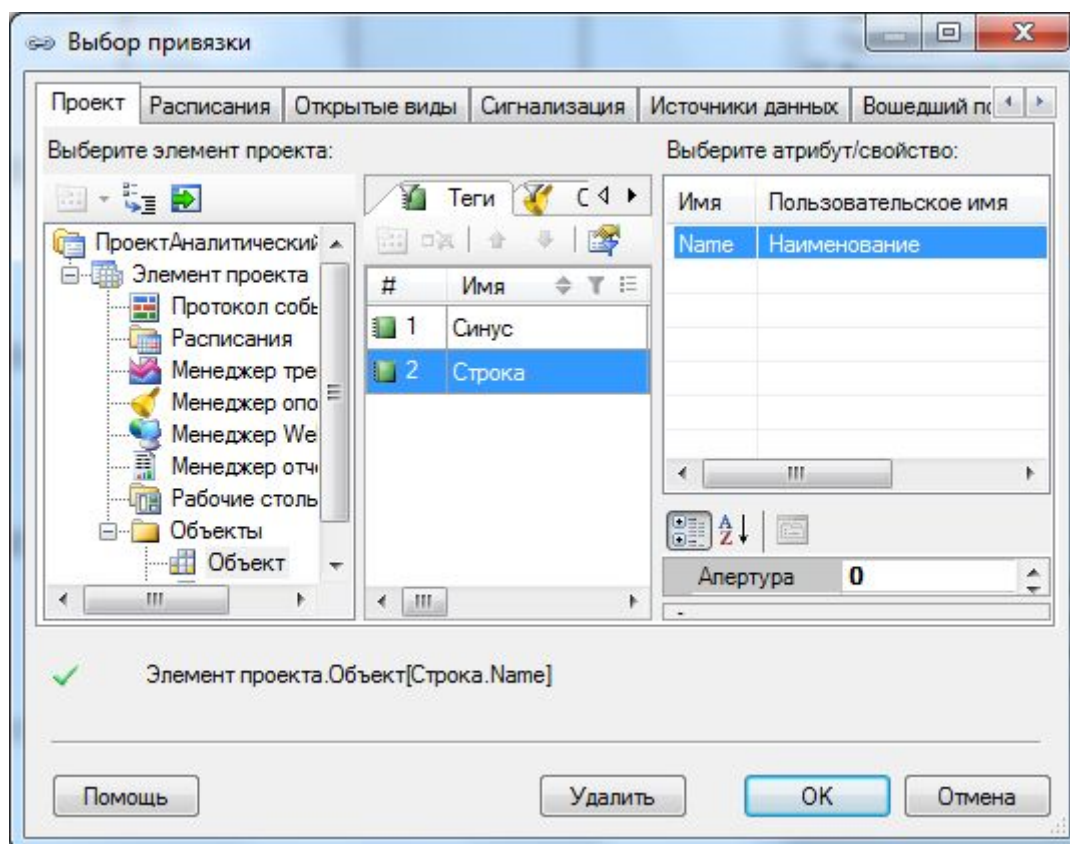


#	Имя	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение
1	Name	Наименование	String	256	<input type="checkbox"/>	0

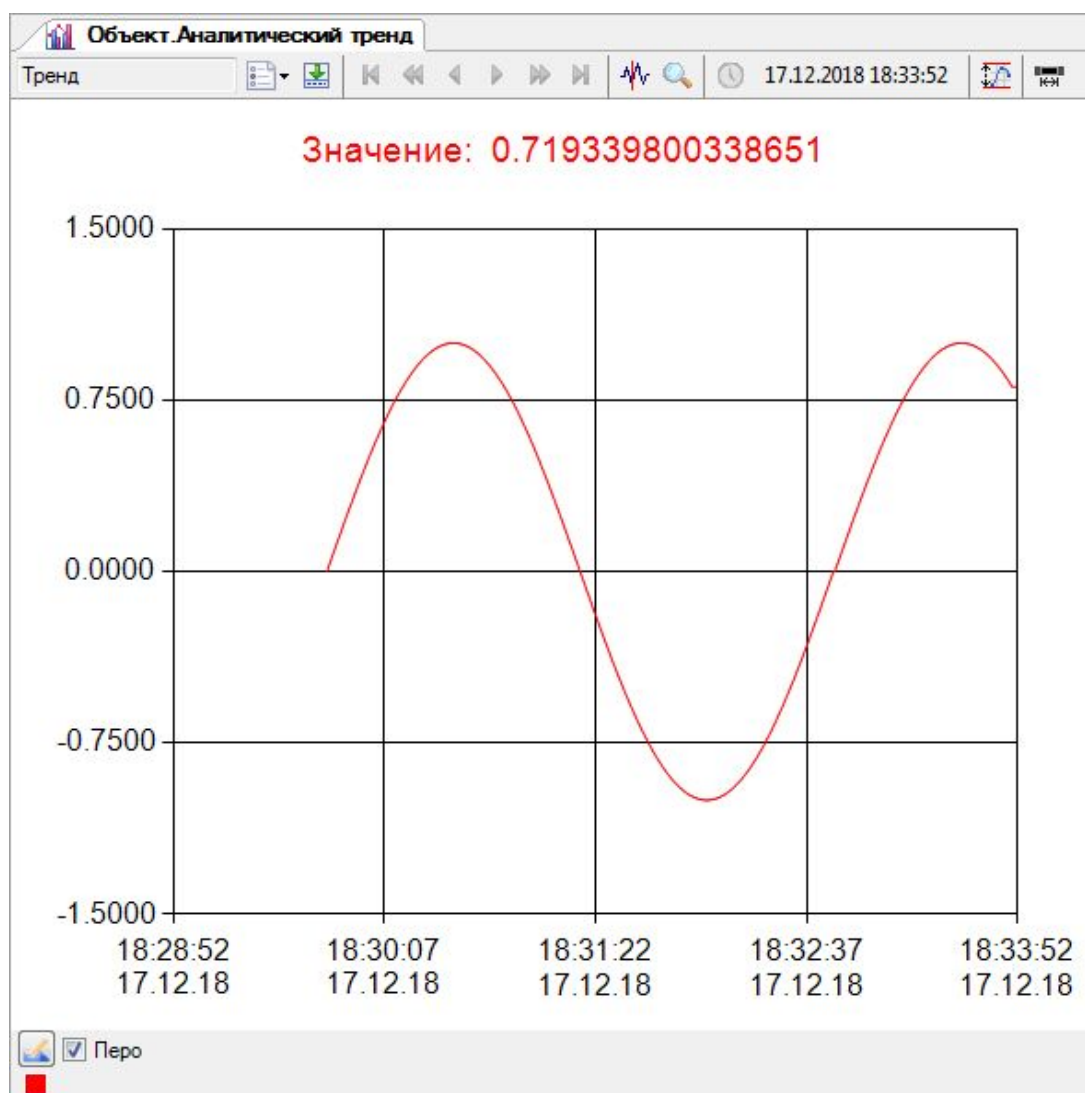
- 2 Для этого же объекта создадим скрипт **Скрипт**, который формирует последовательность символов из строки **Значение:**, за которой следует значение атрибута **Value** тега **Синус** (смотрите текст и аргументы скрипта в скриншоте ниже).



- 3 В виде **Аналитический тренд** объекта **Объект** назначим тип заголовка тренда – **Текущее значение** и в свойстве **Привязка** выберем атрибут **Name** тега **Строка**



- 4 Запускаем проект. В заголовке **Аналитического тренда** отображается требуемая строка



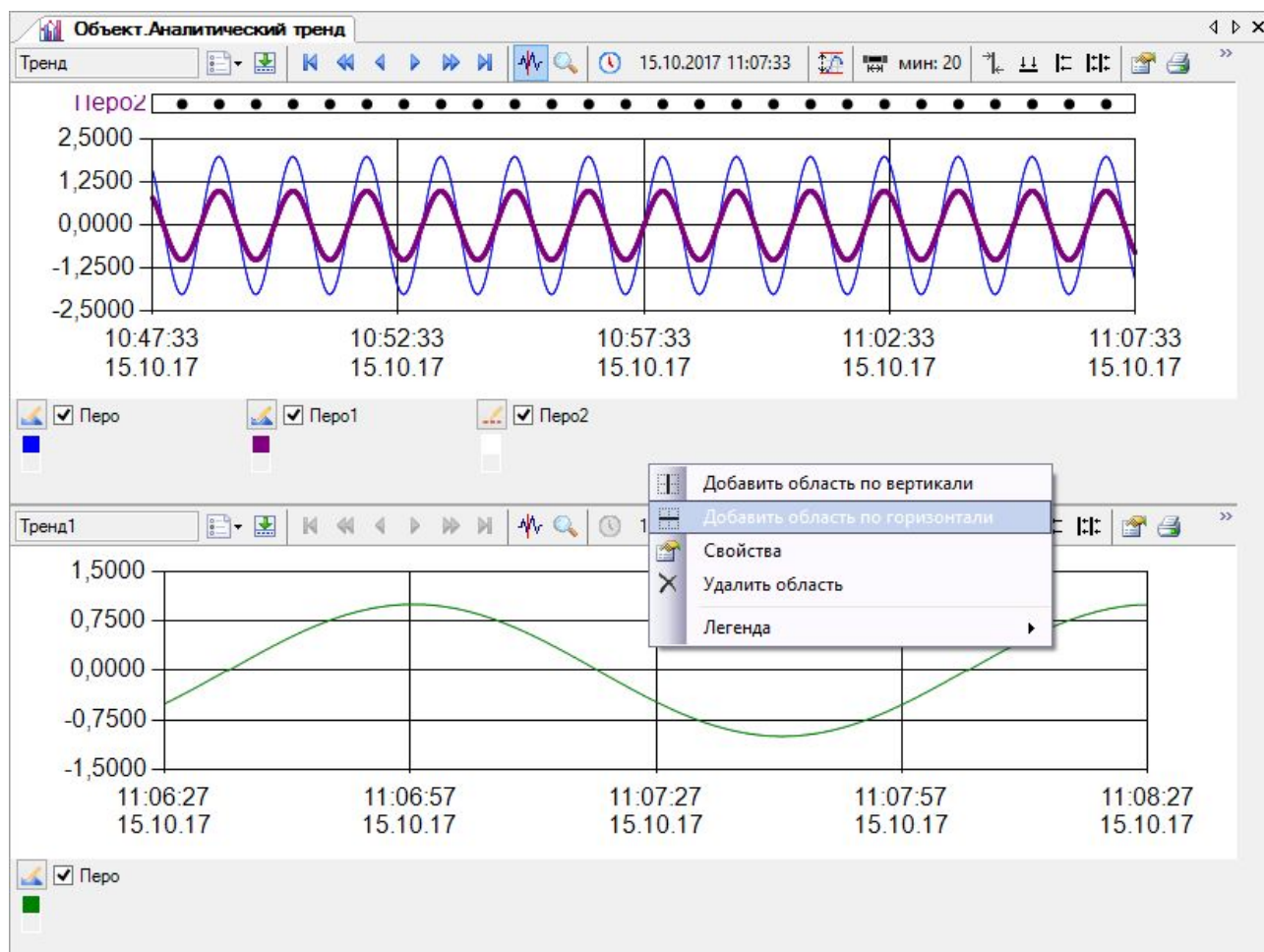
### 6.6.2.3 Область графиков

Количество областей и их положение регулируется при настройке аналитического тренда (задается количество областей по вертикали и горизонтали).

При создании/удалении области существующие области пропорционально масштабируются.

Создание и удаление областей осуществляется при помощи контекстного меню существующей области тренда.





#### 6.6.2.4 Окно настройки аналитического тренда

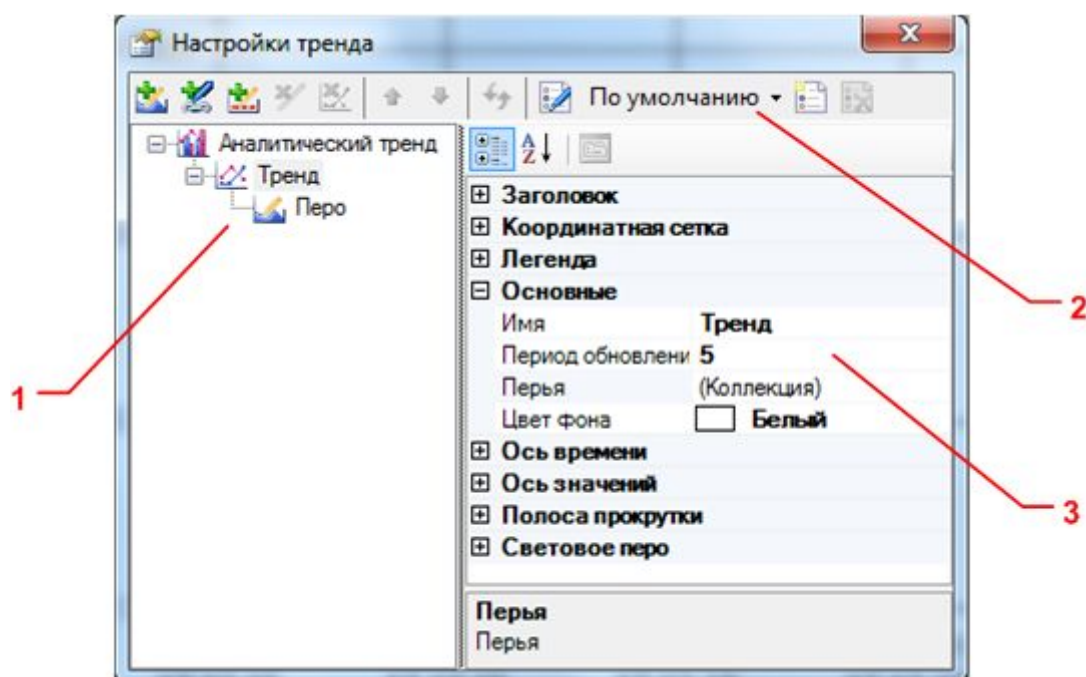
**Окно настройки аналитического тренда** служит для настройки параметров тренда и его перьев, а также для управления группами настроек.

**Группа настроек** – это совокупность настроек всех параметров тренда. Группы настроек хранятся вместе с аналитическим трендом. В любой момент времени пользователь может загрузить одну из существующих групп настроек.












Группа настроек отображается в окне **Настройки тренда**.

Окно настройки содержит три области:

- 1 Дерево группы настройки Аналитического тренда** – отображается активная в данный момент группа настроек



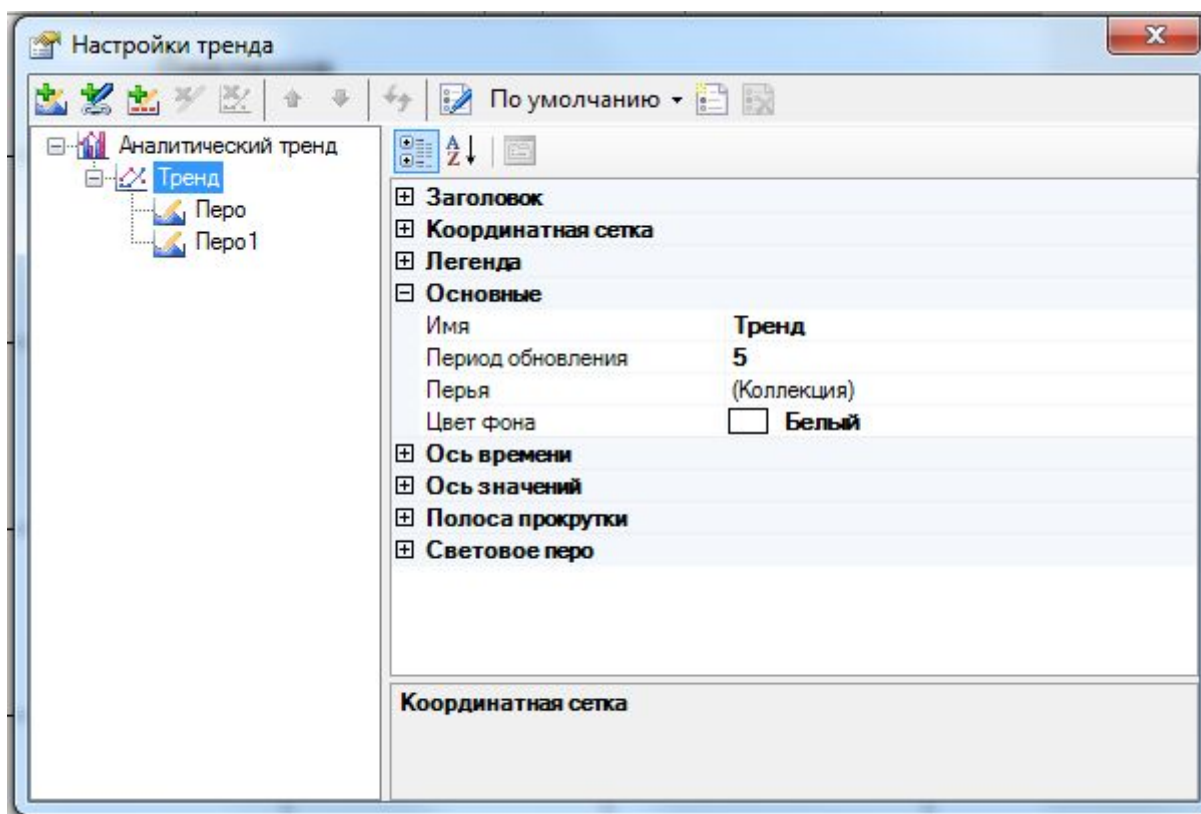
### 2 Панель инструментов – содержит следующие кнопки:

-  – создание нового пера тренда
-  – создание перьев на основе источников привязок
-  – создание нового пера протокола событий
-  – удаление пера тренда или протокола событий
-  – удаление области аналитического тренда
-  – перемещение на одну позицию вверх выделенного пера
-  – перемещение на одну позицию вниз выделенного пера
-  – сброс настроек к исходным. Доступно только в режиме исполнения
-  – переименование группы настроек
-  – создание новой группы настроек на основе текущей
-  – удаление группы настроек

### 3 Свойства объекта, выбранного в дереве.

### 6.6.3 Настройки тренда

Для вызова настроек тренда достаточно щелкнуть левой клавишей мыши по наименованию тренда в дереве группы настроек.



#### 6.6.3.1 Координатная сетка

##### Свойства

- **Толщина рамки, Цвет рамки** – толщина и цвет рамки вокруг области графика тренда
- **Цвет заливки** – цвет заливки области графика тренда
- **Толщина линий сетки, Стилль линий сетки, Цвет линий сетки** – характеристики линий сетки

Координатная сетка		
Толщина рамки	1	
Цвет рамки		Черный
Цвет заливки		Белый
Толщина линий сетки	1	
Стилль линий сетки		Сплошная
Цвет линий сетки		Черный

#### ВНИМАНИЕ!!!

Свойства координатной сетки применяются к области графика тренда и к области перьев протокола событий.

## 6.6.3.2 Легенда

**Легенда** – это панель, содержащая описание списка перьев.



Информация, отображаемая в легенде аналитического тренда:

- 1 Кнопка вызова окна настроек пера
- 2 Переключатель состояния пера (отображать/ не отображать перо)
- 3 Текст или имя пера, если значение параметра **Отображать** свойства пера **Текст** установлено в **Нет**
- 4 Единицы измерения (если значение параметра **Тип значения единиц измерения** свойства пера **Текст** установлено в **Константа** или **Привязка**)
- 5 Цвет пера
- 6 Цвет заливки
- 7 Значение пера и соответствующий ему момент времени (**только в режиме светового пера**)

Легенду можно расположить сверху, снизу или по бокам тренда.

### Свойства легенды

<ul style="list-style-type: none"> <li>• <b>Ширина элемента легенды</b> – ширина области легенды для графических символов и отображаемого текста, например,</li> </ul>	
<ul style="list-style-type: none"> <li>• <b>Расположение легенды</b> – место расположения легенды относительно отображения тренда. По умолчанию – <b>Снизу</b></li> </ul>	

## 6.6.3.3 Основные

### Свойства

<ul style="list-style-type: none"> <li>• <b>Имя</b> – имя области графика тренда</li> <li>• <b>Период обновления</b> – Период обновления данных тренда</li> <li>• <b>Перья</b> – коллекция перьев области аналитического тренда. Для настройки перьев тренда и линий сетки выделить строку свойства и нажать на кнопку в строке</li> <li>• <b>Цвет фона</b> – цвет фона области тренда</li> </ul>	
---	--

## 6.6.3.4 Настройка осей

## Ось времени

## Свойства

- **Окно интервала времени** задает размер интервала (**Значение**) в выбранных единицах (**Тип единицы интервала**)
- **Количество интервалов для быстрого листания** – количество временных окон для быстрого листания
- **Использовать выравнивание** – если используется выравнивание, то время левой границы графика будет всегда выровнено по типу значения временного окна (например, если тип значения временного окна – день, то левая граница будет всегда 0 часов дня)
- **Количество линий сетки, Количество меток оси времени, Цвет меток, Шрифт меток, Ориентация разметки** – определяют отображение оси времени
- **Формат разметки 1, Формат разметки 2** – формат отображения выбранных единиц времени рядом с осью времени

Ось времени	
Окно интервала времени	
Количество интервалов для быс	2
Использовать выравнивание	Нет
Количество линий сетки	3
Количество меток оси времени	3
Цвет меток	Черный
Шрифт меток	Arial; 12
Ориентация разметки	Снизу
Формат разметки 1	ЧЧ:мм:сс
Формат разметки 2	дд/ММ/гг
Ось значений	

## Ось значений

## Свойства

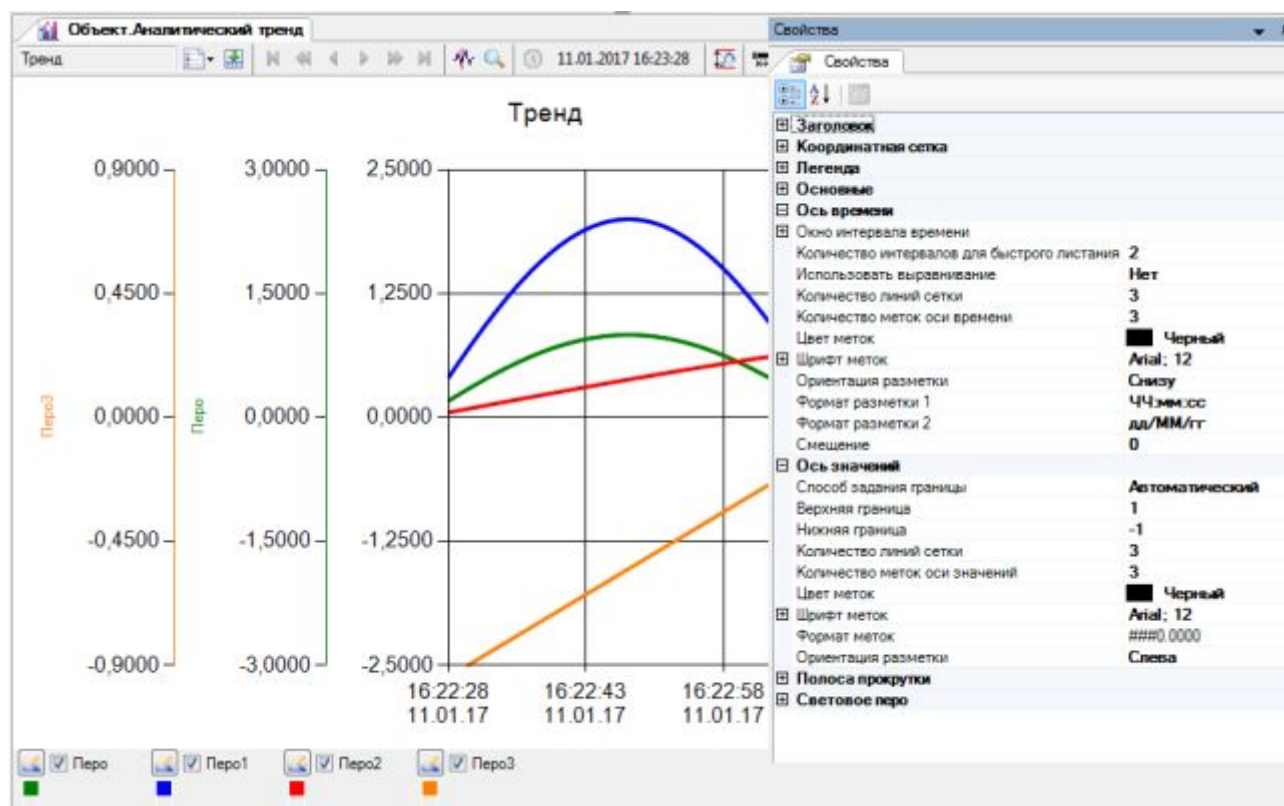
- **Способ задания границы** – если выбран тип **Ручной**, то будут установлены значения границ, которые указаны в поле **Верхняя граница/ Нижняя граница**, иначе (тип **Автоматический**) – границы будут выбраны автоматически. По умолчанию установлено значение – **Автоматический**
- **Верхняя граница, Нижняя граница** – граничные значения
- **Количество линий сетки, Количество меток оси времени, Цвет меток, Шрифт меток, Ориентация разметки** – определяют отображение оси значений

Способ задания границы	Ручной
Верхняя граница	Автоматический
Нижняя граница	Ручной

Ось значений	
Способ задания границы	Автоматический
Верхняя граница	1
Нижняя граница	-1
Количество линий сетки	3
Количество меток оси зн	3
Цвет меток	Черный
Шрифт меток	Arial; 12
Ориентация разметки	Слева



## Пример



### 6.6.3.5 Полоса прокрутки

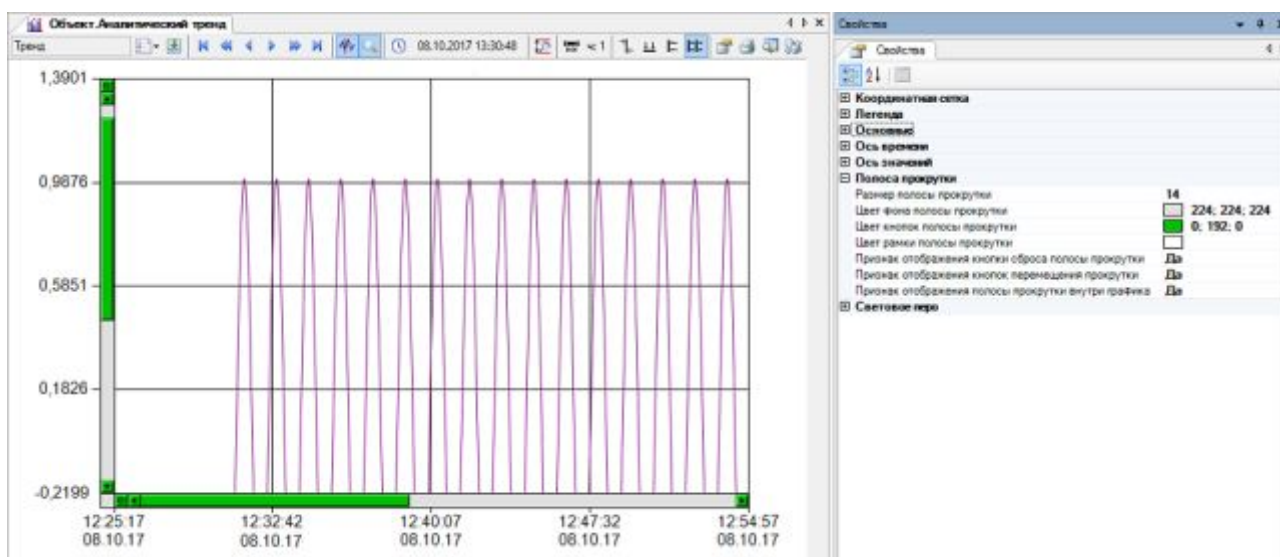
#### Свойства

<ul style="list-style-type: none"> <li>• <b>Размер</b> – размер полос прокрутки (должен быть в диапазоне от 5 до 20, по умолчанию 14)</li> <li>• Другие свойства задают параметры отображения полос прокрутки</li> </ul>	<b>Полоса прокрутки</b>	
	Размер полосы прокрутки	14
	Цвет фона полосы прокрутки	<input type="text"/>
	Цвет кнопок полосы прокрутки	<input type="text"/>
	Цвет рамки полосы прокрутки	<input type="text"/>
	Признак отображения кнопки сброса полосы прокрутки	Да
	Признак отображения кнопок перемещения прокрутки	Да
	Признак отображения полосы прокрутки внутри графика	Да

#### ВНИМАНИЕ!!!

Свойства полосы прокрутки применяются как к оси времени, так и к оси значений. Полосы прокрутки отображаются при масштабировании графика.

## Пример



### 6.6.3.6 Световое перо

**Световое перо** предназначено для получения информации о значениях перьев аналитического тренда в момент времени, который отмечается курсором на графике тренда.




Световое перо отображается как вертикальная линия, перемещаемая по тренду курсором при нажатии левой клавиши мыши.

Значение пера тренда отображается в легенде перьев.

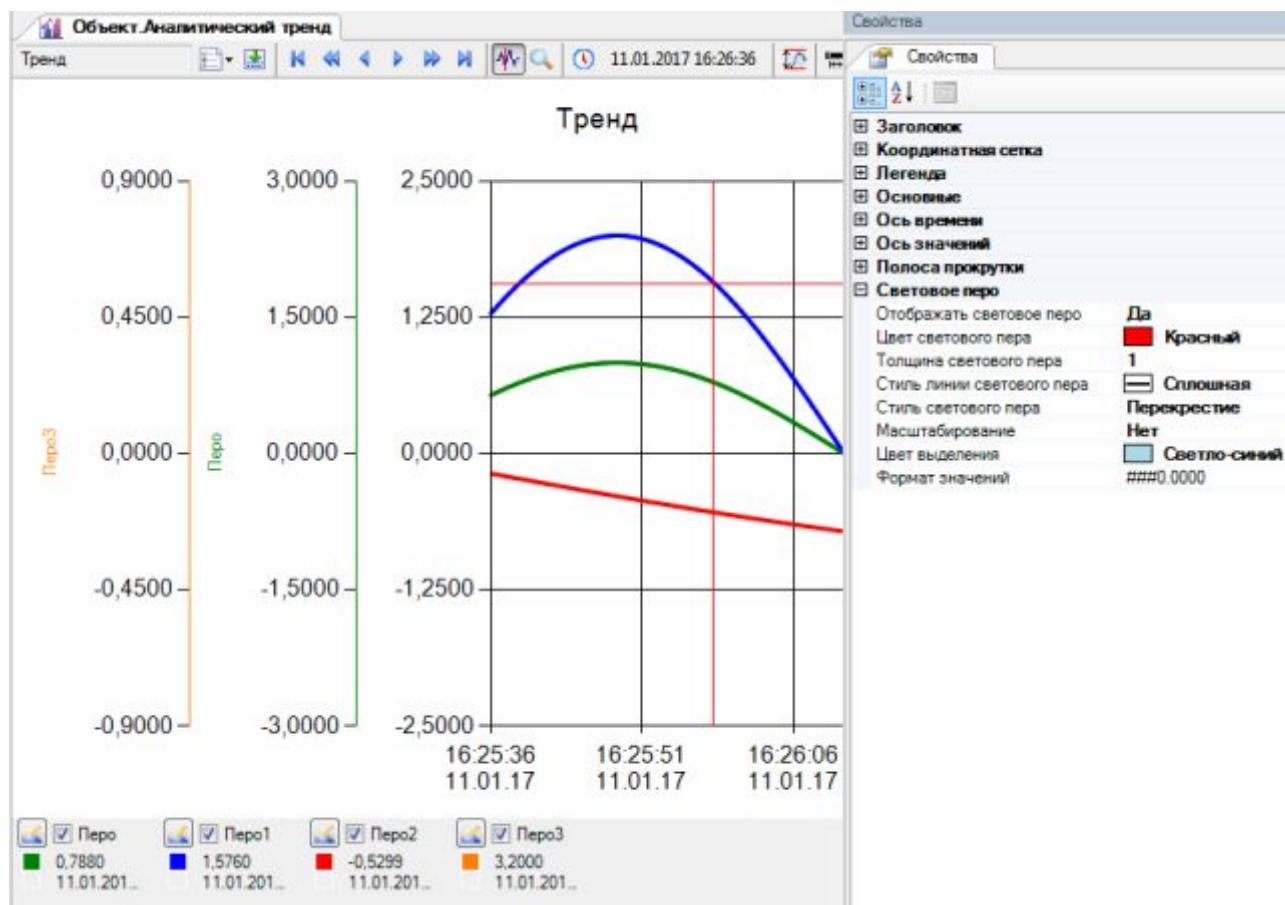
Для перехода в режим светового пера и обратно используйте свойство **Отображать световое перо** или кнопку **Световое перо** на панели инструментов аналитического тренда.

### Свойства

- **Отображать световое перо** – для отображения светового пера на графиках задайте значение **Да**
- Другие свойства задают отображение светового пера
- **Формат значений** – формат значений, выводимых световым пером. **0** – означает, что если на месте символа не будет значения, то в позицию символа будет вставлена цифра 0. **#** – означает, что если на месте символа не будет значения, то в позицию символа ничего вставлено не будет.  
Например: формат - **###0.0000**, исходное число **3,034**, результат – **3,0340**

<b>Световое перо</b>		
Отображать световое перо	Нет	
Цвет светового пера		Красный
Толщина светового пера	1	
Стиль линии светового пера		Сплошная
Стиль светового пера	Перекрестие	
Масштабирование	Нет	
Цвет выделения		Светло-синий
Формат значений	###0.0000	

## Пример



### 6.6.4 Настройки перьев

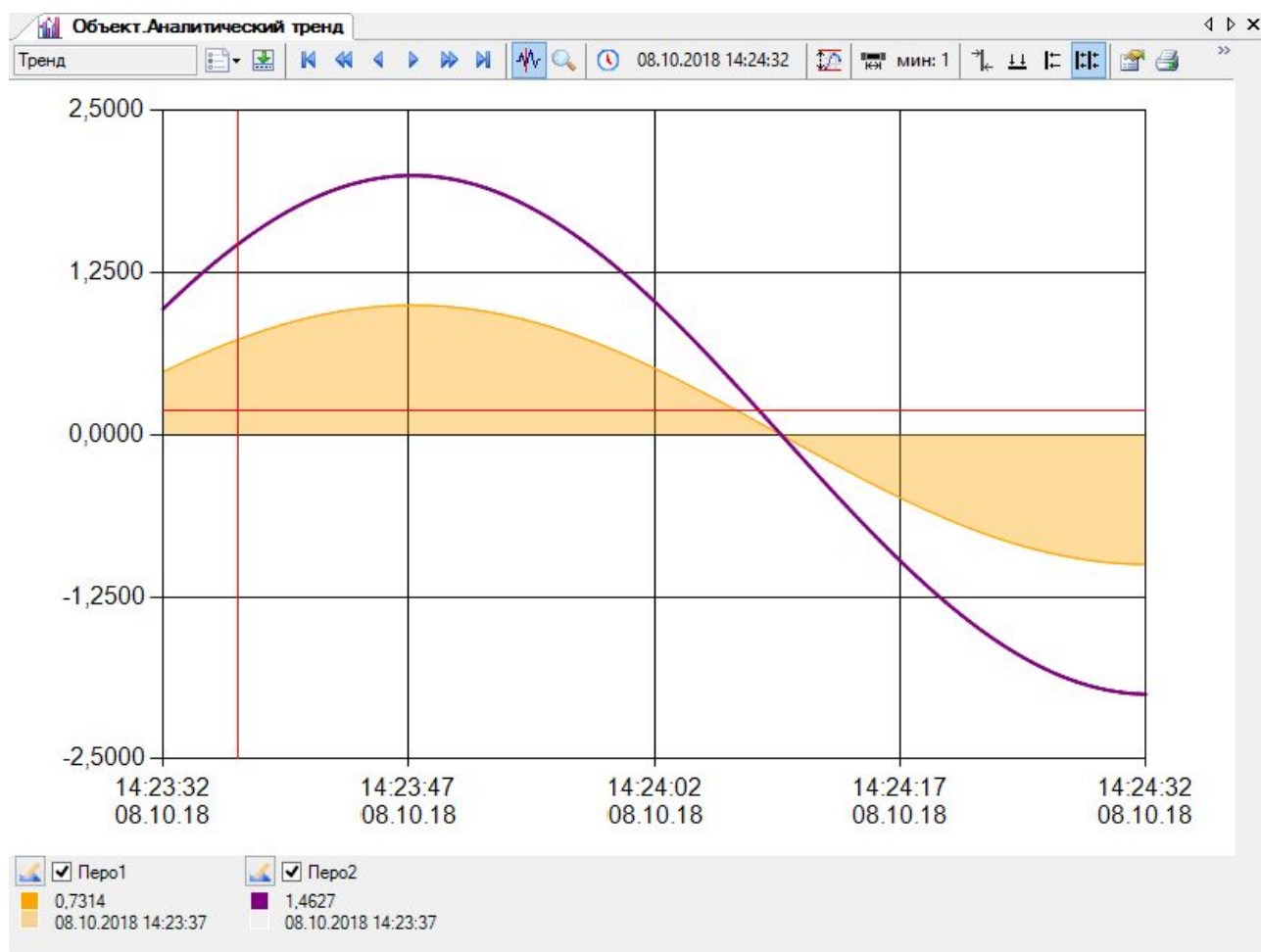
Вид **Аналитический тренд** оперирует двумя типами перьев: перо тренда и перо протокола событий.

#### 6.6.4.1 Перо тренда

Для создания пера тренда выбирается перо или тег **DataRate**.

Создать перо тренда можно с помощью кнопки **Добавить перо тренда** на панели инструментов окна настроек тренда, либо перетаскиванием (**drag&drop**) тега из Редактора проекта (закладка **Теги** объекта) в область аналитического тренда (подробнее смотрите в подразделе 6.6.1 **Создание аналитического тренда**).

Значения пера тренда отображаются в виде графика.



## Свойства

Настройки тренда

Аналитический тренд

Тренд

Перо

Перо1

По умолчанию

**Границы**

**Загрузка пера**

Признак загрузки пера: Константа

Загрузка пера: Да

**Индикатор**

**Основные**

**Ось значений**

**Текст**

Отображать: Да

Тип значения: Привязка

Привязка: ☒ Элемент проекта.Объект[Простой тег.Заголовок]

Тип значения единиц измерения: Привязка

Привязка: ☒ Элемент проекта.Объект[Простой тег.Единицы измерения]

Границы

## Границы

- **Привязка границ** – вызов окна **Привязки границ** по кнопке
- **Показывать границы** – переключатель **Да/ Нет**. По умолчанию – **Нет**
- **Конец шкалы, Верхняя аварийная граница, Верхняя предупредительная граница, Нижняя предупредительная граница, Нижняя аварийная граница, Начало шкалы** – настройка изображения границ:
  - **Стиль линии** – стиль линии отрисовки границы
  - **Толщина линии** – толщина линии границы
  - **Цвет линии** – цвет линии границы

Границы

Привязки границ

Показывать границы

Конец шкалы

Верхняя аварийная граница

Верхняя предупредительная граница

Нижняя предупредительная граница

Нижняя аварийная граница

Начало шкалы

Верхняя аварийная граница

Стиль линии

Толщина линии

Цвет линии

Сплошная

1

Красный

Окно **Настройки границ** обеспечивает задание значений границ типа **Константа** или **Текущее значение** (в этом случае следует настроить привязку к тегу объекта)

Привязки границ

Привязка технологических границ:

Граница	Тип привязки	Значение привязки
Конец шкалы	Не используется	(нет)
ВАГ	Не используется	(нет)
ВПГ	Константа	(нет)
НПГ	Не используется	(нет)
НАГ	Не используется	(нет)
Начало шкалы	Не используется	(нет)

Конец шкалы

Конец шкалы.

Подсказка: выполните двойной щелчок по ячейке "Значение привязки" или нажмите на ней F2 для редактирования.

Помощь

OK

Отмена

## ВНИМАНИЕ!!!

Границы будут отображаться только в режиме видимости одного пера при условии, что установлено свойство **Показывать границы** (значение – **Да**).



- **Признак загрузки пера** – Константа или Привязка
- **Загрузка пера** – переключатель Да/ Нет. В случае Нет информация о перо и значения пера не загружаются и, следовательно, не могут быть визуализированы. По умолчанию – Да
- **Привязка** – привязка к тегу объекта, параметр которого (логическое значение) управляет загрузкой пера

<b>Загрузка пера</b>	
Признак загрузки пера	Константа
Загрузка пера	Да
	Да
	Нет
<b>Загрузка пера</b>	
Признак загрузки пера	Привязка
Привязка	Элемент проекта.1

### Основные свойства

- **Имя пера** – имя пера тренда
- **Стиль пера** – стиль отрисовки графика (тип графика). По умолчанию – Линия
- **Видимость** – видимость пера на графике (Да/ Нет). По умолчанию – Да
- **Кешировать данные** – переключатель Да/ Нет. По умолчанию – Да
- **Достраивать график** – признак достраивания графика прямой линией, если отсутствует последняя точка. Если выставлен стиль пера Сглаженная, то независимо от значения свойства график не достраивается. Значение по умолчанию – Да
- **Множитель** – множитель для масштабирования графика
- **Привязка** – привязка пера к тегу или к перу тренда
- **Режим начала отрисовки** – если выбран тип Автоматическая то время начала отрисовки будет совпадать с левой границей графика
- **Цвет линии, Толщина линии, Стиль линии** – свойства отрисовки линии графика пера тренда
- **Цвет заливки, Показывать заливку** – свойства заливки под пером
- **Настройки отображения точек** – свойства отображения точек на графике в соответствии с качеством значения пера, а также значения пера в текстовом виде
- **Скрипт** – скрипт обработки данных тренда

<b>Основные</b>	
Имя пера	Перо1
Стиль пера	Линия
Видимость	Линия
Кешировать данные	Ломаная
Включать граничные точки	Сглаженная
Достраивать график	Ступенчатая
Множитель	Гистограмма
Привязка	Точечная
<b>Режим начала отрисовки</b>	
Цвет линии	Красно-коричневый
Толщина линии	3
Стиль линии	Сплошная
Цвет заливки	
Показывать заливку	Нет
<b>Настройки отображения точек</b>	
Скрипт	<Редактировать>
<b>Режим начала отрисовки</b>	
Время	04.07.2011 11:04
Начало отрисовки	Автоматическая
<b>Настройки отображения точек</b>	
Показывать точки	Нет
Радиус точек	7
Цвет точек для качества Good	Зеленый
Цвет точек для качества Uncertain	Коричневый
Цвет для точек качества Bad	Синий
Показывать значения точек	Нет
Шрифт для значений точек	Arial; 12
Цвет текста для значений точек	Черный

### ВНИМАНИЕ!!!

Заливка работает не со всеми типами графиков (стилями пера). При установке этого свойства возможна подмена одного стиля на тот, который поддерживает заливку.

### Ось значений

- **Способ задания границы** – Ручной, если границы задаются пользователем, **Автоматический** – границы вычисляются автоматически
- **Ручная граница** – задание типа ручной границы **Константа** или **Текущее значение**
- **Верхняя граница, Нижняя граница** – граничные значения отображения пера
- **Привязка верхней границы, Привязка нижней границы** – привязки граничных значений отображения пера

<b>Ось значений</b>	
Способ задания границы	Автоматический
Ручная граница	Автоматический
Верхняя граница	Ручной
Нижняя граница	-1
<b>Ось значений</b>	
Способ задания границы	Ручной
Ручная граница	Константа
Верхняя граница	1
Нижняя граница	-1
<b>Ось значений</b>	
Способ задания границы	Ручной
Ручная граница	Текущее значение
Привязка верхней границы	<input type="checkbox"/> (нет)
Привязка нижней границы	<input type="checkbox"/> (нет)

## ВНИМАНИЕ!!!

В случае, если свойства одной из осей пера не совпадают со свойствами оси тренда, создается дополнительная ось на тренде для пера.

Если не совпадают свойства оси значений, то соответственно появится новая ось для значений. Аналогично для оси времени.

Например: для тренда установлено свойство: **Ось значений**, а **Способ задания границы** в значение **Автоматический**. Для Пера установлено свойство: **Ось значений**, а **Способ задания границы** в значение **Ручная**. Появится дополнительная ось для пера с ручным способом задания границ.

## Текст

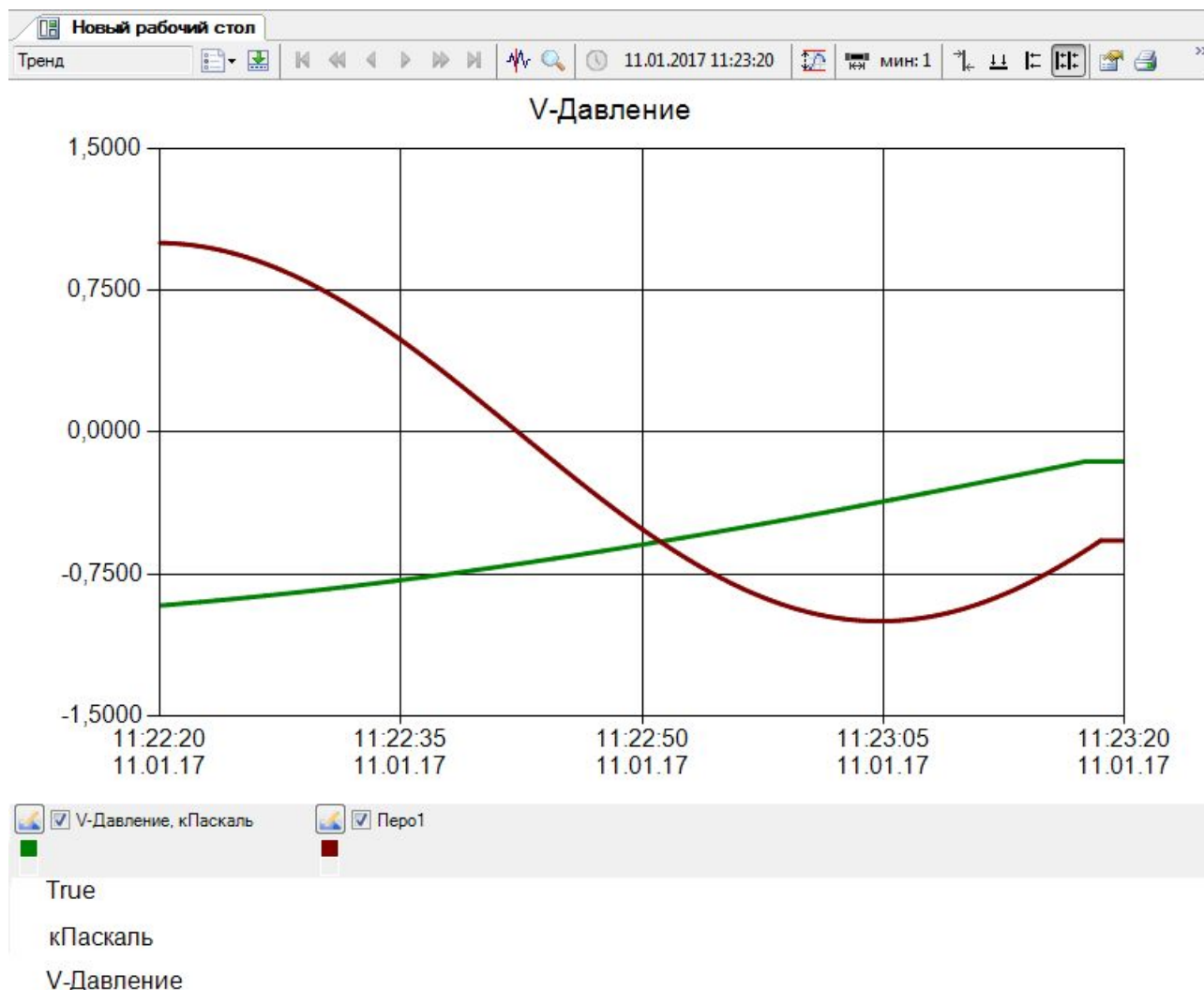
- **Отображать** – переключатель (**Да/Нет**) отображения пера. По умолчанию – **Да**. Если **Нет**, то в легенде пера отображается значение **Имя пера** (свойства **Основные**)
- **Тип значения** – переключатель **Константа/Привязка**
- **Значение** – текст, отображаемый в легенде пера
- **Привязка** – привязка к тегу объекта, параметр которого (строковое значение) отображает в легенде пера наименование пера
- **Тип значения единиц измерения** – переключатель **Не используется/Константа/Привязка**. По умолчанию – **Не используется**
- **Единицы измерения** – наименование единиц измерения
- **Привязка** – привязка к тегу объекта, параметр которого (строковое значение) отображает в легенде пера наименование единиц измерения

<b>Текст</b>	
Отображать	Да
Тип значения	Константа
Значение	Перо1
Тип значения единиц измерения	Не используется
Не используется	
Константа	
Привязка	
<b>Текст</b>	
Отображать	Да
Тип значения	Константа
Значение	Перо1
Тип значения единиц измерения	Константа
Единицы измерения	кПа
<b>Текст</b>	
Отображать	Да
Тип значения	Привязка
Привязка	<input checked="" type="checkbox"/> Элемент при
Тип значения единиц измерения	Привязка
Привязка	<input checked="" type="checkbox"/> Элемент при

## 6.6.4.2 Пример управления заголовком и отображением перьев аналитического тренда

**Задача**

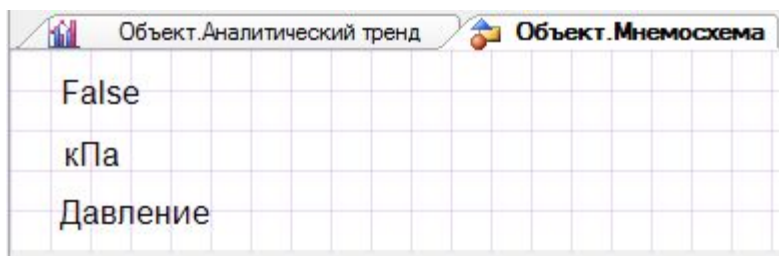
Создать перья аналитического тренда и элементы управления отображением пера, заголовка тренда и легенды.

**Решение**

- 1 Создадим проект с тегами **Синус** и **Синус1** для привязки перьев тренда, а также **Простой тег** атрибуты которого используются для отображения пера (**Загрузка**) и вывода заголовка тренда, текста легенды (**Заголовок**) и единиц измерения (**Единицы измерения**)

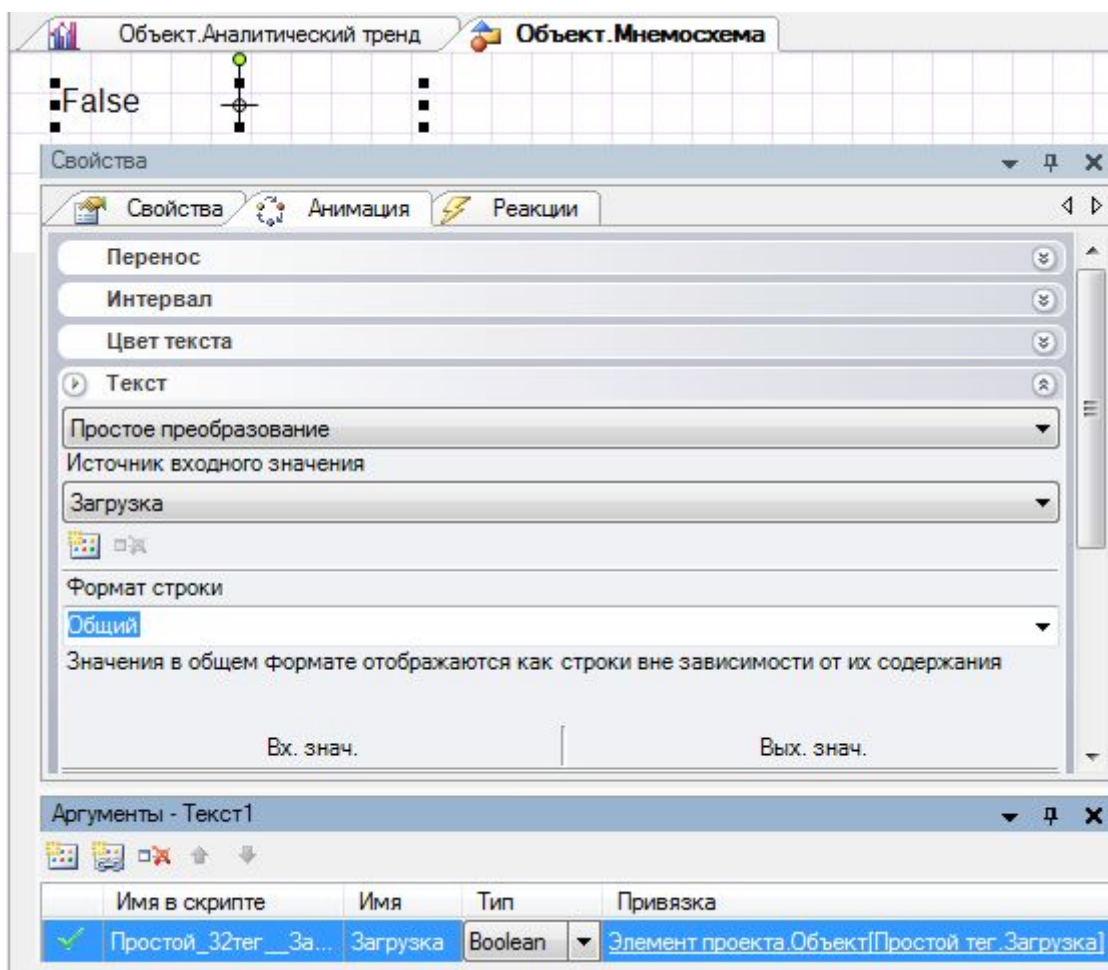
Объект. Простой тег							
#	Имя	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение	
1	Синус						
2	Синус1						
3	Простой тег						
1	Загрузка	Загрузка	Boolean		<input type="checkbox"/>	False	
2	Единицы измерен...	Единицы измерения	String	256	<input type="checkbox"/>	кПа	
3	Заголовок	Заголовок	String	256	<input type="checkbox"/>	Давление	

- 2 Добавим в проект виды **Аналитический тренд** и **Мнемосхема**
- 3 Разместим на **Мнемосхеме** графические элементы **Текст1** (False), **Текст2** (кПа), **Текст3** (Давление) и настроим их **Анимацию** и **Реакции**.

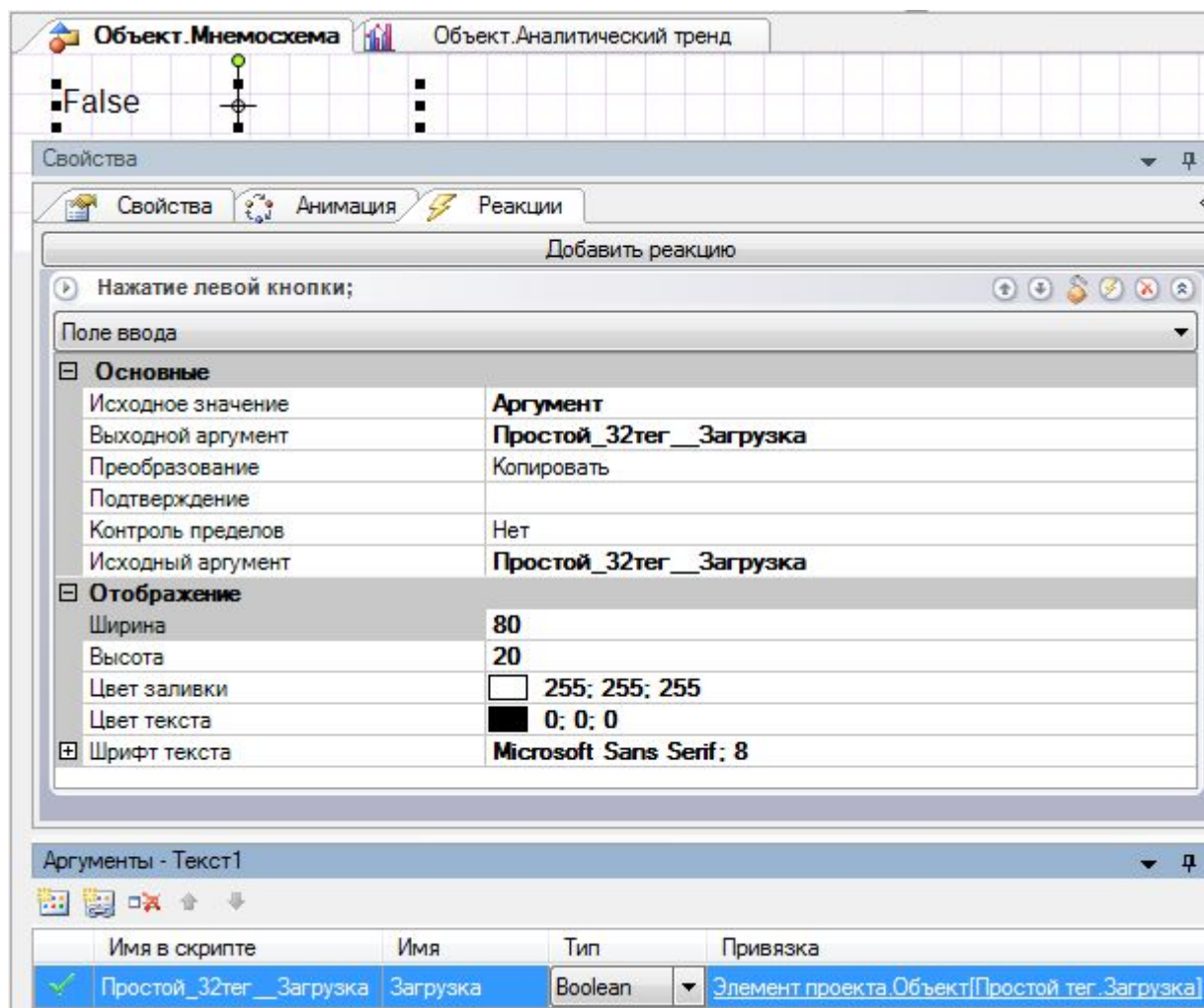


Для поля управления загрузкой пера (элемент **Текст1**) это выглядит так:

- **Текст1.** Анимация **Простое преобразование** атрибута **Текст**



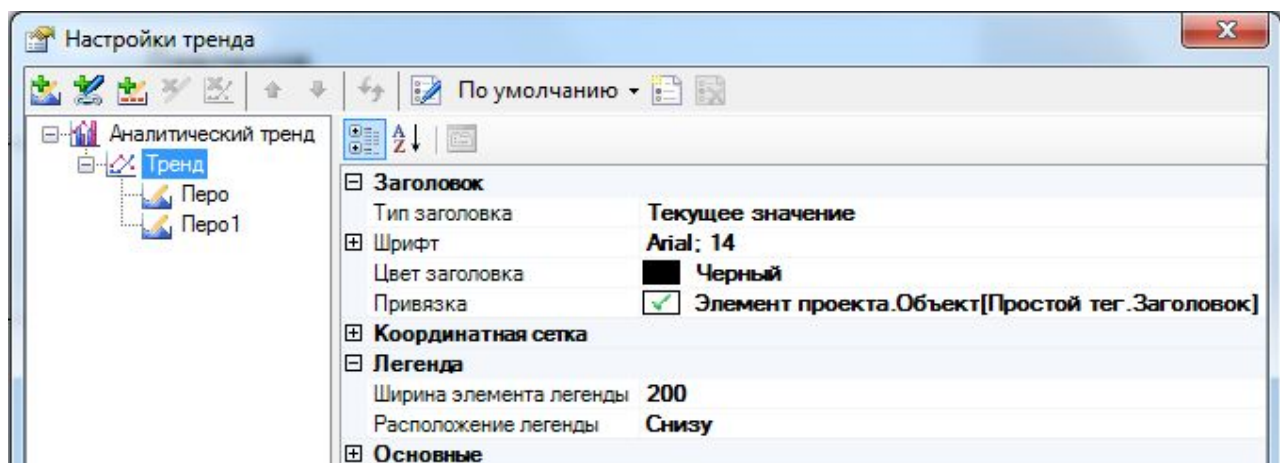
- Текст1. Реакция Поле ввода на событие Нажатие левой кнопки



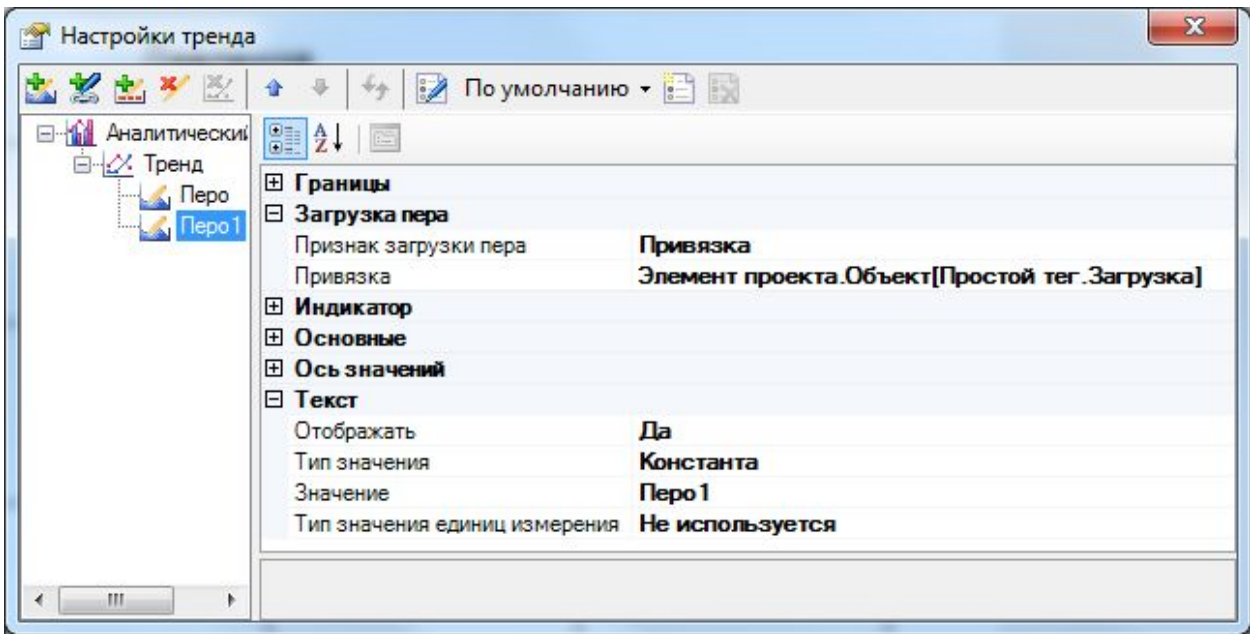
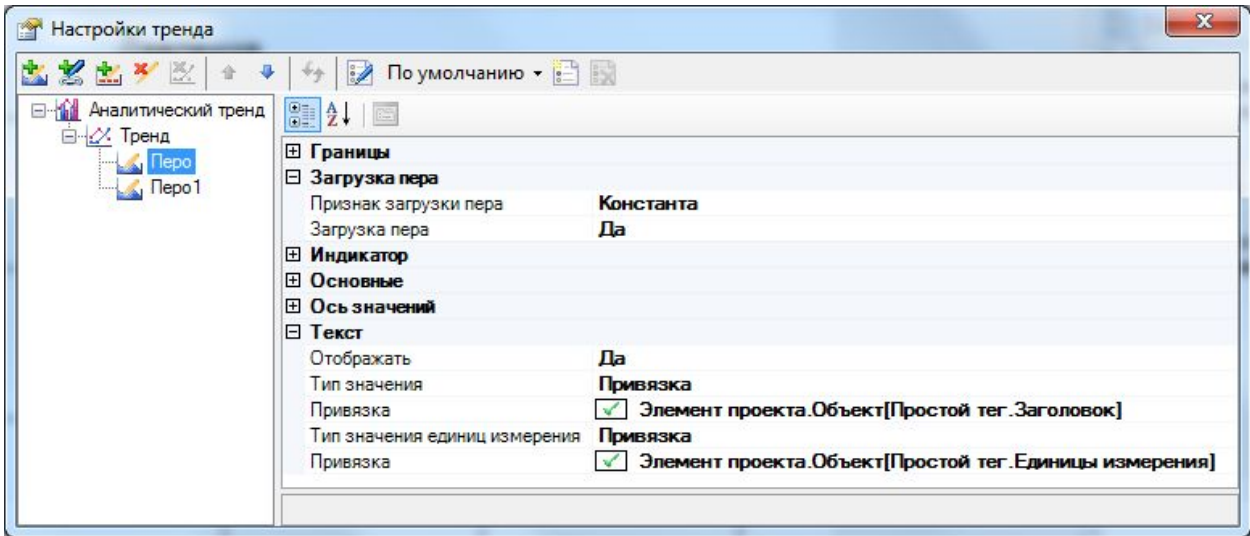
Аналогично настраиваем **Анимацию** и **Реакцию** для элементов **Текст2** (аргумент **Единицы измерения**) и **Текст3** (аргумент **Заголовок**)

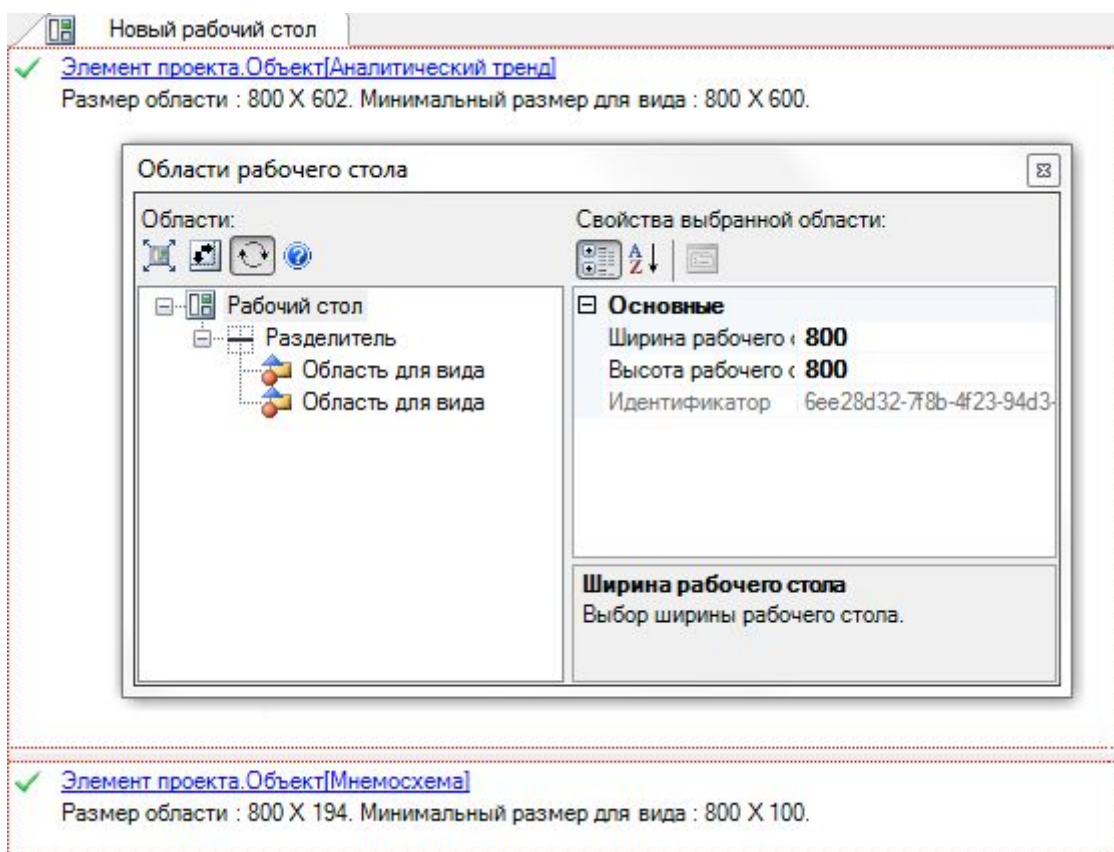
#### 4 Настраиваем вид Аналитический тренд

- Настройка тренда

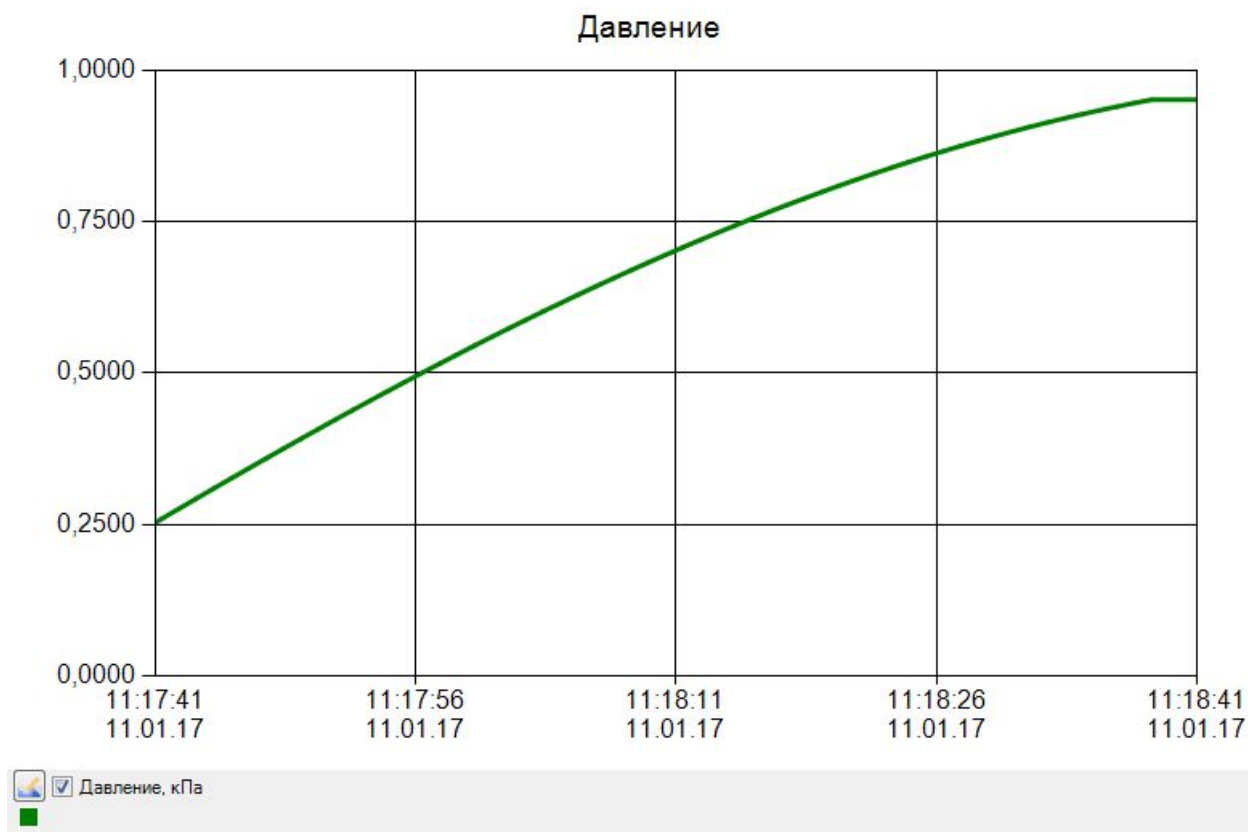




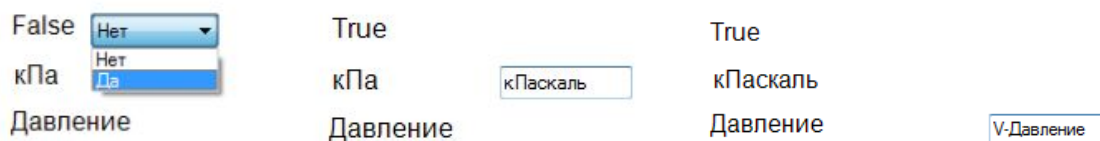




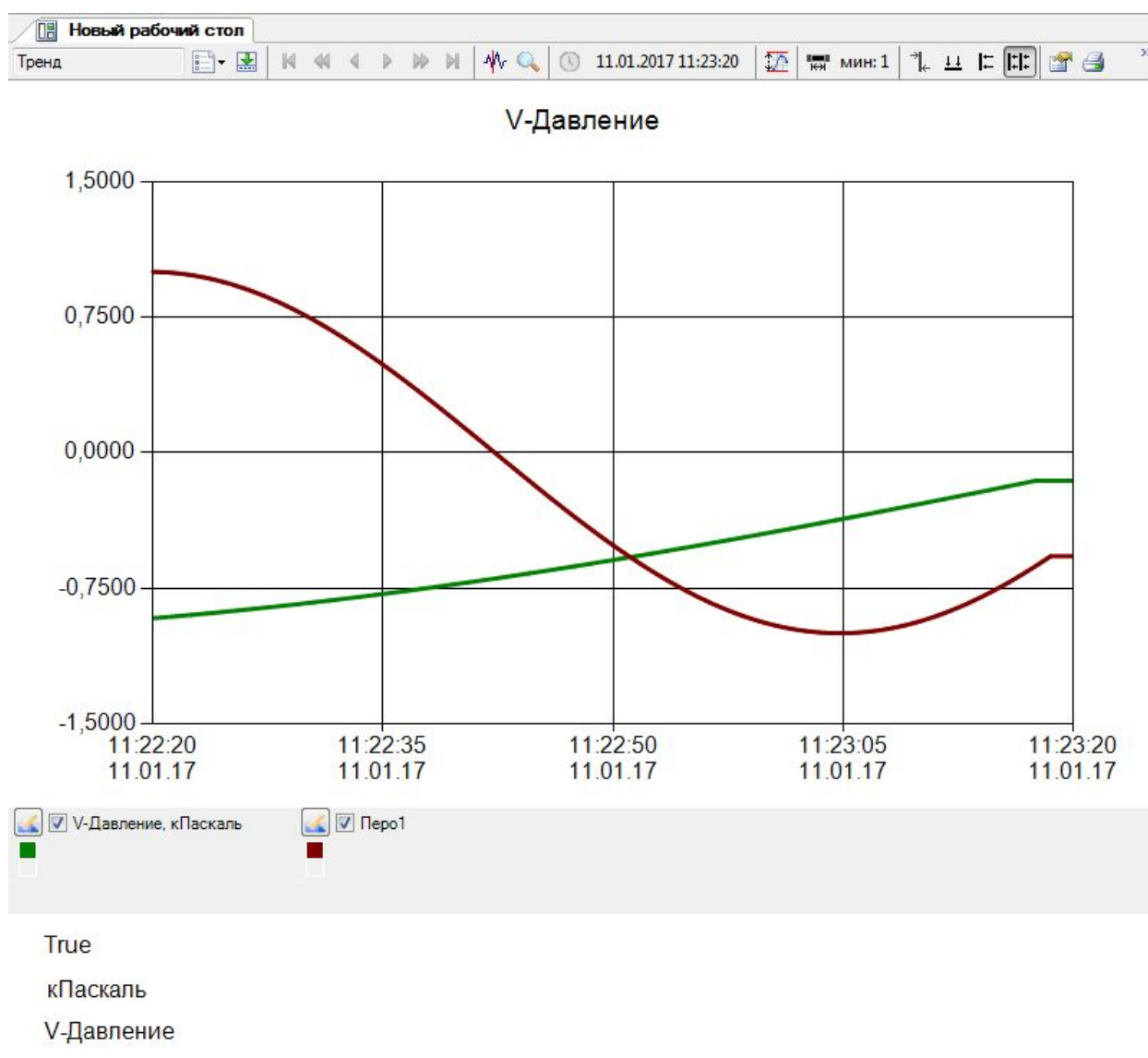
- 6 Запускаем проект на исполнения. Первоначально отображается только одно перо, так как свойство **Загрузка** второго пера установлено в **False**



- 7 Изменим значение свойства **Загрузка** на значение **True**, а также текст легенды и наименования единиц измерения следующим образом:

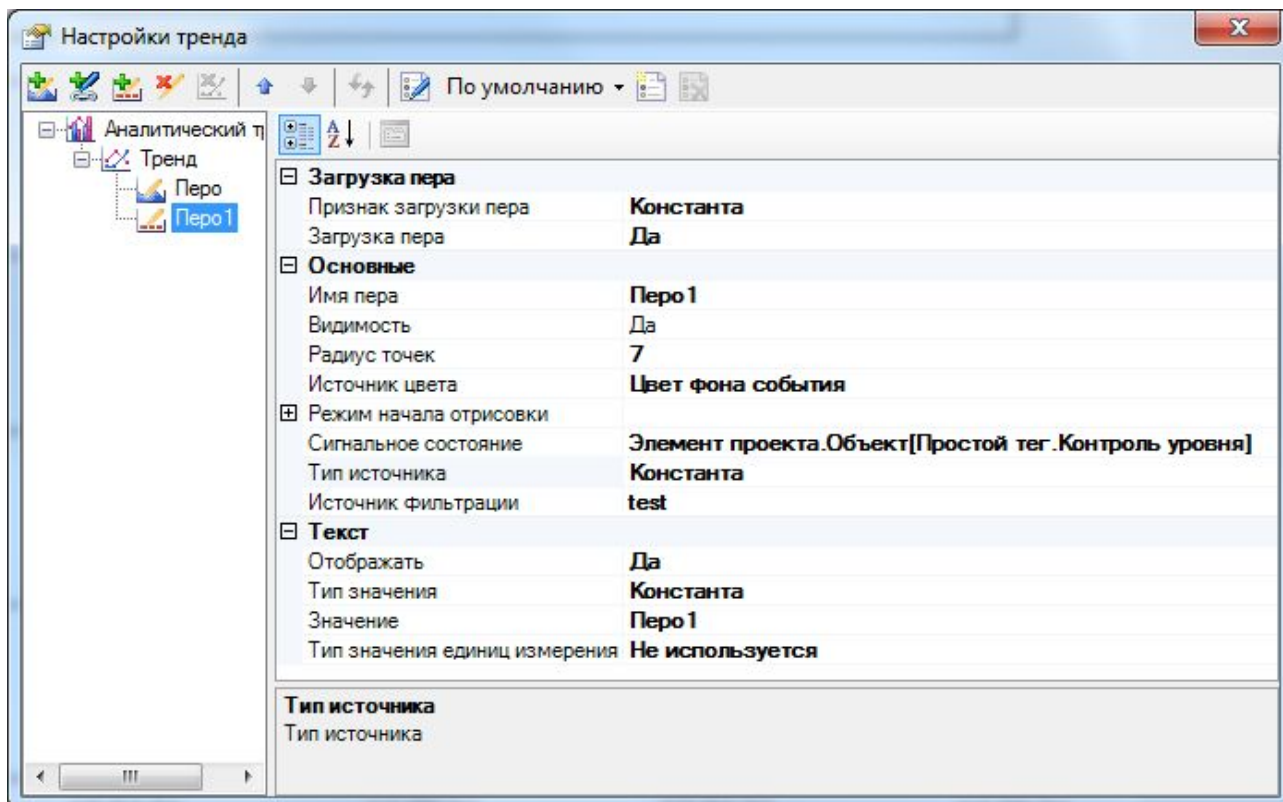


- 8 В результате получим **Аналитический тренд** следующего вида





- **Режим начальной отрисовки.** Если выбран тип **Автоматическая**, то время начала отрисовки будет совпадать с левой границей графика
- **Сигнальное состояние**, с которым связано перо
- **Тип источника** (константа или привязка к атрибуту тега)
- **Источник фильтрации**, связанный с полем **Источник** настроек сигнализации



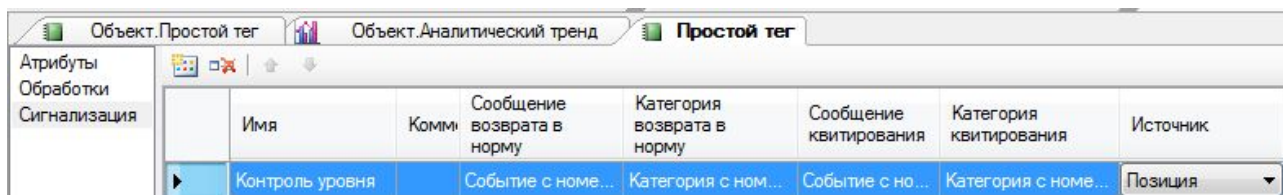
### 6.6.4.4 Пример создания перьев тренда и протокола событий

#### Задача:

Создать перо тренда и перо протокола событий для заданного тега.  
Вывести на график события из протокола событий, связанные с пером тренда.

#### Решение

- 1 Сначала следует создать библиотечный тег и настроить для него сигнализацию:
  - Создайте в библиотечной части проекта тип тега **Простой тег**
  - Перейдите на вкладку **Сигнализация**, выберите тип сигнализации **Контроль уровня** (настройки Протокола событий будут установлены по умолчанию)
  - Укажите в поле **Источник** атрибут **Позиция** тега **Простой тег**.





Сообщение возврата в норму Событие с номером 1 из Словаря событий с аргументами: Позиция,ТЗ

Категория возврата в норму Категория с номером 7 из Словаря категорий

Сообщение квитирования Событие с номером 8 из Словаря событий с аргументами: Позиция

Категория квитирования Категория с номером 2 из Словаря категорий

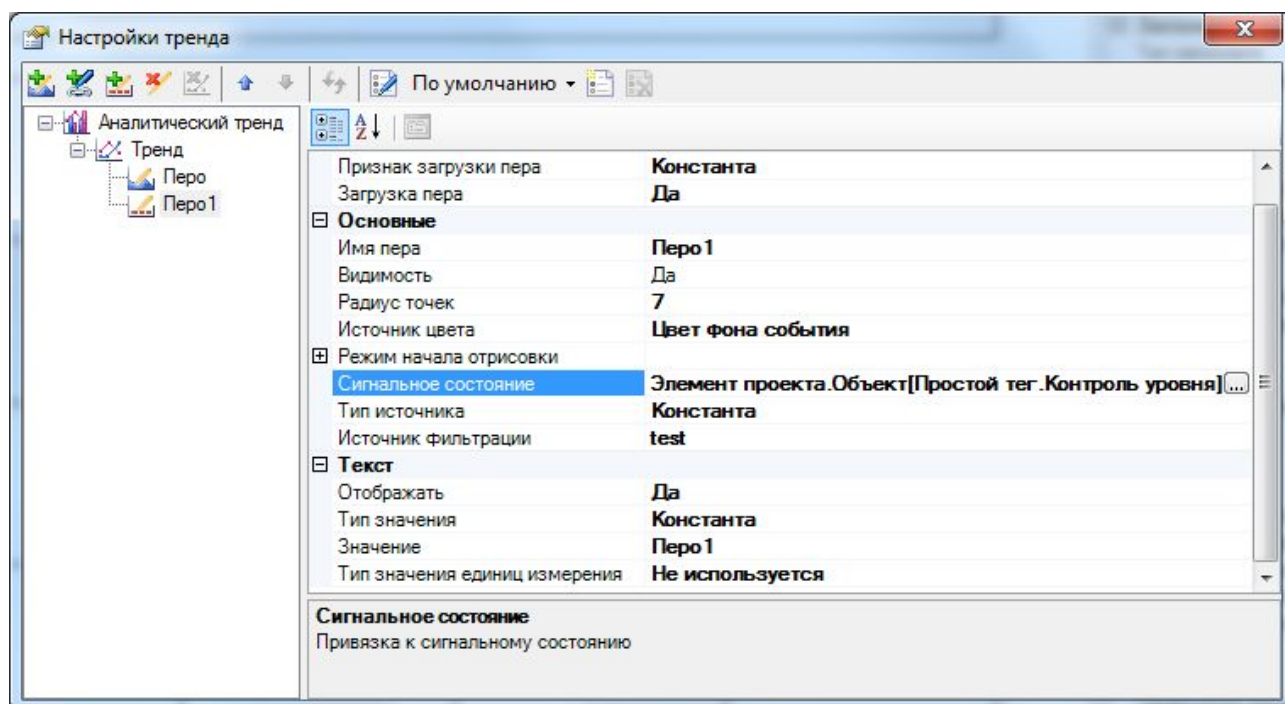
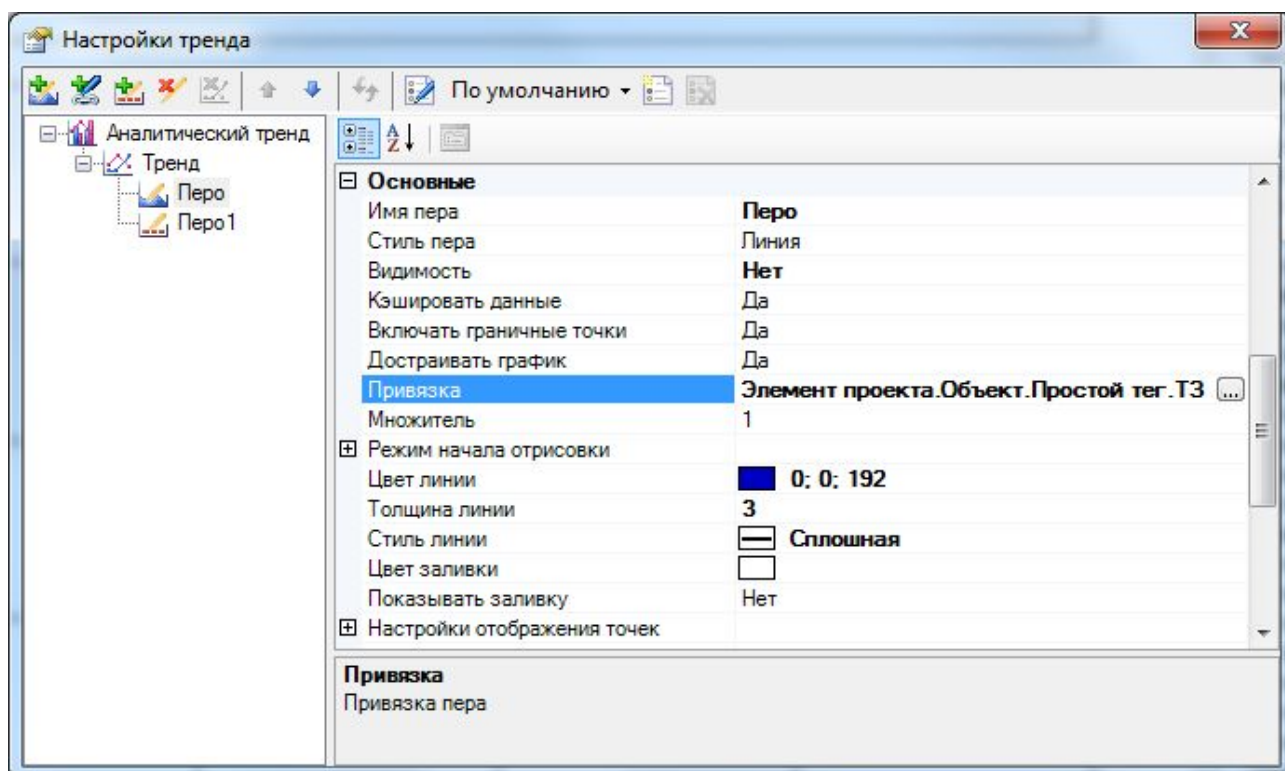
- 2 Для объекта **Объект** создайте теги **Синус** и **Простой тег** (на основе библиотечного тега **Простой тег** и укажите значение атрибута **Позиция**, в примере – **test**)

Объект.Простой тег							
Объект.Аналитический тренд							
	#	Имя	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение
	1	ТЗ	Текущее значение	Double		<input type="checkbox"/>	0
	2	ВАГ	Верхняя предаварийная гран...	Double		<input type="checkbox"/>	0,9
	3	ВПГ	Верхняя предупредительная г...	Double		<input type="checkbox"/>	0,8
	4	НПГ	Нижняя предупредительная г...	Double		<input type="checkbox"/>	-0,8
	5	НАГ	Нижняя предаварийная грани...	Double		<input type="checkbox"/>	-0,9
	6	Позиция	Позиция	String	256	<input type="checkbox"/>	test

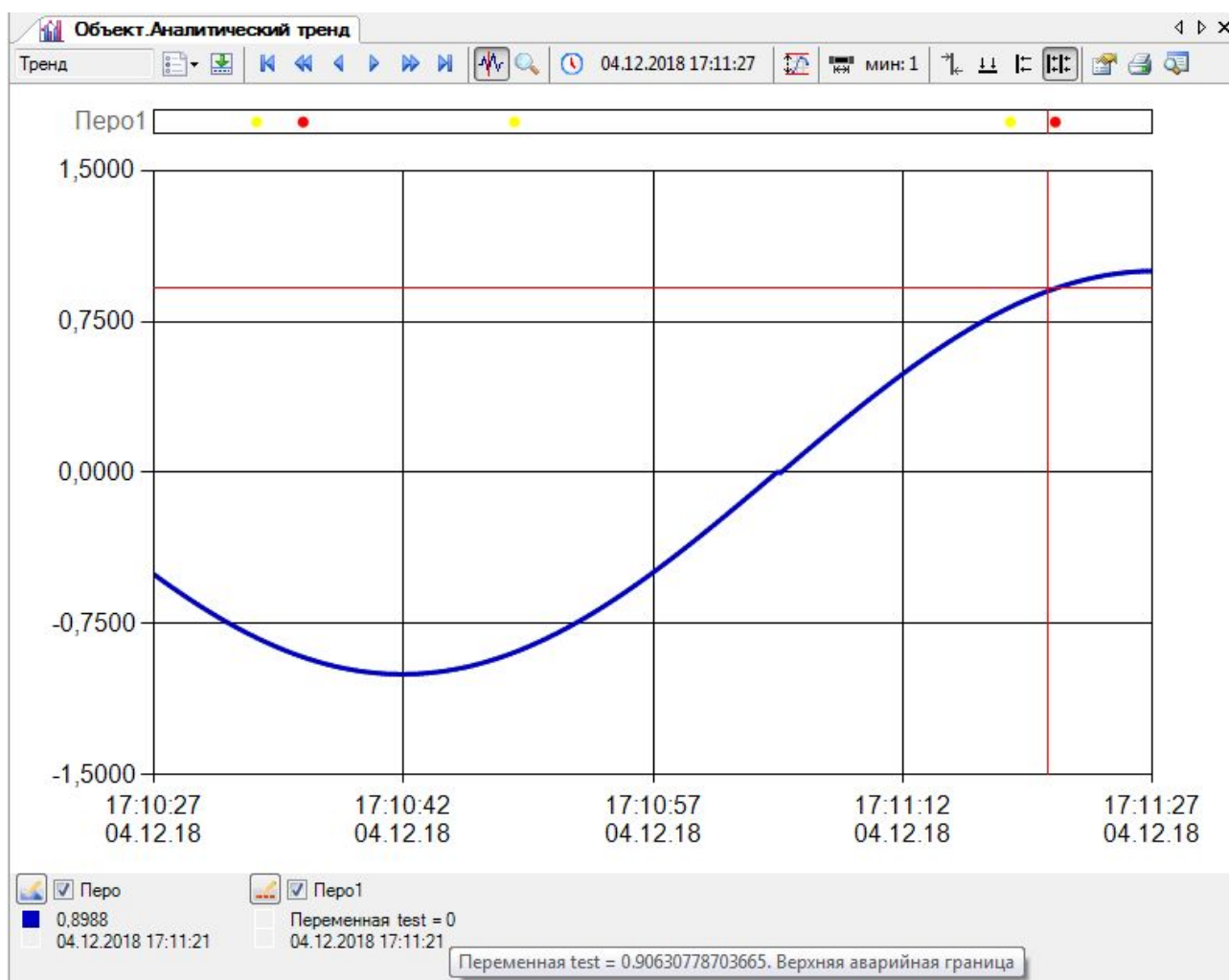
- 3 Свяжите **Простой тег** объекта с тегом **Синус**

Свойства	
Свойства	
Основные	
Атрибуты	(Коллекция)
Имя	Простой тег
Описание	Простой тег
Ссылка на библиотеку	<input checked="" type="checkbox"/> Элемент проекта.Простой тег
Связь с другим тегом	<input checked="" type="checkbox"/> Элемент проекта.Объект[Синус]

- 4 Создайте вид Аналитический тренд и настройте его перья:
- **Перо** – тренд атрибута **ТЗ** тега **Простой тег**
  - **Перо1** – перо **Протокола событий**



- 5 Запустите проект на исполнение. Включите режим светового пера



## 6 Откройте окно Протокола событий

Элемент проекта.Протокол событий

1.Протокол событий

Время	Качество	Серьезность	Название серьезности	Сообщение	Источник
04.12.2018 17:11:21	Хорошее	1000	Очень высокая	Переменная test = 0.9063... test	
04.12.2018 17:11:19	Хорошее	750	Высокая	Переменная test = 0.8090... test	
04.12.2018 17:10:52	Хорошее	250	Низкая	Переменная test = 0.90630778703665. Верхняя аварийная граница	
04.12.2018 17:10:49	Хорошее	750	Высокая	Переменная test = -0.898... test	
04.12.2018 17:10:36	Хорошее	1000	Очень высокая	Переменная test = -0.906... test	
04.12.2018 17:10:33	Хорошее	750	Высокая	Переменная test = -0.809... test	
04.12.2018 17:10:07	Хорошее	250	Низкая	Переменная test = 0.7986... test	
04.12.2018 17:10:04	Хорошее	750	Высокая	Переменная test = 0.8987... test	
04.12.2018 17:09:51	Хорошее	1000	Очень высокая	Переменная test = 0.9063... test	
04.12.2018 17:09:48	Хорошее	750	Высокая	Переменная test = 0.8090... test	
04.12.2018 17:09:21	Хорошее	250	Низкая	Переменная test = -0.798... test	

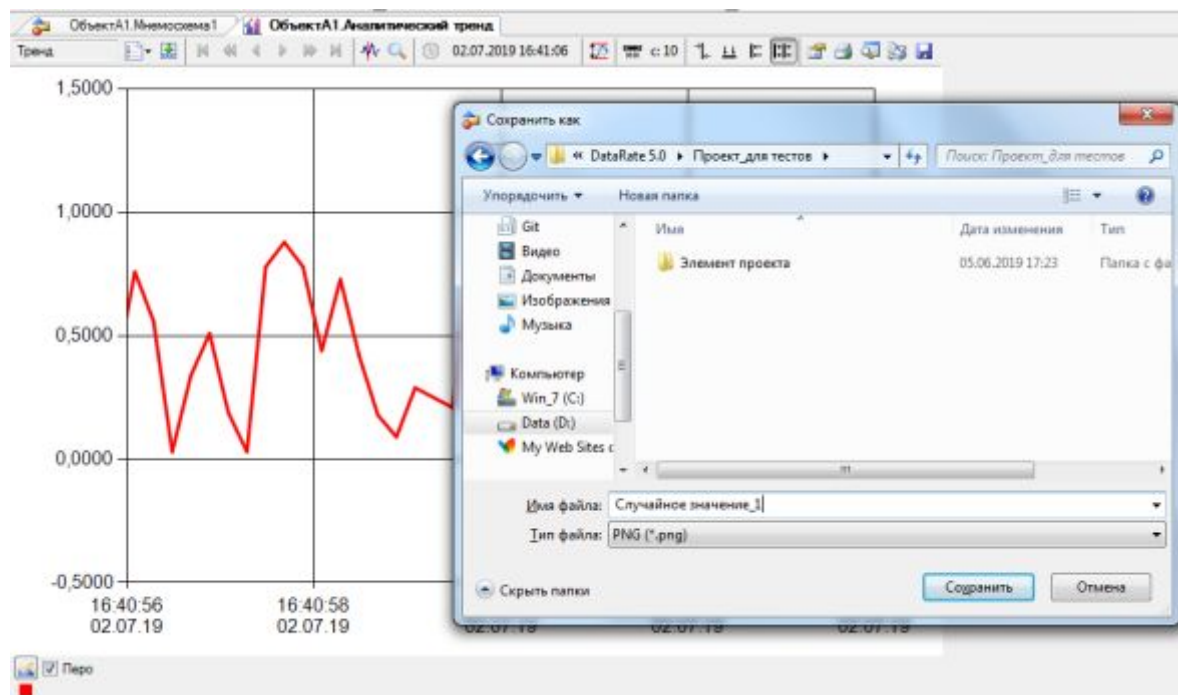
Прием событий прекращен. Событий: 301 Интервал времени: 01.01.0001 3:00:00 - текущее время

### 6.6.5 Сохранение изображения тренда

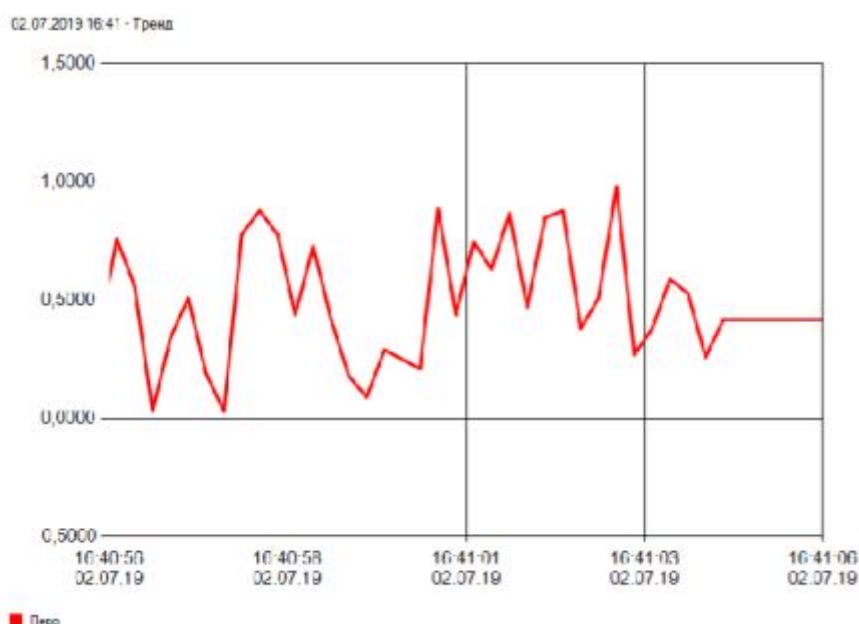
Функция сохранения изображения служит для сохранения изображения аналитического тренда в графическом файле.

Для сохранения изображения следует:

- 1 Нажать в панели инструментов на кнопку **Сохранить изображение тренда**
- 2 В открывшемся окне Сохранить как выбрать папку (нажать на кнопку **Обзор папок**), выбрать тип файла (**PNG, JPEG, Bitmap, EMF, GIF, TIFF**) и задать имя файла.



- 3 Нажать на кнопку **Сохранить**.
- 4 Результирующий файл содержит следующее изображение аналитического тренда

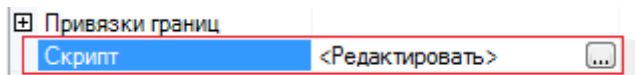


### 6.6.6 Изменение данных скриптом

**Аналитический тренд** поддерживает возможность обработки выводимых данных тренда с помощью скрипта.

Для этого в диалоговом окне **Настройки тренда** следует:

- 1 Выбрать перо тренда, значения которого нужно изменить
- 2 В разделе свойств **Основные** выделить свойство **Скрипт** и нажать на кнопку в поле **<Редактировать>**



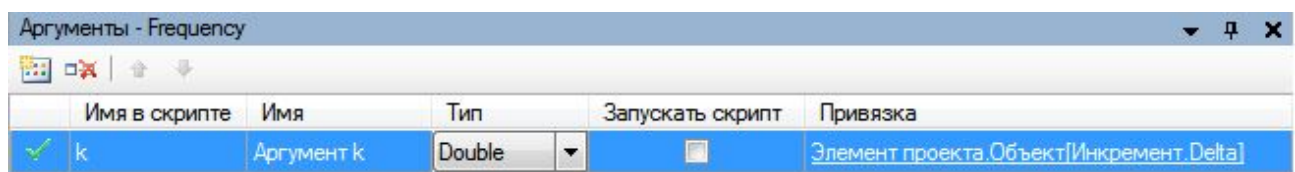
- 3 В появившемся окне редактора скрипта отредактировать код скрипта, изменяющего выводимые значения пера аналитического тренда.

#### Пример

В примере выводимые данные пера аналитического тренда умножаются на коэффициент **k**, который задается атрибутом тега или константой.

Для связи с атрибутом тега следует создать аргумент скрипта:

- 1 Сделать видимым окно **Аргументы**
- 2 Добавить аргумент типа **Double**
- 3 Задать имя аргумента (**Имя в скрипте**) и привязку к вещественному атрибуту тега (**Привязка**).



Во вкладку **Общий код** добавить переменные **k** (коэффициент изменения значения пера тренда) и **last\_k** (для проверки изменения коэффициента):

```
const double k = значение; //«значение» может быть любым числом
double last_k = 0;
```

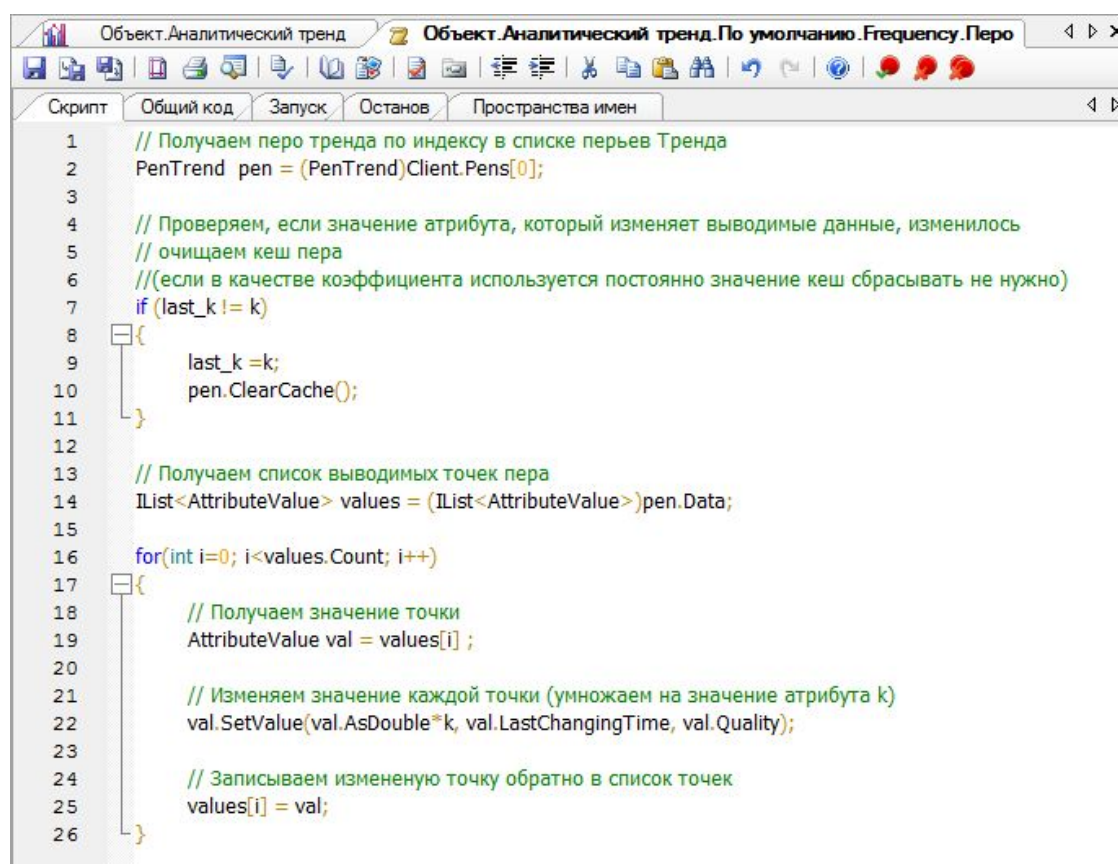
Во вкладке **Скрипт** создать следующий скрипт обработки данных:

```
// Получаем перо тренда по индексу в списке перьев Тренда
PenTrend pen = (PenTrend)Client.Pens[0];
//Проверяем, если значение атрибута, который изменяет выводимые данные
//изменилось, очищаем кеш пера (если в качестве коэффициента используется
//постоянно значение, кеш сбрасывать не нужно)
if (last_k != k)
{
    last_k = k;
    pen.ClearCache();
}
// Получаем список выводимых точек пера
IList<AttributeValue> values = (IList<AttributeValue>)pen.Data;
```



```
//Обрабатываем каждую точку
for(int i=0; i<values.Count; i++)
{
    // Получаем значение точки
    AttributeValue val = values[i] ;
    // Изменяем значение точки (умножаем на значение атрибута k)
    val.SetValue(val.AsDouble*k, val.LastChangingTime, val.Quality);
    // Записываем измененную точку обратно в список точек
    values[i] = val;
}
```

Так выглядит этот скрипт в редакторе скриптов среды разработки **DataRate**



Во вкладке **Пространства имен** должен быть код:

```
Krug.Tools
Krug.Tools.Scripting
Krug.Tools.Scripting.Algorithms
Krug.ObjectModel.AlarmServiceInterfaces
Krug.ObjectModel.HistoricalServiceInterfaces
System
System.Windows.Forms
System.Diagnostics
Krug.ObjectModel
Krug.Tools.Colors
```

Krug.ObjectModel.TimeSeriesAnalysis  
 Krug.ObjectModel.TimeSeriesAnalysis.Model  
 System.Drawing  
 System.Collections.Generic

### 6.6.7 Источники данных привязок

**Источники данных привязок** представляют собой иерархически структурированный список источников данных для перьев аналитического тренда.

#### ВНИМАНИЕ!!!

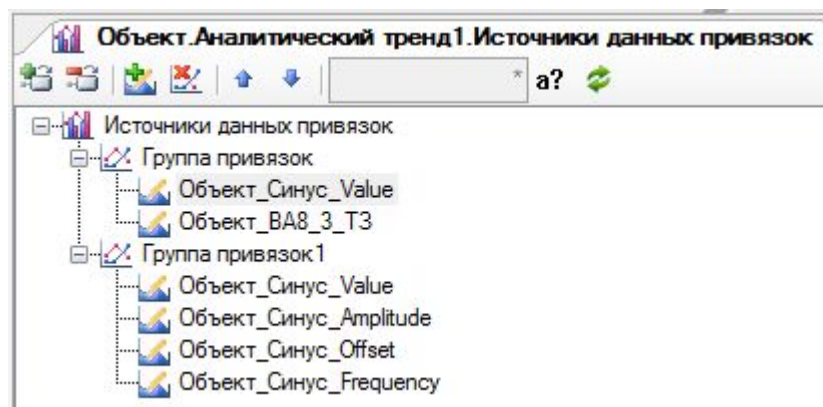
**Источники данных привязок** предназначены для добавления перьев аналитического тренда оператором АСУТП в среде исполнения **DataRate** (на сервере и на клиентах).

Источники данных разбиваются по группам привязок и содержат наименование и привязку.

Каждый аналитический тренд имеет свои источники данных привязок.

#### 6.6.7.1 Редактор источников привязок

**Редактор источников привязок** вызывается из контекстного меню **Источники данных привязок** элемента списка видов объекта **Аналитический тренд**.



Редактор источников привязок обеспечивает:

- Добавление/удаление группы источников привязок
- Добавление/удаление источников привязок
- Перемещение выделенных элементов в дереве источников данных привязок.

#### Добавление группы источников привязок

Для добавления группы источников привязок следует:

- 1 Открыть **Источники данных привязок**
- 2 Нажать кнопку **Добавить группу**
- 3 Задать имя группы в окне **Свойства**.

#### Добавление источников привязок

Для добавления источников привязок следует:

- 1 Открыть **Источники данных привязок**
- 2 Создать группу привязок или выбрать группу из существующих
- 3 Нажать кнопку **Добавить источник привязки**

- 4 В окне **Редактор привязки тренда** выбрать атрибут тега, к которому необходимо привязаться, и нажать кнопку **ОК**.


### Добавление источников привязок с применением технологии Drag&Drop


Для добавления источников привязок с применением **Drag&Drop** следует:

- 1 Открыть **Источники данных привязок**
- 2 В окне **Редактор проекта** выделить объект содержащий теги
- 3 В окне **Редактор проекта** на закладке **Теги** выделить необходимые теги
- 4 Переместив указатель мыши на выделенную область, нажать левую кнопку мыши
- 5 Перетащить указатель в область редактора источников привязок на группу источников привязок
- 6 Отпустить левую кнопку мыши.  
В результате произведенных действий откроется **Окно редактора связей**
- 7 Нажать на кнопку **ОК** (источники привязок для перьев будут добавлены).



Аналогичным образом можно добавлять Источники данных привязок из списка перьев Менеджера трендов.

### Удаление группы или источников привязок


Для удаления группы, необходимо в редакторе источников выбрать элемент иерархического списка и нажать кнопку **Удалить группу** .

Если необходимо удалить только источник привязки, то нажать кнопку **Удалить источник привязки** .

### Порядок следования групп и источников данных привязок

Стрелки **Вверх**  / **Вниз**  на панели инструментов над списком источников данных привязок позволяют задать порядок следования групп и источников данных привязок.

### Фильтрация

Кнопка **а?**  позволяет включить или отключить фильтрацию источников данных привязок. После включения фильтра необходимо задать его маску. Маска фильтра применяется автоматически и список источников перестраивается.

### Обновление списка

Список данных источников привязок обновляется автоматически в соответствии с изменениями в проекте и заданной маской фильтрации. При необходимости, список источников можно обновить вручную.

### Редактирование свойств источников привязок



Источники привязок содержат следующие свойства:




- **Имя** – наименование источника привязок. Наименования источников привязок желательно задавать в понятных оператору АСУТП терминах
- **Имя пера** – при добавлении пера в аналитический тренд, это свойство будет использоваться в названии перьев
- **Привязка** – привязка к атрибуту тега.

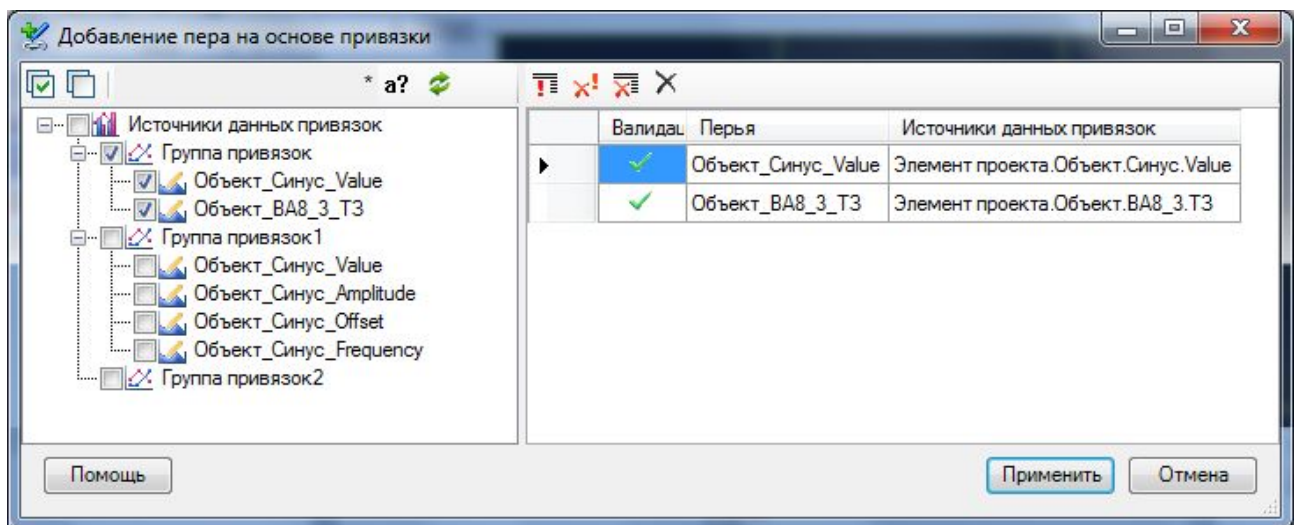
#### 6.6.7.2 Добавление перьев на основе созданных привязок

При выполнении проекта, иерархический список источников привязок доступен в среде исполнения (на сервере) и в среде исполнения на клиентах.

Для добавления пера аналитического тренда следует:

- 1 Открыть **Окно настройки** Аналитического тренда
- 2 Нажать кнопку **Добавление пера на основе привязки** 
- 3 В открывшемся окне **Добавление пера на основе привязки** нужно выбрать один или более источников привязок. Для этого в иерархическом списке источников отметить флажком необходимые элементы или группы  
Для удобства выбора в окне можно применить фильтр элементов списка. Фильтр включается нажатием кнопки **Фильтр** , после чего задать строку фильтра.
- 4 Нажать кнопку **Применить**.

Перед добавлением перьев осуществляется проверка доступности источников данных. При обнаружении невалидных привязок в добавлении перьев будет отказано. Для удаления невалидных источников привязок из списка выделенных необходимо нажать кнопку **Удалить с ошибкой** . Если необходимо удалить другие источники привязок, нужно в списке выбранных источников выделить необходимые и нажать кнопку **Удалить выделенные** . Для очистки всего списка необходимо нажать кнопку **Удалить все выделенные** .



## 6.7 Режимный тренд

**Режимный тренд** DataRate предназначен для визуализации регистрируемых и режимных значений параметров технологического процесса.

**Основные отличия** режимного тренда:

- Отображение «идеального» графика (значения режимных параметров) совместно с «реальным» графиком (реальный тренд, регистрируемые значения параметров)
- Отрисовка значений параметров относительно заданной точки отображения
- Отображение идеального графика в «будущем» времени

Режимный тренд в DataRate визуализируется графическим примитивом **Тренд** или видом **Аналитический тренд**.

**Два способа визуализации** режимного тренда:

- **Интервальный** – идеальный график и реальный тренд отображаются только в области, определяемой смещением от текущего времени
- **Полномасштабный** – идеальный график отображается полностью, а реальный тренд отрисовывается до момента текущего времени.

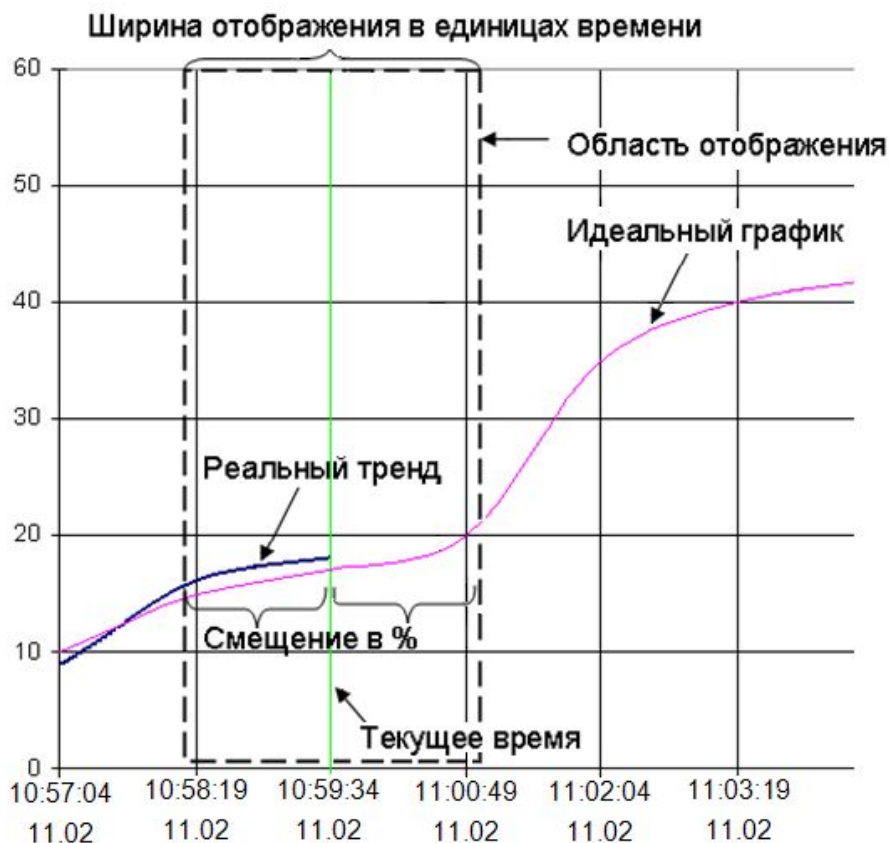
**Два способа задания данных** для режимного тренда:

- **Запись значений в перо тренда из скрипта.** Этот способ позволяет записать значения режимных данных напрямую в перо тренда, и затем визуализировать это перо совместно с пером реальных данных. Если необходимо, возможно сохранить историю режимных данных
- **Создание источника режимных данных.** Источник данных содержит режимные данные и используется в качестве привязки пера тренда. Этот способ позволяет гибко управлять режимными данными в **Среде исполнения DataRate**, в том числе получать их значения из базы данных (**SQL-коннектор**) или задавать с помощью математической функции. Для привязки к перу тренда можно использовать источники данных **SQL коннектор** или **Скрипт**.

При необходимости задавать разные цвета на участках графика режимного тренда следует создать столько источников данных, содержащих соответствующие «куски» режимных данных, сколько цветов планируется использовать.

### 6.7.1 Интервальный режимный тренд

При интервальном отображении видима только та часть тренда, которая попадает в область отображения. Смещение от текущего времени задается в процентах от выбранной ширины шкалы тренда. Границы области отображения смещаются вместе с текущим временем. При этом идеальный график отображается в зависимости от наличия режимных данных за выбранный промежуток времени.





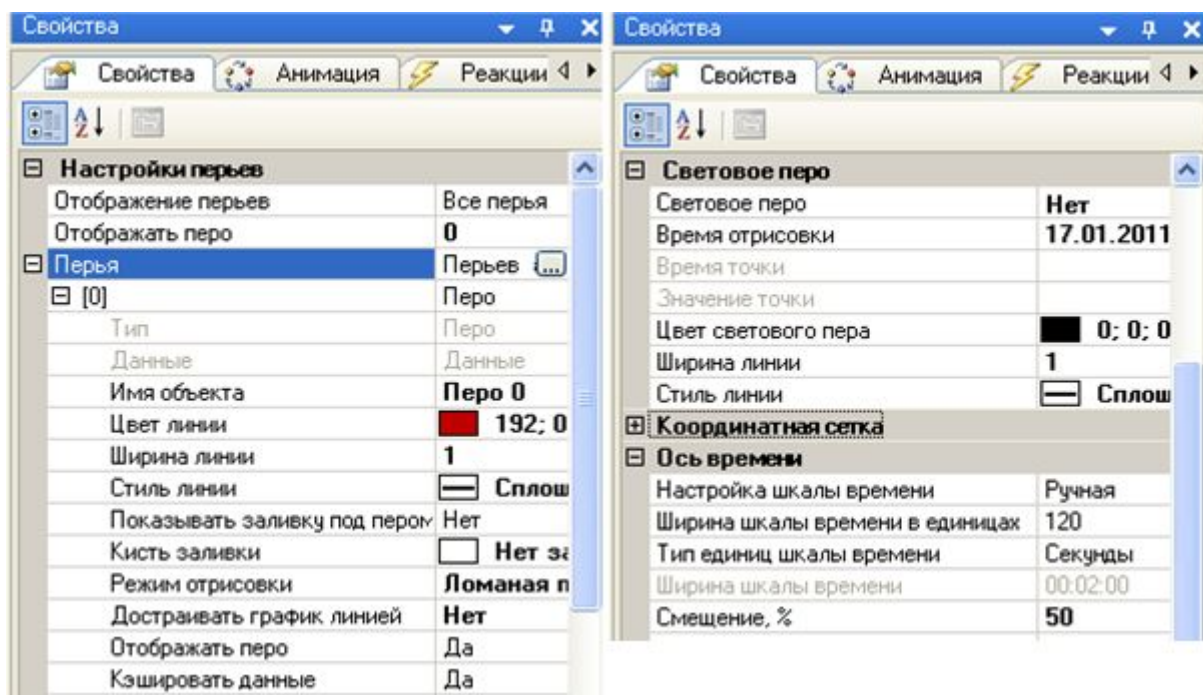
## 6.7.1.1 Настройка тренда

Для визуализации интервального режимного тренда свойства графического примитива **Тренд** или вида **Аналитический тренд** необходимо настроить следующим образом:

- **Режим светового пера выключить**
- **Достраивание графика линией отключить для пера режимного тренда**
- **Смещение задать в % от ширины шкалы времени**
- **Ширину шкалы времени задать в соответствии с требуемым интервалом**

В случае использования примитива **Тренд** для визуализации режимного тренда следует установить значения следующих свойств:

- **Световое перо/Световое перо – Нет**
- **Перья/Достраивать график линией – Нет**
- **Ось времени/Смещение, % – например, 50**
- **Ось времени/Тип единиц шкалы времени – например, Секунды**
- **Ось времени/Ширина шкалы времени в единицах – например, 120 (с учетом типа единицы – 120 секунд или 2 минуты)**



## 6.7.1.2 Пример создания тренда

Для созданных в проекте **DataRate** объекта **Объект** и его вида **Мнемосхема** разработаем интервальный режимный тренд.

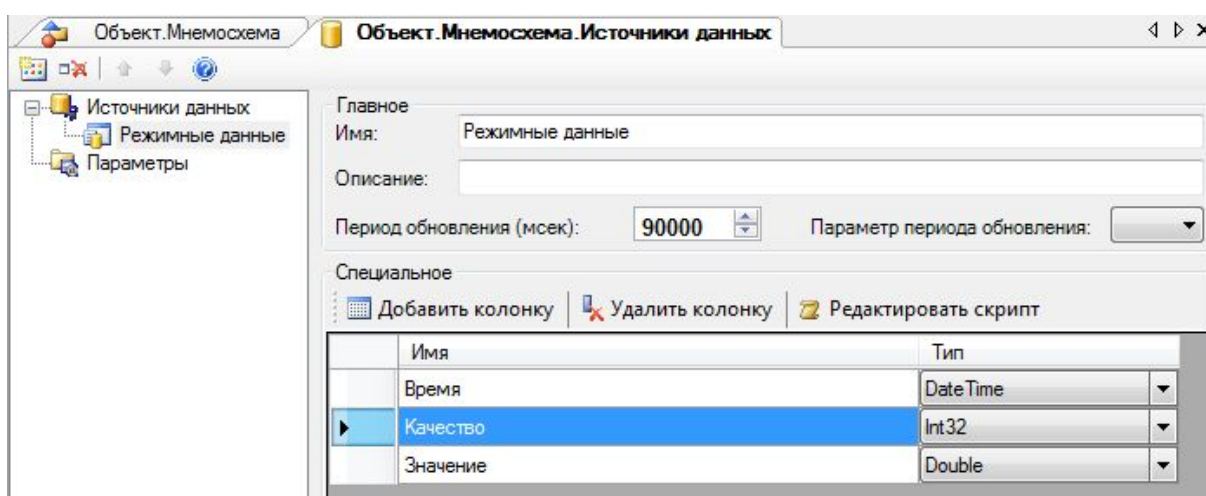
Визуализацию тренда реализуем на основе графического примитива **Тренд** с двумя перьями:

- **Режимные данные.** Задаются с помощью источника данных **Скрипт**
- **Регистрируемые данные.** Задаются с помощью тега-эмулятора **Синус**.

Для создания данного тренда следует:

1 Задать режимные данные:

- В контекстном меню вида **Мнемосхема** выбрать пункт **Источники данных и параметры** и создать источник данных категории **Скрипт** с названием **Режимные данные**
- Установить **Период обновления** (период автоматического обновление источника данных) – **90000 мсек** (90 сек)
- Добавить **3 колонки** значений источника данных:
  - **Время** (тип данных **DateTime**) – временная метка точки (значение должно быть в универсальном времени)
  - **Качество** (тип данных **Int32**) – качество значения точки
  - **Значение** (тип данных **Double**) – значение точки.



- Создать скрипт для формирования режимных данных. Для этого нажать кнопку **Редактировать скрипт**, и в появившемся окне редактора скриптов на вкладке **Скрипт** добавить следующий код:

```
DateTime time = DateTime.UtcNow.AddSeconds(-135);
```

```
// Заполним таблицу значениями функции -1.1*sin(x), где x - угол,
//меняющийся на 4 градуса в секунду
// Функция выбрана для наглядности таким образом, чтобы ее
//график немного отличался от графика синуса.
for (int i=0; i< 270; i++)
{
```

```
    DataRow row = Table.NewRow(); // создаем новую строку для
                                // таблицы
```

```
    row["Время"] = time.AddSeconds(i); // задаем время точки
```

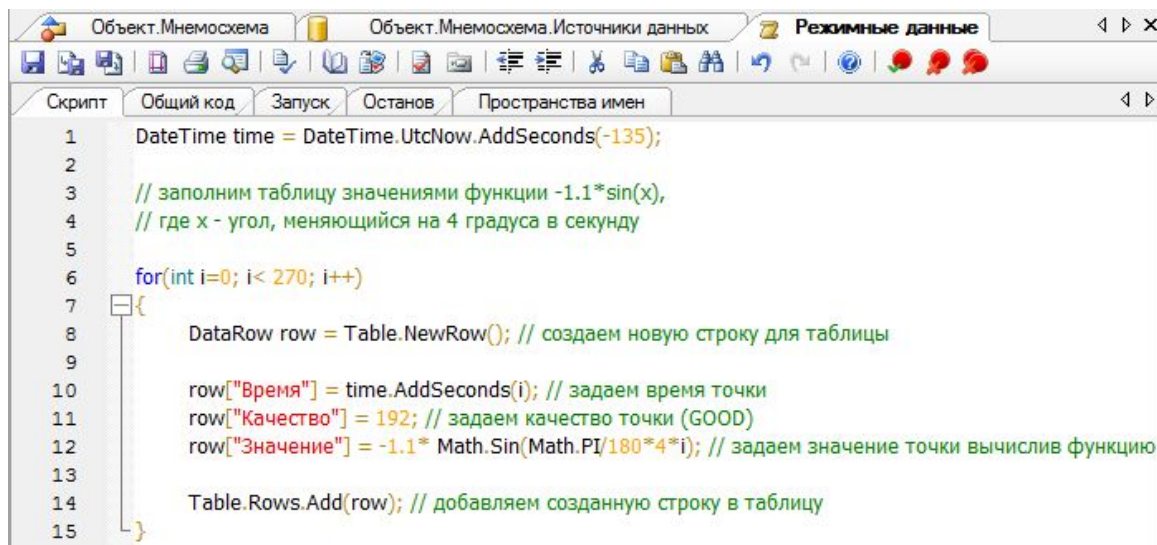
```
    row["Качество"] = 192; // задаем качество точки (GOOD)
```

```
    // задаем значение точки, вычислив данную функцию
```

```
    row["Значение"] = -1.1 * Math.Sin(Math.PI/180*4*i);
```

```
    Table.Rows.Add(row); // добавляем созданную строку в таблицу
```

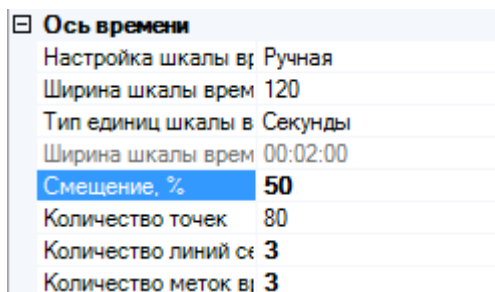
```
}
```

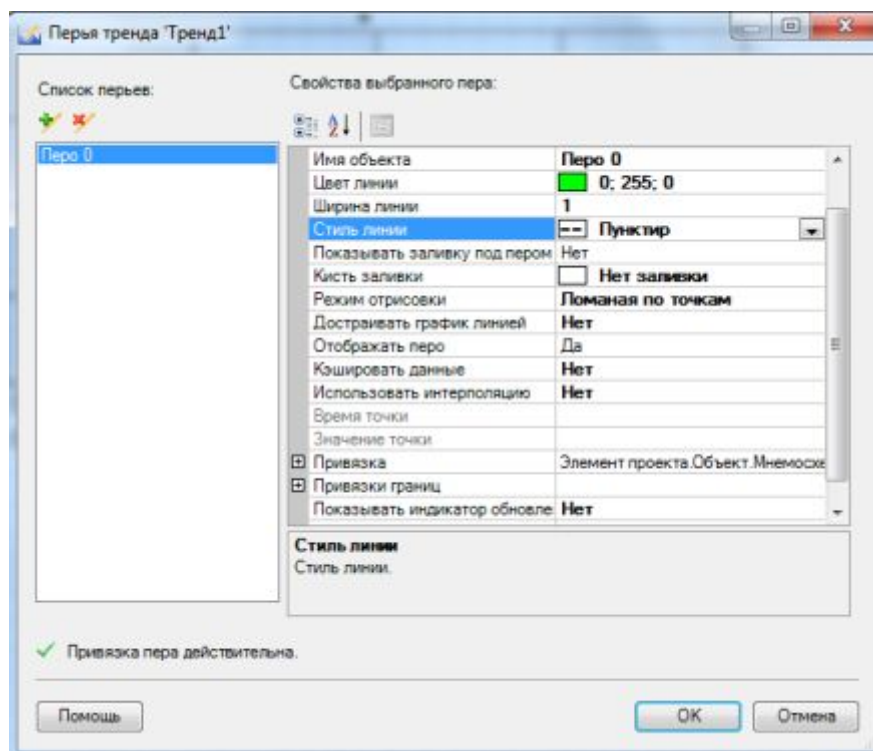
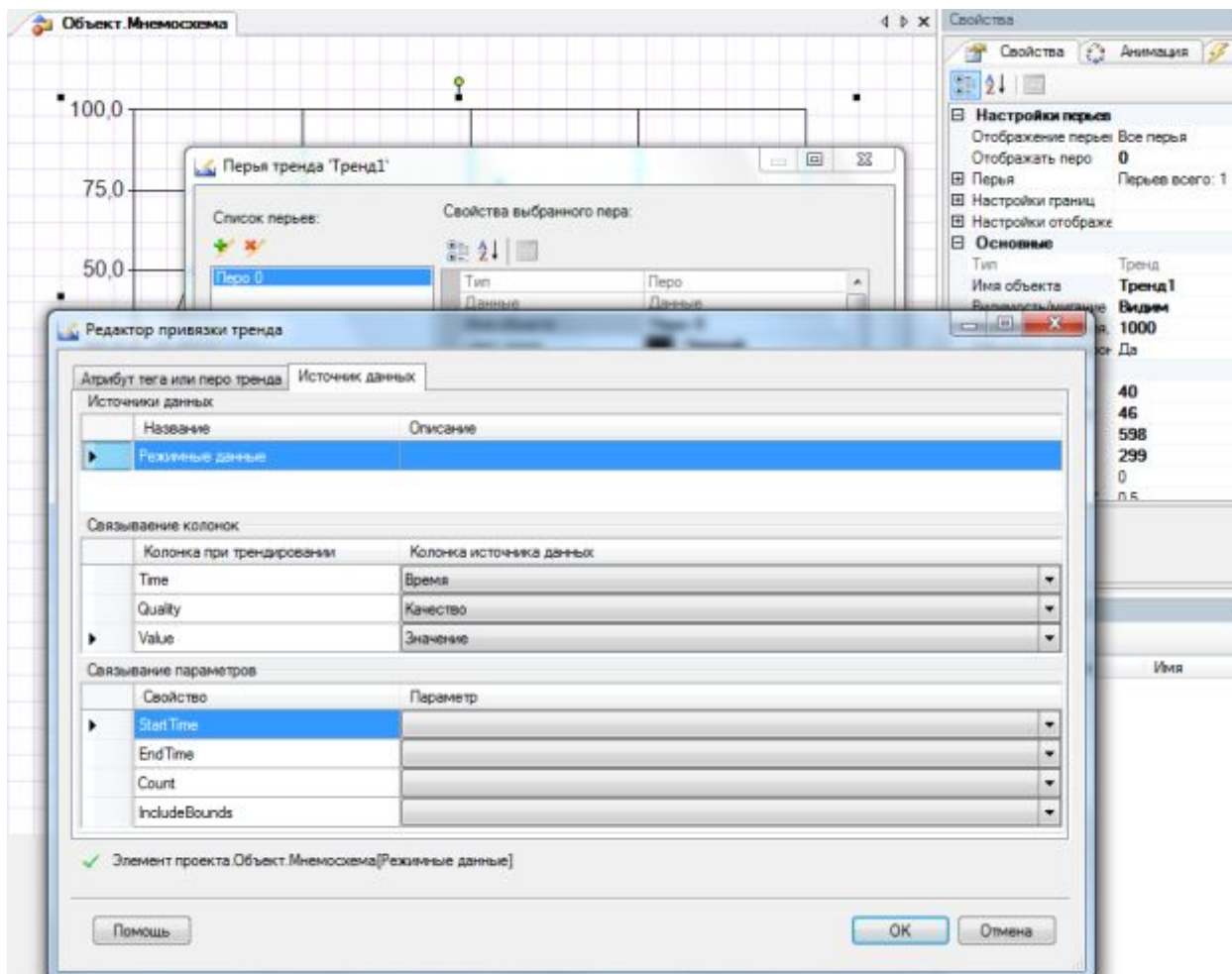


- Закрывать редактор скриптов, окно **Источники данных**, не забыв сохранить изменения.

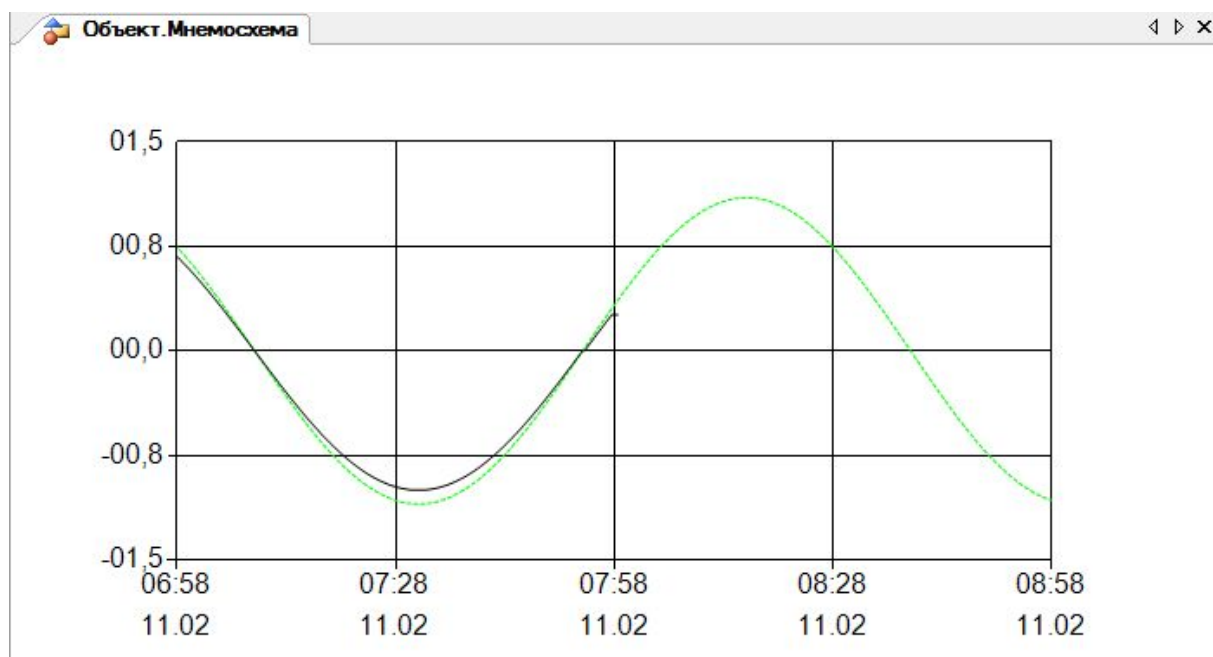
## 2 Для визуализации режимного тренда создать примитив **Тренд**:

- Разместить на мнемосхеме графический примитив **Тренд** (наименование примитива **Тренд1**)
- Открыть диалоговое окно **Перья тренда** и добавить перо для режимных данных
- В окне редактора привязки тренда перейти на вкладку **Источники данных**, выбрать источник **Режимные данные** и привязать колонки источника данных следующим образом: **Время** – **Time**, **Качество** – **Quality**, **Значение** – **Value**
- В качестве «реальных» данных создать для объекта **Объект** тег-эмулятор **Синус** и задать для его визуализации перо в тренде **Тренд1**
- Установить свойство пера **Достраивать график линией** – **Нет**, задать цвет, тип линии и другие свойства пера
- Установить свойство тренда **Смещение** – **50%**.



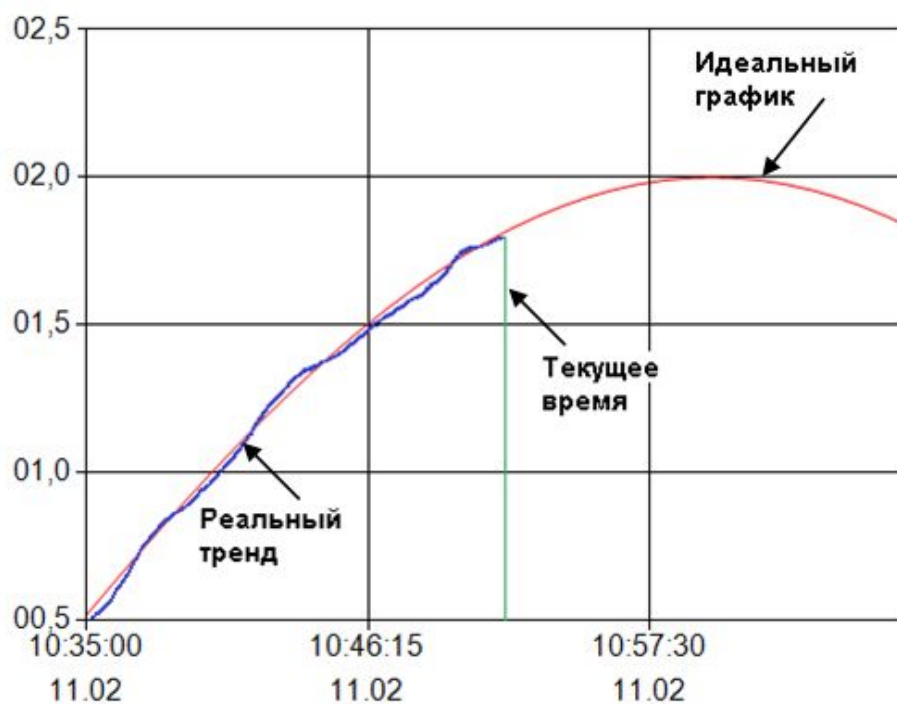


- 3 Запустить созданный проект на имитацию. В результате получим следующий график (зеленый – режимные данные, черный – синус)



#### 6.7.2 Полномасштабный режимный тренд

В полномасштабном режимном тренде временной промежуток шкалы графика тренда фиксирован и задается таким образом, чтобы идеальный график занимал весь график. Тренд регистрируемых данных появляется в процессе регистрации параметров на временном отрезке от начала шкалы тренда до текущего времени.





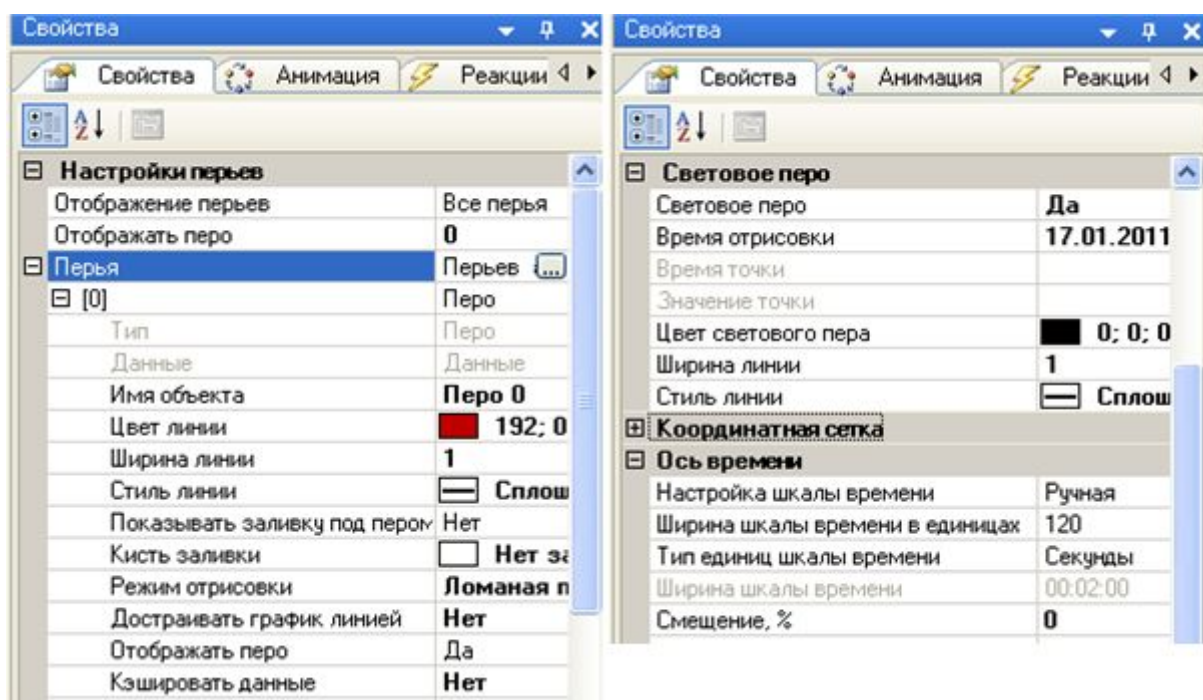
## 6.7.2.1 Настройка тренда

Для визуализации полномасштабного режимного тренда свойства графического примитива **Тренд** или вида **Аналитический тренд** необходимо настроить следующим образом:

- **Режим светового пера** включить
- **Достраивание графика линией** отключить для пера режимного тренда
- **Кеширование** у всех перьев с регистрируемыми данными отключить
- **Начальную точку** задать в соответствии с начальным временем режимных данных
- **Ширину шкалы времени** задать в соответствии с требуемым интервалом

В случае использования примитива **Тренд** для визуализации режимного тренда следует установить значения следующих свойств:

- **Световое перо/Световое перо** – Да
- **Перья/Достраивать график линией** – Нет
- **Перья/Кэшировать данные** – Нет
- **Время отрисовки** (начальная точка) – задать в соответствии с режимными данными
- **Ось времени/Тип единиц шкалы времени** – например, **Секунды**
- **Ось времени/Ширина шкалы времени в единицах** – например, **120** (с учетом типа единицы – **120** секунд или **2** минуты)



## 6.7.2.2 Пример создания тренда

Для созданных в проекте **DataRate** объекта **Объект** и его вида **Аналитический тренд** разработаем **полномасштабный режимный тренд**.

**Визуализацию тренда** реализуем на основе вида **Аналитический тренд** с двумя перьями:

- Режимные данные. Задаются с помощью источника данных **Sql коннектор**
- Регистрируемые данные. Задаются с помощью тега-эмулятора **Синус**.

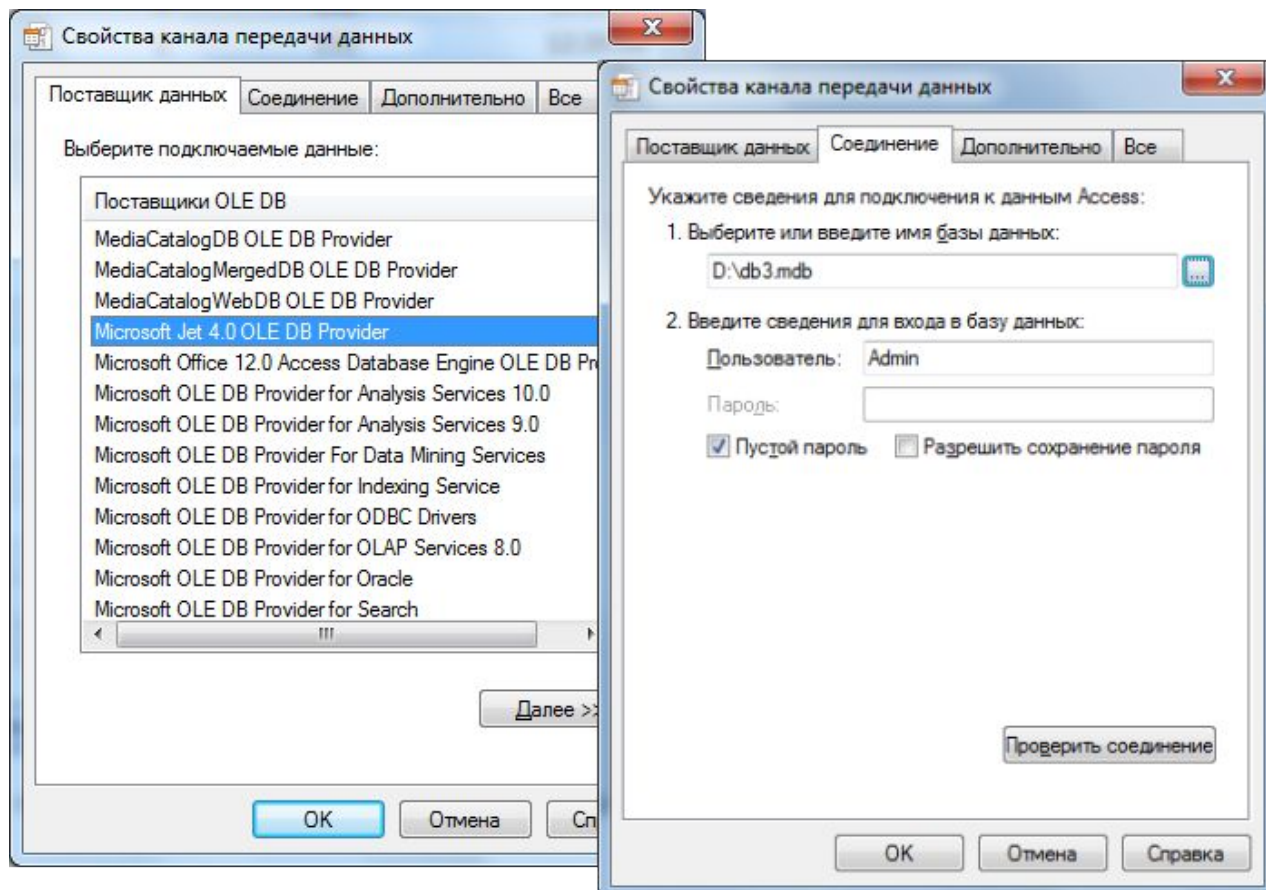
Для создания данного тренда следует:

1 Задать режимные данные:

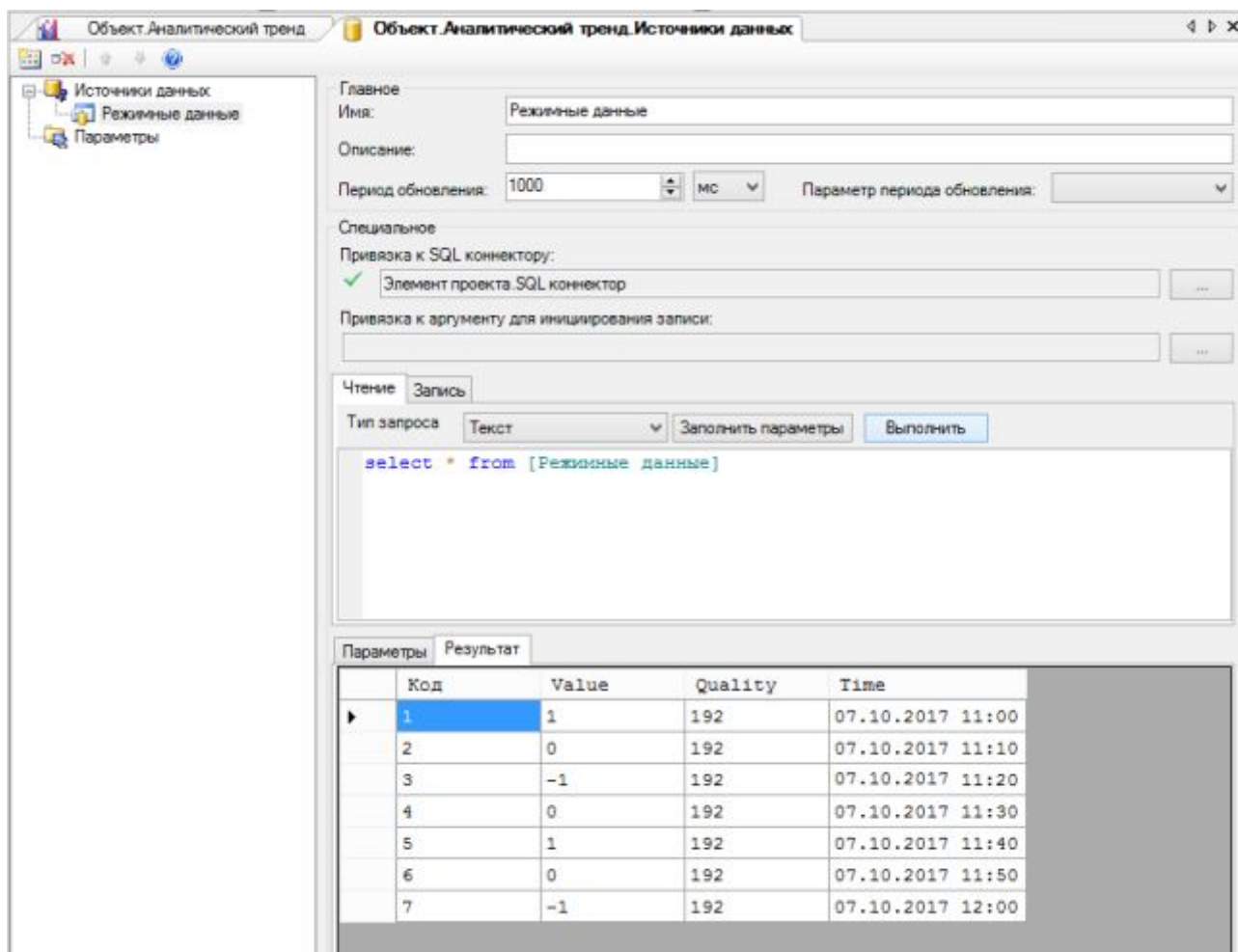
- Создать **Sql коннектор** для получения режимных значений из базы данных **Access** (файл **db3.mdb**). БД содержит таблицу **Режимные данные**, состоящую из 3-х колонок:
  - **Value** – значения точек
  - **Quality** – качества точек
  - **Time** – временные метки точек. **Временные метки всегда должны быть заданы в универсальном формате!**

Код	Value	Quality	Time
1	1	192	07.10.2017 11:00:00
2	0	192	07.10.2017 11:10:00
3	-1	192	07.10.2017 11:20:00
4	0	192	07.10.2017 11:30:00
5	1	192	07.10.2017 11:40:00
6	0	192	07.10.2017 11:50:00
7	-1	192	07.10.2017 12:00:00
8	0	192	07.10.2017 12:10:00
9	1	192	07.10.2017 12:20:00
10	0	192	07.10.2017 12:30:00
11	-1	192	07.10.2017 12:40:00
12	0	192	07.10.2017 12:50:00
13	1	192	07.10.2017 13:00:00
14	0	192	07.10.2017 13:10:00
15	-1	192	07.10.2017 13:20:00
16	0	192	07.10.2017 13:30:00
17	1	192	07.10.2017 13:40:00
18	0	192	07.10.2017 13:50:00
19	-1	192	07.10.2017 14:00:00
20	0	192	07.10.2017 14:10:00
21	1	192	07.10.2017 14:20:00
22	0	192	07.10.2017 14:30:00
23	-1	192	07.10.2017 14:40:00
24	0	192	07.10.2017 14:50:00
25	1	192	07.10.2017 15:00:00

- Связать **Sql коннектор** с базой данных режимных значений



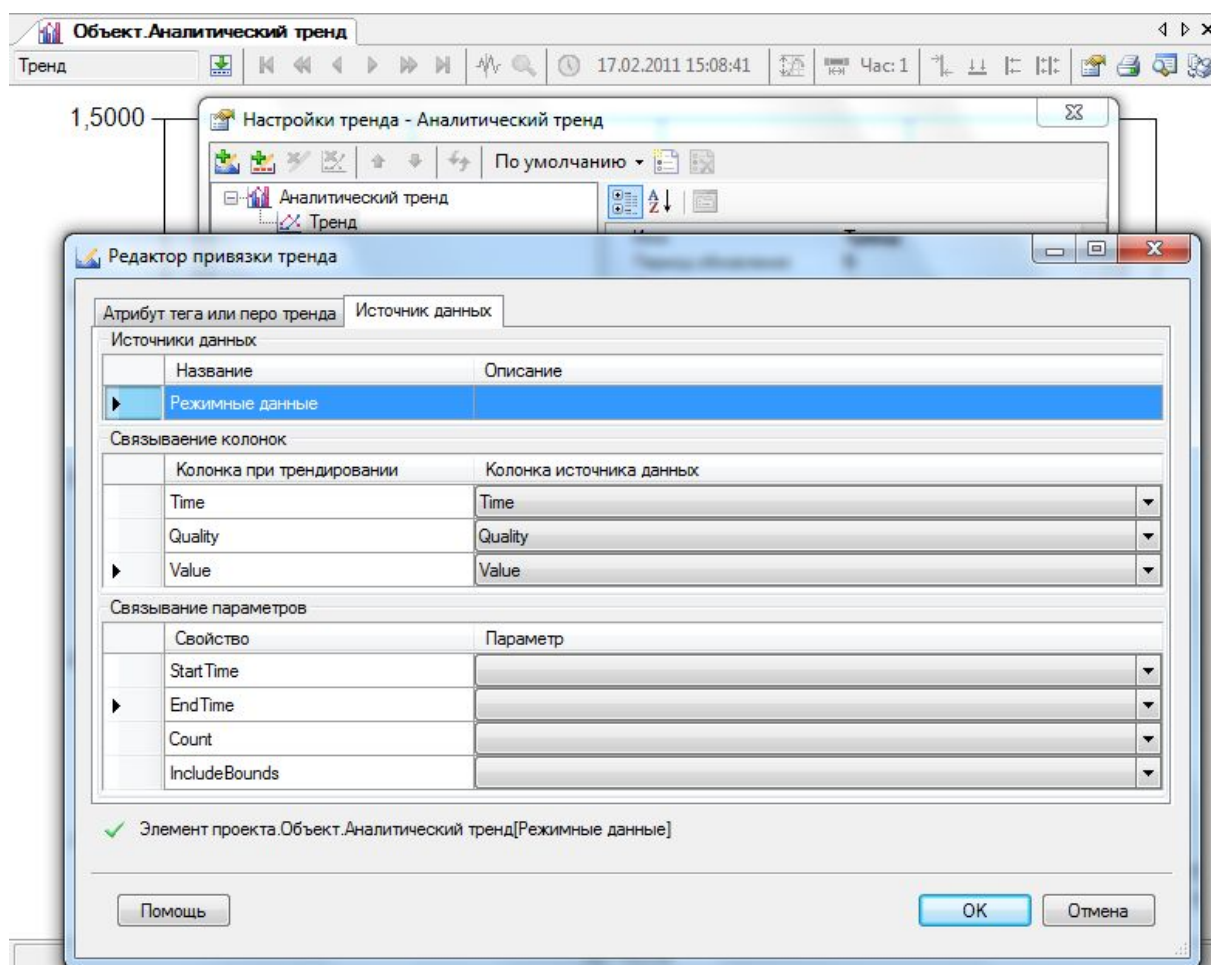
- Для вида **Аналитический тренд** создать источник данных типа **Sql коннектор** с названием **Режимные данные**. Привязать источник данных к **Sql коннектору** и задать **sql-запрос** к БД с режимными данными



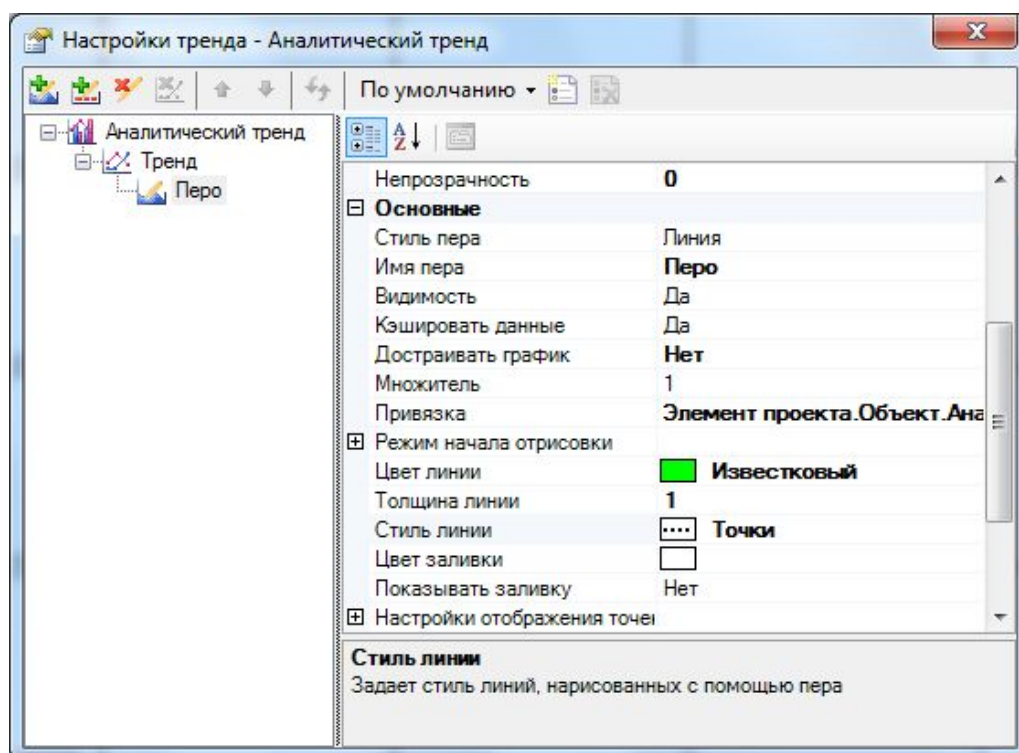
- Закреть окно **Источники данных**, сохранив изменения. Режимные данные готовы.

### 2 Добавить перья в аналитический тренд:

- Открыть окно **Настройки тренда** и добавить новое перо, привязав его к источнику данных **Режимные данные** следующим образом:

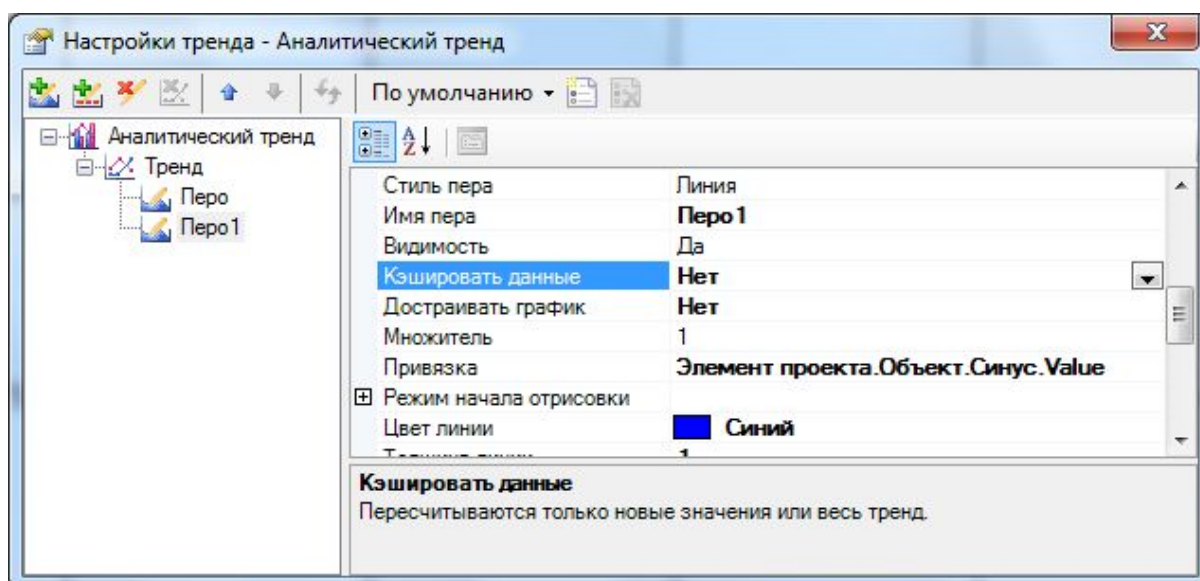


- Задать цвет, тип линии и другие свойства пера. Свойство **Достраивать график линией** установить – **Нет**, свойство **Отображать световое перо** – **Да**. Сохранить изменения

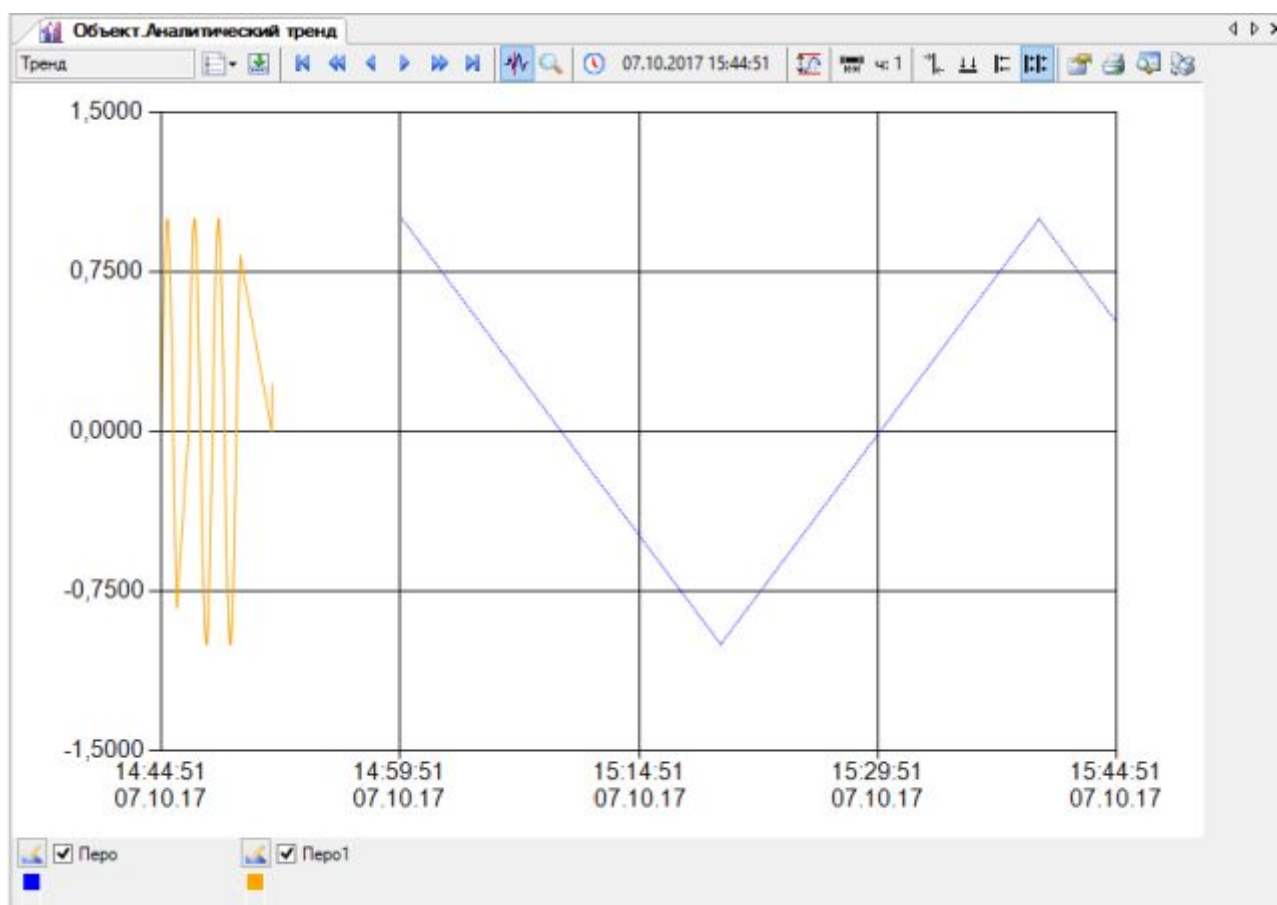




- В качестве "реальных" данных создать тег-эмулятор **Синус**. Добавить в аналитический тренд новое перо, привязав его к тегу **Синус**.
- Задать свойства пера **Кэшировать данные** и **Достраивать график** – **Нет**!



- 3 Запустить созданный проект на имитацию. В результате получим следующий график (зеленый – режимные данные, синий – синус)





## 6.8 Доступ к трендам из скрипта

Для работы с Менеджером трендов из скрипта предназначен вспомогательный статический класс **HistoricalServiceHelper**.

Метод этого класса **ReadRaw** позволяет читать значения из трендов, а методы **Append**, **AppendRange** – записывать значения в тренды.

Для того чтобы использовать методы класса **HistoricalServiceHelper**, в закладке **Пространства имен** редактора скриптов добавьте строку:

[Krug.ObjectModel.HistoricalServiceInterfaces](#)

### 6.8.1 Чтение значений из трендов

Для чтения значений атрибута тега из архивного (исторического) тренда выбранного объекта за определенный временной интервал предназначена функция **ReadRaw**.

Функция **ReadRaw** возвращает список значений типа **AttributeValue** и вызывается скрипте одним из следующих способов:

- **с указанием объекта** (идентификатор **ObjectID**), которому принадлежит трендируемый атрибут тега

[HistoricalServiceHelper.ReadRaw\(  
Client, ObjectID, TagName, AttributeName, SettingGroupName,  
StartTime, EndTime, IncludeBounds\);](#)

где **Client** – владелец скрипта;

**ObjectID** – идентификатор объекта, которому принадлежит трендируемый атрибут тега (тип **Guid**);

**TagName** – название тега, которому принадлежит трендируемый атрибут тега;

**AttributeName** – название трендируемого атрибута, у которого запрашивается история;

**SettingsGroupName** – название группы настроек, которая используется при трендировании атрибута тега;

**StartTime** – начальное время интервала запроса данных (тип **DateTime**);

**EndTime** – конечное время интервала запроса данных (тип **DateTime**);

**IncludeBounds** – параметр определяющий, включать ли в выборку данных граничные точки (тип **bool**, значение **true** – включать).

- **с указанием информационного пути** (идентификатор **InfoPath**) к объекту, которому принадлежит трендируемый атрибут тега

[HistoricalServiceHelper.ReadRaw\(  
Client, InfoPath, TagName, AttributeName, SettingGroupName,  
StartTime, EndTime, IncludeBounds\);](#)

где **InfoPath** – информационный путь к объекту, которому принадлежит трендируемый атрибут тега. Путь может быть абсолютный или относительный.

- **без указания объекта**, которому принадлежит трендируемый атрибут тега. В этом случае трендируемый атрибут ищется у текущего объекта, которому принадлежит скрипт.

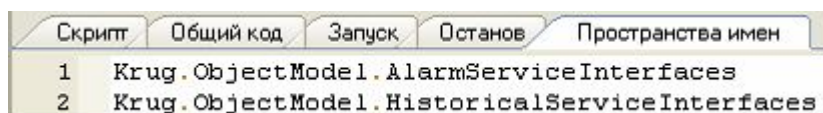
[HistoricalServiceHelper.ReadRaw\(  
Client, TagName, AttributeName, SettingGroupName,  
StartTime, EndTime, IncludeBounds \);](#)

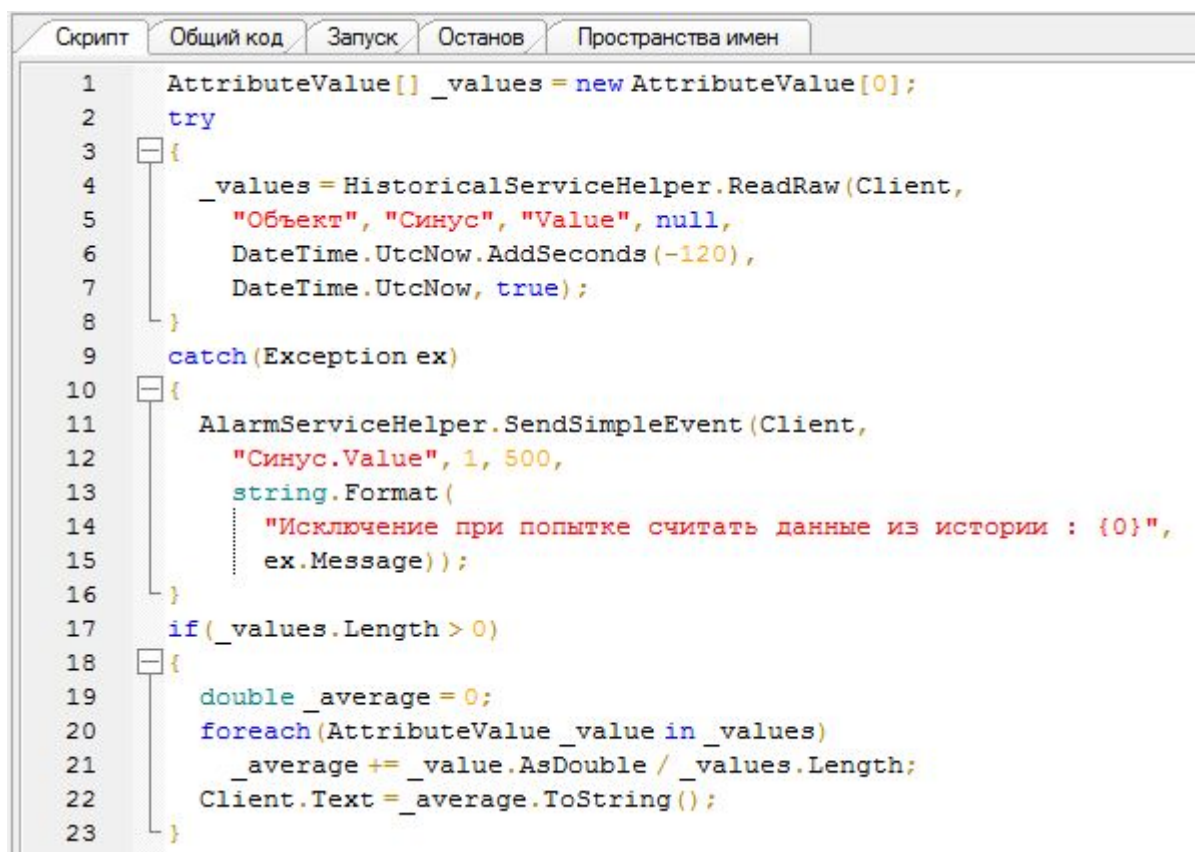
## Пример. Чтение значений исторического тренда

Данный пример демонстрирует чтение значений атрибута **Value** затрендрованного тега **Синус** за интервал времени в 2 минуты от текущего времени. На основе полученных значений атрибута тега вычисляется среднее значение и производится вывод результата в графический примитив **Надпись** или **Текст**. В случае ошибки получения данных из тренда формируется сообщение в протокол событий. Получение значений и вычисление среднего реализуются с помощью скрипта, назначенного на анимацию свойства **Текст** графического примитива. Код скрипта приведен ниже:

```
AttributeValue[] _values = new AttributeValue[0];
try
{
    _values = HistoricalServiceHelper.ReadRaw(Client,
        "Объект",
        "Синус",
        "Value",
        null,
        DateTime.UtcNow.AddSeconds(-120),
        DateTime.UtcNow, true);
}
catch(Exception ex)
{
    AlarmServiceHelper.SendSimpleEvent(Client,
        "Синус.Value" , 1, 500,
        string.Format(
            "Исключение при попытке считать данные из истории : {0}",
            ex.Message));
}
if(_values.Length > 0)
{
    double _average = 0;
    foreach(AttributeValue _value in _values)
        _average += _value.AsDouble / _values.Length;
    Client.Text = _average.ToString();
}
```

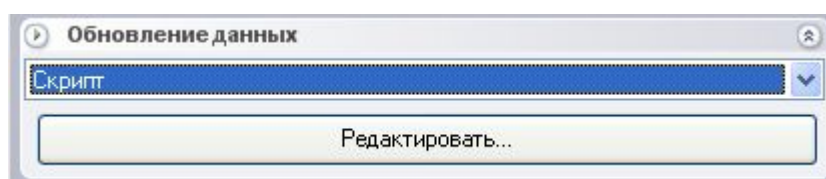
Скрипт в Интегрированной среде разработки **DataRate**





### 6.8.2 Обновление данных

Обновление данных пера менеджера трендов во время исполнения проекта возможно для примитивов **Тренд** и **Табличный тренд**. Для этого необходимо во вкладке **Анимации** примитива **Тренд** активизировать анимацию **Обновление данных**. Анимация **Обновление данных** может задаваться скриптом или простым преобразованием.



#### Пример

Рассмотрим, как при помощи скрипта увеличить в 10 раз значение атрибута тега пера примитива **Тренд**. Для созданных в проекте объекта **Объект**, вида **Мнемосхема** и примитива **Тренд**:

- 1 Создать аргумент с привязкой к свойству **Данные(Data)** пера примитива **Тренд**
- 2 Задать для примитива **Тренд** анимацию **Обновление данных** и выбрать тип анимации **Скрипт**
- 3 Добавить следующий код в скрипт:
 

```

IList<AttributeValue> values = (IList<AttributeValue>)Data;
for(int i=0; i<values.Count; i++)
{
    AttributeValue val = values[i];
    val.SetValue(val.AsDouble * 10, val.LastChangingTime, val.Quality);
    values[i] = val;
}
      
```

### 6.8.3 Запись значений в тренд

#### Функция Append

Функция **Append** предназначена для добавления нового значения в конец тренда выбранного атрибута тега. Функция не возвращает значений и вызывается в скрипте одним из следующих способов:

- **с указанием объекта** (идентификатор **ObjectID**), которому принадлежит трендируемый атрибут тега

```
HistoricalServiceHelper.Append(
    Client,
    ObjectID,
    TagName,
    AttributeName,
    SettingGroupName,
    Value
);
```

где

**Client** – владелец скрипта

**ObjectID** – идентификатор объекта, которому принадлежит трендируемый атрибут тега (тип **Guid**)

**TagName** – название тега, которому принадлежит трендируемый атрибут

**AttributeName** – название трендируемого атрибута, в тренд которого записывается значение

**SettingsGroupName** – название группы настроек, с которой трендируется выбранный атрибут тега

**Value** – значение атрибута, которое добавляется в конец тренда (типа **AttributeValue**).

- **с указанием информационного пути** (идентификатор **InfoPath**) к объекту, которому принадлежит трендируемый атрибут тега

```
HistoricalServiceHelper.Append(
    Client,
    InfoPath,
    TagName,
    AttributeName,
    SettingGroupName,
    Value
);
```

где

**InfoPath** – информационный путь к объекту, которому принадлежит трендируемый атрибут тега. Путь может быть абсолютный или относительный.

- **без указания объекта**, которому принадлежит трендируемый атрибут тега. В этом случае трендируемый атрибут ищется у текущего объекта, которому принадлежит скрипт.

```
HistoricalServiceHelper.Append(
    Client,
    TagName,
    AttributeName,
    SettingGroupName,
    Value
);
```

**Пример. Запись значения в тренд**

В примере случайное значение `_value` записывается в конец тренда атрибута **Value** тега **Простой тег** объекта **Объект**.

```

Скрипт  Общий код  Запуск  Останов  Пространства имен
1  Krug.ObjectModel.AlarmServiceInterfaces
2  Krug.ObjectModel.HistoricalServiceInterfaces

Скрипт  Общий код  Запуск  Останов  Пространства имен
1  Random _rnd = new Random();
2  AttributeValue _value = new AttributeValue(_rnd.NextDouble());
3  try
4  {
5      HistoricalServiceHelper.Append(Client,
6          "Объект", "Простой тег", "Value", null, _value);
7  }
8  catch (Exception ex)
9  {
10     AlarmServiceHelper.SendUserEvent(
11         Client, "Простой тег", "Value", 1, 500,
12         string.Format(
13             "Исключение при попытке записать значение в историю : {0}",
14             ex.Message));
15 }

```

**Функция AppendRange**

Функция **AppendRange** предназначена для добавления списка значений в конец тренда выбранного атрибута тега. Функция не возвращает значений.

Вызов функции **AppendRange** аналогичен вызову функции **Append**.

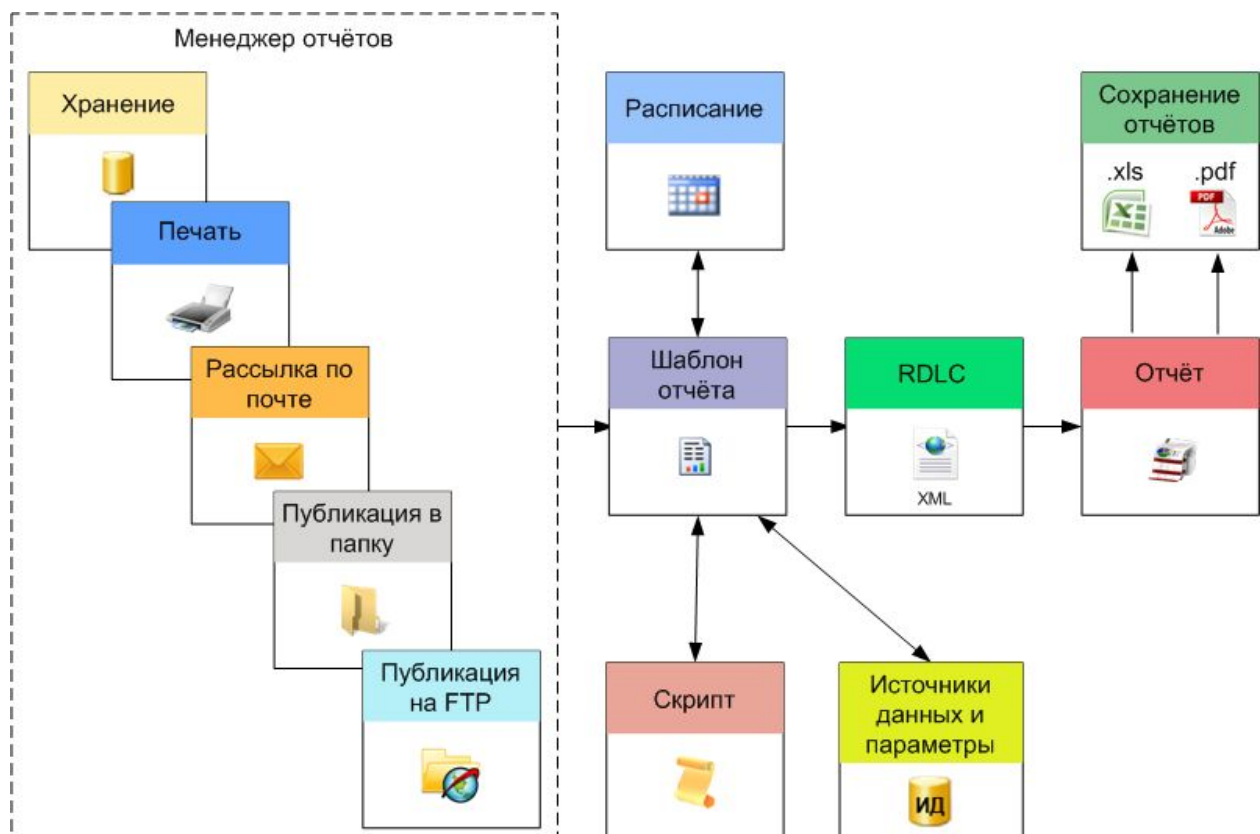




## 7 МОДУЛЬ ОТЧЕТОВ

Модуль отчетов **DataRate** обеспечивает:

- Визуальное создание шаблонов любых необходимых отчетов
- "Привязку" элементов шаблона отчета к различным источникам данных
- Хранение шаблонов отчетов в Среде исполнения
- Создание и хранение «снимка данных» при генерации отчета (по событиям/расписаниям)
- Отображение отчетов в «толстых» (например, **DataRate**) и «тонких» (например, **Web-Контроль**) клиентах
- Печать отчетов (по событиям/расписаниям)
- Рассылку отчетов по электронной почте
- Сохранение отчетов в документах различных форматов: **XML**, **PDF**, **MS Excel** и других
- Настройка отчетов пользователем самостоятельно
- Права доступа к отчетам.



Программные компоненты модуля отчетов:

- **Дизайнер отчетов** – предназначен для визуального создания шаблонов отчетов любой степени детализации. Дизайнер отчетов интегрирован в Среду разработки **DataRate**
- **Менеджер отчетов** – предназначен для создания по запросу или по расписанию самого отчета по заданному шаблону. Менеджер отчетов конфигурируется в Среде разработки и интегрируется в Среду исполнения **DataRate**
- **Компоненты для публикации отчетов** в различные системы – e-mail, web, «толстые клиенты» для просмотра отчетов

- **Расширяемая система источников данных** (теги **DataRate**, теги **WideTrack** и другие) – подключаемые к **Менеджеру отчетов** модули, расширяющие его возможности в получении данных из различных систем
- **Расширяемая система хранилищ данных и шаблонов отчетов.**

Шаблон отчета сохраняется в **DataRate** как **XML**-документ – файл с расширением **RDLC**. Это обеспечивает работу с отчетами, созданными не только с помощью дизайнера отчетов **DataRate**, но и с помощью других средств.

### ВНИМАНИЕ!!!

Работа с модулем отчётов возможна только при наличии соответствующего установленного компонента.

## 7.1 Шаблон отчета

**Шаблон отчета** – это макет документа, создаваемый в дизайнере отчетов. Шаблон отчета является видом объекта (как, например, Мнемосхема) и может размещаться на рабочих столах и отображаться в окне предварительного просмотра.

**Базовые элементы шаблона:** линия, прямоугольник, текст, диаграмма, таблица, рисунок. Элементы можно вкладывать друг в друга и свободно размещать в любом месте отчета.

**Структура шаблона** отчета включает следующие секции: **тело отчета**, **верхний и нижний колонтитулы**. Содержимое колонтитулов повторяется на каждой странице отчета.

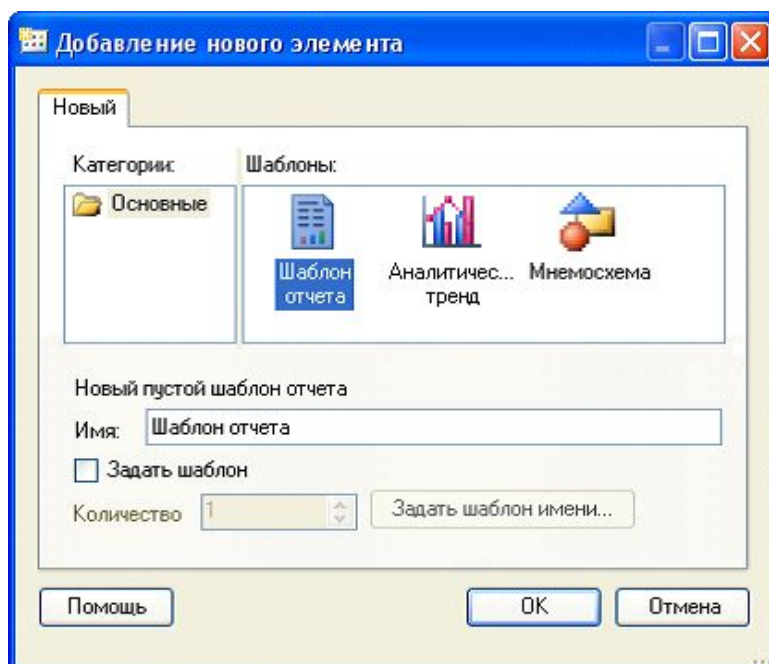
**Два способа создания шаблона отчета:**

- Создать новый (пустой) шаблон отчета и вручную добавить элементы
- Импортировать существующий шаблон отчета (из файла в rdlc-формате).

### Создание нового шаблона отчета

Для создания нового шаблона отчета следует:

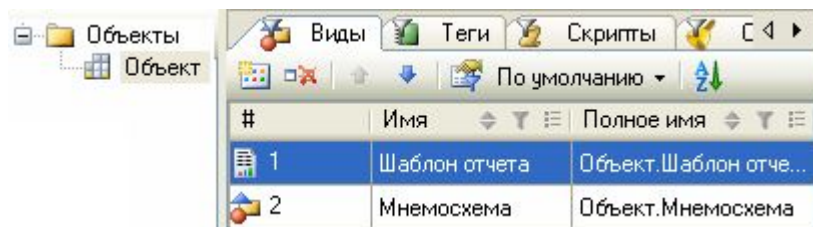
- 1 Создать новый объект (команда **Создать/Объект** в контекстном меню узла **Объекты** дерева проекта)
- 2 В контекстном меню нового объекта выбрать **Создать/Вид**
- 3 В появившемся окне **Добавление нового элемента** выбрать категорию **Основные** и шаблон **Шаблон отчета**
- 4 Задать **Имя** шаблона отчета и нажать **ОК**.



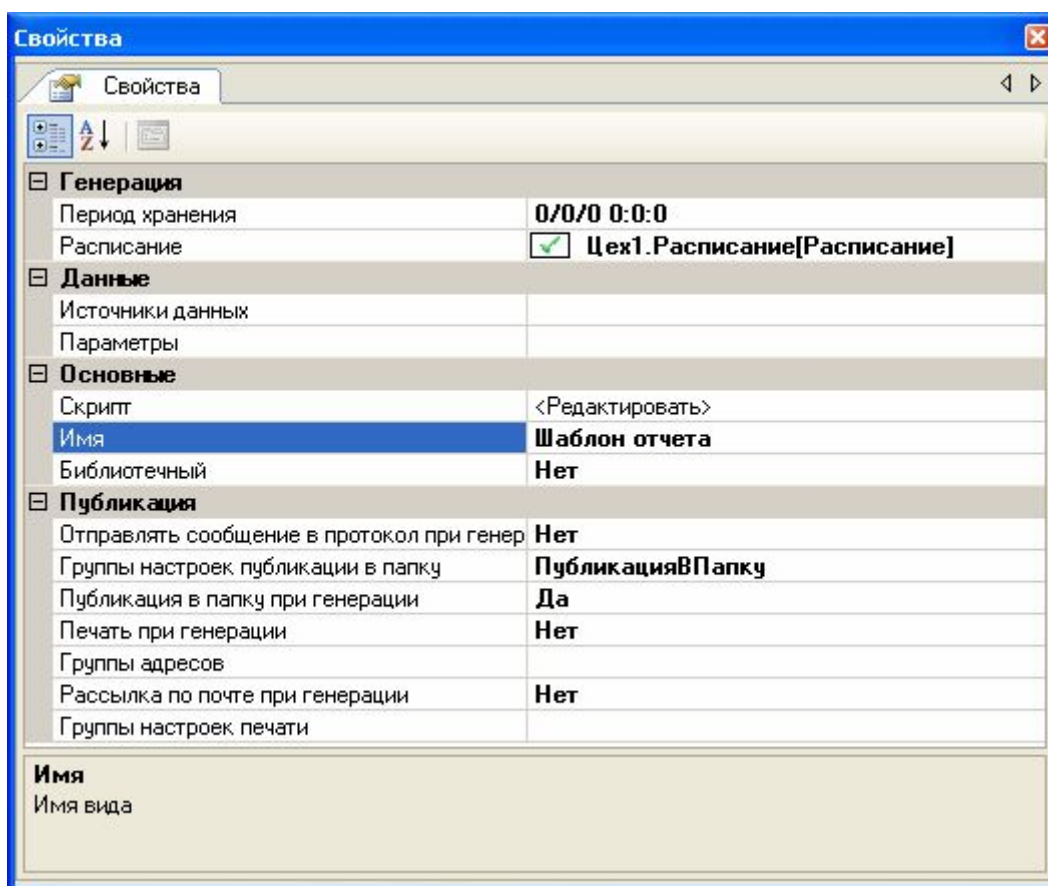
Для создания нескольких шаблонов отчета воспользуйтесь флажком **Задать шаблон**. Введите **Количество** шаблонов отчета и задайте шаблон имени (кнопка **Задать шаблон имени**).

### Свойства отчета в редакторе проекта

Созданный шаблон отчета будет отображен в редакторе проекта.



В окне **Свойства** пользователь может задать **параметры генерации, хранения и публикации отчета**, создаваемого на основе данного шаблона (описание приведено в разделе «Менеджер отчетов»).

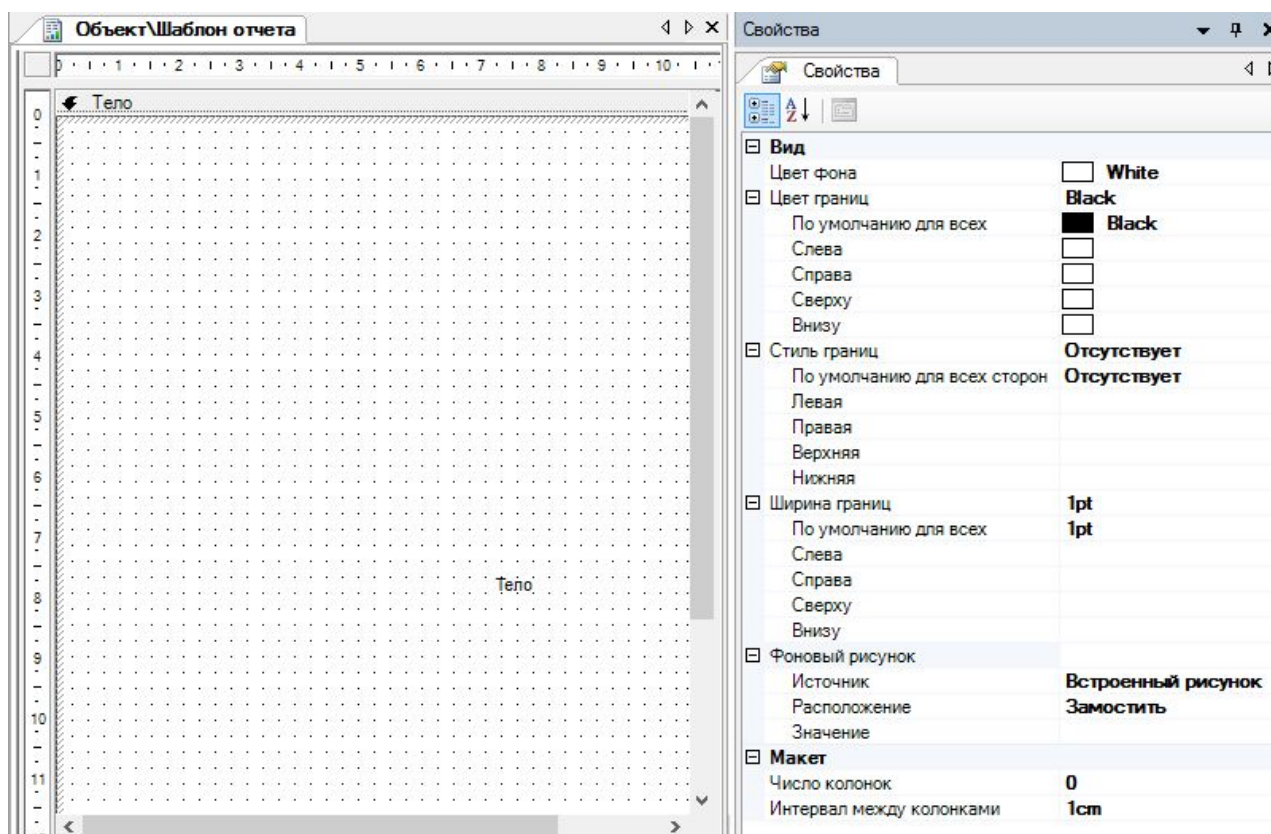


Свойство **Скрипт** предназначено для управления шаблоном отчета из кода скрипта (описание приведено в разделе «Генерация отчета»).

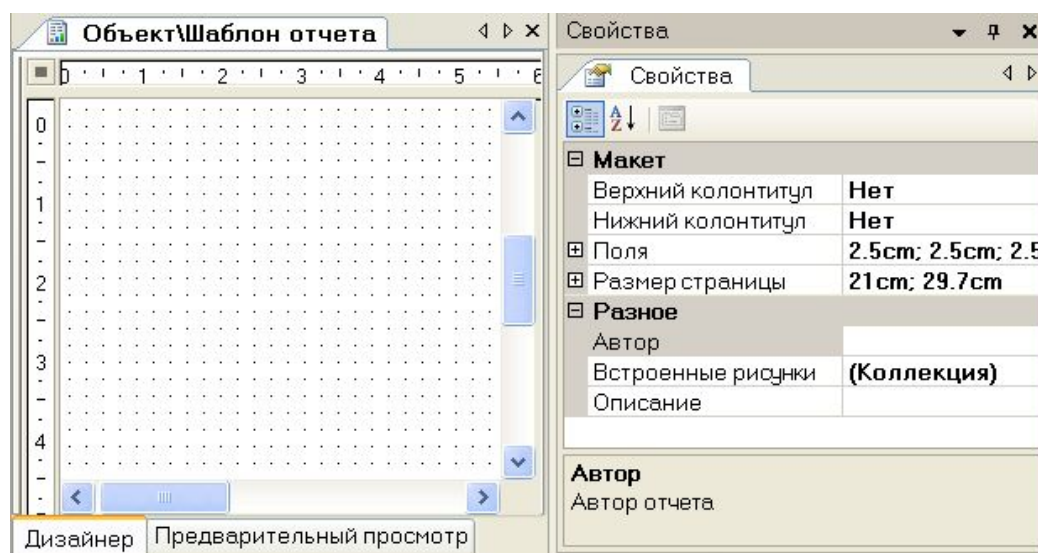
### Свойства отчета в дизайнера отчетов

В дизайнера отчетов можно задать следующие свойства:

- **Параметры границ и фон** отчета можно задать в окне свойств, которое вызывается щелчком левой клавишей мыши в области секции шаблона отчета (например, в теле отчета)



- **Параметры страницы, использование колонтитулов и встроенных рисунков, сведения об авторе** разработки отчета можно задать в окне свойств, которое вызывается щелчком левой клавишей мыши на границе секции шаблона или из контекстного меню дизайнера отчетов по команде **Выбрать отчет**.



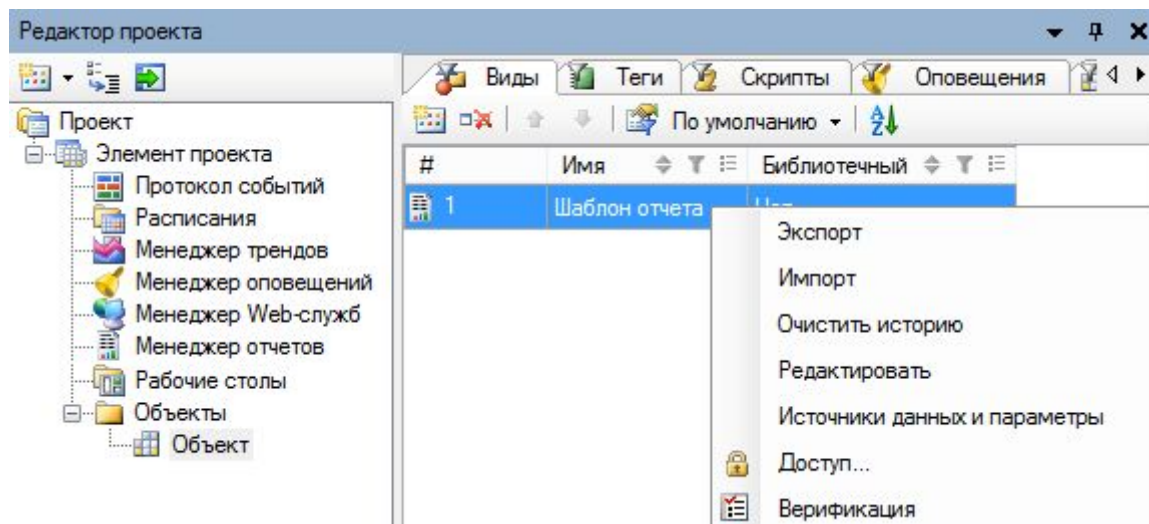
## Контекстное меню отчета в редакторе проекта

**Контекстное меню шаблона** отчета (вызывается щелчком правой клавиши мыши на имени шаблона) кроме пунктов **Вырезать**, **Копировать**, **Вставить**, **Удалить** содержит и специфичные пункты меню:

- **Экспорт** – открывает окно задания файла для экспорта шаблона
- **Импорт** – открывает окно импорта шаблона из файла
- **Очистить историю** – удаляет все снимки данных отчета



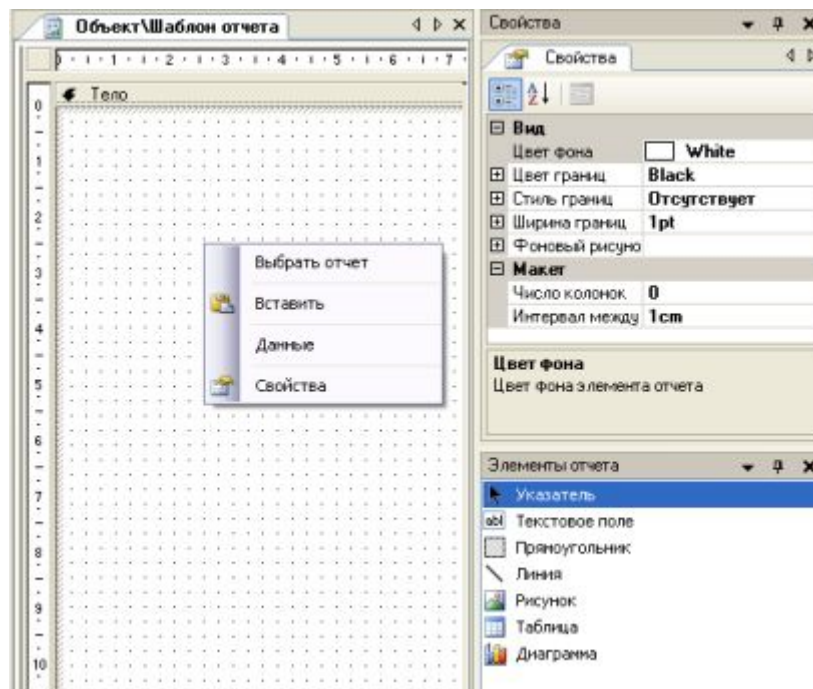
- **Редактировать** – открывает окно дизайнера отчета для редактирования шаблона
- **Доступ** – настройка доступа к шаблону
- **Источники данных и параметры** – открывает окно редактирования источников данных отчета и параметров
- **Просмотр** – открывает шаблон отчета для предварительного просмотра (доступен только в режиме исполнения)



### Контекстное меню отчета в дизайнера отчетов

Контекстное меню дизайнера отчетов (вызывается щелчком правой клавиши мыши в области отчета) содержит пункты меню:

- **Выбрать отчет** – отображает параметры страницы, использование колонтитулов и встроенных рисунков в окне свойств
- **Вставить** – копирует в отчет содержимое буфера обмена
- **Данные** – открывает окно **Источники данных**
- **Свойства** – открывает окно свойств.



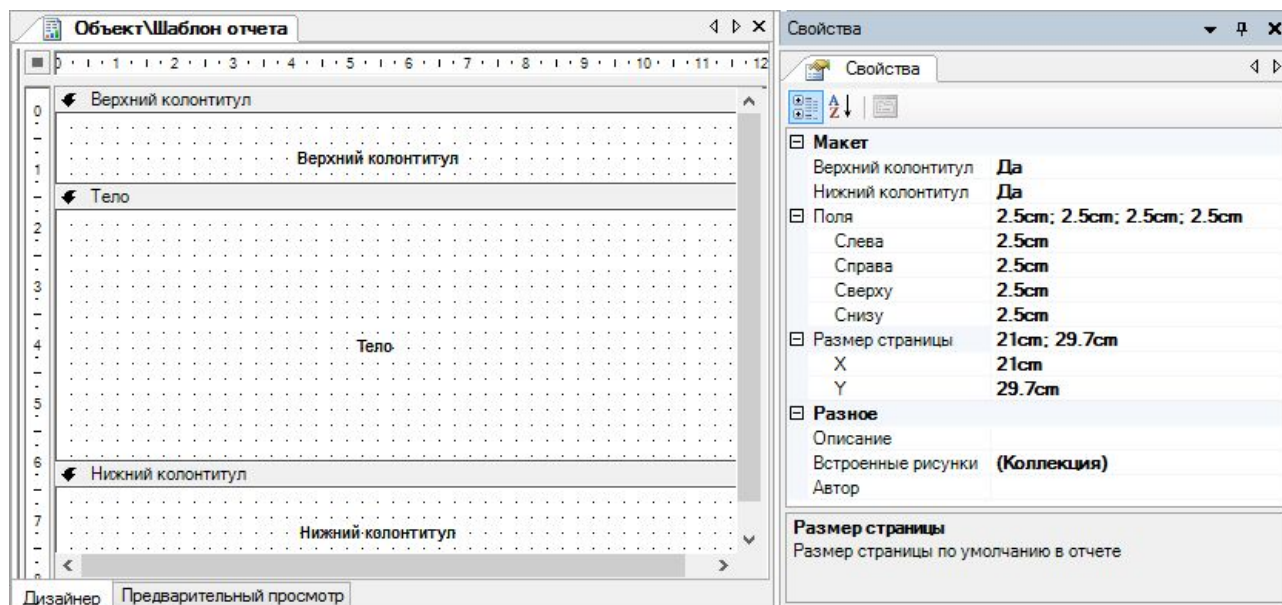
### Создание отчета на основе библиотечного шаблона

Для создания шаблона отчета из библиотечного шаблона следует:

- 1 «Перетащить» шаблон из библиотеки (Библиотека объектов) в проект
- 2 Внести необходимые дополнения и изменения в шаблон.

## 7.1.1 Секции отчета

**Секция отчета** – область страницы отчета: верхний колонтитул, тело отчета (основная часть) и нижний колонтитул.

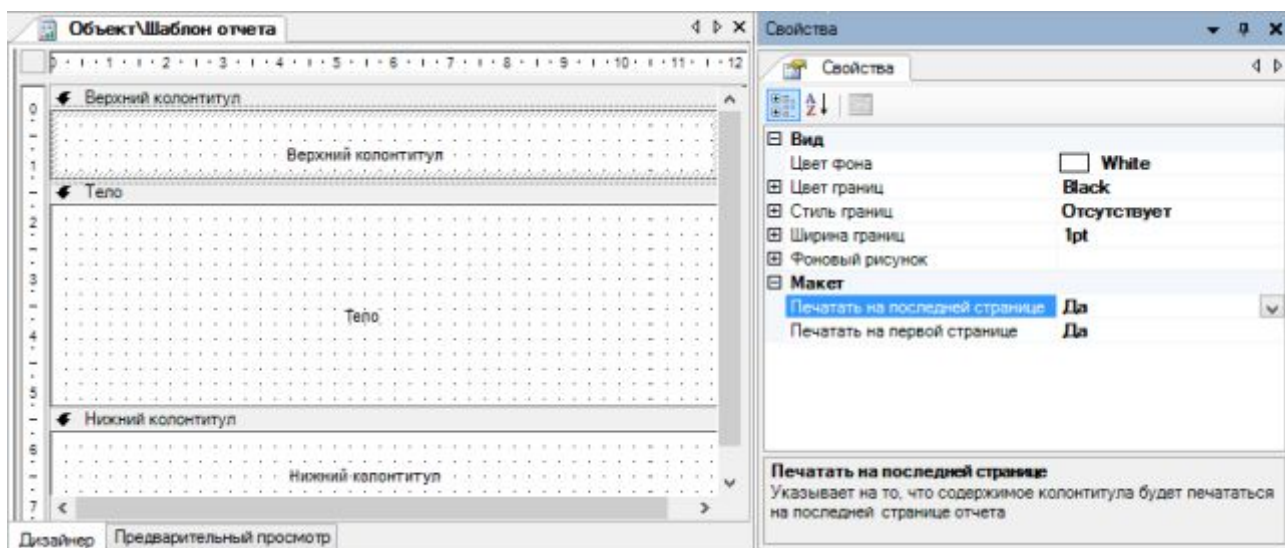


### Колонтитулы

Колонтитулы не являются обязательными секциями шаблона отчета.

Свойства **Верхний колонтитул**, **Нижний колонтитул** определяют наличие или отсутствие колонтитулов на странице.

Управлять отображением колонтитулов на первой и последней странице отчета можно с помощью свойств **Печать на первой странице**, **Печать на последней странице** в окне свойств колонтитула, которое вызывается щелчком левой клавиши мыши в области колонтитула.



Отображение значений и изображений в колонтитулах имеет ряд особенностей. Рассмотрим некоторые из них.

### Добавление номера страницы и даты создания отчета в колонтитул

Для того чтобы отобразить номер и общее количество страниц, создайте в колонтитуле текстовое поле (**Текстовое поле**) и добавьте в его свойство **Значение (Свойства/Данные)** следующее выражение:

= "**Страница**" & Globals.PageNumber & " из " & Globals.TotalPages

Для отображения даты создания отчета добавьте текстовое поле и в его свойстве **Значение** задайте выражение: = Format(Globals.ExecutionTime, "d")

### Отображение значений в колонтитуле

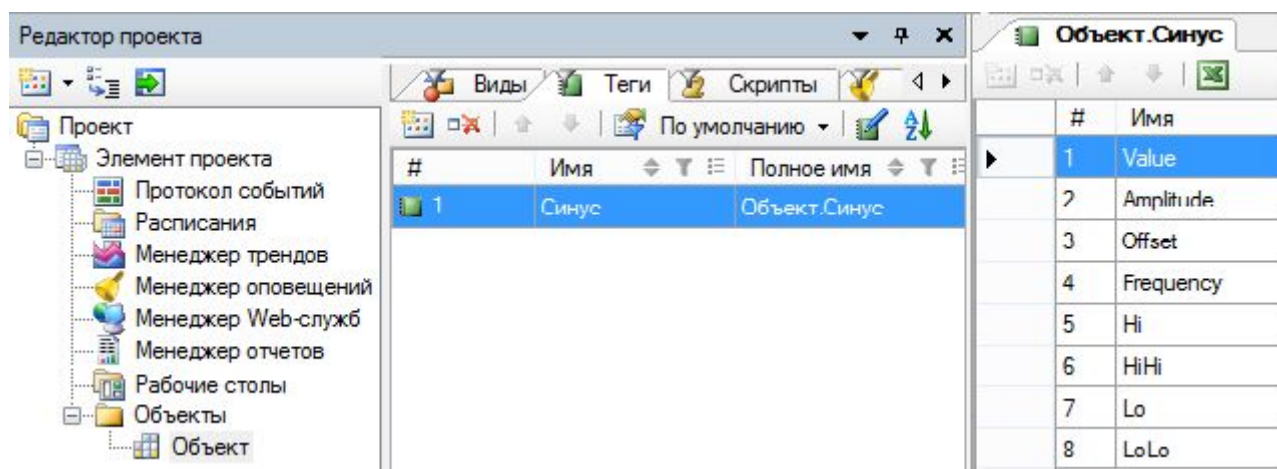
#### ВНИМАНИЕ!


Из текстового поля в колонтитуле нельзя напрямую ссылаться на поле данных. Поэтому не включайте в колонтитуле прямую ссылку на поля из набора данных.

#### Пример

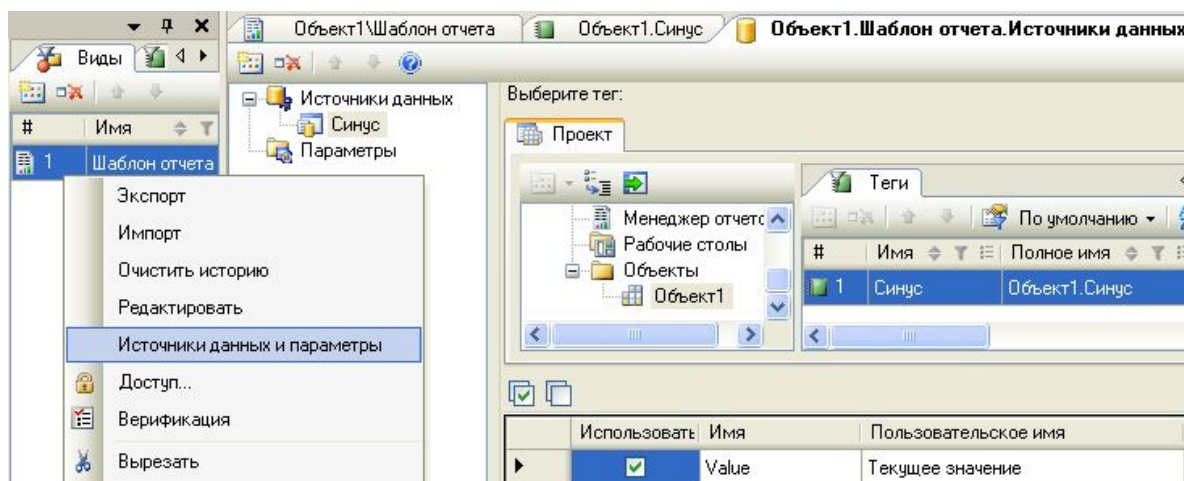
Создадим отчет, в котором выводятся значения тега **Синус**. Значение синуса вычисляется автоматически при каждом обращении к данному тегу и сохраняется в свойстве **Value**. Для создания отчета следует:


- 1 Создать новый объект **Объект1** – команда **Создать/Объект** в контекстном меню узла **Объекты** дерева проекта
- 2 Создать тег **Синус** – в контекстном меню **Объект1** выбрать **Создать/Тег** и на вкладке **Новый** указать шаблон **Синус**



- 3 Создать шаблон отчета – в контекстном меню объекта **Объект1** выбрать **Создать/Вид** и затем **Шаблон отчета**
- 4 Задать источник данных для шаблона отчета – в контекстном меню вида **Шаблон отчета** выбрать **Источники данных и параметры**, а затем добавить (кнопка ) источник данных категории **Атрибут тега**, задать ему имя **Синус** и связать с атрибутом **Value** тега **Синус**

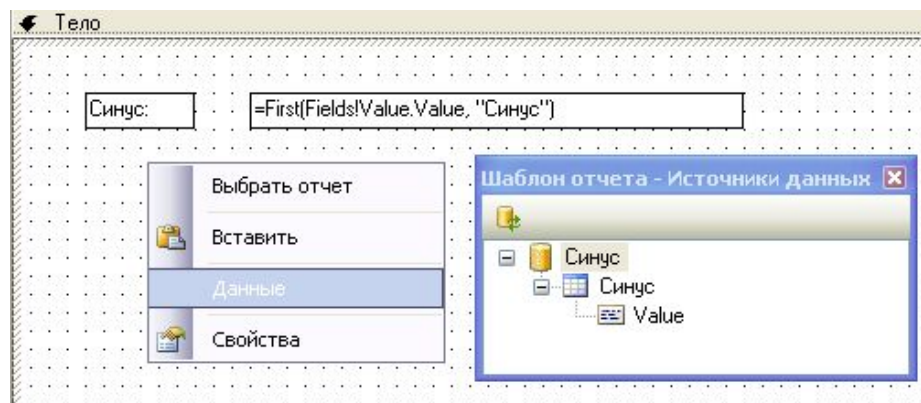




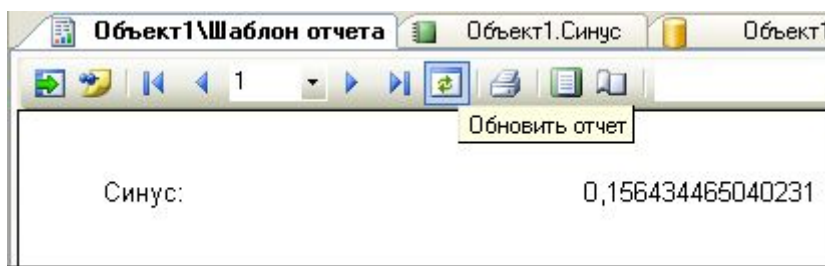
- 5 В теле шаблона отчета создать текстовые поля (кнопка  на панели инструментов среды разработки) со следующими свойствами:

- **Имя** – **TextReportItem** (отображает статический текст **Синус:**)  
**Значение** – **Синус:**
- **Имя** – **TextReportItem1** (отображает значение **Value** тега **Синус**)  
**Значение** – **=First(Fields!Value.Value, "Синус")**

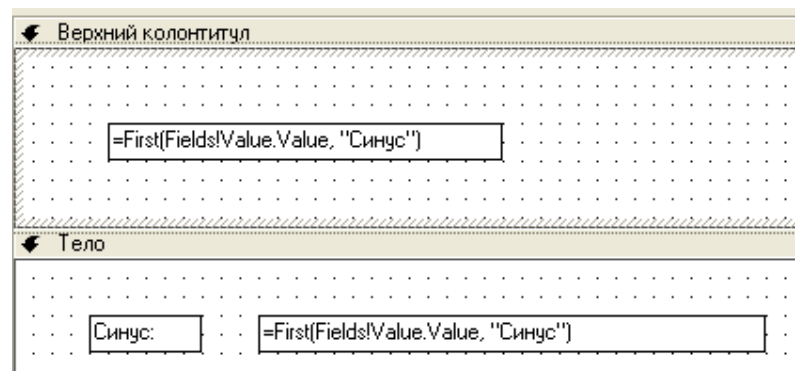
Для задания свойства **Значение** откройте окно **Шаблон отчета - Источники данных** (пункт **Данные** контекстного меню тела отчета) и перетащите из дерева узел **Value** в текстовое поле. В данном примере функция **First** возвращает значение **Value** тега **Синус**.



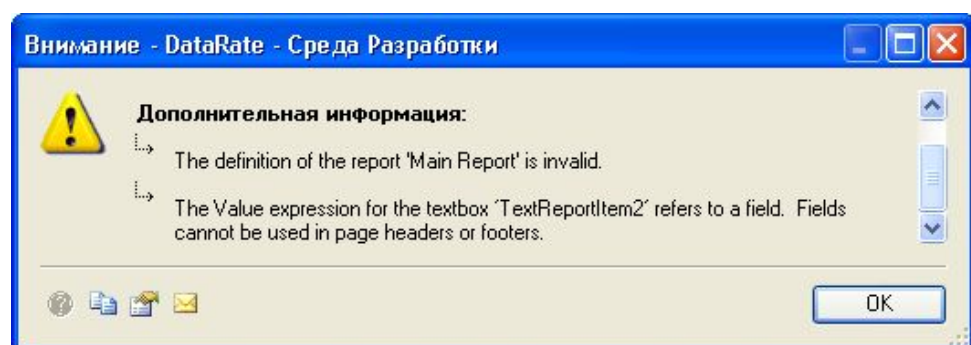
- 6 Запустить проект на имитацию и перейти на закладку **Предварительный просмотр** в дизайнера отчета. Нажимая на кнопку **Обновить отчет**, наблюдать как меняется значение синуса



- 7 Добавить верхний колонтитул с текстовым полем (в примере его имя **TextReportItem2**), свойство **Значение** которого содержит выражение:  
**=First(Fields!Value.Value, "Синус")**

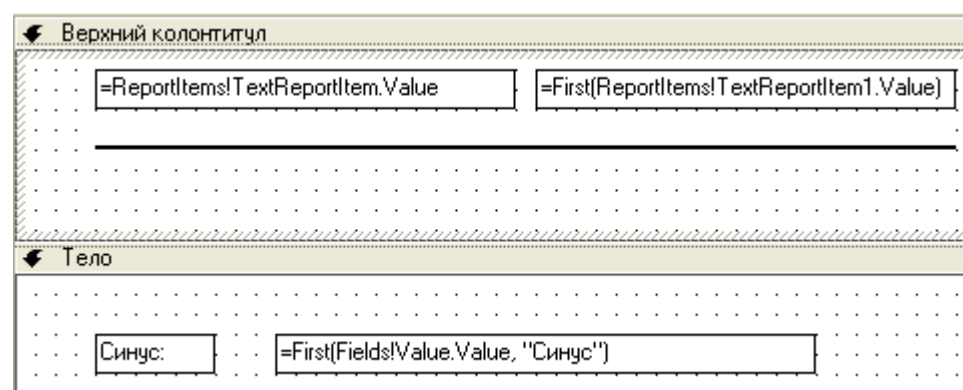


Если выполнить предварительный просмотр шаблона отчета, то результатом будет сообщение об ошибке




- 8 Для того чтобы **правильно отобразить значение в колонтитуле**, следует создать в колонтитуле текстовые поля, которые ссылаются на значения полей тела отчета. В данном примере:

- **Имя** – **TextReportItem2** (отображает значение текстового поля **TextReportItem**)  
**Значение** – **=ReportItems!TextReportItem.Value**
- **Имя** – **TextReportItem3** (отображает значение текстового поля **TextReportItem1**)  
**Значение** – **=First(ReportItems!TextReportItem1.Value)**

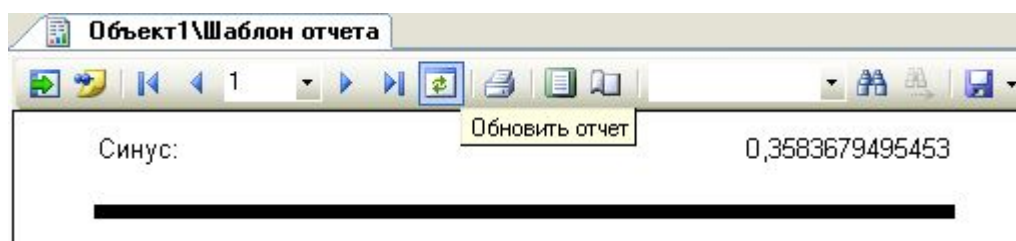


Скроем изображение текстовых полей в теле отчета. Для этого достаточно установить свойство **Видимости/Скрыть** в значение **Да**.

Для того чтобы отделить колонтитул добавим в шаблон отчета горизонтальную линию.

- 9 Запустить проект на выполнение и перейти в режим **Предварительный просмотр** дизайнера отчета. Нажимая на кнопку  – **Обновить отчет**, наблюдаем как в колонтитуле меняется значение синуса.

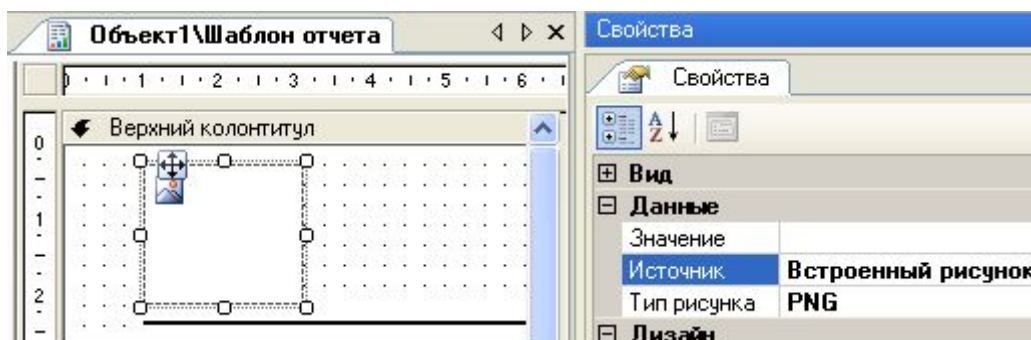




## Отображение рисунка в колонтитуле

### ВНИМАНИЕ!!!

Для размещения рисунка в колонтитуле используйте в качестве источника рисунка – Встроенный рисунок.



## 7.1.2 Элементы отчета

**Элементы отчета** предназначены для отображения в отчете текстов, графических изображений, технологических, экономических и других данных.

Размещение элементов в шаблоне отчета осуществляется с панели **Элементы отчета**.

Открыть панель **Элементы отчета** можно щелчком левой клавиши мыши по кнопке **Элементы отчета** на панели инструментов или в меню **Вид**

### Графические элементы

Для создания визуальных эффектов внутри отчета можно использовать **прямоугольники, линии и изображения**.

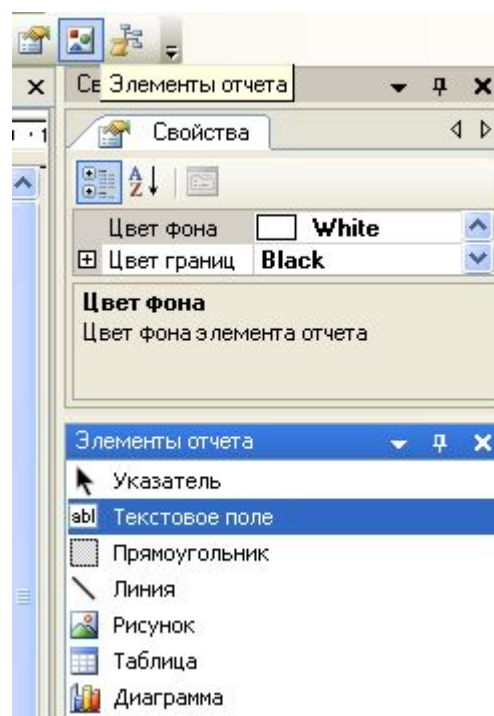
Линии и прямоугольники "не связаны" с данными.

Прямоугольник можно использовать как контейнер для других элементов. При перемещении прямоугольника элементы внутри него перемещаются вместе с ним.

Изображения могут быть статическими или основываться на значениях из наборов данных (значения атрибутов тегов, данные, полученные из базы данных по SQL-запросу, и другие).

### ВНИМАНИЕ!!!

При размещении элементов отчета в прямоугольнике необходимо учесть изменения области этих элементов, которые могут произойти при подготовке



отчета. Например, таблица "развернется" таким образом, чтобы разместить все возвращаемые запросом данные, и передвинет вниз элементы, расположенные ниже таблицы.

Чтобы привязать элемент к месту, можно расположить элемент отчета внутри прямоугольника так, чтобы верхний край элемента находился выше нижнего края таблицы.

Если необходимо зафиксировать элемент отчета, поместите его в отдельном прямоугольнике – прямоугольник не меняет своего положения рядом с областью данных.

### Области данных

Элементы отчета **Таблица** и **Диаграмма**, отображающие значения из наборов данных – это **области данных**.

Области данных могут быть вложены в другие области данных.

Каждая область данных связана с одним набором данных.

При использовании нескольких областей данных в отчете можно установить настройку, чтобы в каждой из областей использовался собственный набор данных.

Если отчет использует только один набор данных, все области данных используют его по умолчанию.

**Пустые области данных.** Если набор данных для области данных не возвращает никаких значений, эта область не отображается при просмотре. Вместо этого подготавливается к просмотру текстовое поле, содержащее значение свойства **Сообщение при отсутствии данных** области данных.

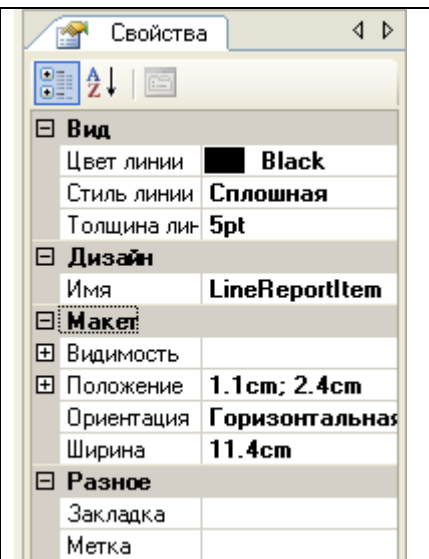
#### 7.1.2.1 Линия

**Линия** — это графический элемент отчета, который может находиться в любом месте страницы.

Для размещения линии в отчете необходимо перенести элемент **Линия** с панели элементов отчета в отчет.

### Свойства

- **Цвет, стиль и толщина линии** – вид линии
- **Имя** – наименование элемента отчета (линии)
- **Видимость/Переключатель** – имя текстового элемента, используемого в качестве переключателя видимости для данного элемента отчета. Щелчок мышкой на назначенном текстовом элементе переключает состояние видимости каждого экземпляра данного элемента отчета. Если элемент, назначенный в качестве переключателя, будет скрыт, данный элемент также станет невидимым
- **Видимость/Скрыть** – указывает на то, что элемент отчета будет скрыт
- **Положение** – Позиция верхнего левого угла элемента отчета относительно контейнера, задается значениями X (по горизонтали) и Y (по вертикали)
- **Ориентация** – Ориентация на плоскости: горизонтальная или вертикальная
- **Ширина** – размер элемента отчета определяется позицией нижнего правого угла, задается значениями по вертикали и горизонтали
- **Закладка** – ассоциированная с данным элементом закладка; может быть использована в качестве параметра для навигации



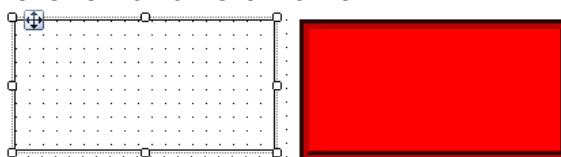
- **Метка** – метка для идентификации элемента отчета

### 7.1.2.2 Прямоугольник

**Прямоугольник** – элемент отчета, который можно использовать как графический элемент и как контейнер, содержащие другие элементы отчета.

Если внутрь прямоугольника поместить другие элементы отчета, то они будут перемещаться вместе с ним. Это позволяет группировать многочисленные текстовые поля и другие элементы отчета в блоки. В прямоугольник, как в контейнер, могут быть добавлены любые элементы отчета: текстовые поля, линии, прямоугольники, таблицы и диаграммы.

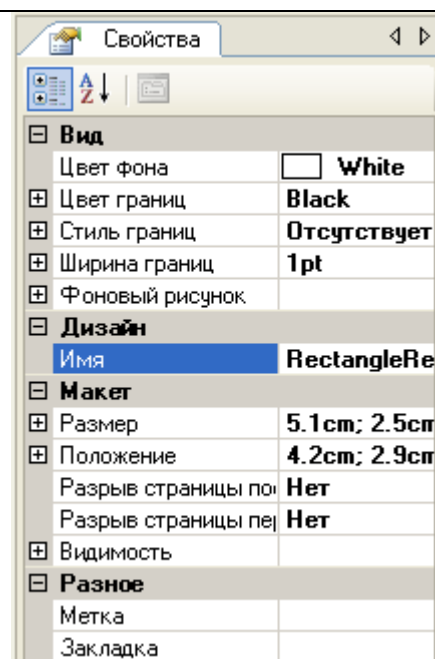
Для размещения прямоугольника в отчете необходимо перенести элемент **Прямоугольник** с панели элементов отчета в отчет.



После того как прямоугольник окажется на шаблоне отчета, пользователь может в него добавлять другие элементы. Для этого необходимо выбрать элемент отчета и перетащить его на область прямоугольника.

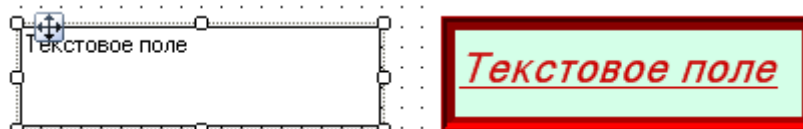
#### Свойства

- **Цвет фона; цвет, стиль и ширина границ; фоновый рисунок** – вид прямоугольника
- **Имя** – наименование элемента отчета (прямоугольника)
- **Размер** – позиция правого нижнего угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Положение** – позиция верхнего левого угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Разрыв страницы после** – указывает на то, что за прямоугольником в отчете следует разрыв страницы
- **Разрыв страницы перед** – указывает на то, что прямоугольнику будет предшествовать разрыв страницы
- **Видимость/Переключатель** – имя текстового элемента, используемого в качестве переключателя видимости для данного элемента отчета. Щелчок мышкой на назначенном текстовом элементе переключает состояние видимости каждого экземпляра данного элемента отчета. Если элемент, назначенный в качестве переключателя, будет скрыт, данный элемент также станет невидимым
- **Видимость/Скрыть** – указывает на то, что элемент отчета будет скрыт
- **Метка** – метка для идентификации элемента отчета
- **Закладка** – ассоциированная с данным элементом закладка; может быть использована в качестве параметра для навигации



## 7.1.2.3 Текстовое поле

**Текстовое поле** – это элемент отчета, содержащий символьное значение: статический текст или результат вычисления выражения.

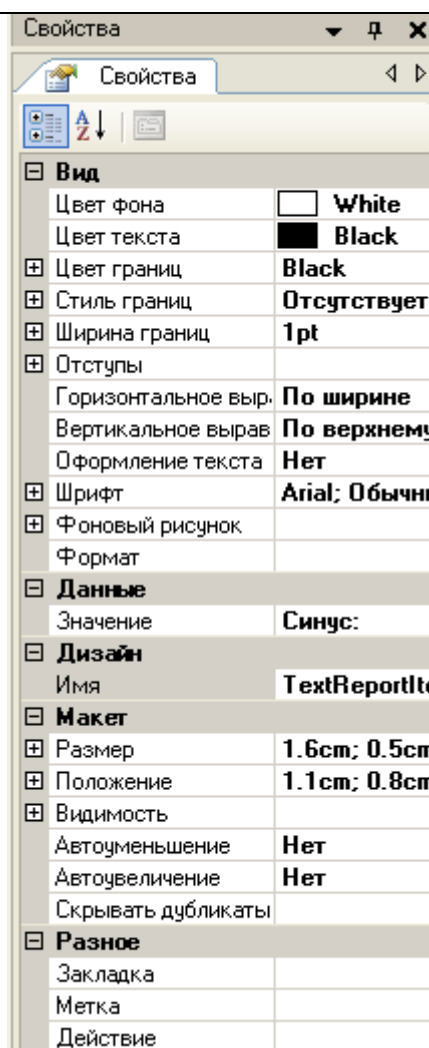


В текстовых полях отображаются любые текстовые данные отчета. Ячейки таблиц для отображения данных по умолчанию содержат текстовое поле. Текстовое поле можно поместить в любую секцию отчета.

Для размещения текстового поля в отчете необходимо перенести элемент **Текстовое поле** панели элементов отчета в отчет и растянуть поле ввода текста до нужных размеров. Чтобы создать текстовое поле фиксированного размера следует выполнить щелчок левой клавишей мыши в области шаблона отчета, а затем двойной щелчок по элементу отчета **Текстовое поле**.

**Свойства**

- **Цвет фона и текста; цвет, стиль и ширина границ; отступы, горизонтальное и вертикальное выравнивание; оформление текста; шрифт; фоновый рисунок; формат** – вид текста и фона
- **Значение** – статический текст, или указатель на поле источника данных или выражение для вычисления значения
- **Имя** – наименование элемента отчета (текстового поля)
- **Размер** – позиция правого нижнего угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Положение** – позиция верхнего левого угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Видимость/Переключатель** – имя текстового элемента, используемого в качестве переключателя видимости для данного элемента отчета. Щелчок мышкой на назначенном текстовом элементе переключает состояние видимости каждого экземпляра данного элемента отчета. Если элемент, назначенный в качестве переключателя, будет скрыт, данный элемент также станет невидимым
- **Видимость/Скрыть** – указывает на то, что элемент отчета будет скрыт
- **Автоуменьшение/Автоувеличение** – определяет, должен ли размер текстового поля автоматически уменьшаться/увеличиваться в зависимости от размера текста, содержащегося в нем
- **Скрывать дубликаты** – определяет, будет ли значение текстового поля автоматически скрываться, если оно совпадает со значением предыдущего элемента, входящего в эту же область данных. Данное свойство применяется, когда текстовое поле находится в ячейке таблицы.



- **Метка** – метка для идентификации элемента отчета
- **Закладка** – ассоциированная с данным элементом закладка; может быть использована в качестве параметра для навигации
- **Действие** – переход на закладку. Закладка, на которую нужно сделать переход, может задаваться именем закладки либо выражением, результатом которого будет имя закладки.

### Примеры выражений для свойства Значение

Выражение	Описание
= "Это пример текста"	Отображает строку: "Это пример текста "
= Fields!UnitCost.Value	Отображает значение поля <b>UnitCost</b>
= Fields!TotalUnits.Value * Fields!UnitCost.Value	Умножает значение поля <b>TotalUnits</b> на значение поля <b>UnitCost</b> и отображает результат
= Sum(Fields!TotalUnits.Value)	Вычисляет сумму значений поля <b>TotalUnits</b> для всех строк в группе

### ВНИМАНИЕ!!!

Строку символов (статический текст) можно поместить в поле **Значение**, не используя выражений (без знака равенства или кавычек).

#### 7.1.2.4 Таблица

**Таблица** – это область данных, в которой значения располагаются в столбцах и строках.

Таблицы могут содержать любое количество столбцов. Количество столбцов конкретной таблицы фиксировано, а число строк в таблице зависит от значений набора данных.

Ячейки данных в пределах таблицы могут охватывать несколько столбцов.

Для упорядочения данных в таблице можно добавить группирование и сортировку.

С помощью конструктора отчетов можно определить табличные отчеты, содержащие верхние и нижние колонтитулы таблиц и групп, а также строки с подробными сведениями. Область данных таблицы можно добавить в новый пустой отчет или в существующий отчет.

К таблицам могут быть добавлены группы, которые упорядочивают данные в соответствии с указанными полями или выражениями. Чтобы добавить таблицу в отчет следует:

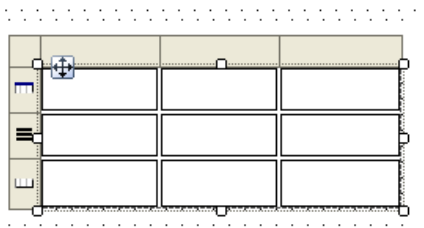
- 1 Перенести элемент **Таблица** из панели элементов отчета в отчет
- 2 Добавить столбцы и строки.

### Форматирование таблицы

#### Маркеры

В дизайнера отчетов работа со столбцами, строками и всей таблицей сводится к оперированию с маркерами. Маркеры столбцов и строк представляют собой серые поля, отображаемые при выборе таблицы в верхней ее части и слева:

- **Маркеры столбцов** расположены в верхней части таблицы
- **Маркеры строк** расположены сбоку от таблицы
- **Угловой маркер** расположен на пересечении маркеров столбцов и строк.

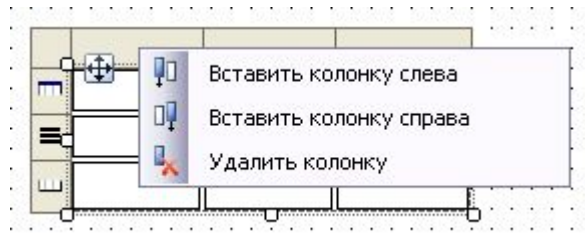




Для выполнения большинства действий со столбцами, строками и таблицей нужно вызвать контекстное меню (щелкнуть правой кнопкой мыши на маркере столбцов, маркере строк или на угловом маркере соответственно).

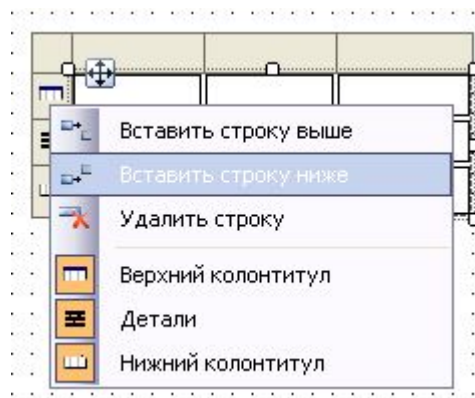
### Вставка/удаление столбцов

Для вставки/удаления столбца необходимо щелкнуть правой кнопкой мыши по маркеру столбца и выбрать необходимый пункт контекстного меню.



### Вставка/удаление строк

Для вставки/удаления строки необходимо щелкнуть правой кнопкой мыши по маркеру строки и выбрать необходимый пункт контекстного меню.



### Включение/отключение колонтитулов

Для включения или выключения показа колонтитулов таблицы следует выбрать в контекстном меню любой строки необходимый пункт меню (**Верхний колонтитул/Нижний колонтитул**).

### Объединение/разбиение ячеек

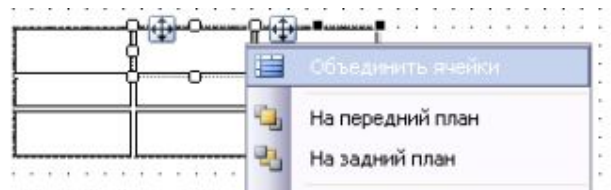
Несколько смежных ячеек (только по столбцам) в пределах таблицы можно объединить в одну ячейку.

При объединении сохраняются данные только первой из объединяемых ячеек, данные в других ячейках удаляются.

Объединяемые ячейки можно разделить обратно на исходные.

Для объединения ячеек следует:

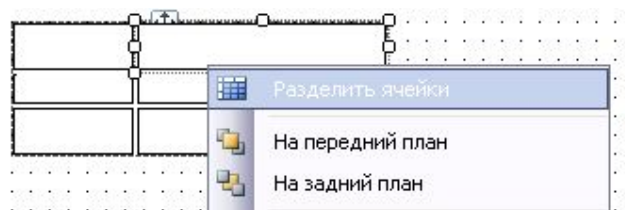
- 1 Щелчком левой клавишей мыши при нажатой клавише **Shift** выбрать нужные для объединения ячейки строки



- 2 Щелчком правой кнопкой мыши вызвать контекстного меню и выбрать пункт **Объединить ячейки**

Чтобы разбить ячейку после объединения следует:

- 1 Щелчком левой клавишей мыши выбрать нужную для разбиения ячейку строки
- 2 Щелчком правой кнопкой мыши вызвать контекстного меню и выбрать пункт **Разбить ячейки**



## Свойства

- **Цвет фона; цвет, стиль и ширина границ; отступы, фоновый рисунок** – вид границ и фона таблицы
- **Данные** – указывает имя источника данных или текст сообщения при отсутствии данных
- **Имя** – наименование элемента отчета (текстового поля)
- **Видимость** – имя текстового элемента, используемого в качестве переключателя видимости для данного элемента отчета, или указание на то, что элемент отчета будет скрыт
- **Разрыв страницы перед/после** – указание о разрыве страниц отчета
- **Положение** – позиция верхнего левого угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Размер** – позиция правого нижнего угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Нижний колонтитул, Верхний колонтитул** и другие свойства задают отображения колонтитулов
- **Отображать все вместе** – указывает на то, что вся область целиком (все повторяющиеся секции) должна, по возможности, располагаться на одной странице
- **Закладка** – ассоциированная с данным элементом закладка; может быть использована в качестве параметра для навигации
- **Метка** – метка для идентификации элемента отчета

Свойства	
<b>Вид</b>	
Цвет фона	White
Цвет границ	Black
Стиль границ	Сплошная
Ширина границ	1pt
Фоновый рисунок	
<b>Данные</b>	
Сообщение при отсутствии данных	
Имя источника данных	
<b>Дизайн</b>	
Имя	TableReportIt
<b>Макет</b>	
Видимость	
Разрыв страницы перед	Нет
Разрыв страницы после	Нет
Положение	3.7cm; 4cm
Размер	5.3cm; 1.9cm
Нижний колонтитул	Да
Верхний колонтитул на каждой стр.	Нет
Верхний колонтитул	Да
Секция деталей	Да
Верхний колонтитул зафиксирован	Нет
Отображать все вместе	Нет
Нижний колонтитул на каждой стр.	Нет
<b>Разное</b>	
Закладка	
Метка	

## Добавление данных в таблицу

По умолчанию каждая ячейка таблицы содержит текстовое поле.

Для задания значения в ячейке можно:

- Задать выражение
- Изменить тип элемента в ячейке
- Связать ячейку с источником данных.

Каждая область данных таблицы связана с определенным набором данных.

Если отчет содержит один набор данных, то таблица при помещении ее в отчет автоматически связывается с этим набором.

Если отчет содержит несколько наборов данных, то необходимо связать таблицу с нужным набором данных. Для этого задайте имя набора данных в свойстве таблицы **Имя источника данных**.

Если таблица связана с набором данных, то данные из других наборов данных могут быть помещены в ячейку этой таблицы только с использованием функции агрегирования.

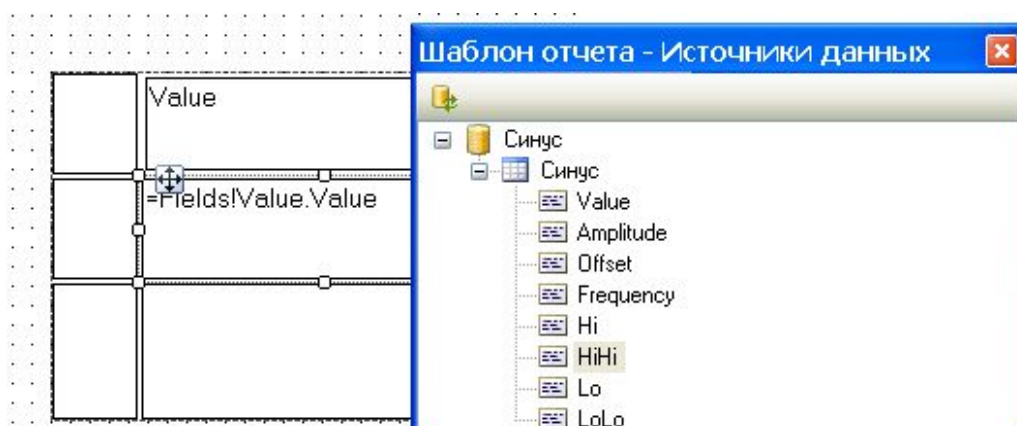
При перемещении из окна **Шаблон отчета - Источники данных** мышью поля набора данных, не связанного с данной таблицей, в ячейку этой таблицы автоматически создается функция агрегирования **Sum**.

#### Как связать ячейку таблицы с источником данных

Для добавления значений в таблицу из источника данных следует:

- 1 Выбрать в контекстном меню отчета или дизайнера пункт **Данные**
- 2 Перетащить из появившегося окна **Источники данных** требуемый источник данных в ячейку таблицы.

В данном примере источником данных является атрибут **Value** тега **Синус**



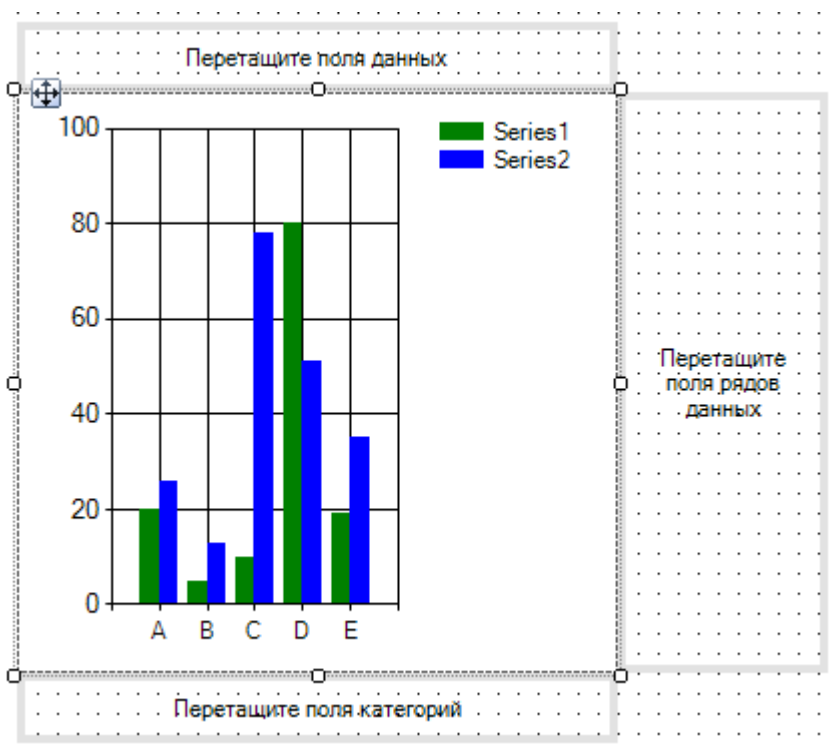
#### 7.1.2.5 Диаграмма

**Диаграмма** – это область отчета, которая отображает данные в графическом представлении.

**Тип диаграммы** определяет, какая геометрической фигура используется для отображения данных: отрезок, круг или другая фигура.

#### Данные диаграммы

Данные диаграммы сгруппированы в три области: **поле данных**, **поле категорий** и **поле рядов данных**.



### Поля данных

#### **ВНИМАНИЕ!!!**

**При создании диаграммы в поле данных надо обязательно задать хотя бы один источник данных.**

Значения данных в поле данных определяют размер элемента диаграммы для каждой группы категорий.

Например, данные определяют высоту столбца в гистограмме или размер сегмента в круговой диаграмме.

Поля данных являются статическими. Если определяется одно поле данных и не задаются группы рядов, то один элемент диаграммы отображается для каждой группы категорий. Например, одна гистограмма с одним полем данных отображает один столбец для каждой группы категорий. Если определяется несколько значений, диаграмма отображает элемент диаграммы для каждого поля данных. Если предусмотрено несколько полей данных, условное обозначение диаграммы отображает имя каждого из них.

В большинстве диаграмм данные группируются по категориям. В этом случае в диаграмме необходимо использовать статистическое выражение или функцию.

Статистическое выражение не требуется, если не выполняется группирование данных, поскольку предусмотрено одно значение категорий для каждого значения в наборе данных. Тем не менее, при использовании групп рядов статистическое выражение обязательно.

### Категории

Категории используются для группирования данных. Они предоставляют метки для элементов диаграммы. Например, на гистограмме метки категорий размещаются на оси X по одной на каждый набор столбцов.

Категории могут быть вложенными. Если определено несколько категорий, каждая из них вкладывается в другую категорию. Например, на гистограмме, на которой выводятся продукты по моделям, первая группа категорий будет моделью, а вторая — продуктом. Гистограмма отобразит группирование продуктов по моделям на горизонтальной оси.

### Группы рядов данных

Группы рядов необязательны. Можно определить группу рядов для добавления нового измерения данных в отчет. Например, на гистограмме, отображающей продажи по продуктам, можно добавить группу рядов для вывода продаж по годам для каждого продукта. Метки групп рядов помещаются в условные обозначения диаграммы.

Группы рядов являются динамическими. В диаграмме, использующей группы рядов, элемент диаграммы отображается для каждой группы рядов по каждой категории.

## • Свойства

- **Цвет фона, палитра** – цвет фона и элементов диаграммы
- **Ширина элемента** – относительная ширина в % для столбцов и линеек в гистограммах и линейчатых диаграммах
- **Тип, подтип диаграммы** – идентификация диаграммы (гистограммы, графики, линейчатые, круговые диаграммы и другие)
- **Цвет, стиль и ширина границ** – вид границ диаграммы
- **Цвет и вид градиентной заливки фона** – заливка фона диаграммы
- **Данные** – указывает имя источника данных или текст сообщения при отсутствии данных
- **Имя** – наименование элемента отчета (текстового поля)
- **Разрыв страницы перед/после** – указание о разрыве страниц отчета
- **Отображать все вместе** – указывает на то, что вся область целиком (все повторяющиеся секции) должна, по возможности, располагаться на одной странице
- **Видимость** – имя текстового элемента, используемого в качестве переключателя видимости для данного элемента отчета, или указание на то, что элемент отчета будет скрыт
- **Положение** – позиция верхнего левого угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Размер** – позиция правого нижнего угла элемента отчета относительно контейнера, задается значениями **X** (по горизонтали) и **Y** (по вертикали)
- **Заголовок** – заголовок диаграммы. Включает свойства **Текст, Выравнивание, Оформление текста, Шрифт, Цвет текста**
- **Легенда** – описание легенды диаграммы
- **Свойства координатной плоскости** – описание координатной плоскости диаграммы
- **3-х мерный вид** – свойства объемного представления диаграммы
- **Ось категорий, ось значений** – описание осей диаграммы
- **Закладка** – ассоциированная с данным элементом закладка; может быть использована в качестве параметра для навигации
- **Метка** – метка для идентификации элемента отчета

Свойства

**Вид**

Цвет фона	<input type="checkbox"/> White
Палитра	Стандартная
Ширина элемента	0
Тип	Гистограмма
Подтип	Обычная
Цвет границ	Black
Стиль границ	Отсутствует
Ширина границ	1pt
Конечный цвет градиентн	<input type="checkbox"/> White
Вид градиентной заливки	Отсутствует

**Данные**

Сообщение при отсутстви	
Имя источника данных	

**Дизайн**

Имя	ChartReportItem
-----	-----------------

**Макет**

Разрыв страницы после	Нет
Разрыв страницы перед	Нет
Отображать все вместе	Нет
Видимость	
Положение	3.2cm; 4.5cm
Размер	7.6cm; 5.1cm
Заголовок	
Легенда	
Свойства координатной п	
3-х мерный вид	
Ось категорий (X)	
Ось значений (Y)	

**Разное**

Закладка	
Метка	



## Тип диаграммы

- **Гистограмма** – данные выводятся в виде набора вертикальных столбцов. Включает сведения о гибридных диаграммах с гистограммами и графиками
- **Линейчатая** – данные выводятся в виде набора горизонтальных линий
- **График** – данные выводятся в виде набора точек, соединенных линией
- **Круговая** – данные выводятся в виде круга, разделенного на сектора в зависимости процентов от целого
- **Точечная** – данные выводятся в виде набора точек
- **Пузырьковая** – данные выводятся в виде набора «укрупненных» точек
- **С областями** – данные выводятся в виде набора точек, соединенных линией с областями, закрашенными ниже этой линии
- **Кольцевая** – данные выводятся в виде кольца, разделенного на участки в зависимости процентов от целого.

Тип	Гистограмма
Подтип	Гистограмма
Цвет границ	Линейчатая
Стиль границ	График
Ширина границ	Круговая
Конечный цвет градиент	Точечная
Вид градиентной заливки	Пузырьковая
Данные	С областями
Сообщение при отсутст	Кольцевая

## Легенда

- **Цвет, ширина границы и стиль рамки** – цвет, ширина и стиль линии для рамки у легенды диаграммы
- **Второй цвет, Градиентная заливка, цвет заливки** – параметры заливки легенды
- **Оформление текста, Цвет текста, Шрифт** – параметры текста легенды
- **Расположение** – расположение надписей внутри легенды
- **Позиция легенды в диаграмме** – расположение легенды в диаграмме
- **Отображается** – указывает на то, должна ли легенда отображаться в диаграмме
- **Внутри координатной области** – указывает не то, что легенда отображается внутри координатной области диаграммы

Легенда	
Цвет границы	Black
Ширина границы	1pt
Стиль рамки	Отсутствует
Второй цвет	White
Градиентная заливка	Отсутствует
Цвет заливки	White
Оформление текста	Нет
Расположение	В столбец
Позиция легенды в диаг	Справа вверх
Отображается	Да
Цвет текста	Black
Шрифт	Arial; Обычны
Внутри координатной об	Нет

## Свойства координатной плоскости

- **Цвет заливки** – цвет заливки для координатной области диаграммы
- **Градиентная заливка, Второй цвет** – стиль и второй цвет градиентной заливки для координатной области диаграммы
- **Стиль границы, Толщина линии, Цвет границы** – параметры границы координатной области

Свойства координатной пл	
Цвет заливки	White
Градиентная заливка	Отсутствует
Второй цвет	White
Стиль границы	Отсутствует
Толщина линии	1pt
Цвет границы	Black

## 3-х мерный вид

- **Толщина стенок** – толщина внешних стенок в %
- **Затенение** – стиль отрисовки затенения
- **Группировать в кластеры** – указывает на необходимость группирования рядов данных в кластеры
- **Стиль отрисовки** – кубический или цилиндрический стиль отрисовки
- **Горизонтальный поворот, Вертикальный поворот** – угол поворота диаграммы в горизонтальной/вертикальной плоскости
- **Проекция** – тип проекции (перспективная или ортографическая), используемый при трехмерном отображении диаграммы
- **Используется** – указывает на использование трехмерного режима
- **Перспективность** – % перспективности

3-х мерный вид	
Толщина стенок	0
Затенение	Отсутствует
Группировать в кластеры	Нет
Стиль отрисовки	Кубический
Горизонтальный поворот	0
Проекция	Перспективная
Используется	Нет
Вертикальный поворот	0
Перспективность	0

## Ось категорий (X), Ось значений (Y)

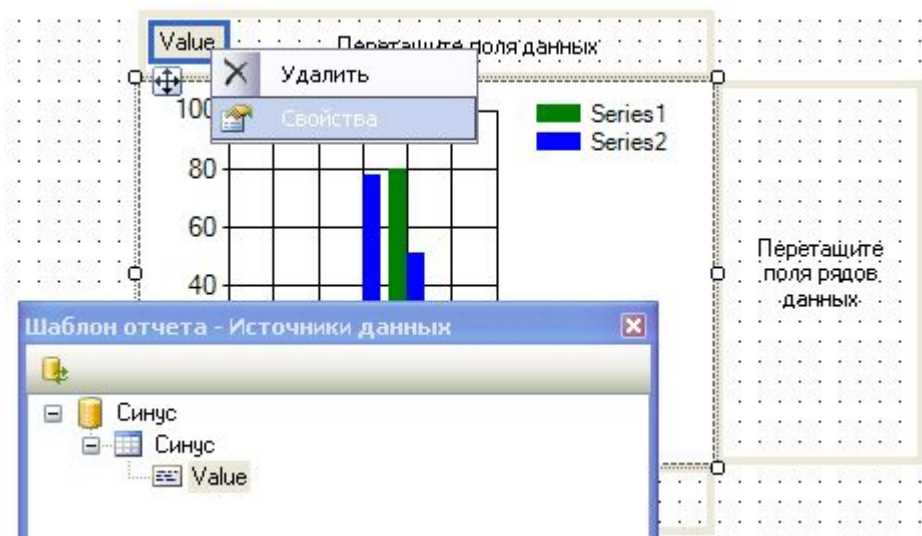
- **Сетка первого порядка** – параметры отображения линий сетки первого порядка для данной координатной оси. Включает свойства **Цвет, стиль, толщина линии** сетки; **Отображать линии сетки**
- **Стиль линии** – стиль линии оси значений
- **Сетка второго порядка** – параметры отображения линий сетки второго порядка для данной координатной оси
- **Оформление меток** – стиль оформления меток на оси
- **Формат меток** – строка, задающая стиль форматирования меток на координатной оси, данные выводятся в виде набора вертикальных столбцов. Включает сведения о гибридных диаграммах с гистограммами и графиками
- **Поле** – указывает на необходимость создания пустого поля на координатной оси. Размер поля вычисляется автоматически на основе текущего значения масштаба и количества точек
- **Интервальный режим** – если значение свойства **Да**, то метки на координатной оси выставляются через равные интервалы автоматически. Если – **Нет**, то метки на координатной оси показывают реальные значения точек (при этом, если точек очень много метки вообще не отображаются)

Ось значений (Y)	
Сетка первого порядка	
Стиль линии	Сплошная
Сетка второго порядка	
Логарифмическая шкала	Нет
Заголовок	
Толщина линии	1pt
Оформление меток	Нет
Цвет текста	Black
Шрифт	Arial; Обыч
Цвет линии	Black
Формат меток	
Максимальное значение	
Стиль маркера второго порядка	Маркер отс
Единица второго порядка	
Стиль маркера первого порядка	Маркер отс
Отображать метки	Да
Поле	Нет
Единица первого порядка	
Интервальный режим	Да
Минимальное значение	
Чередование заливки	Нет
Инвертированная	Нет
Точка пересечения	

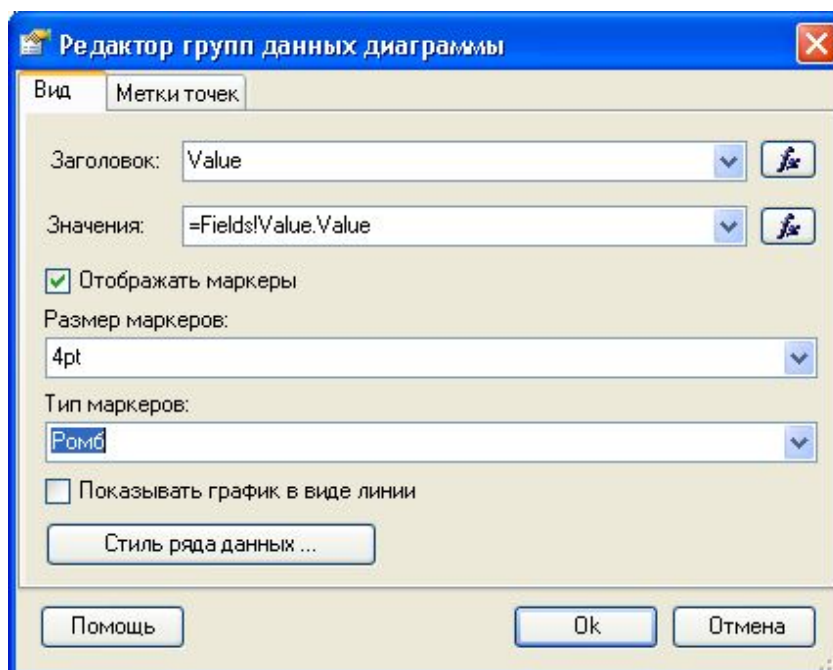
## Группы данных

Для настройки отображения данных следует:

- 1 Перетащить данные из источника данных в поле данных. В примере – атрибут **Value** тега **Синус**
- 2 Выбрать пункт **Свойства** из контекстного меню поля данных



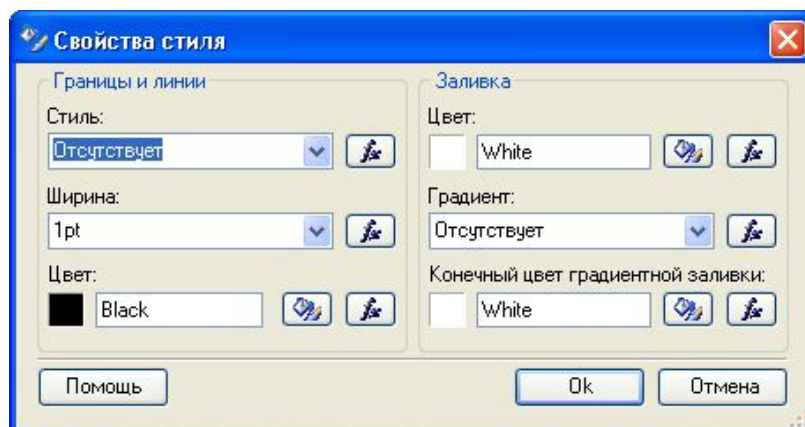
- 3 В появившемся окне **Редактор групп данных диаграммы** задать параметры отображения данных.



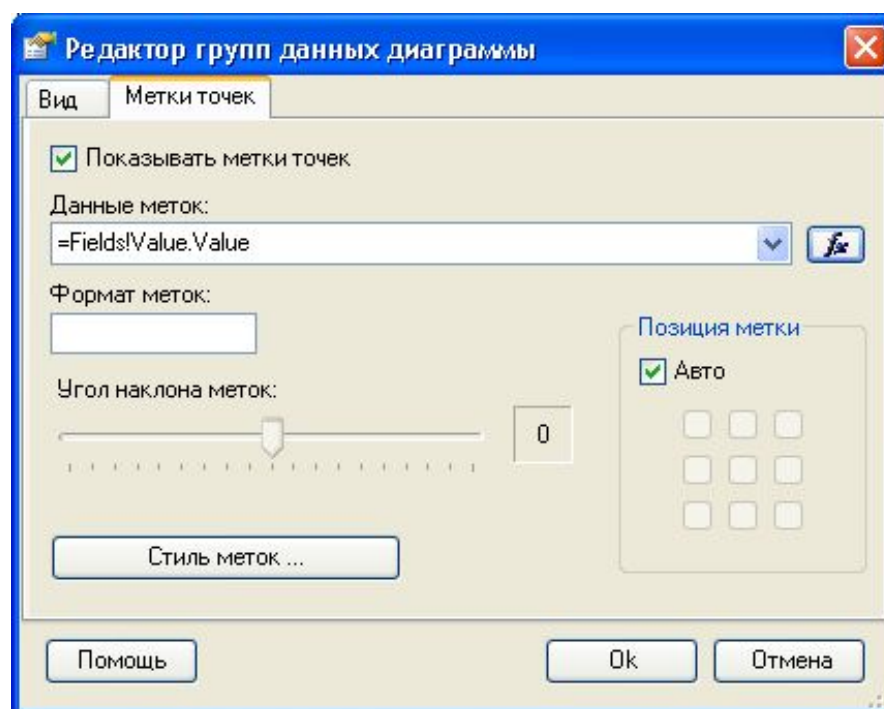
### Вкладка Вид

- **Заголовок** – выражение, используемое для получения заголовка
- **Значение** – выражение, используемое для получения значений группы данных

- **Отображать маркеры** – если данный пункт выбран то, на диаграмме будут отображаться маркеры, заданного размера (**Размер маркеров**) и типа (**Тип маркеров**)
- **Показывать график в виде линии** – признак доступен только для гистограмм
- **Стиль ряда данных...** – кнопка настройки стиля отображения данных, вызывает окно **Свойства стиля**.  
Свойства заливки используются для заливки столбцов гистограммы, если не выбрана опция **Показывать график в виде линии**.

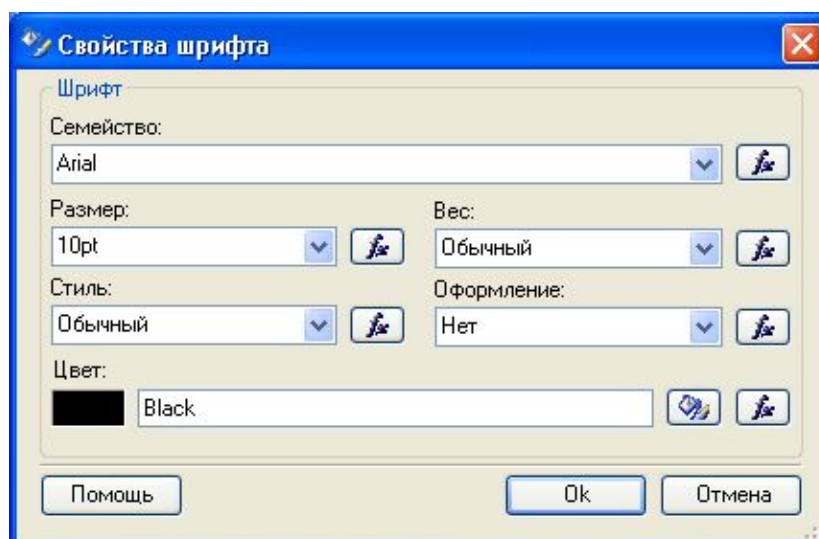


#### Вкладка Метки точек



- **Показывать метки точек** – включает или выключает показ меток на диаграмме
- **Данные меток** – данные, которые будут отображаться в метке
- **Формат меток** – форматирующие символы для форматирования вывода меток
- **Угол наклона метки** – угол наклона текста метки.
- **Позиция метки** – позиция метки относительно элемента диаграммы.

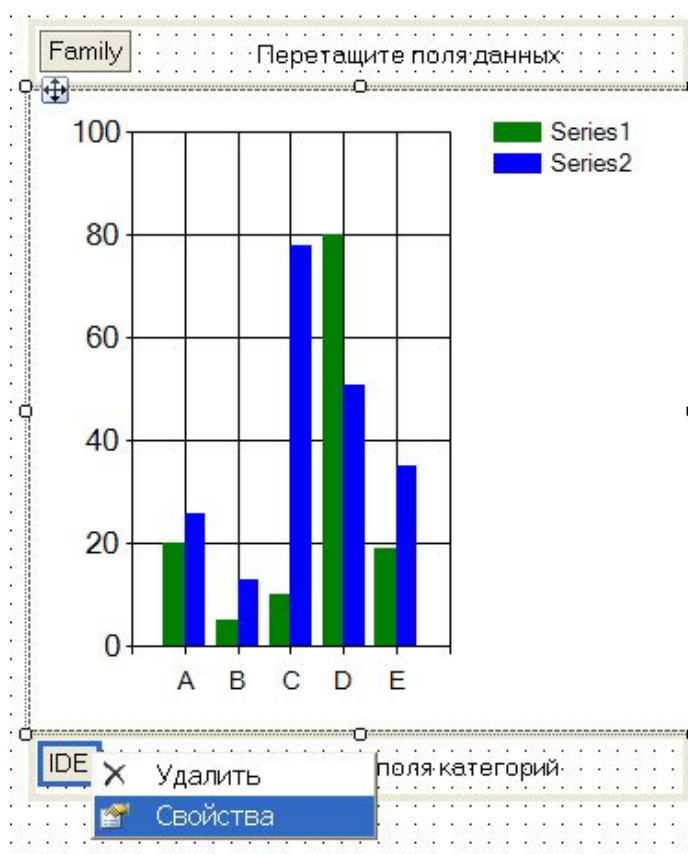
- **Авто** – позиция метки будет выбрана автоматически
- **Стиль меток** – настройка стиля меток. Вызывает окно **Свойство шрифта**



### Группировка и сортировка данных

Для настройки группировки и сортировки данных следует:

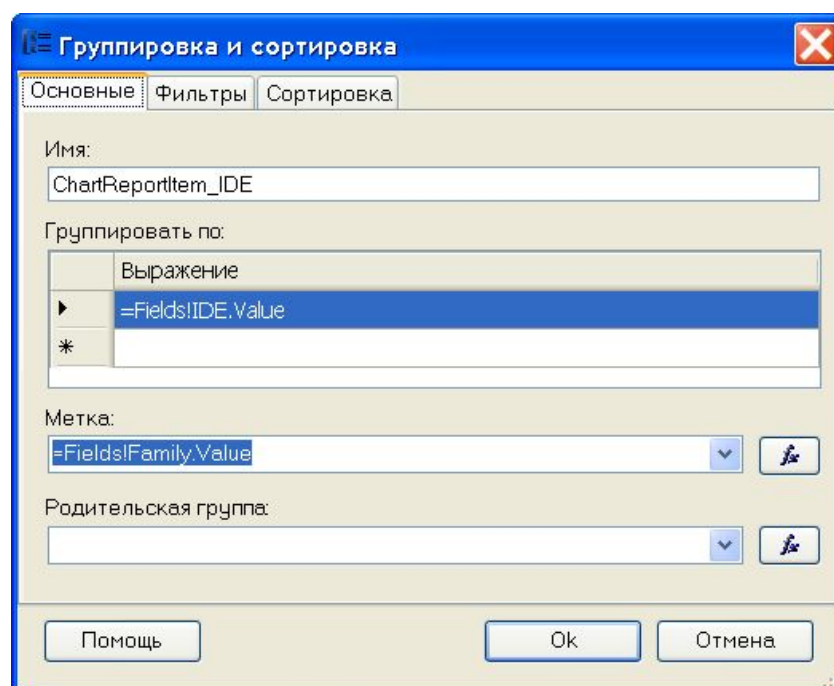
- 1 Перетащить необходимые данные в поле категорий
- 2 Выбрать пункт **Свойства** из контекстного меню поля категорий
- 3 В появившемся окне **Группировка и сортировка** задать параметры отображения данных.



#### Вкладка Основные

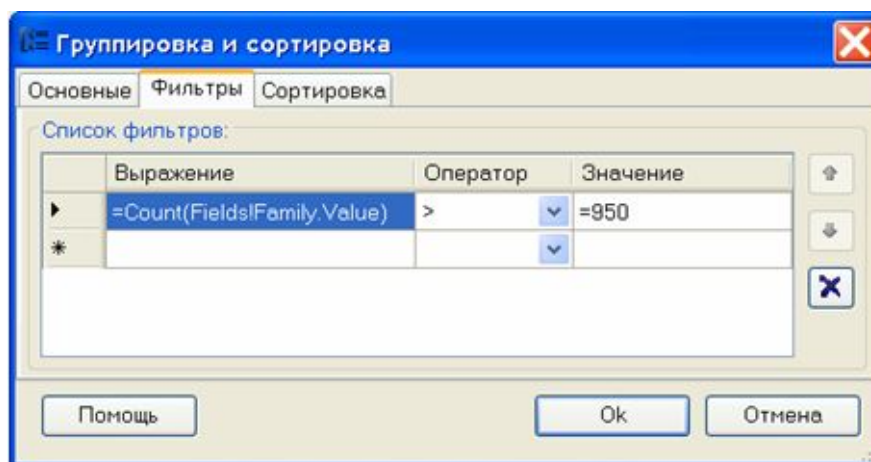
- **Имя** – имя группы. Должно быть уникальным в пределах отчета
- **Группировать по** – выражение для группировки
- **Метка** – значение поля, отображаемое на диаграмме
- **Родительская группа** – родительская группа используется для иерархических запросов.





### Вкладка Фильтры

Фильтр состоит из трех элементов: выражение, оператор сравнения и значение, с которым происходит сравнение.



### ВНИМАНИЕ!!!

Элементы фильтра **Выражение** и **Значение** задаются с помощью выражений.

Условий фильтрации может быть несколько. Для смены позиции условия фильтрации служат кнопки:



– переместить условие фильтрации на одну позицию вверх



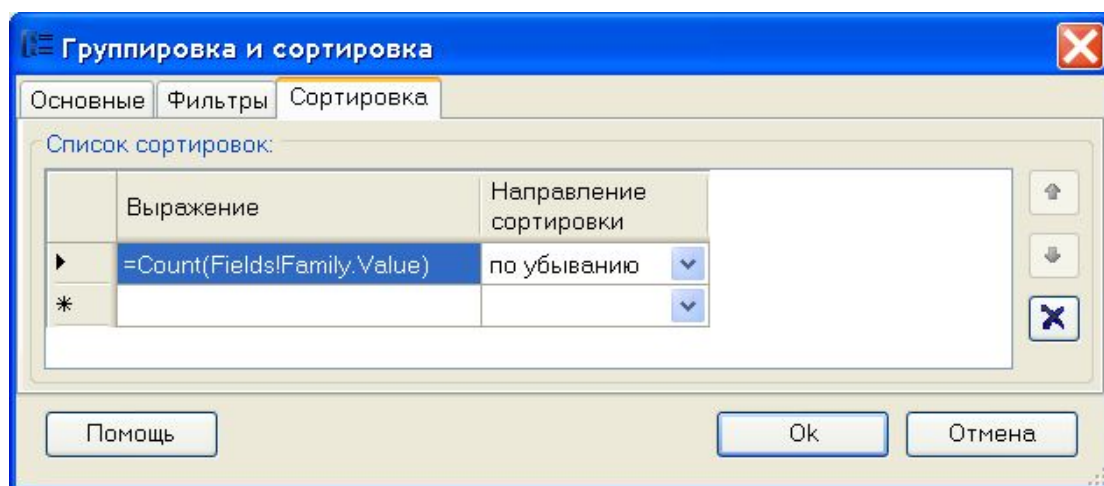
– переместить условие фильтрации на одну позицию вниз



– удалить условие фильтрации.

## Вкладка Сортировка

Условие сортировки состоит из выражения и направления сортировки (по убыванию или по возрастанию).



Условий сортировки может быть несколько. Для смены позиции условия сортировки служат кнопки:

- переместить условие сортировки на одну позицию вверх,
- переместить условие сортировки на одну позицию вниз.
- удалить условие сортировки.

### 7.1.2.5.1 Гистограммы

**Гистограмма** показывает значения полей данных и группы рядов как наборы вертикальных столбцов, сгруппированных по категориям.

Значения представлены высотой столбцов, измеренной по оси Y. По оси X откладываются метки категорий.

Гистограммы обычно используются для сравнения значений категорий.

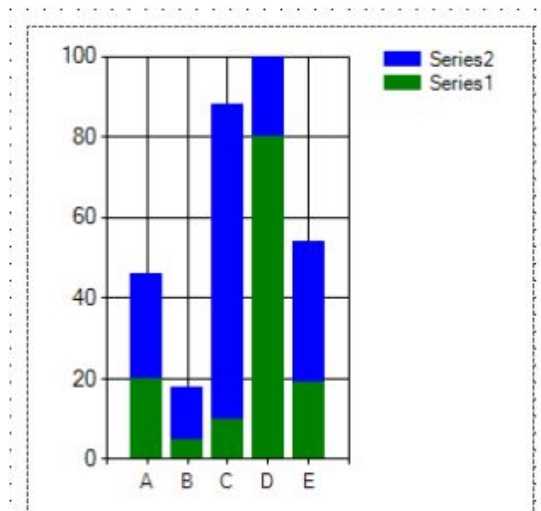
#### Определение гистограмм

Область диаграммы	Обязательность	Описание
Данные	Обязательно	Значения полей данных определяют высоту столбцов в этом ряду. Метки значений отображаются на оси Y. Значения появляются в виде, определяемом типом гистограммы.
Группы категорий	Необязательно	Категории выводятся в виде столбцов или групп столбцов на оси X. Несколько групп вкладываются друг в друга.
Группы рядов	Необязательно	Ряды выводятся в виде отдельных столбцов диаграммы. Каждый ряд также отображается в условных обозначениях диаграммы.

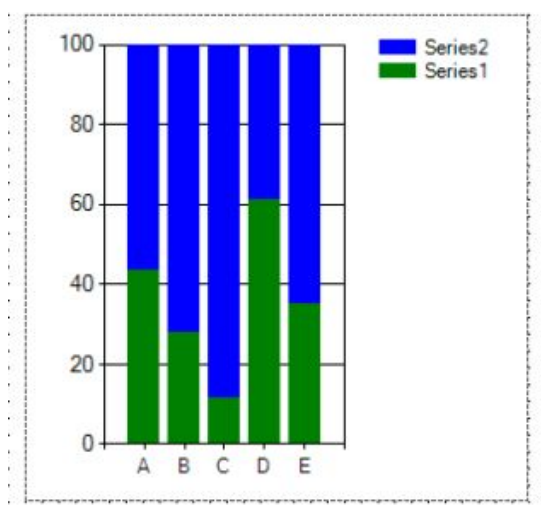
## Типы гистограмм

В **DataRate** используются следующие типы гистограмм:

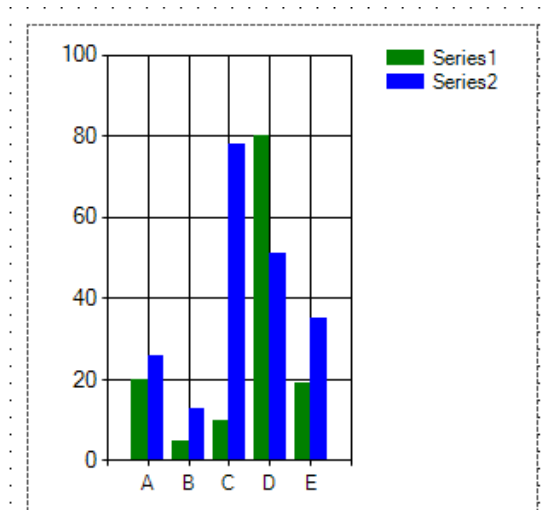
- **Обычная гистограмма** показывает значения полей данных, как отдельные столбцы, сгруппированные по категориям. Высоту каждого столбца определяет значение поля данных



- **Гистограмма с накоплением** показывает значения всех полей данных, расположенные друг над другом в отдельном столбце для каждой категории. Высота каждого столбца определяется суммой всех значений полей данных для этой категории.



- **Нормированная гистограмма** показывает все значения полей данных, расположенные друг над другом в отдельном столбце для каждой категории. Высота каждого столбца всегда является полной высотой диаграммы. Значения в полях данных используются для определения процента от целого. Это процентное отношение определяет размер значения поля данных в столбце. На оси Y откладываются значения от 0 до 100. Значения каждого поля данных появляются в виде раздела внутри столбца



## Гибридные диаграммы с гистограммами и графиками

**DataRate** позволяет создать диаграмму, которая включает как гистограмму (столбцы данных), так и графики (линии данных).

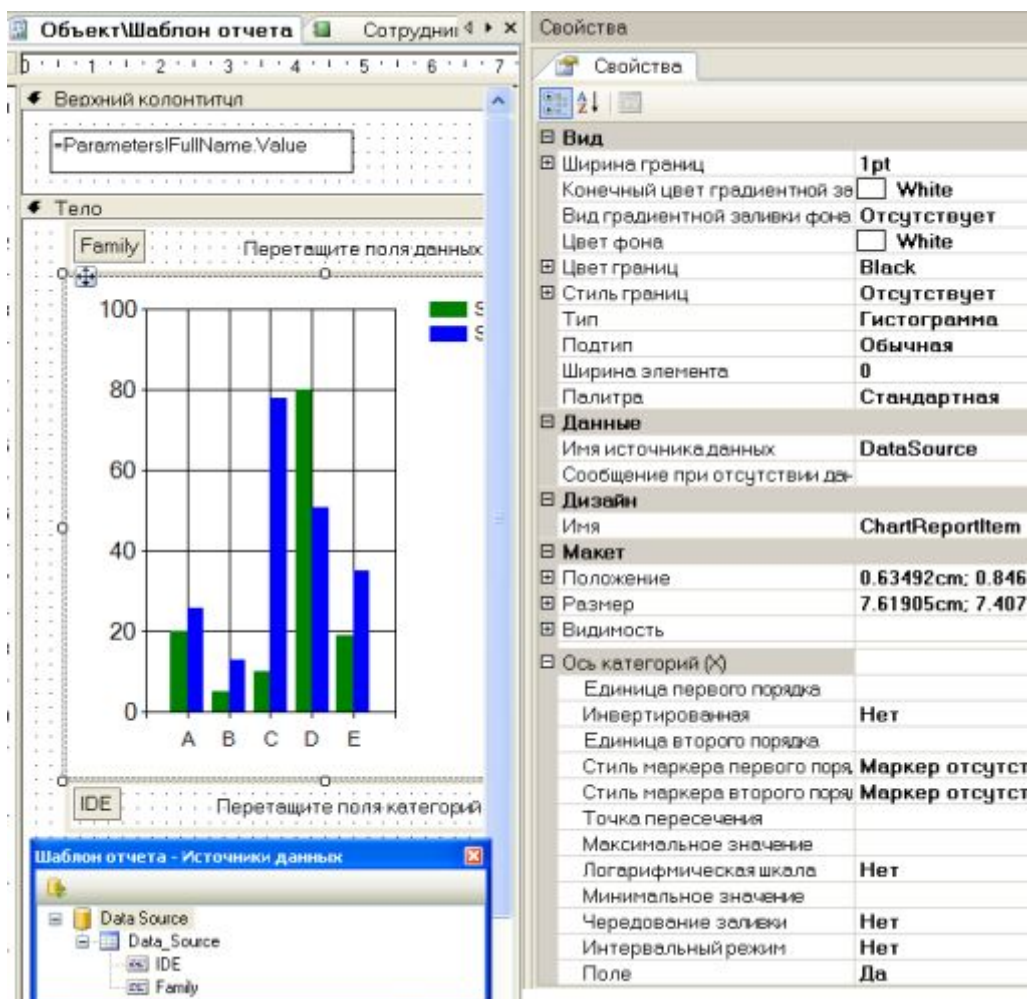
Такая диаграмма создается на основе гистограммы с несколькими полями данных. Для того чтобы отобразить их в виде линии, необходимо установить свойство **Показать график в виде линии** для одного или нескольких полей данных.

### Пример

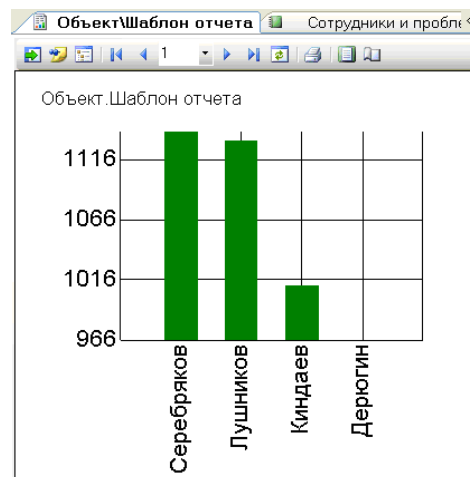
Для формирования данных гистограммы в проект добавлен **SQL-коннектор** с запросом к базе данных:

```
SELECT tblProblem.IDE, tblEmployee.Family
FROM tblProblem
LEFT JOIN tblEmployee
ON tblProblem.IDE = tblEmployee.IDemployee
```

В отчете используется обычная гистограмма с одним полем данных.



Результаты запроса с использованием параметров фильтрации и сортировки отображаются на гистограмме следующим образом



#### 7.1.2.5.2 Линейчатые диаграммы

**Линейчатая диаграмма** отображает значения полей данных, как последовательность горизонтальных прямоугольников, сгруппированных по категориям.

Значения данных представлены шириной полосы по оси X. Метки категорий отображаются на оси Y.

Линейчатые диаграммы обычно используются, чтобы сравнивать значения различных категорий.

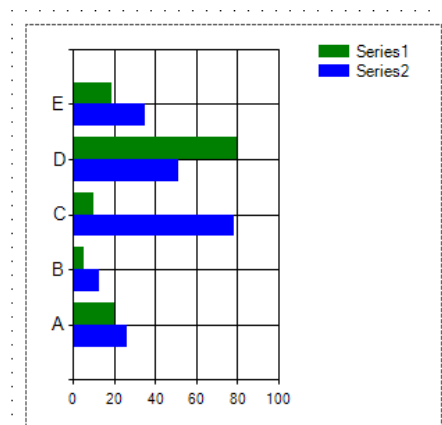
#### Определение гистограмм

Область диаграммы	Обязательность	Описание
Данные	Обязательно	Значения в полях данных определяют ширину полосы для этого поля. Значения меток отображаются на оси X. Каждое значение поля данных отображается как отдельная полоса в соответствии с типом диаграммы.
Группы категорий	Необязательно	Категории отображаются как полосы или группы полос на оси Y. Несколько групп вкладываются друг в друга.
Группы рядов	Необязательно	Ряды в диаграмме отображаются как отдельные полосы. Каждый ряд также отображается в условных обозначениях диаграммы.

#### Типы линейчатых диаграмм

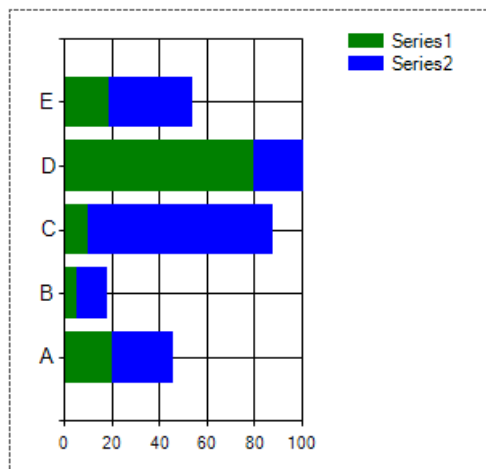
В **DataRate** используются три типа линейчатых диаграмм:

- **Линейчатая диаграмма** отображает каждое значение поля данных, как отдельную горизонтальную полосу, группируя их по категориям. Ширина каждой полосы определяется значением поля данных.

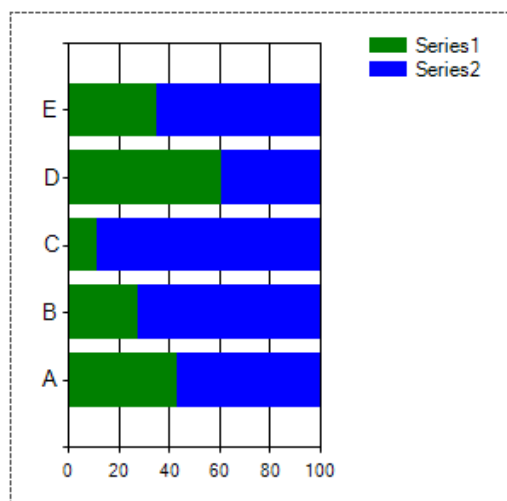




- **Линейчатая диаграмма с накоплением** отображает значения полей данных упорядоченно, по одному на каждую категорию. Ширина каждой полосы определяется общей суммой всех значений полей данных в категории



- **Нормированная линейчатая диаграмма** отображает значения полей данных упорядоченно, по одному на каждую категорию. Ширина каждой полосы всегда равняется полной ширине всей диаграммы. Значения полей данных используются для определения процента от целого. Эти процентные показатели определяют размеры полосы для поля данных. На оси X находятся значения от 0 до 100. Значение каждого поля данных отображается как часть полосы.



## 7.1.2.5.3 Графики

**График** отображает значения данных в виде точек, соединенных линий. По оси Y откладываются значения данных. По оси X откладываются метки категорий.

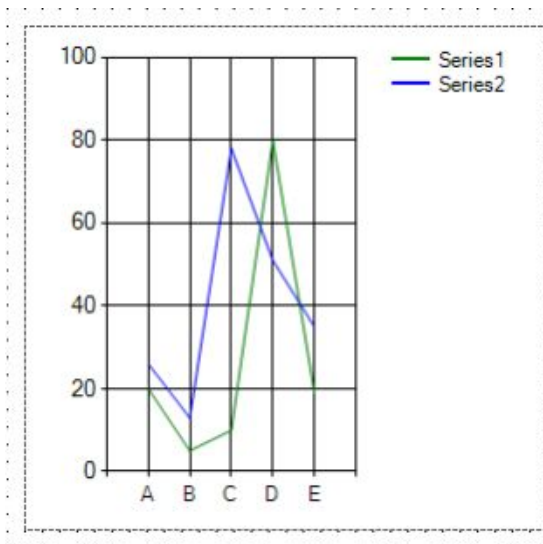
### Определение графика

Область диаграммы	Обязательность	Описание
Данные	Обязательно	Значения в рядах данных определяют высоту линии этого ряда. Метки значений отображаются на оси Y. Каждый ряд значений отображается отдельной линией.
Группы категорий	Необязательно	Категории отображаются в виде меток на оси X. Несколько групп вкладываются друг в друга. В графиках категории обычно связаны со временем.
Группы рядов	Необязательно	На графике ряды отображаются как отдельные линии. Каждый ряд также отображается в условных обозначениях диаграммы.

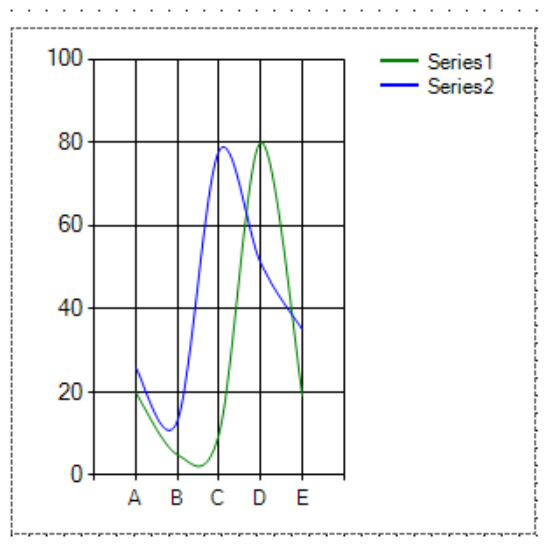
## Типы графиков

В **DataRate** используются следующие типы графиков:

**Линия**



**Гладкий график**



## Пример

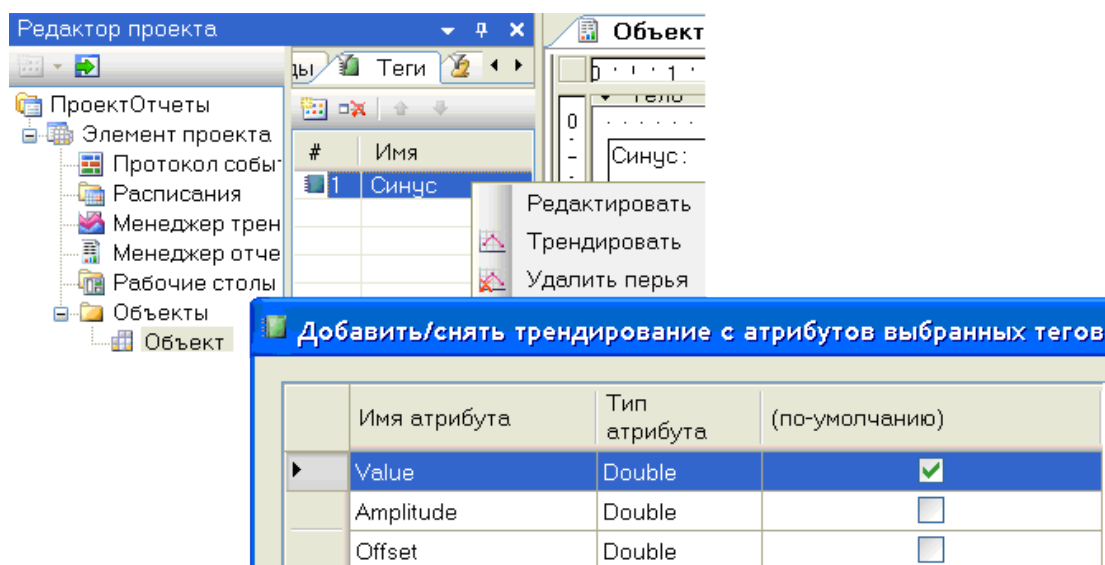
Добавление графика тренда тега в отчет.

В проекте **Отчеты** создан объект **Объект** с тегом **Синус** и видом **Шаблон отчета**.

Для добавления графика тренда в отчет следует:

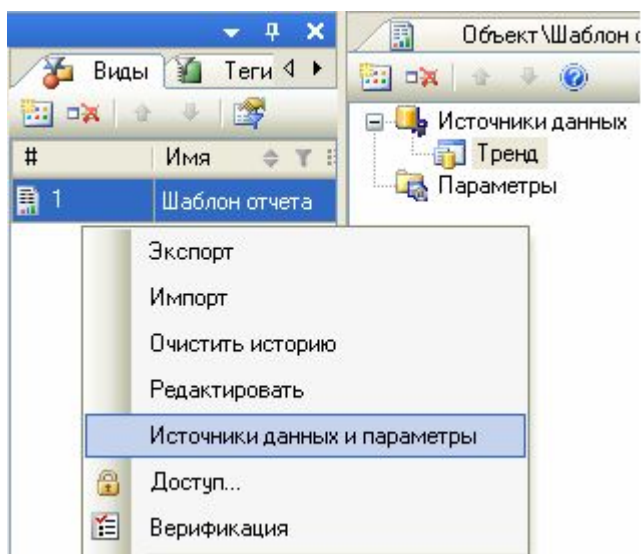
- 1 Создать тренд тега.


В контекстном меню тега **Синус** выбрать пункт **Трендрить** и в появившемся окне **Добавить/снять трендрование у атрибутов выбранных тегов** поставить галочку у атрибута **Value**

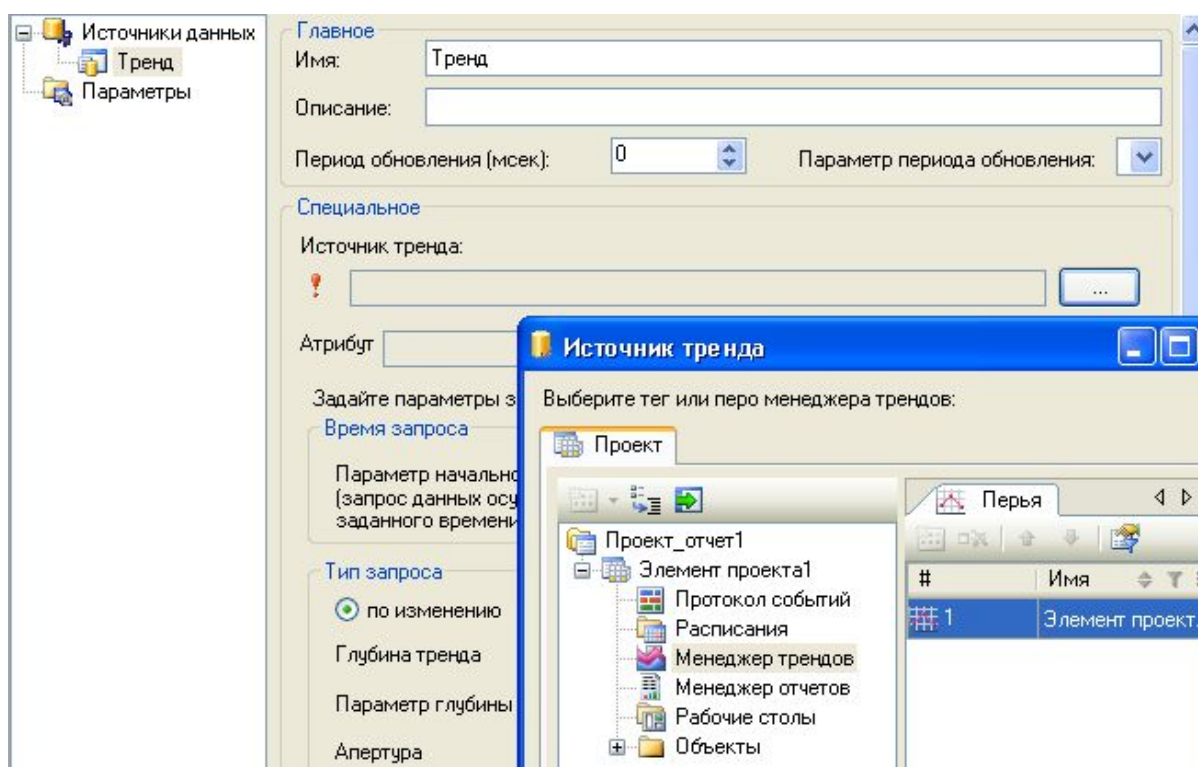


После этого **Менеджер трендов** создаст **Перо 1** для формирования тренда.

- 2 Создать источник данных отчета. В контекстном меню вида **Шаблон отчета** выбрать пункт **Источники данных и параметры**, щелкнуть по кнопке  на панели закладки **Объект\Шаблон отчета**. **Источники данных**, в появившемся окне **Добавление нового элемента** выбрать шаблон источника данных **Тренд**

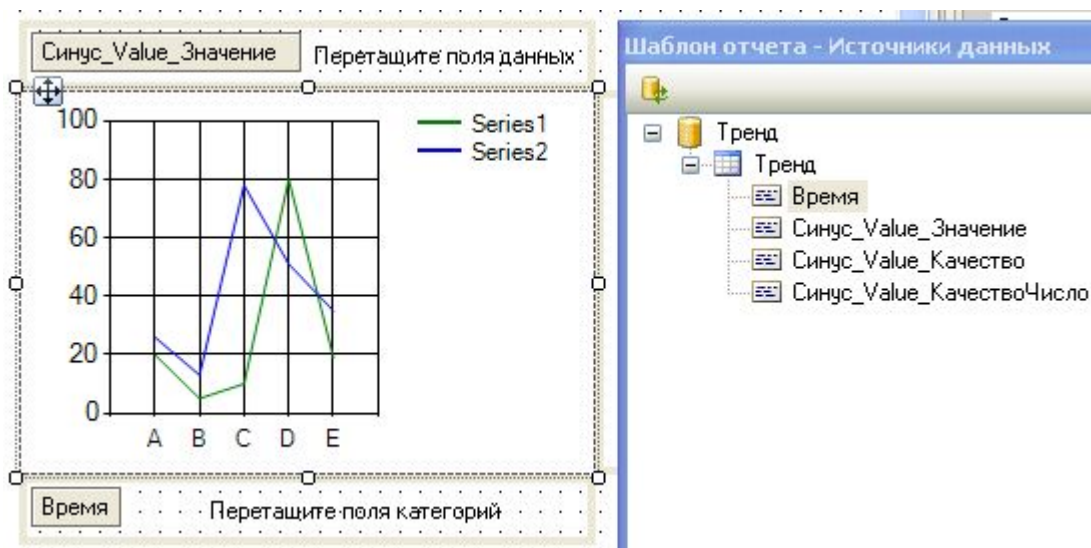


- 3 Связать тренд с пером Менеджера трендов. Для этого щелкнуть по кнопке , в появившемся окне **Источник тренда** выбрать перо 1 менеджера трендов и нажать **ОК**
- 4 Задать параметры запроса тренда:  
**Тип запроса** – по изменению  
**Глубина тренда** – 1/Минуты



- 5 Добавить в шаблон отчета диаграмму:
  - Тип – **График**
  - Подтип – **Обычная**

- 6 В контекстном меню шаблона отчета выбрать пункт **Данные** и перетащить из появившегося окна **Источники данных** в поле данных диаграммы атрибут **Синус\_Value\_Значение** и в поле категорий – атрибут **Время**



- 7 Настроить свойства графика.

**Заголовок:**

- Шрифт – Arial; Обычный; 12pt; Жирный
- Текст – Тренд тега Синус
- Выравнивание – По центр

**Легенда:**

- Расположение – В строку
- Позиция легенды в диаграмме – Внизу по центру
- Отображается – Да

**Ось категорий (X):**

- Формат меток – mm:ss

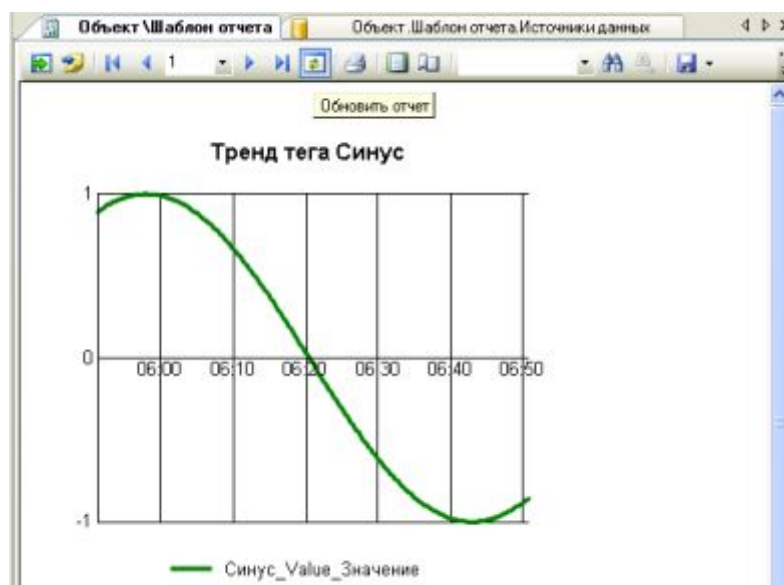
**Ось значений (Y):**

- Максимальное значение – 1
- Минимальное значение – -1
- Точка пересечения – 0

- 8 Сохранить проект

- 9 Запустить проект на выполнение и перейти в режим **Предварительный просмотр**.

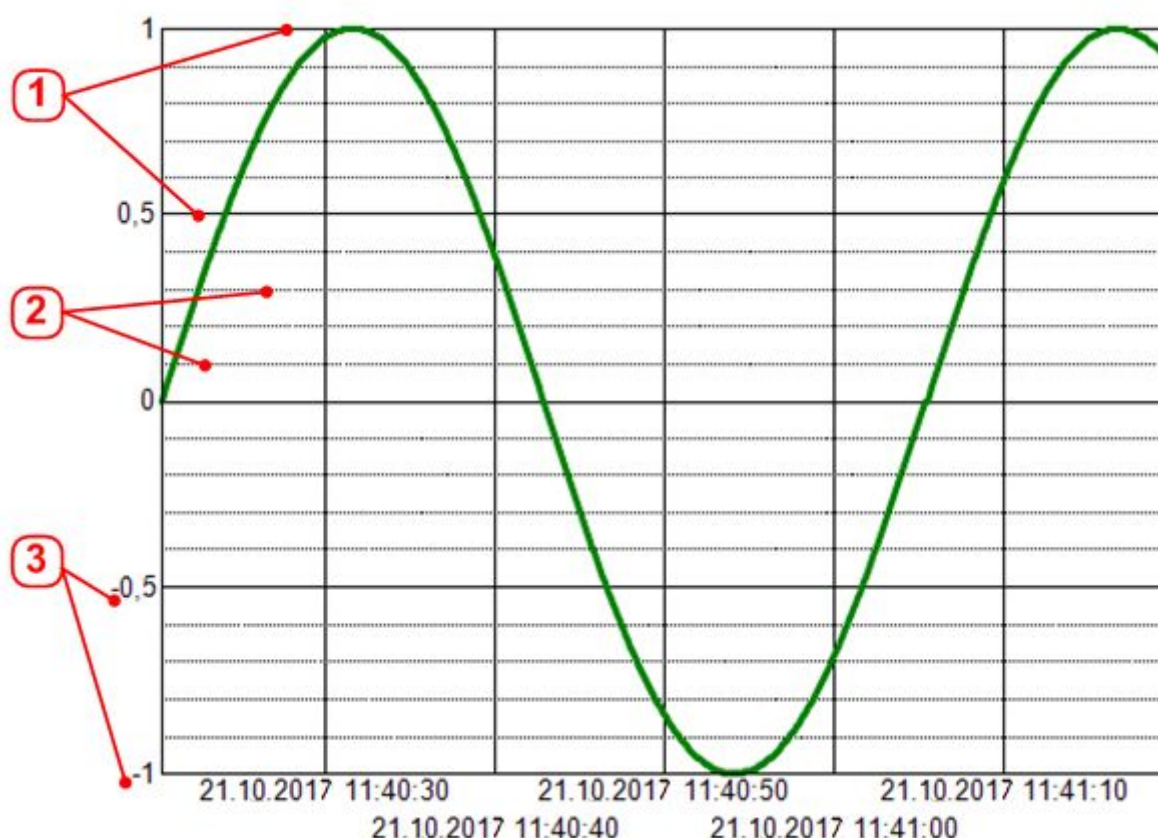
- 10 Нажимая на кнопку **Обновить отчет**, наблюдать как формируется тренд в отчете:



## Настройка осей значений X и Y

Для отображения на графике сетки первого и второго порядка установите для свойств **Ось значений (Y)** или **Ось значений (X)** следующие параметры:

- **Единица первого порядка** = 0.5. Единица измерения для маркеров координатной оси первого порядка
- **Единица второго порядка** = 0.1. Единица измерения для маркеров координатной оси второго порядка
- **Максимальное значение** = 1. Максимальное значение, представляемое на оси координат
- **Минимальное значение** = -1. Минимальное значение, представляемое на оси координат
- **Интервальный режим** - Да. Метки на координатной оси выставляются автоматически через равные интервалы
- **Сетка первого порядка** -> **Отображать линии сетки** - Да. Линии сетки первого порядка должны отрисовываться
- **Сетка второго порядка** -> **Отображать линии сетки** - Да. Линии сетки второго порядка должны отрисовываться
- **Сетка второго порядка** -> **Стиль линии** - Точечно-пунктирная. Стиль линий сетки второго порядка
- **Отображать метки** - Да. Метки на на оси координат должны отображаться.



- 1 – Линии сетки первого порядка
- 2 – Линии сетки второго порядка
- 3 – Метки на оси координат.



### Отображение значений тега на графике

Для того чтобы график не был перегружен изображениями значений тега, изменим некоторые параметры:

- **Апертура = 0.100**

В этом случае для построения используются только те значения тренда, абсолютная разность между которыми больше апертуры. Для задания апертуры следует в контекстном меню вида **Шаблон отчета** выбрать пункт **Источники данных и параметры** и в появившемся окне задать значение апертуры

- **Частота = 500**

В редакторе проекта щелкнуть по названию тега **Синус** и задать в строке **Frequency (Частота)** в столбце **Значение** – **500**.

Для размещения маркеров значений и самих значений на графике следует:

- 1 Выбрать из контекстного меню поля данных (**Синус\_Value\_Значение**) пункт **Свойства**

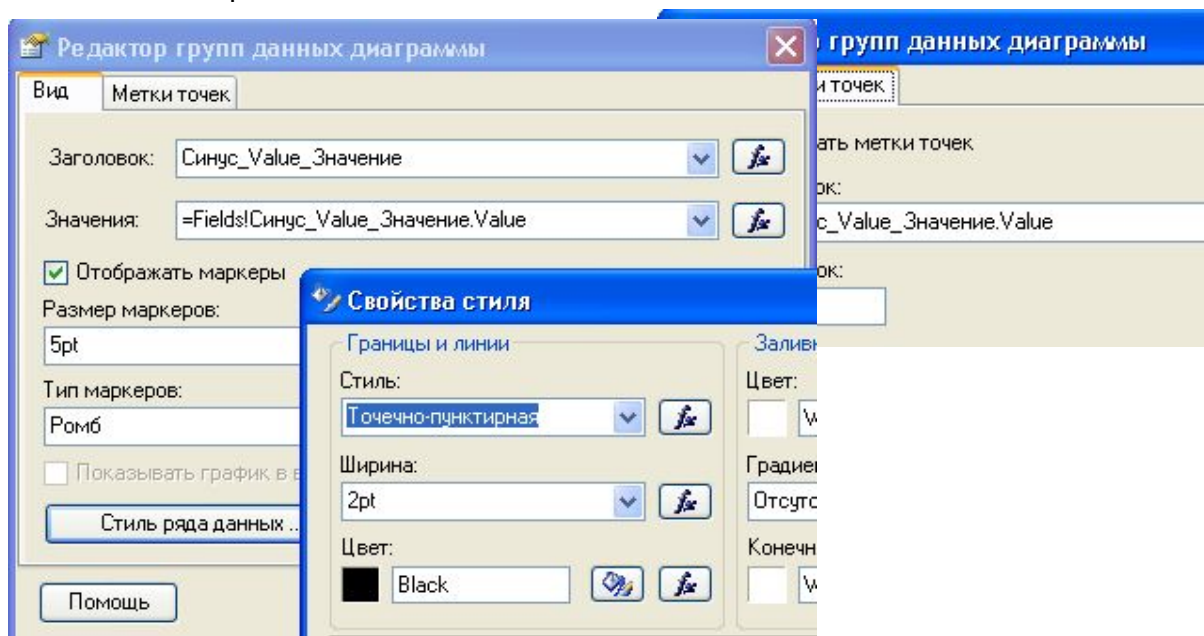
- 2 В появившемся окне **Редактор групп данных диаграммы** установить:

Закладка **Вид**:

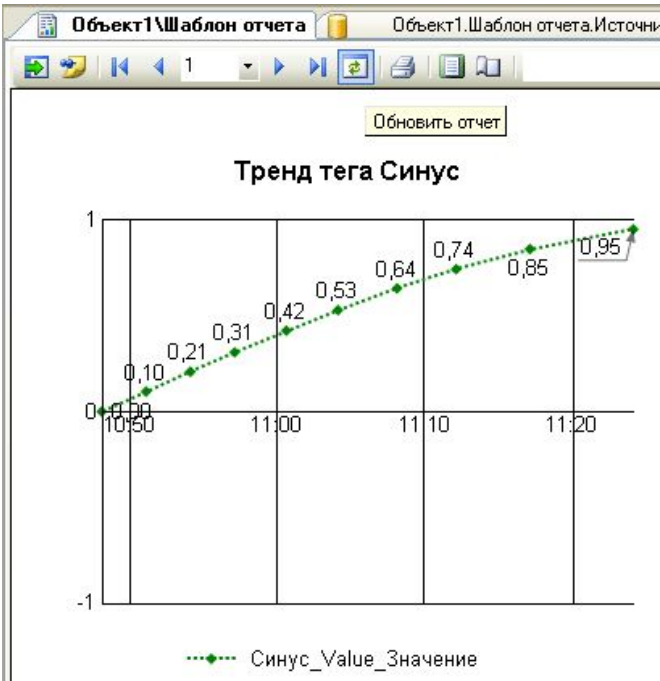
- Отображать маркеры – **поставить галочку**
- Размер маркеров – **5**
- Тип маркеров – **Ромб** (выбирается из списка)
- Стиль ряда данных – **Точечно-пунктирная, 2 pt**

Закладка **Метки точек**:

- Показывать метки точек – **поставить галочку**
- Формат меток – **0.00**



- 3 Запустить проект на выполнение и перейти в режим **Предварительный просмотр**

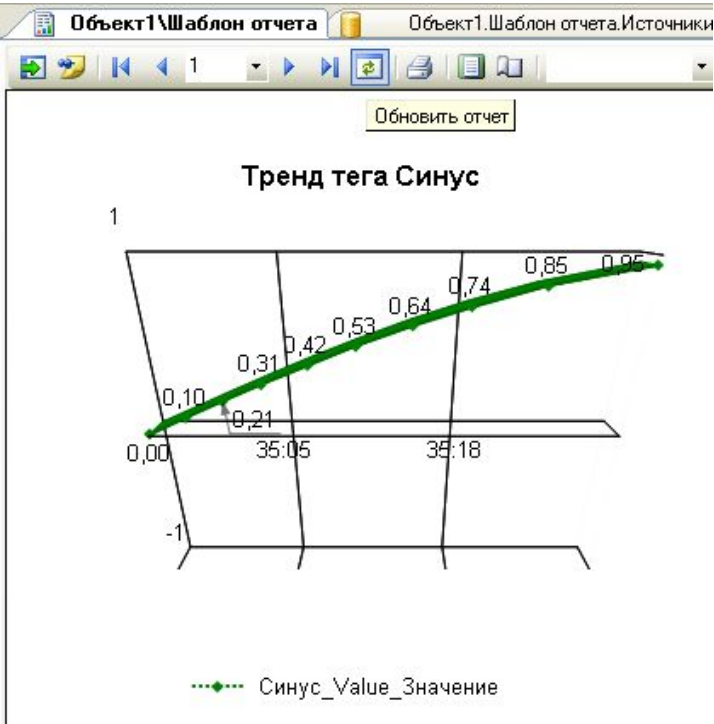


**Объемное изображение**

Для объемного изображения графика задайте значения свойств **3-х мерный вид** (на пример такие).

3-х мерный вид	
Толщина стенок	0
Затенение	Отсутствует
Группировать в кластер	Нет
Стиль отрисовки	Кубический
Горизонтальный поворот	0
Проекция	Перспективная
Используется	Да
Вертикальный поворот	25
Перспективность	25

Изображение тренда при заданных значениях будет следующим.



## 7.1.2.5.4 Круговые диаграммы

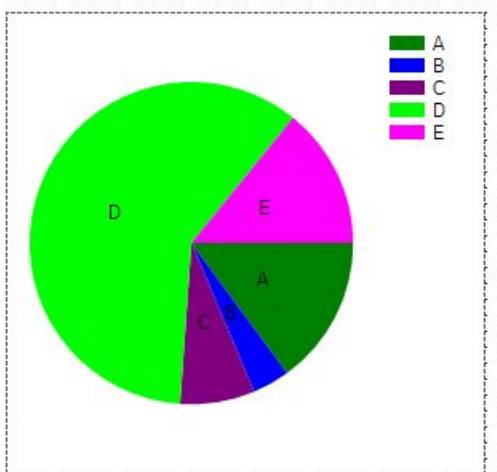
**Круговые диаграммы** отображают значения данных, как процентные части целого. Категории представляются отдельными сегментами. Размер каждого сегмента определяется соответствующим ему значением. Круговые диаграммы обычно используются для отображения процентных соотношений.

**Определение круговых диаграмм**

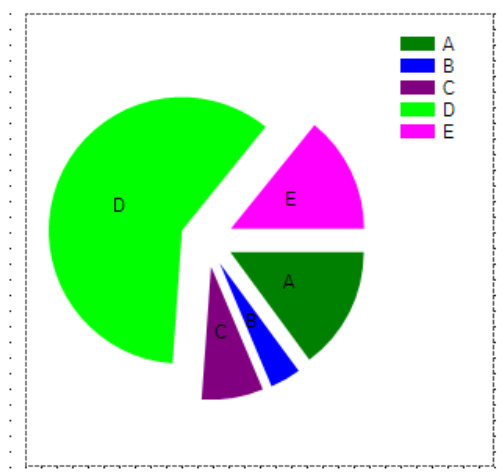
Область диаграммы	Обязательность	Описание
Данные	Обязательно	Значения в рядах данных определяют размер каждого сегмента круговой диаграммы. Несколько рядов значений отображаются как отдельные сегменты.
Группы категорий	Необязательно	Категории отображаются, как отдельные сегменты круга. В описании диаграммы отображается каждая категория.
Группы рядов	Необязательно	Ряды отображаются как отдельные сегменты круга. Каждый ряд также отображается в описании диаграммы. Настройки оси X и Y для круговых диаграмм не используются. Если существует несколько групп категорий или рядов, то метки групп отображаются в описании диаграммы.

**Типы круговых диаграмм**

В **DataRate** используются два типа круговых диаграмм: круговые диаграммы и разрезанные круговые диаграммы

**Круговая диаграмма**

**Круговая диаграмма** отображает значения полей данных, группы категорий, группы рядов в виде сегментов круга. Размер сегмента определяется значениями поля данных, как процентной долей от суммы всех значений.

**Разрезанная круговая диаграмма**

**Разрезанная круговая диаграмма** отображает долю каждого значения в одном целом, выделяя при этом отдельные значения. Она идентична обычной круговой диаграмме, за исключением того, что сегменты сдвинуты от центра круга. Результаты в таких диаграммах отображаются как пространства между сегментами диаграммы

## 7.1.2.5.5 Диаграммы с областями

**Диаграмма с областями** отображает значения полей данных в виде соединенных линий и точек с закрашенными ниже этой линии областями. Значения представлены высотой точки, измеренной по оси Y. По оси X откладываются метки категорий.

Диаграммы с областями обычно используются для сравнения меняющихся со временем значений.

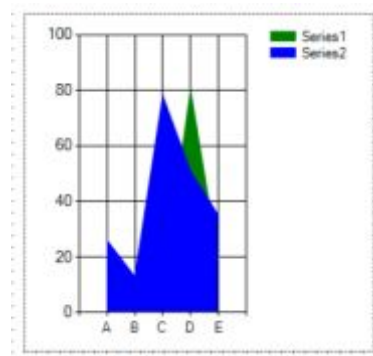
### Определение диаграмм с областями

Область диаграммы	Обязательность	Описание
Данные	Обязательно	Значения полей данных определяют высоту строк в этих последовательностях. Метки значений отображаются на оси Y. Каждая последовательность значений отображается как отдельная область.
Группы категорий	Необязательно	Категории отображаются в виде меток на оси X. Несколько групп вкладываются друг в друга. В диаграммах с областями категориями обычно являются отрезки времени.
Группы рядов	Необязательно	Последовательности в этих диаграммах отображаются в виде отдельных областей. Каждый ряд также отображается в условных обозначениях диаграммы.

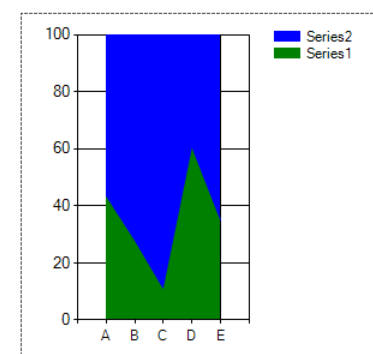
### Типы диаграмм с областями

В **DataRate** используются следующие типы диаграмм с областями:

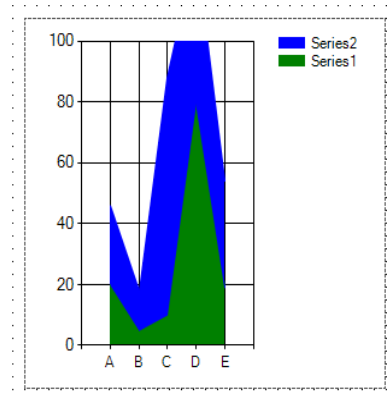
- **Диаграмма с областями** отображает значения полей данных в виде отдельных прямых линий с закрашенными под ними областями. Высота отображения каждой точки данных определяется ее значением.



- **Накопительная диаграмма с областями** отображает значения полей данных в виде отдельных прямых линий с закрашенными под ними областями. Высота первой последовательности основана на значении каждой точки ее данных. Высота дополнительных последовательностей основана на значениях точек данных в последовательности плюс значения точек данных нижестоящей последовательности. Высота самой верхней линии основана на общей сумме значений для каждой категории. Метки значений отображаются на оси Y.



- **Нормированная диаграмма с областями** отображает в виде отдельных прямых линий с закрашенными под ними областями. Каждая область отображается сверху нижестоящей области. Высота каждой точки данных основана на процентном отношении к общему количеству всех значений данной категории. Высота верхней линии всегда является общей высотой диаграммы. Ось Y показывает значения от 0 до 100.



### ВНИМАНИЕ!!!

В некоторых диаграммах с областями области, обозначающие последовательности данных, могут быть скрыты областями значений других последовательностей значений. Чтобы избежать подобного эффекта, выберите диаграмму с областями с накоплением или нормированную с областями, либо добавьте к ней трехмерные эффекты.

#### 7.1.2.5.6 Кольцевые диаграммы

**Кольцевые диаграммы** отображают значения полей данных в виде процентов от целого. Категории представлены отдельными срезами.

Кольцевые диаграммы обычно используются для отображения процентных отношений. По функциям кольцевые диаграммы совпадают с круговыми диаграммами.

#### Определение кольцевых диаграмм

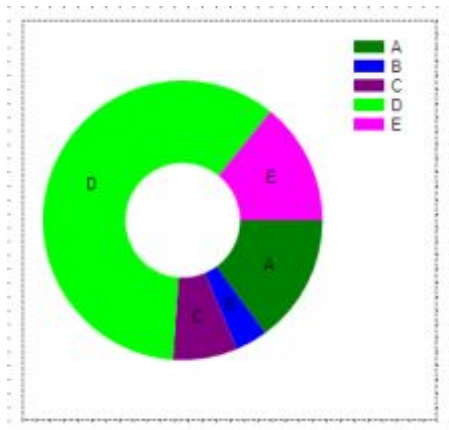
Область диаграммы	Обязательность	Описание
Данные	Обязательно	Значения в рядах данных определяют размер среза в кольцевой диаграмме. Несколько рядов значений отображаются в виде отдельных срезов.
Группы категорий	Необязательно	Категории отображаются в виде отдельных срезов кольцевой диаграммы. В условных обозначениях диаграммы отображается каждая категория.
Группы рядов	Необязательно	Ряды отображаются в виде отдельных срезов кольцевой диаграммы. Каждый ряд также отображается в условных обозначениях диаграммы.

#### Типы кольцевых диаграмм

В **DataRate** используются два типа кольцевых диаграмм: кольцевая диаграмма и разрезанная кольцевая диаграмма.

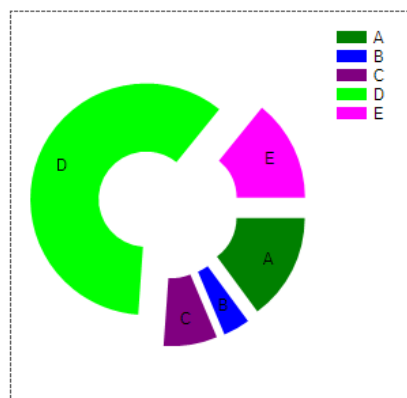


Кольцевая диаграмма



**Кольцевая диаграмма** отображает значения полей данных, группы рядов и ряды значений в виде срезов кольца. Размер среза определяется значением поля данных, как процентного отношения к сумме всех значений.

Разрезанная кольцевая диаграмма



**Разрезанная кольцевая диаграмма** идентична обычной кольцевой диаграмме во всем, кроме того, что срезы сдвинуты от центра кольца. Это приводит к появлению промежутков между срезами кольцевой диаграммы.

### ВНИМАНИЕ!!!

Параметры осей X (абсцисс) и Y (ординат) в кольцевых диаграммах не учитываются. Если существует несколько категорий или групп рядов, то в условных обозначениях диаграммы отображаются метки групп.

#### 7.1.2.6 Рисунок

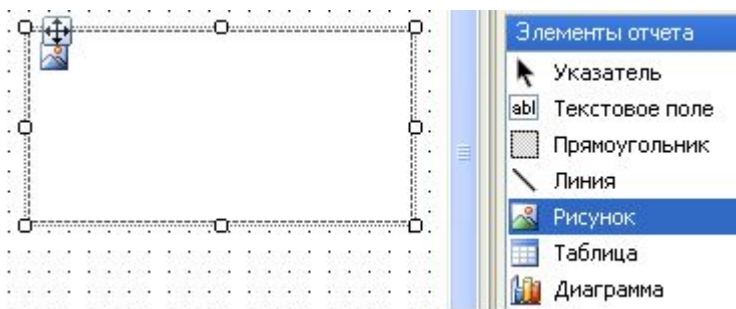
**Рисунок** – это элемент отчета, содержащий изображение, которое хранится во внешнем по отношению к отчету источнике или внедрено в отчет.

Рисунок можно также использовать в качестве фона некоторых элементов отчета. При использовании встроенного рисунка необходимо учесть, что данные изображений хранятся в определении отчета и не существуют как отдельные файлы. Встроенные изображения всегда доступны для отчета, но они не могут быть общими.

#### Размещение в отчете

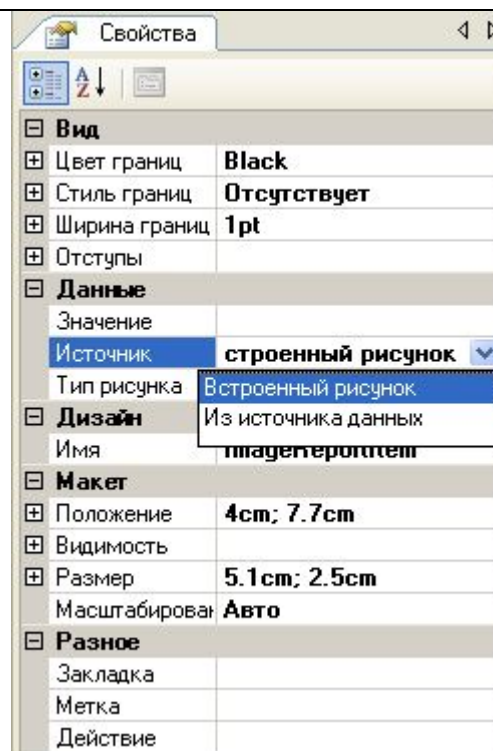
Для размещения изображения в отчете следует:

- 1 Перенести элемент **Рисунок** с панели элементов отчета в отчет
- 2 Задать источник рисунка – выбрать свойство **Источник** и указать встроенный рисунок или рисунок из источника данных.



## Свойства

- **Цвет, Стиль и Ширина границ** рисунка
- **Отступы** – отступы от границ рисунка
- **Значение** – статический текст или указание на поле источника данных, или выражение для вычисления значений данных
- **Источник** – источник для рисунка
- **Тип рисунка** – тип изображения (BMP, PNG и другие)
- **Имя** – наименование элемента отчета
- **Положение, Размер** – позиция верхнего левого и правого нижнего углов рисунка относительно контейнера (X – по горизонтали и Y – по вертикали)
- **Видимость** – указывает на то, что элемент отчета будет видим или скрыт. Если задано значение **Переключатель**, то это имя текстового элемента, используемого в качестве переключателя видимости для данного элемента отчета. Щелчок мышкой на назначенном текстовом элементе переключает состояние видимости каждого экземпляра данного элемента отчета. Если элемент, назначенный в качестве переключателя, будет скрыт, данный элемент также станет невидимым
- **Масштабирование** – способ масштабирования изображения, если его размер не соответствует размеру элемента отчета
- **Закладка** – ассоциированная с данным элементом закладка; может быть использована в качестве параметра для навигации
- **Метка** – метка для идентификации элемента отчета
- **Действие** – переход на закладку



## Фоновые рисунки

Элемент отчета **Рисунок** может использоваться, как фоновый рисунок в теле отчета, в прямоугольнике, в текстовом поле или в таблице. Фоновый рисунок имеет те же свойства, что и обычный рисунок.

Можно указать шаблон заполнения, по которому изображение сформирует фон для элемента отчета.

В **DataRate** используются следующие варианты расположения фонового рисунка:

- Замостить
- Без повторения
- Повторять по оси X
- Повторять по оси Y
- Выражение.

Для добавления фонового рисунка следует:

- 1 Открыть окно свойств элемента отчета и выбрать свойство **Фоновый рисунок**
- 2 Выбрать источник рисунка
- 3 Задать значение и расположение фонового рисунка.

### ВНИМАНИЕ!!!

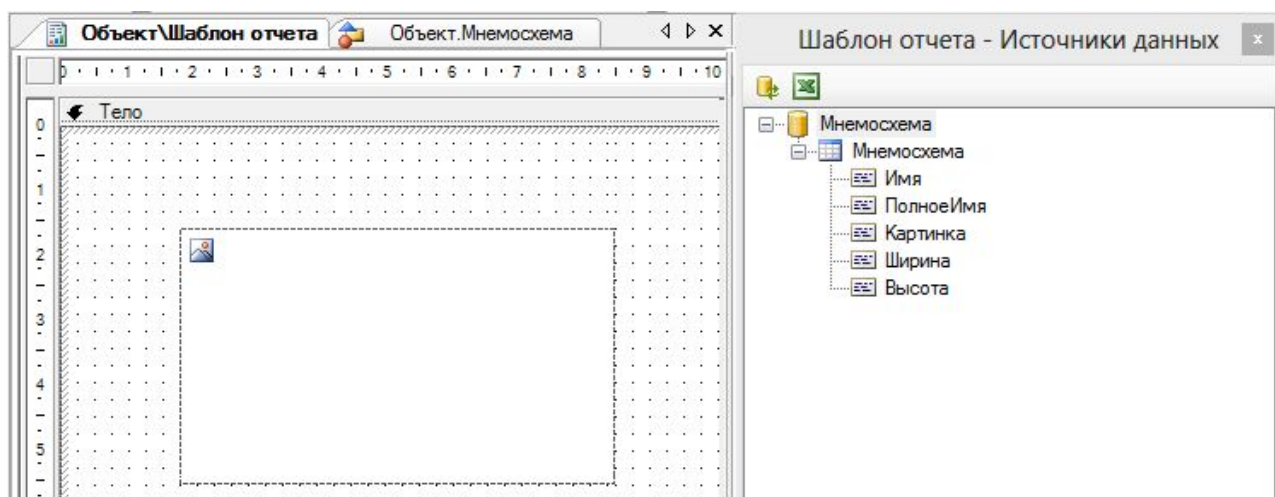
Для колонтитулов отчета тип источника фонового рисунка может быть только – Встроенный рисунок.

### Добавление изображения в отчет из источника данных

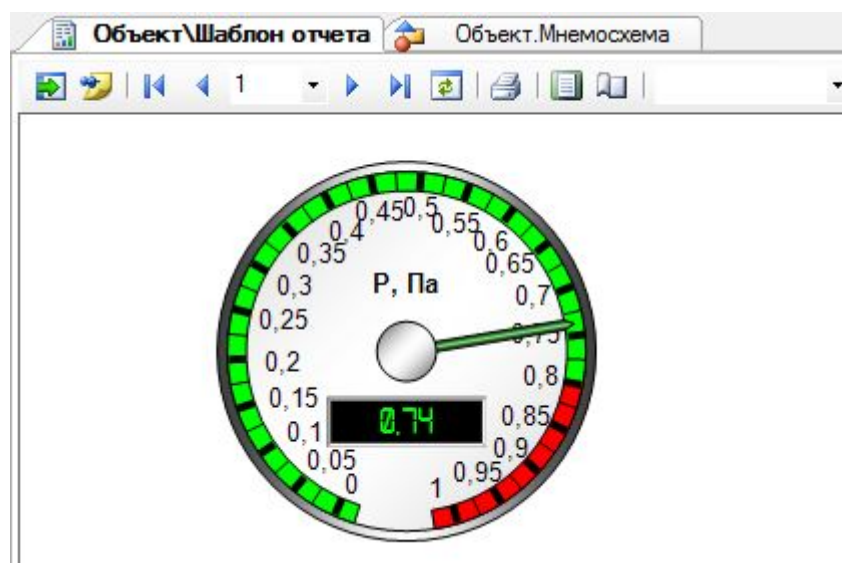
В отчет можно добавить изображения из источников данных, поддерживающих рисунки. Такими источниками могут быть **SQL-коннектор** и **Мнемосхема**.

Для отображения мнемосхемы в отчете следует:

- 1 Создать источник данных на основе элемента **Мнемосхема**
- 2 Создать элемент отчета **Рисунок**
- 3 Захватить и перетащить поле **Картинка** источника данных на созданный элемент отчета **Рисунок**
- 4 Выбрать из списка в свойстве **Источник** значение **Из источника данных**



- 5 Перейти в режим **Предварительный просмотр** и посмотреть, как мнемосхема отображается в отчете (пример отображения мнемосхемы в отчете приведен ниже).



### 7.1.3 Встроенные рисунки

Если требуется сохранить рисунок в определении отчета, рисунок должен иметь значение **Встроенный рисунок** свойства **Источник**. В этом случае дизайнер отчетов кодирует изображение и сохраняет его, как текст, в определении отчета. Использование встроенного рисунка обеспечивает отчету полный доступ к нему, но также увеличивает размер определения отчета.

Для встраивания рисунка в отчет предварительно создают коллекцию рисунков. Изображения из коллекции встроенных рисунков могут использоваться в качестве фоновых рисунков элемента отчета (текстовое поле, прямоугольник и таблица), секции отчета или как источник для элемента отчета **Рисунок**.

#### Как создать коллекцию рисунков отчета

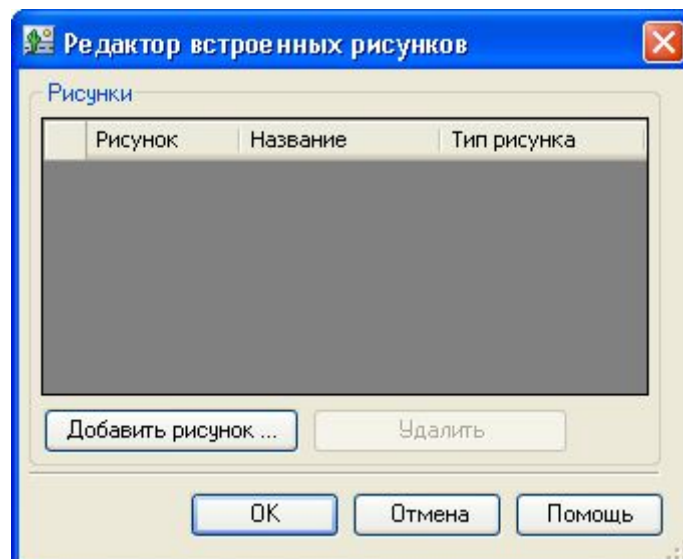
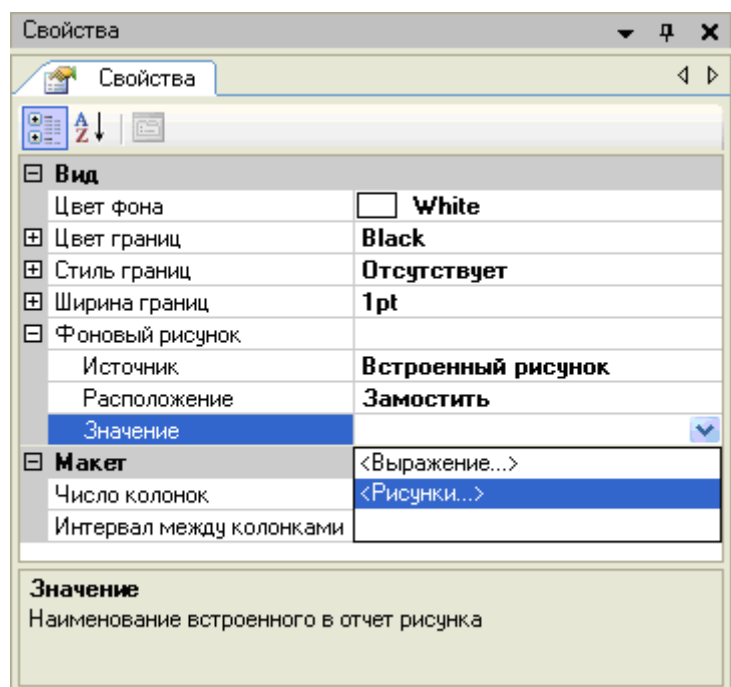
Для создания коллекции рисунков следует:

- 1 В шаблоне отчета выбрать свойство **Фоновый рисунок/Значение**
- 2 Щелкнуть мышью в строке **Рисунки**
- 3 В появившемся окне **Редактор встроенных рисунков** нажать на кнопку **Добавить рисунок...**
- 4 В диалоговом окне открытия файла выбрать исходный файл рисунка.

Имя встроенного рисунка может не совпадать с именем исходного файла, т.е. на имя накладываются ограничения в соответствии со спецификацией **RDL**.

Для загрузки встроенного рисунка в модуле отчетов

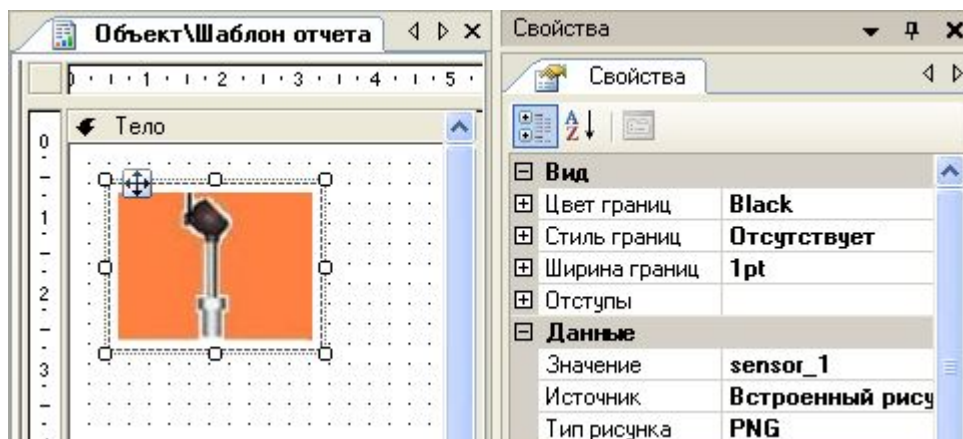
**DataRate** могут быть использованы файлы в формате **BMP**, **JPEG**, **GIF** и **PNG**.



### Как добавить встроенный рисунок в отчет

Для добавления рисунка в отчет следует:

- 1 Перенести элемент **Рисунок** с панели элементов отчета в отчет
- 2 В свойстве **Источник** (раздел свойств **Данные**) выбрать из списка значение **Встроенный рисунок**
- 3 В свойстве **Значение** (раздел свойств **Данные**) выбрать из списка значение **Картинка**
- 4 В появившемся окне **Редактор встроенных рисунков** выбрать встраиваемый рисунок (или добавить новый рисунок в коллекцию и затем включить его в отчет)



### 7.1.4 Форматирование элементов отчета

Форматирование элементов отчета позволяет сделать отчет более привлекательным, удобным и информативным. Форматирование включает:

- Стиль
- Условное форматирование
- Числовые значения и даты.

#### 7.1.4.1 Стиль

**Свойства стиля** соответствуют спецификации **RDL** (Report Definition Language) версии 2.0, существуют для каждого элемента отчета и включают: стиль и цвет границ, стиль шрифта, заполнение.

Некоторые стили доступны не для всех элементов отчета. Например, свойства стиля шрифта применимы только к элементу **Текстовое поле**, потому что только этот элемент может содержать текст.

Можно использовать одни и те же свойства стиля границ и свойства заполнения для всех аспектов элемента, либо применить различные стили к каждому аспекту элемента. Например, **Текстовое поле** имеет свойство **Стиль границ**, которое состоит из свойств **По умолчанию** для всех сторон и стилей границ всех четырех сторон текстового поля: **Левая**, **Правая**, **Верхняя**, **Нижняя**.



## Наличие свойств стиля у элементов отчета

Свойство стиля	Элемент отчета					
	Текстовое поле	Линия	Прямо-угольник	Рисунок	Таблица	Диаграмма
Цвет фона	Да	Да	Да	Да	Да	Да
Вид градиентной заливки фона	Нет	Нет	Нет	Нет	Нет	Да
Конечный цвет градиентной заливки фона	Нет	Нет	Нет	Нет	Нет	Да
Фоновый рисунок	Да	Нет	Да	Нет	Да	Нет
Цвет границ	Да	Нет	Да	Да	Да	Нет
Стиль границ	Да	Нет	Да	Да	Да	Да
Ширина границ	Да	Нет	Да	Да	Да	Да
Цвет текста	Да	Нет	Нет	Нет	Нет	Нет
Шрифт	Да	Нет	Нет	Нет	Нет	Нет
Формат	Да	Нет	Нет	Нет	Нет	Нет
Высота строки	Да	Нет	Нет	Нет	Нет	Нет
Отступы	Да	Нет	Нет	Да	Нет	Нет
Горизонтальное выравнивание	Да	Нет	Нет	Нет	Нет	Нет
Оформление текста	Да	Нет	Нет	Нет	Нет	Нет
Вертикальное выравнивание	Да	Нет	Нет	Нет	Нет	Нет
Цвет линии	Нет	Да	Нет	Нет	Нет	Нет
Стиль линии	Нет	Да	Нет	Нет	Нет	Нет
Толщина линии	Нет	Да	Нет	Нет	Нет	Нет

## Свойства стиля

Свойство стиля	Описание
Цвет фона	Цвет фона элемента отчета. Задается выражением или цветом RGB.
Вид градиентной заливки фона	Направление, в котором отображен фоновый градиент.
Конечный цвет градиентной заливки фона	Конечный цвет фонового градиента. Если он не задан, у элемента нет фонового градиента.
Фоновый рисунок	Рисунок, отображаемый в качестве фона элемента отчета. Может быть встроенным рисунком или рисунком из внешнего источника
Цвет границ	Задается выражением или цветом RGB, как цвет границ сторон элемента отчета и значения по умолчанию.
Стиль границ	Стиль границ элемента отчета. Задается как стиль границ сторон элемента отчета и значения по умолчанию. Стиль границ может быть задан выражением или наименованием стиля. Например: пунктирный, штриховой или сплошной.
Ширина границ	Ширина границ элемента отчета. Задается как ширина границ сторон элемента отчета и значения по умолчанию. Ширина границ может быть задана выражением или значением в сантиметрах (Cm), дюймах(in), миллиметрах(mm), пикселах(px), точках(1/72 дюйма).
Цвет текста	Цвет текста в элементе отчета. Цвет текста может быть задан выражением или цветом RGB.
Шрифт	Шрифт текста элемента отчета.
Наименование	Имя шрифта, используемого для текста в элементе. В свойстве <i>Наименование</i> шрифта представлен список всех шрифтов, доступных на локальном компьютере. При переносе шаблона отчета на другой компьютер выбранной шрифт может быть не установлен, тогда при отображении отчета будет использован шрифт Arial.
Размер	Размер шрифта в пунктах. Значение по умолчанию 10 pt. Диапазон значений от 1 pt до 200 pt.
Стиль	Стиль шрифта, используемого для текста в элементе, например курсив.
Толщина	Толщина шрифта, используемого для текста в элементе.
Формат	Применяемая к элементу строка форматирования платформы Microsoft .NET Framework, например \$ для валюты. В строке формата не должны использоваться функции форматирования строк, например FormatNumber. Для подробной информации о форматах см. Числовые значения и даты
Высота строки	Высота строки текста. Если не указана, устанавливается в соответствии с размером шрифта.
Отступы	Количество пробелов, вставляемых между границей элемента отчета и текстом или рисунком элемента. Значения: <b>Внизу, Слева, Справа, Вверху</b>
Горизонтальное выравнивание	Горизонтальное выравнивание текста в элементе отчета. Например: по левому краю, по правому краю, по центру.
Оформление текста	Указывает оформление для шрифта текста (например, подчеркнутый)
Вертикальное выравнивание	Вертикальное выравнивание текста в элементе отчета (по верхнему краю, по середине, по нижнему краю).

Свойство стиля	Описание
Цвет линии	Цвет элемента отчета линия. Цвет линии может быть задан выражением или цветом RGB.
Стиль линии	Стиль элемента отчета линия.
Толщина линии	Толщина элемента отчета линия.

#### 7.1.4.1 Условное форматирование

##### Контекстное изменение стиля элемента отчета

Отчет можно сконструировать таким образом, что к элементам отчета будут применяться различные стили в зависимости от данных отчета.

Чтобы сделать стили динамическими, используйте пункт **<Выражение...>** для свойств стиля элемента.

##### Пример

Отображение отрицательных значений красным цветом.

Для того чтобы в текстовом поле (например, поле **Temperature** источника данных) показывать отрицательные значения красным, откройте окно свойств и задайте следующее выражение для свойства **Цвет текста**:

```
=iif(Fields!Temperature.Value < 0, "Red", "Black")
```

##### Пример

Создание отчета с выделением четных и нечетных строк

Чтобы применить эффект выделения четных и нечетных строк (чередование цветов через строку) в табличном отчете, задайте следующее выражение для свойства **Цвет фона** в строке детализации для каждого текстового поля:

```
=iif(RowNumber(Nothing) Mod 2, "PaleGreen", "White")
```

#### 7.1.4.2 Числовые значения и даты

Для форматирования значений **DataRate** использует строки форматирования **Microsoft .NET Framework** и значение свойства **Формат**.

##### ВНИМАНИЕ!!!

Для форматирования используйте выражение. Если, например, значение текстового поля не задано выражением, т.е. не начинается со знака равенства (=), то текст в поле интерпретируется как статический текст и не форматируется.

##### Форматирование чисел

Следующая таблица содержит список стандартных строк форматирования **Microsoft .NET Framework** для числовых значений.

Строка формата	Имя
C или c	Денежный формат
E или e	Научный формат
F или f	Фиксированный формат
G или g	Общий формат
N или n	Числовой формат
P или p	Процентный формат
R или r	Полный обход
X или x	Шестнадцатеричный формат

В строки формата можно добавлять указатель точности, определяющий число цифр справа от десятичного разделителя. Например, указатель точности «#,###» определяет одну цифру в целой части и две - после десятичной запятой. Строка **NO** указателя точности преобразует число так, что после десятичного разделителя не будет ни одной цифры.

### Форматирование дат

Следующая таблица содержит список стандартных строк форматирования **Microsoft .NET Framework** для дат.

Строка формата	Имя
<b>d</b>	Короткая дата
<b>D</b>	Полная дата
<b>t</b>	Короткое время
<b>T</b>	Полное время
<b>f</b>	Полная дата-время (короткое время)
<b>F</b>	Полная дата-время (полное время)
<b>g</b>	Общая дата-время (короткое время)
<b>G</b>	Общая дата-время (полное время)
<b>M</b> или <b>m</b>	День месяца
<b>R</b> или <b>r</b>	Шаблон RFC1123
<b>Y</b> или <b>y</b>	Месяц года

Можно также использовать собственные строки форматирования, например, **dd/MM/yy**.

Дополнительные сведения о строках форматирования можно найти в разделе **Formatting Types** на веб-узле [msdn.microsoft.com](http://msdn.microsoft.com).

## 7.2 Разбивка отчета на страницы

Для управления разбивкой на страницы необходимо указать в определении отчета свойства, описывающие страницу. Способ разбивки на страницы зависит от содержания отчета. Наличие разрывов страниц зависит от формата генерации отчета.

По умолчанию элементы отчета не содержат разрывов страницы.

Разрывы страниц определяют размещение содержимого на страницах отчета. Можно устанавливать разрывы страницы до или после элементов отчета, задавая соответствующие свойства элементов. Разрыв страницы можно добавить в начале или конце прямоугольника, таблицы или диаграммы.

### Как добавить разрывы страницы

Для добавления разрыва страницы в начало или конец элемента отчета установите его свойство **Разрыв страницы перед** или **Разрыв страницы после** в значение **Да**.

### Публикация отчета и разрывы страниц

- При публикации отчета в формате **Excel** используются мягкие (или логические) переносы страниц, положение которых вычисляется во время выполнения. Страницы Excel рассчитываются как листы одной книги. Сгенерированные отчеты в формате Excel не ориентируются на физические страницы.

- Формат **XML** не поддерживают разбивку на страницы. Свойства, относящиеся к страницам, которые задаются в отчете, при просмотре отчета в этих форматах не учитываются.
- При публикации отчета в формате **Adobe PDF** и в формате изображения используют постраничный формат.

Постраничные отчеты (такие как PDF и изображение), создающие страницы одинакового размера для всего отчета, автоматически расставляют разрывы страниц.

Для добавления разрывов страниц на основе размера страницы используются следующие свойства Высота страницы и Ширина страницы. Они используются при подготовке отчетов в формате PDF или изображений для расстановки регулярных вхождений разрывов страниц, местоположение которых зависит от физических размеров.

### ВНИМАНИЕ!!!

**Ширина отчета может превышать ширину страницы.**

Если такой отчет генерируется в формат, поддерживающей задание размеров страниц, то результирующий отчет может быть расширен на несколько страниц (по горизонтали).

В том случае, когда необходимо, чтобы отчет занимал в ширину одну страницу, но результат содержит несколько страниц, проследите, чтобы ширина отчета не превышала ширину страницы.

## 7.3 Предварительный просмотр

Предварительный просмотр отчета позволяет отображать содержание отчета в соответствии с выбранным снимком данных.

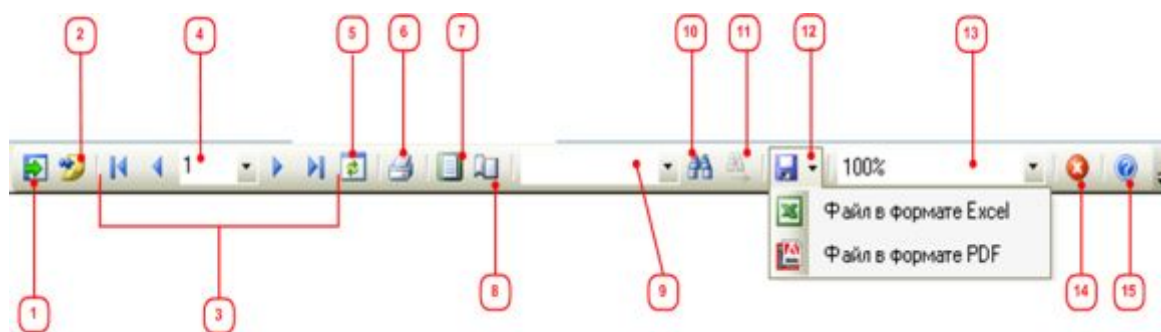
Дата/время	Наименование	Значение
23.03.2015 10:21:50	ПараметрЦелое	10
23.03.2015 10:21:40	ПараметрЛогический	Да
23.03.2015 10:21:30		
23.03.2015 10:21:20	ПараметрЦелое	10
23.03.2015 10:18:20		
23.03.2015 10:05:50	ПараметрЛогический	True
23.03.2015 10:05:40		
23.03.2015 10:05:30	ПараметрСтрока	qwerty фывап
23.03.2015 10:05:20	ПараметрДатаВремя	18.03.2015 10:55:02
23.03.2015 9:24:20		
23.03.2015 9:24:10	ПараметрByte	12
23.03.2015 9:08:30		
23.03.2015 9:08:20	ПараметрМульти	10
23.03.2015 9:08:10		
23.03.2015 9:07:40	ПараметрСписковый	10
23.03.2015 9:07:30		
23.03.2015 9:07:20	CreationTime	23.03.2015 10:21:56
23.03.2015 9:07:10		
22.03.2015 17:55:20	FullName	Объект.Шаблон отчета
22.03.2015 17:55:10		
	ПараметрByte	12



Область предварительного просмотра отчета включает:

- 1 **Поле просмотра отчета** отображает шаблон отчета
- 2 **Список снимков данных** содержит сохраненные снимки данных доступные к просмотру.
- 3 **Панель инструментов** предназначена для управления предварительным просмотром отчетов
- 4 **Список параметров отчета** позволяет отображать и редактировать параметры отчета

### Панель инструментов



Панель инструментов предварительного просмотра отчета содержит следующие элементы управления:

- 1 Кнопка **Показать/скрыть список снимков данных**. После нажатия на эту кнопку в левой части окна предварительного просмотра отчетов появляется список дат и времени создания снимков данных.  
При выборе снимка данных из списка в поле отображения отчета выводится отчет, соответствующий этому снимку данных.  
В списке снимков данных отображаются только существующие на данный момент снимки данных в соответствии с периодом хранения снимков данных отчета (Хранение снимков данных)
- 2 Кнопка **Показать/скрыть список параметров отчета**. После нажатия на эту кнопку в верхней части окна предварительного просмотра отчетов появляется список параметров (Параметры отчета)
- 3 **Элементы перемещения по отчету**. Позволяют перемещаться по страницам отчета. Перемещение возможно только в режиме предварительного просмотра
- 4 Выпадающий список **Выбор текущей страницы отчета**. Полное количество страниц отображается, если навести курсор на этот элемент панели инструментов
- 5 Кнопка **Обновить отчет**. Позволяет обновить содержимое отчета и список снимков данных
- 6 Кнопка **Печать отчета**. Позволяет вызвать стандартное диалоговое окно печати документа, в котором пользователь может выбрать принтер и страницы отчета которые должны быть напечатаны
- 7 Кнопка **Предварительный просмотр** печати отчета. Предварительный просмотр позволяет отобразить отчет в том виде, в котором он будет напечатан
- 8 Кнопка **Настройки страницы**. Позволяет вызвать стандартное диалоговое окно настроек страницы, которое позволяет выбрать размен бумаги и отступы для отчета
- 9 Поле ввода **Текст для поиска**. В этом поле Пользователь может ввести текст для его поиска в теле отчета. Чтобы перейти к следующему найденному слову необходимо нажать кнопку с надписью **"Найти далее"**

- 10** Кнопка **Найти**. Иницирует поиск в теле отчета текста, содержащегося в поле ввода текста для поиска  
Условия поиска:
- поиск чувствителен к регистру
  - поиск начинается с секции, выделенной в данный момент
  - шаблоны поиска и булевы операторы не поддерживаются
  - в поиске участвует только не скрытые элементы шаблона
- 11** Кнопка **Найти далее**. Предназначен для поиска текста поле того, как текст уже был один раз найден
- 12** Выпадающее меню **Экспортировать отчет**. Отчет может быть экспортирован в формат электронной таблицы **Excel** или файл формата **PDF**. После выбора экспорта пользователю будет предложено выбрать место размещения файла экспортируемого отчета
- 13** Выпадающий список **Масштаб отображения отчета**. Позволяет выбрать масштаб отображения отчета при просмотре
- 14** Кнопка **Остановить загрузку отчета**. Пользователь может остановить процесс загрузки отчета в случае большого количества снимков данных или большого размера отчета
- 15** Кнопка **Справка**. Осуществляет вызов данного раздела справочной системы DataRate.

## 7.4 Взаимодействие с пользователем

Для удобной навигации по отчету шаблон отчета может содержать **ссылки и закладки**.

Элементы отчета **Текстовое поле** и **Рисунок** могут содержать один из двух типов ссылки:

- **Ссылка-закладка** содержит ссылку на закладку в текущем отчете
- **Ссылка на URL-адрес (гиперссылка)** содержит ссылку на веб-страницу.

### Создание ссылки на закладку

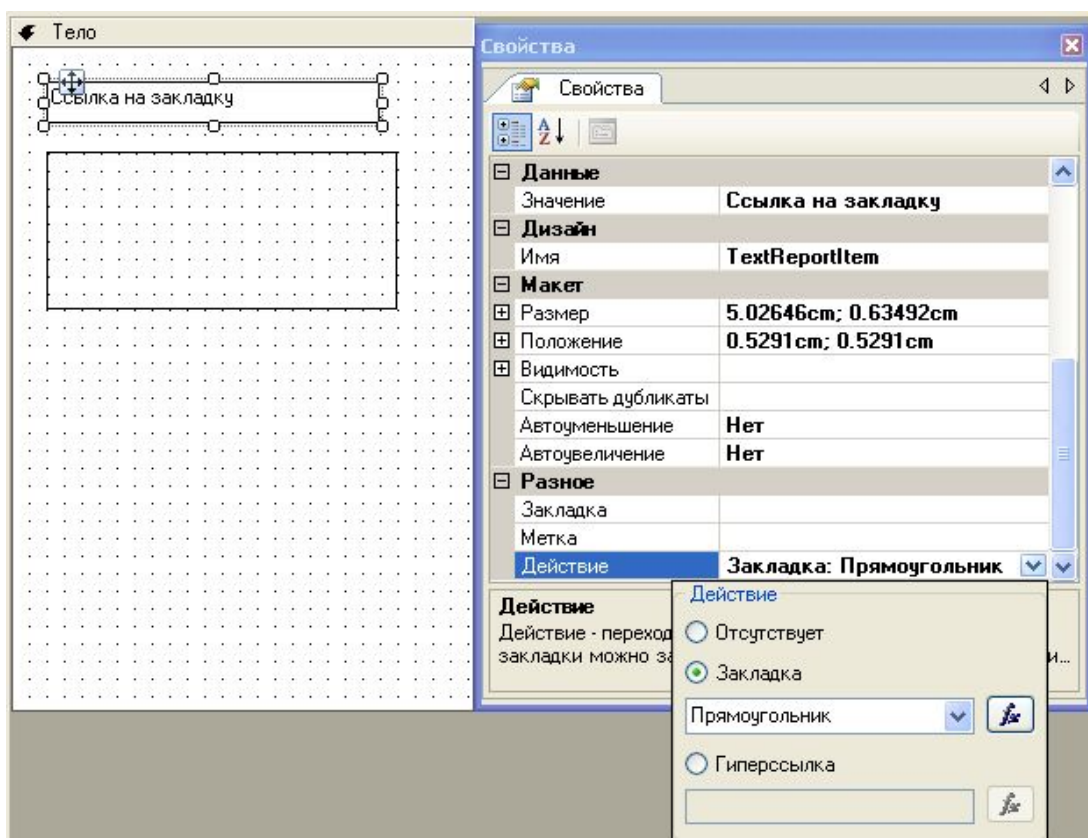
Для создания ссылки на закладку необходимо создать закладку на элементе отчета (свойство **Закладка**) и выбрать элемент, щелчок по которому будет инициировать переход на закладку (свойство **Действие**).

Для создания закладки следует:

- 1 Выделить элемент отчета
- 2 Задать значение свойства **Закладка** - ввести **наименование** закладки.  
Наименование закладки может быть любой строкой, уникальной для всего отчета. Если наименование закладки не уникально, то с ним будет связана первая совпадающая закладка.

Для создания ссылки на закладку следует:

- 1 Выбрать элемент **Текстовое поле** или **Рисунок**
- 2 Выбрать свойство **Действие** и раскрыть выпадающий список действий
- 3 Отметить действие **Закладка** и ввести наименование закладки.



Ссылка на закладку может быть задана выражением. Можно добавить несколько ссылок на одну закладку.

Наименования закладок устанавливаются на элементы отчета по отдельности.

### Создание гиперссылки

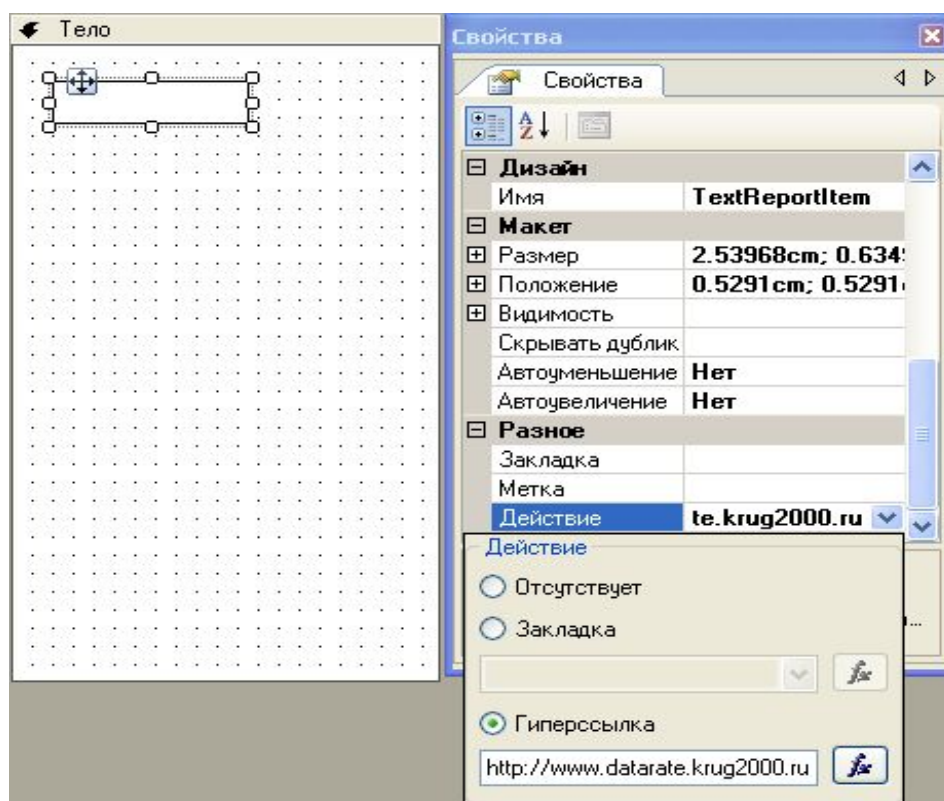
**Гиперссылкой** может служить **статический URL-адрес** или **выражение**, результатом вычисления которого будет такой адрес. Если в источнике данных есть поле, содержащее URL-адрес, выражение может содержать это поле.

### ВНИМАНИЕ!!!

**Гиперссылку можно задать только в элементах Текстовое поле и Рисунок.**

Для создания гиперссылки следует:

1. Выбрать элемент **Текстовое поле** или **Рисунок**, к которому нужно добавить гиперссылку
2. Выбрать свойство **Действие** и раскрыть выпадающий список действий
3. Отметить действие **Гиперссылка**
4. Ввести в поле ввода текст гиперссылки или выражение, результатом которого является **URL-адрес**.



### Управление видимостью элементов

Каждый элемент отчета имеет набор свойств **Видимость – Скрыть, Переключатель** – которые определяют, видим элемент или скрыт.

Эти свойства можно использовать для скрытия элементов отчета, условного скрытия данных, основанных на других данных отчета, а также щелчком по элементу делать другие элементы отчета видимыми или невидимыми.

Например, можно создать отчет с углубленной детализацией, который показывает при загрузке данные сводки, а строки подробностей отображает, когда пользователь щелкает определенное текстовое поле.

### ВНИМАНИЕ!!!

Управлять видимостью элемента отчета с помощью другого элемента возможно при следующих условиях:

- Управляющий и управляемый элементы находятся в теле отчета
- Управляющий элемент (управляет видимостью другого элемента) является текстовым полем.

### Пример

Элемент отчета управляет видимостью другого элемента отчета.

Для управления видимостью элемента отчета следует:

1. Выбрать элемент отчета, который должен "скрываться" – в примере элемент **Рисунок** с именем **ImageReportItem**
2. Создать управляющий элемент – **Текстовое поле** с именем **TextReportItem** и значением **Текст1**



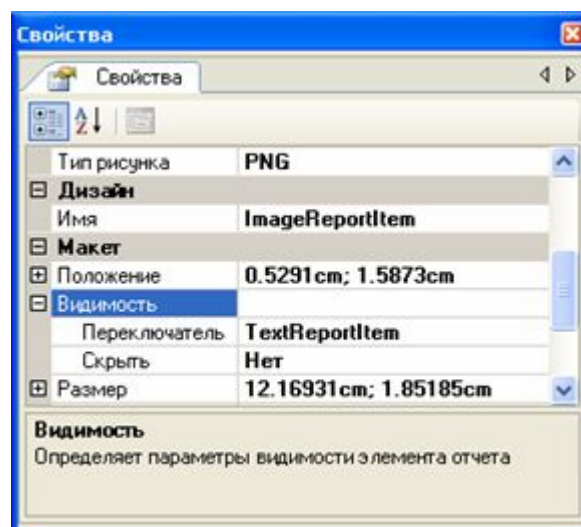
- 3 Для этого элемента выбрать из выпадающего списка свойства видимости **Переключатель** и задать управляющий элемент – в примере элемент **Текстовое поле** с именем **TextReportItem** и значением **Текст1**

- 4 Для проверки работы перейти в режим **Предварительный просмотр**. Рядом с управляющим элементом появится управляющее поле:

+ – управляемый элемент видим

- – управляемый элемент не видим.


Щелчок по этому управляющему полю изменяет видимость управляемого элемента – рисунка **ImageReportItem**.



### Схема документа

Схема документа отображает структуру документа и дает пользователям возможность перемещаться к конкретным областям отчета.

Когда осуществляется просмотр отчета, содержащего схему документа, рядом с отчетом появляется отдельная боковая панель. При выборе элементов схемы документа отчет обновляется и отображается та его область, которая соответствует выбранному элементу.

Для отображения схемы документа при предварительном просмотре нажмите на кнопку  на панели инструментов в окне предварительного просмотра.

### ВНИМАНИЕ!!!

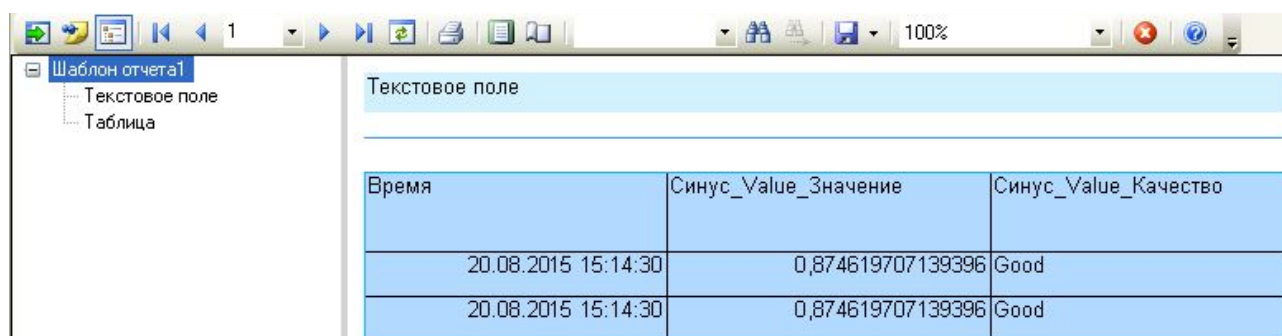
Схема документа формируется только в том случае, если в шаблоне отчета расставлены метки на элементы отчета, в противном случае схема не будет отображаться.



### Создание схемы документа

Для создания схемы документа задайте наименование "структурной единицы" документа (раздел, подраздел, пункт, подпункта и т.п.) в свойстве **Метка** тем элементам отчета, которые должны появиться в схеме.

Если у каких-либо элементов отчета уже есть метка, схема документа будет создана автоматически при просмотре отчета пользователем.



### Экспорт схемы документа

При экспорте отчета в формате **PDF** обращаются со схемой документа, как с панелью закладок.

Все элементы схемы документа размещаются один за другим на панели. Иерархия списка ссылок отсутствует.

При экспорте отчета в формате **Excel** схема документа представлена в виде именованной таблицы, содержащей иерархию ссылок. Разделы отчета помещаются на отдельные страницы, которые затем включаются в ту же книгу, что и схема документа.

### ВНИМАНИЕ!!!

При публикации отчета в виде в графических форматах, а также в XML схема документа не используется.

## 7.5 Автоматическое изменение размера и позиционирование

При размещении элементов на странице отчета дизайнер отчетов показывает область, которую занимает элемент. Однако, при генерации отчета области, которые занимают элементы отчета, могут изменяться.

Например, области данных таблиц и диаграмм динамически определяются в соответствии с данными, полученными из источника данных.

В **DataRate** приняты определенные правила автоматического изменения размеров и положения элементов отчета в процессе его генерации.

### Свойства, влияющие на размещение элементов отчета

В таблице перечислены основные элементы отчета и их свойства, которые влияют на то, как эти элементы выглядят на странице.

Элемент отчета	Примечания
Секция отчета	Свойства размера: <b>Ширина</b> и <b>Высота</b> . Ограниченное поле с белым фоном, появляющееся в макете, представляет собой секцию отчета. Для удаления нежелательного пустого пространства уменьшите площадь белого фона. Увеличение значения свойства <b>Ширина границ</b> секции отчета сокращает площадь, предназначенную для размещения элементов отчета.
Страница отчета	Для отчетов, которые посылаются в средства отображения или сохраняются в файл, поддерживающие физические страницы, используйте свойства: <b>Ширина страницы</b> , <b>Высота страницы</b> , а для печати еще и настройки страницы.
Текстовое поле	Свойства автоматического изменения размера в зависимости от содержимого: <b>Автоувеличение</b> и <b>Автоуменьшение</b> . Можно настроить размер по вертикали. Автоматическое увеличение по горизонтали не поддерживается.
Таблица	Число строк может увеличиваться в соответствии со связанным набором данных.
Прямоугольник	Действует как контейнер. При помещении в прямоугольник нескольких элементов они равноправны в отношении друг друга. Прямоугольники могут быть вложены для создания связей контейнеров «предок-потомок». Прямоугольники увеличиваются в соответствии с динамическими элементами в области данных.

## Правила расширения элементов отчета

Элементы отчета автоматически расширяются в соответствии с содержимым, сопоставленным из связанного набора данных, выражения или текста.

Элемент отчета растет в соответствии со своими данными и отодвигает другие элементы (находящиеся в том же секции отчета) по следующим правилам:

- Каждый элемент передвигается вниз для сохранения минимального пространства между собой и элементами, которые заканчиваются над ним.
- Каждый элемент передвигается вправо для сохранения минимального пространства между собой и элементами, заканчивающимися слева от него.
- Секции отчета растут в соответствии с ростом элементов потомков.

### ВНИМАНИЕ!!!

Сочетание такого поведения, свойств ширины и высоты элементов отчета, размеров тела отчета, определений высоты и ширины страницы, настроек границ, а также способа разбиения на страницы, — все это вместе определяет то, как элементы отчета размещаются на странице.

## 7.6 Использование выражений

**Выражения** используются в свойствах элементов отчета и предназначены для получения, вычисления, отображения, группирования, сортировки, параметризации и форматирования содержимого отчета.

### Синтаксис

Выражение начинается со знака равенства (=), за которым следует сочетание идентификаторов (наименований) полей, констант, функций и операторов.

Результатом выражения является единственное значение, вычисляемое «обработчиком отчета».

Например, полное имя отчета можно получить с помощью следующего выражения:

=Parameters!FullName.Value

### Классификация

В **DataRate** выражения классифицируются по их функциям в отчете:

- **Статистическая обработка данных** – вычисления суммы, среднего значения, процентной доли или произведения по отдельному набору строк
- **Условное форматирование** – форматирование текста или фона в зависимости от заданной логики
- **Условный текст** – заголовок отчета варьируется в зависимости от пользователя, запросившего отчет
- **Сцепление текста** – создание текста из нескольких констант и полей источника данных
- **Фильтрация данных** отчета
- **Группирование и сортировка** данных.
- **Динамическое формирование содержимого** верхнего и нижнего колонтитулов страницы.

#### 7.6.1 Создание выражений

Выражения создаются автоматически (например, при перетаскивании поля источника данных в элемент отчета) или могут задаваться пользователем.

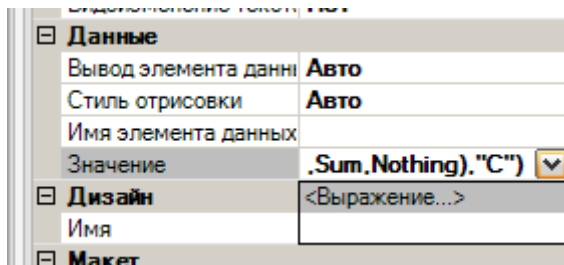
Выражения пишутся на языке **Microsoft Visual Basic for Applications**.

Выражение можно создать с помощью **Редактора выражений** или ввести непосредственно в текстовое поле, поле группирования или поле сортировки выражений.

### Редактор выражений

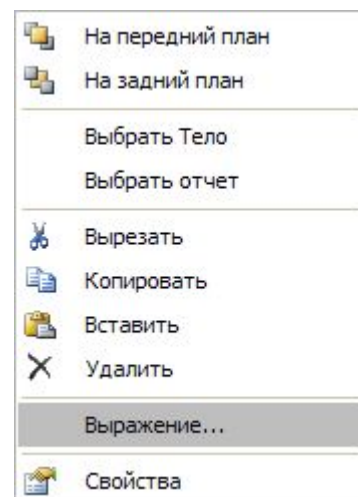
Редактор выражений предназначен для визуального создания выражений и может быть вызван одним из следующих способов:

- Выбрать **<Выражение...>** из раскрывающегося списка свойства элемента отчета или полей диалоговых окон

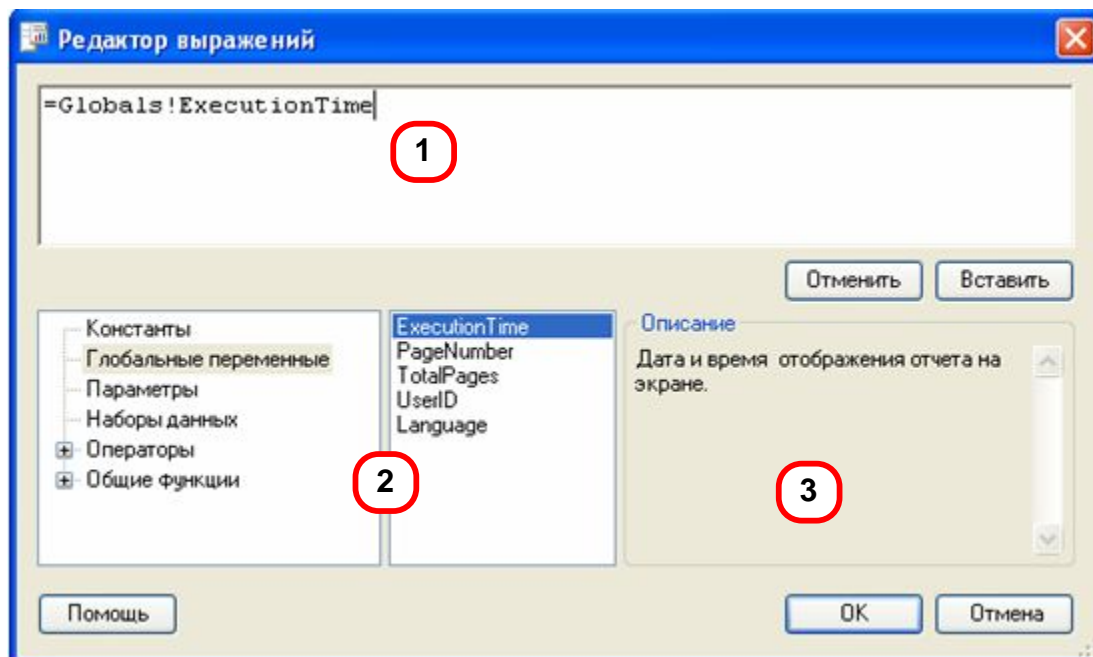


или

Выбрать пункт **Выражение ...** из контекстного меню элемента отчета (вызывается щелчком правой кнопкой мыши по элементу)



## Диалоговое окно Редактор выражений



Окно редактора выражений содержит:

- 1 **Область ввода и отображения** (код **Microsoft Visual Basic for Applications**)  
Область ввода выражения располагается на самом верху окна редактора и предназначена для ввода и отображения выражения.  
Если свойство элемента отчета содержит выражение, то при открытии редактора, выражение элемента отображается в данной области.  
Можно изменять выражение или вставлять в него функции, операторы, константы, поля и параметры из глобальной коллекции. Все сделанные изменения будут отображаться в области ввода.
- 2 **Дерево компонент и список функций или переменных**  
Дерево включает компоненты выражения такие, как константы, переменные, поля источников данных, **операторы**, функции. Выбор типа компонента определяет набор компонент, доступных для добавления в выражение.  
Например, выбор **Глобальные переменные** делает доступными переменные: **ExecutionTime** (дата и время создания отчета), **PageNumber** (номер текущей страницы отчета), **TotalPages** (число страниц в отчете), **UserID** (ID пользователя), **Language** (язык пользователя).
- 3 **Панель Описание**  
На этой панели отображается текст помощи по выбранному элементу, функции или **оператору**, а также примеры их использования.

Управляющие элементы окна – кнопки **Отменить**, **Вставить**, **Помощь**, **ОК**, **Отмена**:

- **Отменить** отменяет последние изменения сделанные в окне кода
- **Вставить** действует аналогично двойному клику на выбранном элементе выражения в дереве категорий, т.е. вставляет выбранный элемент выражения в окно кода в позицию курсора
- **Помощь** вызывает помощь по редактору выражений
- **ОК** сохраняет выражение элемента отчета и закрывает окно редактора
- **Отмена** закрывает окно без сохранения изменений

### Допустимые ссылки в выражениях

Приводимая ниже таблица показывает типы ссылок, которые можно включать в выражение: какие ссылки являются встроенными, а какие ссылки нужно определить для обработчика отчетов, чтобы вызовы функций могли быть разрешены во время генерации отчета.

Элементы	Описание функций и создание для них ссылок
Функции отчетов	Встроенные. Функции, предоставляющие статистические значения элементов отчета и прочие программы, поддерживающие статистическую обработку. Реализацию <b>Aggregate</b> предоставляет каждый поставщик данных. Дополнительные сведения приведены в разделе «Использование функций в выражениях».
Коллекции отчетов	Встроенные. <b>Globals, User, Fields, ReportItems, Datasources, Datasets</b> . Дополнительные сведения приведены в разделе «Применение глобальных коллекций в выражениях».
Параметры	Параметры, заданные для отчета.
Библиотека программ этапа исполнения Visual Basic for Applications	Встроенная. Дополнительные сведения в разделе «Visual Basic Run-Time Library Members» на узле <a href="http://msdn.microsoft.com">msdn.microsoft.com</a> .
<b>System.Math</b>	Встроенные. Дополнительные сведения приведены в разделе «Математические функции».
<b>System.Convert</b>	Встроенные. Дополнительные сведения приведены в разделе «Функции преобразования типов».

## 7.6.2 Работа с выражениями

### Использование выражений для доступа к полям источника данных

В выражении можно обратиться к полю источника данных и отобразить его значение в текстовом поле. Для связи полученного поля с элементом отчета выражение должно включать:

- Глобальную коллекцию **Fields**
- **Имя** поля
- Свойство **Value**.

При перетаскивании поля источника данных в свойство **Значение** элемента отчета **Текстовое поле** автоматически добавляется выражение:

**=Fields!<Имя поля>.Value.**

#### Пример 1

Задано выражение: **=Fields!LastName.Value**

Результат : строка – значение поля **LastName**. Например, **Иванов**

Если выражение используется для доступа к полю базы данных, то результатом будет множество строк столбца **LastName** таблицы базы данных. Например, **Иванов, Петров, Сидоров**.

#### Пример 2. Комбинирования констант и значений из нескольких полей данных

Задано выражение: **=Fields!FirstName.Value & " " & Fields!LastName.Value**

Результат : строка – значение поля **FirstName** пробел значение поля **LastName**. Например, **Иван Иванов**



## Использование встроенных функций

В выражениях можно обращаться к общим (встроенным) функциям. Например, статистические функции (**Min, Max, First, Last, Sum, Count** и т.п.) обычно используются в текстовых полях верхнего или нижнего колонтитулов (Секции отчета). Полный список встроенных функций, к которым можно обращаться из выражений, приведен в разделе Использование функций в выражениях.

### Пример 3. Подсчет количества и суммы значений поля источника данных

Подсчитаем количество и сумму значений тренда тега **Синус** (пример создания проекта с этим тегом приведен в разделе «Шаблон отчета/Диаграмма/Графики»).

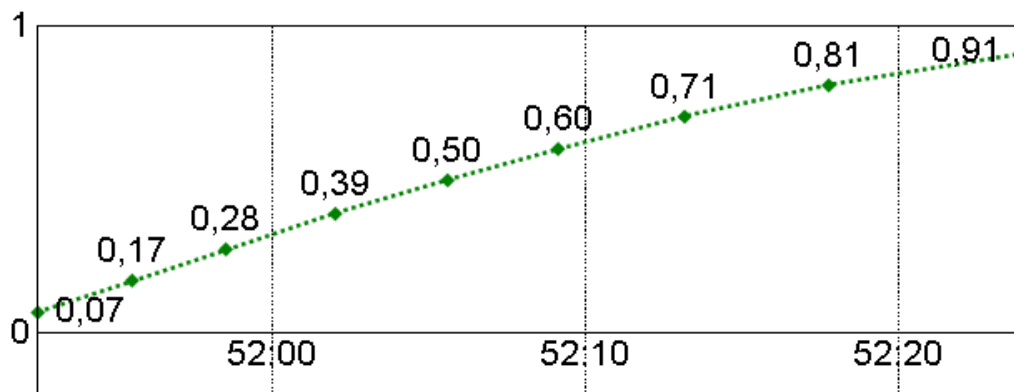
Добавим в отчет два элемента **Текстовое поле** и зададим для них свойство **Значение** с помощью выражения:

- Выражение 1:  
`= "Количество значений поля Синус_Value_Значение: " & Count(Fields!Синус_Value_Значение.Value)`
- Выражение 2:  
`= "Сумма значений поля Синус_Value_Значение : " & Sum(Fields!Синус_Value_Значение.Value)`

Результат :

Количество значений поля Синус\_Value\_Значение: 9

Сумма значений поля Синус\_Value\_Значение : 4,44125739207347



### Пример 4. Форматирование числовых значений

Для форматирования числового значения используется функции **Format** и строки форматирования **Microsoft .NET Framework** (смотрите раздел «Форматирование элементов отчета/Числовые значения и даты»).

Задано выражение:

`= "Сумма значений поля Синус_Value_Значение: " & Format(Sum(Fields!Синус_Value_Значение.Value), "#.##")`

Результат:

**Сумма значений поля Синус\_Value\_Значение: 4,44**

### Использование пространств имен .NET FRAMEWORK и системных пространств имен

В выражениях можно обращаться к встроенным пространствам имен библиотеки времени выполнения **Microsoft.VisualBasic**, а также к пространствам имен **.NET Framework: System.Convert** и **System.Math**.

Для обращения к другим пространствам имен среды **.NET Framework CLR** необходимо использовать полный идентификатор пространства имен, например **System.Text.StringBuilder**.

### Использование условного форматирования

Выражения могут использоваться для управления внешним видом элементов отчета.

#### Пример 5. Управление цветом текста в зависимости от значения поля источника данных

Задано выражение, которое определяет свойство **Color** текстового поля:

`=I if(Fields!TotalDue.Value < 10000,"Red","Black")`

Результат: если значение поля **TotalDue** меньше **10000**, то значение текстового поля отображается **красным цветом**, в противном случае – **черным цветом**.

### Использование выражений для фильтрации и сортировки

Выражения можно использовать для задания фильтров и сортировок данных. Фильтры применяются к данным после получения значений данных из источника данных.

#### **ВНИМАНИЕ!!!**

**Фильтрация и сортировка доступна только в диаграммах.**

### Использование встроенных коллекций

В выражениях можно обращаться к встроенным классам коллекций.

В дополнение к рассмотренной ранее коллекции **Fields** можно создавать выражения, обращающиеся к элементам следующих коллекций: **Globals**, **User**, **ReportItems**, **DataSets** и **DataSources**.

При использовании этих коллекций в выражении они предоставляют доступ к свойствам отчета и данным, определяемым динамически в процессе обработки отчета.

Дополнительные сведения приведены в разделе «Применение глобальных коллекций в выражениях».

### Использование параметров языковых стандартов

Для определения способа отображения отчета выражение может обращаться к параметрам языковых стандартов на клиентском компьютере через значение переменной **User.Language**.

Например, можно создать отчет, использующий выражение с запросом, различающимся в зависимости от значения языкового стандарта. Этот запрос может изменяться и извлекать локализованную информацию из разных столбцов в зависимости от установленного языка.

Выражения на основе **User.Language** можно также использовать в языковых параметрах отчета или его элементов.

### ВНИМАНИЕ!!!

Хотя в отчете можно изменять языковые параметры, необходимо следить за всеми проблемами отображения, которые могут при этом возникнуть. Например, изменение параметра языковых стандартов отчета может изменить в нем не только формат даты, но и формат денежной единицы. Если в поле валюты не использовано преобразование, это может привести к отображению в отчете неверного символа валюты.

Во избежание этого устанавливайте языковые настройки для отдельных элементов, которые нужно изменять, или настройте этот элемент на отображение денежных данных на определенном языке.

### 7.6.3 Применение глобальных коллекций в выражениях

В **DataRate** из выражений можно ссылаться на следующие глобальные коллекции:

- **DataSources**
- **DataSets**
- **Fields**
- **Globals**
- **ReportItems**
- **User**.

#### Как обратиться к коллекции из выражения

Чтобы обратиться к коллекции из выражения, необходимо использовать стандартный синтаксис языка **Visual Basic** для элемента коллекции.

Ниже приведены примеры синтаксиса обращения к элементам коллекции:

- **Collection!ObjectName.Property** или **Collection!ObjectName("Property")**  
=Fields!Sales.Value  
=Fields!Sales("Value")
- **Collection("ObjectName").Property**  
=Fields("Sales").Value
- **Collection("Member")**  
=User("Language")
- **Collection.Member**  
=User.Language

#### Сводка по глобальным коллекциям

В следующей таблице описаны все глобальные коллекции и указано, можно ли к коллекции обращаться из выражения.

Элементы коллекций **Globals** и **Users** возвращают тип **variant**.

Глобальная коллекция	Пример	Описание
<b>Fields</b>	<code>=Fields!Sales.Value</code>	Представляет коллекцию полей набора данных, доступных для отчета. Доступна после получения значений данных из источника данных.
<b>ReportItems</b>	<code>=ReportItems("Textbox1").Value</code>	Представляет коллекцию текстовых полей для элемента отчета, например для текстовых полей, содержащихся в области данных таблицы, ее верхнем или нижнем колонтитуле. Доступна во время обработки отчета.
<b>Globals</b>	<code>=Globals!ExecutionTime</code>	Представляет глобальные переменные, полезные для отчетов, например, время генерации отчета или номер страницы. Доступна всегда.
<b>User</b>	<code>=User.UserID</code> или <code>=User.Language</code>	Представляет коллекцию сведений о пользователе, выполняющем отчет, например языковые настройки или идентификатор пользователя. Доступна всегда. Параметр <b>User!UserID</b> часто используется для фильтрации результатов в отчетах.

Если нужно использовать глобальную переменную в выражении, требующем определенный тип данных, необходимо сначала привести переменную.

Например, чтобы преобразовать тип данных времени выполнения **variant** в формат **Date** следует использовать преобразование: `CDate(Globals!ExecutionTime)`

## Fields

В каждом наборе данных отчета содержится одна коллекция **Fields**. Это коллекция полей, которые отображаются, когда в окне **Источники данных** разворачивается поле данных.

Коллекция **Fields** обычно используется для отображения данных в текстовых полях отчета, но может также использоваться и для других элементов, свойств и функций отчета. Имена полей должны быть уникальными в пределах коллекции, но одно и то же имя поля может существовать в нескольких коллекциях.

Во время обработки отчета, по мере обработки по очереди каждого элемента отчета, коллекция **Fields**, связанная с текущим элементом отчета, привязывается к базовым исходным данным. Данные, возвращаемые при обработке отчета, для конкретного поля могут иметь одно значение или значение для каждой строки в наборе данных. Тип полученных данных — с одним значением или с несколькими — определяется типом элемента отчета, который содержит ссылку на поле. Для ссылок из области данных (таблица или диаграмма) возвращается значение для каждой строки в наборе данных. Для ссылок, которые не являются частью области данных, возвращается одиночное значение.

Для обработки коллекции поля в наборе данных с помощью статистической функции отчета включите параметр **scope**, который определяет набор данных.

Чтобы проверить взаимосвязи элементов отчета с набором данных, выберите элемент отчета и просмотрите его свойства.

Дополнительные сведения о функциях отчетов и параметре **scope** приведены в разделе Использование функций в выражениях.

### Свойства

По умолчанию элементы в коллекции **Fields** имеют два свойства: **Value** и **IsMissing**. При перетаскивании поля из окна **Источники данных** на элемент в шаблоне отчета, содержимому элемента отчета по умолчанию присваивается свойство поля **Value**.

Для области данных значению каждой строки в наборе данных присваивается значение `=Fields!FieldName.Value`.

Содержимому элемента отчета, который не является частью области данных, присваивается одиночное значение, например, `=First(Fields!FieldName.Value)`.

Свойство **IsMissing** указывает, содержится ли поле, определенное для набора данных, в полях, полученных из источника данных. Поле может быть проверено в выражении поля с помощью языка **Visual Basic**. Свойство **Value** для отсутствующих полей имеет значение **Nothing**.

Для проверки отсутствия поля можно использовать выражение:

```
=IIF(Fields!FieldName.IsMissing, "Field was not returned by query",
Fields!FieldName.Value))
```

Поскольку функция **IIF** языка **Visual Basic** вычисляет все элементы инструкции перед возвратом результата, обращение к отсутствующему полю может вызвать ошибку. В следующем примере показано, как можно выполнить вызов функции языка **Visual Basic**, обращающейся к полю, которое может отсутствовать:

```
=IIF(Sum(Fields!FieldName.Value) IS NOTHING, 0,
Sum(Fields!FieldName.Value))
```

### Globals

Коллекция **Globals** содержит глобальные переменные отчета. Следующая таблица описывает элементы коллекции **Globals**.

Элемент	Тип	Описание
ExecutionTime	<b>DateTime</b>	Дата и время, когда началось выполнение отчета.
PageNumber	<b>Integer</b>	Номер текущей страницы. Может использоваться только в верхних и нижних колонтитулах страницы.
ReportFolder	<b>String</b>	Полный путь к папке, содержащей отчет. Не включает в себя URL-адрес сервера отчетов.
TotalPages	<b>Integer</b>	Общее количество страниц в отчете. Может использоваться только в верхних и нижних колонтитулах страницы.

### Примеры использования коллекции Globals

- Выражение, помещенное в текстовое поле нижнего колонтитула отчета, отображает **номер страницы** и **общее количество страниц в отчете**  
`=Globals.PageNumber & " of " & Globals.TotalPages`
- Выражение отображает **текущую дату и время**. Время представляется в формате короткой даты платформы **Microsoft .NET Framework**  
`=Globals.ExecutionTime`



Если добавить форматирование "d", то будет отображаться только дата:  
`=Format(Globals.ExecutionTime, "d")`

### ReportItems

Коллекция **ReportItems** содержит текстовые поля в отчете. Элементы коллекции **ReportItems** имеют только одно свойство: **Value**. Значение элемента **ReportItems** может использоваться для отображения или вычисления данных из другого поля в отчете. Чтобы получить доступ к значению текущего текстового поля, можно использовать глобальную переменную **Me.Value** или просто **Value**, встроенную в язык **Visual Basic**.

#### ВНИМАНИЕ!!!

В функциях отчета, таких как **First** и во всех статистических функциях, используйте полный синтаксис – **Me.Value**.

#### Примеры использования коллекции ReportItems:

- Выражение, помещенное в текстовое поле, отображает значение текстового поля с именем **Textbox1**  
`=ReportItems("Textbox1").Value`
- Выражение, помещенное в свойство **Color** текстового поля **ReportItem**, отображает текст черным цветом, если значение > 0, в ином случае — красным:  
`=If(Me.Value > 0,"Black","Red")`
- Выражение, помещенное в текстовое поле верхнего или нижнего колонтитула, отображает первое значение текстового поля **LastName** на каждой странице подготовленного к просмотру отчета:  
`=First(ReportItems("LastName").Value)`

### User

Коллекция **User** содержит данные о пользователе, выполняющем отчет. Эту коллекцию можно использовать для фильтрации данных, появляющихся в отчете. Например, отобразить только данные текущего пользователя или отобразить идентификатор пользователя в заголовке отчета.

#### Пример использования коллекции User:

Следующее выражение, помещенное в текстовое поле, объединяет статический текст с идентификатором пользователя:

`= " for " & User!UserID`

Следующая таблица описывает элементы коллекции **User**.

Элемент	Тип	Описание
Language	String	Язык пользователя, выполняющего отчет. Например, en-US.
UserID	String	Идентификатор пользователя, выполняющего отчет. Если используется проверка подлинности Windows, это значение является учетной записью домена текущего пользователя. Значение определяется модулем безопасности служб Reporting Services, который может использовать проверку подлинности Windows или нестандартную проверку подлинности.

### 7.6.4 Примеры выражений

#### Работа с датами

- Функция **Today()** возвращает текущую дату.  
Эту функцию можно указать в текстовом поле для вывода даты формирования отчета или в качестве параметра фильтрации данных на основе текущей даты.
- Функция **DateAdd()** может оказаться полезной для вычисления диапазона дат на основе одного параметра.  
Например, выражение вычисляет дату, которая отстоит на 6 месяцев вперед от даты, содержащейся в параметре **StartDate**:  
`=DateAdd(DateInterval.Month, 6, Parameters!StartDate.Value)`
- Функция **Year()** выводит год указанной даты.  
Эту функцию можно использовать для группирования по датам или для вывода года в качестве метки для набора дат. Приведенное выражение возвращает год заданной группы дат заказов:  
`=Year(Fields!OrderDate.Value)`
- Для работы с датами можно также пользоваться функцией **Month()** и другими (описание приведено в разделе «Использование функций в выражениях/Функции даты и времени»).

#### Работа со строками

- Выражение возвращает значения двух полей – каждое на отдельной строке в одном текстовом поле:  
`=Fields!FirstName.Value & vbCrLf & Fields!LastName.Value`
- Функция **Format()** позволяет отформатировать даты и числа в символьном виде.  
Например, выражение отображает значения параметров **StartDate** и **EndDate** в длинном формате даты:  
`=Format(Parameters!StartDate.Value, "D") & " through " & Format(Parameters!EndDate.Value, "D")`

#### ВНИМАНИЕ!!!

Если значением текстового поля является *дата* или *число*, то для форматирования поля лучше назначить значение свойству **Format**, а не вызывать функцию **Format**.

- Функции **Right()**, **Len()** и **InStr()** полезны для извлечения вложенных строк.  
Например, для усечения строки вида *ДОМЕН\имя\_пользователя* до *имя\_пользователя*. Выражение возвращает часть параметра **User**, расположенную справа от символа обратной косой черты (\):  
`=Right(Parameters!User.Value, Len(Parameters!User.Value) - InStr(Parameters!User.Value, "\"))`
- Следующее выражение вычисляет то же значение, что и предыдущее, при помощи элементов класса **String** среды **.NET Framework** (вместо функций языка **Visual Basic for Applications**):  
`=Parameters!User.Value.Substring(Parameters!User.Value.IndexOf("\") + 1, Parameters!User.Value.Length-Parameters!User.Value.IndexOf("\")-1)`

- Функция **Join()** сцепляет выбранные значения параметра **MySelection** в одну строку, которая может использоваться в качестве выражения для значения текстового поля в элементе отчета:  
`= Join(Parameters!MySelection.Value)`
- Функция **Regex()** платформы **.NET Framework System.Text.RegularExpressions** полезна при изменении формата существующих строк.  
 Например, выражение изменяет формат десятизначного телефонного номера в поле **Phone** с формы 425-555-8080 на (425) 555-8080:  
`= System.Text.RegularExpressions.Regex.Replace(Fields!Phone.Value, "(\\d{3})[ -]*(\\d{3})[ -]*(\\d{4})", "($1) $2-$3")`

### Преобразования типов данных

- Выражение отображает количество значений, выбранных для многозначного параметра **MySelection**:  
`= CStr(Parameters!MySelection.Count)`

### Условный выбор

- Функция **lif** возвращает одно из двух значений, в зависимости от того, истинно ли указанное выражение.  
 Например, выражение вызывает функцию **lif**, которая возвращает логическое значение **True**, если значение **LineTotal** превышает **100**, в противном случае – **False**:  
`= Iif(Fields!LineTotal.Value > 100, True, False)`
- Выражение использует несколько функций **lif** (так называемые «вложенные **lif**»), возвращая в зависимости от значения **PctComplete** одно из трех значений:  
`= Iif(Fields!PctComplete.Value >= .8, "Green", Iif(Fields!PctComplete.Value >= .5, "Amber", "Red"))`
- Выражение в зависимости от значения **PctComplete** возвращает одно из трех значений, но использует для этого функцию **Switch()**, которая возвращает значение, связанное с первым выражением списка, результат которого равен **True**:  
`= Switch(Fields!PctComplete.Value >= .8, "Green", Fields!PctComplete.Value >= .5, "Amber", Fields!PctComplete.Value < .5, "Red")`
- Выражение проверяет значение поля **ImportantDate** и возвращает **Red**, если прошло больше недели, в противном случае – **Blue**:  
`= Iif(DateDiff("d", Fields!ImportantDate.Value, Now()) > 7, "Red", "Blue")`  
 Это выражение позволяет управлять свойством **Color** текстового поля в элементе отчета.
- Выражение проверяет значение поля **PhoneNumber** и, если его значение **Nothing**, возвращает строку **No Value** вместо значения **Nothing**:  
`= Iif(Fields!PhoneNumber.Value Is Nothing, "No Value", Fields!PhoneNumber.Value)`  
 Это выражение позволяет управлять содержимым текстового поля в элементе отчета.
- Выражение проверяет значение поля **Department** и возвращает либо имя вложенного отчета, либо **Nothing**:  
`= Iif(Fields!Department.Value = "Development", "EmployeeReport", Nothing)`  
 Это выражение может использоваться для условной детализации вложенных отчетов

- Выражение проверяет, содержится ли в поле значение **Nothing**:  
`=If(IsNothing(Fields!LargePhoto.Value),True,False)`  
Это выражение может использоваться для управления свойством **Скрыть** элемента **Рисунок**.

### Работа с данными

- Выражение выводит сумму данных **Order**:  
`=Sum(Fields!LineTotal.Value, "Order")`  
Это выражение может быть полезно в колонтитулах табличной группы.
- Выражение выводит номер строки, начиная с первой строки самой внешней области данных. Ключевое слово **Nothing** указывает, что функция начнет нумерацию строк с первой строки самой внешней области данных:  
`=RowNumber(Nothing)`  
Чтобы начать нумерацию с вложенной области данных, укажите в выражении ее имя.

### Параметры

- Выражение, при указании в качестве значения параметра по умолчанию, выбирает идентификатор пользователя, запустившего данный отчет:  
`=User!UserID`
- Выражение задает время **генерации снимка данных** :  
`=Parameters!CreationTime.Value`

### Колонтитулы

- Выражение отображает текущую дату. Выражение можно указать в текстовом поле в нижнем колонтитуле или в теле отчета. Дата представляется в формате короткой даты платформы **.NET Framework**:  
`=Format(Globals.ExecutionTime, "d")`
- Выражение, если его поместить в текстовое поле в нижнем колонтитуле отчета, выводит номер страницы и общее число страниц, содержащихся в отчете:  
`=Globals.PageNumber & " of " & Globals.TotalPages`
- Выражение, указанное для текстового поля в левой части верхнего колонтитула, выводит первое значение текстового поля **LastName**, содержащееся на данной странице:  
`=First(ReportItems("LastName").Value)`
- Выражение, указанное для текстового поля в правой части верхнего колонтитула, выводит последнее значение текстового поля **LastName**, содержащееся на данной странице:  
`=Last(ReportItems("LastName").Value)`
- Выражение, помещенное в верхний или нижний колонтитул, выводит в текстовом поле сумму значений поля **Cost**:  
`=Sum(ReportItems("Cost").Value)`

### ВНИМАНИЕ!!!

При ссылке из колонтитула в выражении может быть указан только один элемент отчета.

### Разрывы страниц

- Выражение, указанное в качестве выражения группы, назначает номер каждому набору из 25 строк. Если для этой группы определен разрыв страницы, данное выражение будет выводить его через каждые 25 строк.

`=Int((RowNumber(Nothing)-1)/25)`

### Изменение свойств

#### Форматирование

- Выражение, указанное в свойстве **Color** текстового поля, изменяет цвет текста в зависимости от значения поля **Profit**:

`=Iif(Fields!Profit.Value < 0, "Red", "Black")`

- При указании следующего выражения в свойстве **BackgroundColor** элемента отчета в области данных цвет фона каждой строки изменяется от бледно-зеленого до белого:

`=Iif(RowNumber(Nothing) Mod 2, "PaleGreen", "White")`

Возможно, что при использовании выражения для конкретной области понадобится указать набор данных для статистической функции:

`=Iif(RowNumber("Employees") Mod 2, "PaleGreen", "White")`

#### Видимость

- Выражение для определения видимости строк. Строки отображаются только для значений поля **PctQuota** больших 90%:

`=Iif(Fields!PctQuota.Value > .9, False, True)`

- Следующее выражение при установленных свойствах таблицы **Видимость** и **Скрытый** показывает таблицу, только если в ней больше 12 строк:

`=IIF(CountRows() > 12,true,false)`

### 7.6.5 Использование функций в выражениях

Многие выражения в отчете содержат вызовы функций. Они позволяют форматировать данные, применять логические операции и производить доступ к метаданным отчета.

**DataRate** обеспечивает работу с выражениями, использующими функции из библиотеки **Visual Basic** этапа исполнения, а также из пространств имен **System.Convert** и **System.Math**. Допустимо также использование классов платформы **Microsoft .NET Framework**, включая **System.Text.RegularExpressions**.

Дополнительные сведения о функциях языка **Visual Basic for Applications**, поддерживаемых в выражениях, можно найти в документации «Visual Basic Run-Time Library» на веб-узле [msdn.microsoft.com](http://msdn.microsoft.com).

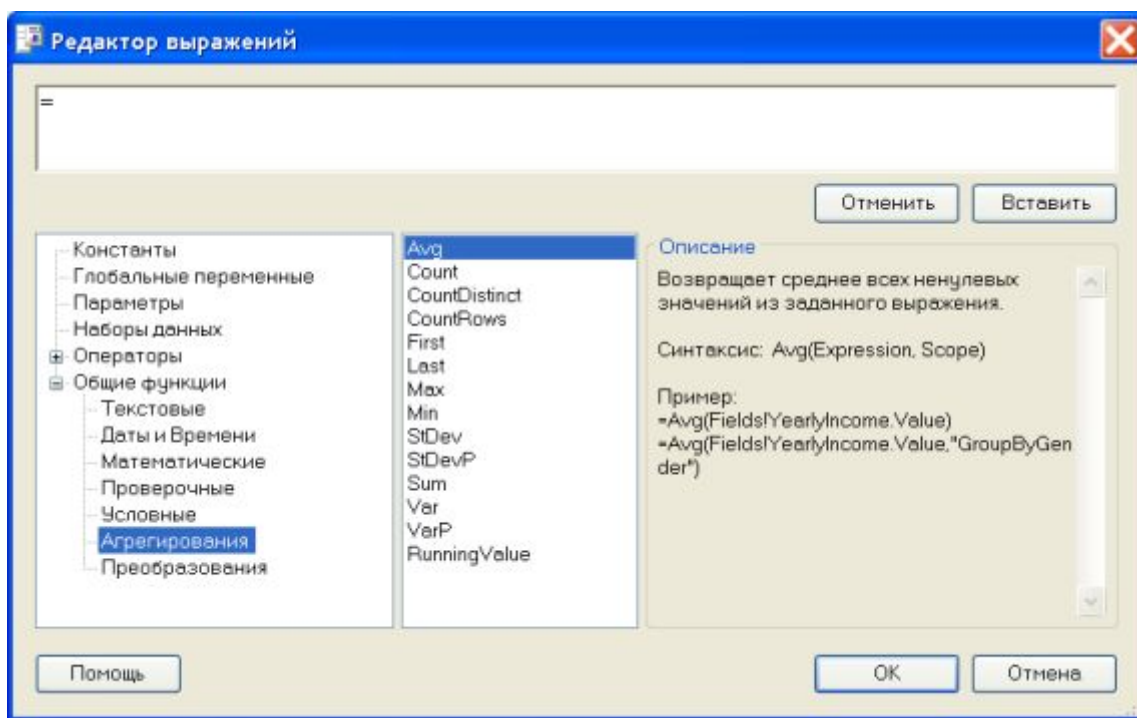
Для манипуляции значениями данных отчета в выражениях можно использовать общие (встроенные) функции.

В редакторе выражений доступны следующие типы встроенных функций:

- Функции работы со строками
- Функции даты и времени



- Математические функции
- Функции проверки типов
- Функции выполнения условия
- Функции преобразования
- Функции агрегирования



### 7.6.5.1 Функции работы со строками

Следующие таблицы описывают функции форматирования текста. Функции форматирования можно использовать в выражениях для любого элемента отчета.

#### Функции преобразования символа в код символа и обратно

Функция	Описание	Пример использования
<b>Asc</b>	Возвращает целочисленное, значение представляющее код символа.	<code>=Asc(Fields!Description.Value)</code>
<b>AscW</b>	Возвращает целочисленное, значение представляющее код символа.	<code>=AscW(Fields!Description.Value)</code>
<b>Chr</b>	Возвращает символ, соответствующий введенному коду.	<code>=Chr(65)</code>
<b>ChrW</b>	Возвращает символ, соответствующий введенному коду.	<code>=ChrW(241)</code>

#### Функции изменения регистра

Функция	Описание	Пример использования
<b>LCase</b>	Возвращает строку, приведенную к нижнему регистру.	<code>=LCase(Fields!Description.Value)</code>
<b>UCase</b>	Возвращает строку или символ, приведенный к верхнему регистру.	<code>=UCase(Fields!Description.Value)</code>

## Функции работы с пробелами

Функция	Описание	Пример использования
<b>LTrim</b>	Возвращает строку, содержащую копию заданной строки без символов пробелов в начале строки.	=LTrim(Fields!Description.Value)
<b>RTrim</b>	Возвращает строку, содержащую копию заданной строки без символов пробелов с правого края строки.	=RTrim(Fields!Description.Value)
<b>Trim</b>	Возвращает строку, содержащую копию заданной строки без символов пробелов с обеих сторон строки.	=Trim(Fields!Description.Value)
<b>Space</b>	Возвращает строку, содержащую заданное количество пробелов.	=Space(3)

## Функции форматирования

Функция	Описание	Пример использования
<b>Format</b>	Возвращает форматированную строку.	=Format(Globals!ExecutionTime, "Long Date")
<b>FormatCurrency</b>	Возвращает выражение, форматированное в денежный формат определенный в панели управления.	=FormatCurrency(Fields!YearlyIncome.Value,0)
<b>FormatDateTime</b>	Возвращает строковое выражение, представляющее число в формате даты/времени.	=FormatDateTime(Fields!BirthDate.Value, DateFormat.ShortDate)
<b>FormatNumber</b>	Возвращает выражение форматированное как число.	=FormatNumber(Fields!Weight.Value,2)
<b>FormatPercent</b>	Возвращает выражение, форматированное как процент и оканчивающееся символом %.	=FormatPercent(Fields!Sales.Value/Sum(Fields!Sales.Value, "DataSet1"),0)

## Функции поиска и фильтрации

Функция	Описание	Пример использования
<b>GetChar</b>	Возвращает символ в строке по заданному индексу.	<code>=GetChar(Fields!Description.Value, 5)</code>
<b>InStr</b>	Возвращает индекс начальной позиции первого вхождения строки в другой строке.	<code>=InStr(Fields!Description.Value, "car")</code>
<b>InStrRev</b>	Возвращает позицию первого вхождения одной строки в другой, начиная с правого края строки.	<code>=InStrRev(Fields!Description.Value, "car")</code>
<b>Left</b>	Возвращает строку, содержащую определенное количество символов с левого края строки.	<code>=Left(Fields!Description.Value,4)</code>
<b>Right</b>	Возвращает строку, содержащую заданное количество символов с правой стороны строки.	<code>=Right(Fields!Description.Value,4)</code>
<b>Len</b>	Возвращает число содержащее количество символов в строке или количество байт необходимое для сохранения строки.	<code>=Len(Fields!Description.Value)</code>
<b>Mid</b>	Возвращает строку, содержащую заданное количество символов из строки.	<code>=Mid(Fields!Description.Value,3,4)</code>
<b>Filter</b>	Возвращает массив, содержащий строки соответствующие критерию. Параметры: <ul style="list-style-type: none"> <li>• <b>sourcearray</b> - одномерный массив, элементы которого являются строками.</li> <li>• <b>match</b> - образец поиска. Строка, вхождение которой ищется в каждом элементе исходного массива.</li> <li>• <b>include</b> - необязательный аргумент булевого типа. По умолчанию имеет значение True, означающее, что элементы, удовлетворяющие образцу, входят в результирующий массив. Если задано значение False, то результирующий массив составляется из элементов, не удовлетворяющих образцу.</li> <li>• <b>compare</b> - вид сравнения: vbBinaryCompare – двоичное побитовое, vbTextCompare – текстовое.</li> </ul>	<code>=Filter(Parameters!MultivalueParameter.Value, "3", True, vbBinaryCompare)</code>

Функция	Описание	Пример использования
<b>Split</b>	<p>Возвращает массив, содержащий определенное количество подстрок. Предполагается, что исходная строка состоит из элементов (подстрок), разделенных специальными символами-разделителями.</p> <p>Параметры:</p> <ul style="list-style-type: none"> <li>• <b>expression</b> – строковое выражение, результат которого задает строку-источник, состоящую из символов и разделителей</li> <li>• <b>delimiter</b> – строка, задающая последовательность символов, используемых в качестве разделителя. Если параметр опущен, то по умолчанию в качестве разделителя выступает пробел</li> <li>• <b>limit</b> – необязательный параметр, позволяющий ограничить число возвращаемых элементов. По умолчанию его значение равно -1, означающее выделение всех элементов</li> <li>• <b>compare</b> – необязательный параметр, имеющий стандартный смысл во всех операциях над строками (vbBinaryCompare – двоичное побитовое, vbTextCompare – текстовое)</li> </ul>	<code>=Split(Fields!ListWithCommas.Value, ",")</code>

#### Другие функции

Функция	Описание	Пример использования
<b>Join</b>	<p>Восстанавливает строку по ее элементам, хранящимся в массиве, добавляя разделители в момент их объединения.</p> <p>Если второй аргумент функции разделитель опущен, то элементы разделяются пробелами.</p>	<code>=Join(Parameters!MultivalueParameter.Value, ";")</code>
<b>Replace</b>	Возвращает строку, в которой заданные подстроки были заменены другими строками определенное количество раз.	<code>=Replace(Fields!Description.Value, "tube", "headlight")</code>

Функция	Описание	Пример использования
<b>StrReverse</b>	Возвращает строку, в которой инвертирован порядок символов.	<code>=StrReverse(Fields!Description.Value)</code>
<b>StrComp</b>	Возвращает -1, 0 или 1 основываясь на результате сравнения строк.	<code>=StrComp(Fields!Description.Value,First(Fields!Description.Value))</code>
<b>StrConv</b>	<p>Возвращает строковое значение, преобразованное указанным способом. Функция StrConv имеет следующие аргументы:</p> <ul style="list-style-type: none"> <li>• строка - преобразуемое строковое выражение. Обязательный аргумент</li> <li>• тип_преобразования - сумма значений (Integer), определяющих тип преобразования. Обязательный аргумент</li> <li>• идентификатор - идентификатор языка (системный идентификатор языка используется по умолчанию). Необязательный аргумент.</li> </ul> <p>Аргумент тип_преобразования может принимать следующие значения:</p> <ul style="list-style-type: none"> <li>• vbUpperCase – преобразование знаков строки в верхний регистр.</li> <li>• vbLowerCase – преобразование знаков строки в нижний регистр.</li> <li>• vbProperCase – преобразование первой буквы каждого слова строки в прописную</li> <li>• vbUnicode Преобразование строки в Юникод в соответствии с системной кодовой страницей по умолчанию.</li> <li>• vbFromUnicode Преобразование строки из Юникода в знаки системной кодовой страницы по умолчанию.</li> </ul>	<code>=StrConv(Fields!Description.Value, vbProperCase)</code>
<b>StrDup</b>	Возвращает строку из символа, повторенного заданное количество раз.	<code>=StrDup(3,"M")</code>



## 7.6.5.2 Функции даты и времени

Функция	Описание	Пример использования
<b>CDate</b>	Конвертирует в тип данных Date	=CDate(Fields!BirthDate.Value)
<b>DateAdd</b>	Возвращает объект Date, который содержит дату и время, к которым был добавлен интервал.	=DateAdd("d",3,Fields!BirthDate.Value) =DateAdd(DateInterval.Day,3,Fields!BirthDate.Value)
<b>DateDiff</b>	Возвращает разницу между двумя интервалами в формате Long.	=DateDiff("yyyy",Fields!BirthDate.Value,"1/1/2007") =DateDiff(DateInterval.Year,Fields!BirthDate.Value,"1/1/2007")
<b>DatePart</b>	Возвращает заданный компонент данного значения Date в формате Integer.	=DatePart("q",Fields!BirthDate.Value,0,0) =DatePart(DateInterval.Quarter,Fields!BirthDate.Value,FirstDayOfWeek.System,FirstWeekOfYear.System)
<b>DateSerial</b>	Возвращает объект Date, представляющий заданный год, месяц, день и время относительно (00:00:00).	=DateSerial(DatePart("yyyy",Fields!BirthDate.Value)-10,DatePart("m",Fields!BirthDate.Value)+3,DatePart("d",Fields!BirthDate.Value)-1)
<b>DateString</b>	Возвращает или устанавливает текущую дату в формате строки.	=DateString() =DatePart("m",DateString())
<b>DateValue</b>	Возвращает объект Date, содержащий информацию представленную строкой, относительно (00:00:00).	=DateValue("January 15, 2007")
<b>Day</b>	Возвращает число от 1 до 31, представляющее собой день месяца.	=Day(Fields!BirthDate.Value)
<b>FormatDateTime</b>	Возвращает строковое выражение, представляющее дату/время.	=FormatDateTime(Fields!BirthDate.Value,DateFormat.ShortDate)
<b>Hour</b>	Возвращает число от 0 до 23. Представляет собой час дня.	=Hour(Fields!BirthDate.Value)
<b>Minute</b>	Возвращает число от 0 до 59. Представляет минуту часа.	=Minute(Fields!BirthDate.Value)

Функция	Описание	Пример использования
<b>Month</b>	Возвращает число от 1 до 12. Представляет месяц в году.	<code>=Month(Fields!BirthDate.Value)</code>
<b>MonthName</b>	Возвращает имя заданного месяца в виде строки.	<code>=MonthName(10,True)</code> <code>=MonthName(Month(Fields!BirthDate.Value),False)</code>
<b>Now</b>	Возвращает объект Date, содержащий текущую дату и время в системе.	<code>=Now()</code> <code>"This time tomorrow is " &amp; DateAdd("d",1,Now())</code>
<b>Second</b>	Возвращает число от 0 до 59. Представляет секунду в минуте.	<code>=Second(Fields!BirthDate.Value)</code>
<b>TimeOfDay</b>	Возвращает или устанавливает текущее время дня в системе. Задается объектом в формате Date.	<code>=TimeOfDay()</code> <code>"Time of the day is " &amp; TimeOfDay()</code>
<b>Timer</b>	Возвращает число в формате Double, представляющее количество секунд прошедших с полночи.	<code>=Timer()</code> <code>"Number of seconds since midnight " &amp; Timer()</code>
<b>TimeSerial</b>	Возвращает Date, представляющий заданный час, минуту и секунду относительно 1 Января 1-го года.	<code>=TimeSerial(DatePart("h", Fields!BirthDate.Value),DatePart("n", Fields!BirthDate.Value),DatePart("s", Fields!BirthDate.Value))</code> <code>=TimeSerial(23,49,52)</code>
<b>TimeString</b>	Возвращает или устанавливает день и время в системе в формате String.	<code>=TimeString()</code>
<b>TimeValue</b>	Возвращает объект Date, содержащий информацию о времени в виде строки, относительно 1 Января 1-го года.	<code>=TimeValue("16:20:17")</code> <code>=TimeValue(Fields!BirthDate.Value)</code>
<b>Today</b>	Возвращает или устанавливает текущую дату в системе. Текущая дата в формате Date.	<code>=Today()</code> <code>"Tomorrow is " &amp; DateAdd("d",1,Today())</code> <code>"Tomorrow is " &amp; DateAdd(DateInterval.Day,1,Today())</code>
<b>Weekday</b>	Возвращает число представляющее день недели.	<code>=Weekday(Fields!BirthDate.Value,0)</code> <code>=Weekday(Fields!BirthDate.Value, FirstDayOfWeek.System)</code>

Функция	Описание	Пример использования
<b>WeekdayName</b>	Возвращает имя дня недели в формате String.	<code>=WeekdayName(2,True,0)</code> <code>=WeekDayName(DatePart("w", Fields!BirthDate.Value),True,0)</code> <code>=WeekDayName(DatePart(DateInterval.Weekday, Fields!BirthDate.Value),True, FirstDayOfWeek.System)</code>
<b>Year</b>	Возвращает число от 1 до 9999. Представляет год.	<code>=Year(Fields!BirthDate.Value)</code>

### 7.6.5.3 Математические функции

Функция	Описание	Пример использования
<b>Abs</b>	Возвращает абсолютное значение числа с плавающей точкой.	<code>=Abs(-5.5)</code> <code>=Abs(Fields!YearlyIncome.Value - 80000)</code>
<b>Acos</b>	Возвращает угол по значению косинуса.	<code>=Acos(.5)</code> <code>=Acos(Fields!Angle.Value)</code>
<b>Asin</b>	Возвращает угол синуса.	<code>=Asin(.5)</code> <code>=Asin(Fields!Angle.Value)</code>
<b>Atan</b>	Возвращает угол тангенса.	<code>=Atan(45)</code> <code>=Atan(Fields!Tangent.Value)</code>
<b>Atan2</b>	Возвращает угол тангенса заданного разностью двух значений.	<code>=Atan2(3,5)</code> <code>=Atan2(Fields!CoordinateY.Value, Fields!CoordinateX.Value)</code>
<b>BigMul</b>	Возвращает полный результат умножения двух 32 разрядных чисел.	<code>=BigMul(2147483647,-2147483647)</code> <code>=BigMul(Fields!Int32Value.Value, Fields!Int32Value.Value)</code>
<b>Ceiling</b>	Возвращает ближайшее целое число, большее или равное заданному числу с плавающей точкой.	<code>=Ceiling(34.3352)</code> <code>=Ceiling(Fields!YearlyIncome.Value / 7)</code>
<b>Cos</b>	Возвращает косинус заданного угла.	<code>=Cos(67)</code> <code>=Cos(Fields!Angle.Value)</code>
<b>Cosh</b>	Возвращает гиперболический косинус заданного угла.	<code>=Cosh(67)</code> <code>=Cosh(Fields!Angle.Value)</code>
<b>Exp</b>	Возвращает число, возведенное в степень.	<code>=Exp(5)</code> <code>=Exp(Fields!IntegerCounter.Value)</code>
<b>Fix</b>	Возвращает целую часть числа.	<code>=Fix(-9.25)</code>

<b>Функция</b>			<code>=Fix(Fields!YearlyIncome.Value / -3)</code>
<b>Floor</b>	Возвращает ближайшее целое число, меньшее или равное заданному числу с плавающей точкой.	Пример использования	<code>=Floor(4.67)</code> <code>=Floor(Fields!YearlyIncome.Value / 12)</code>
<b>Int</b>	Округляет до наименьшего целого.		<code>=Int(-93.4)</code> <code>=Int(Fields!YearlyIncome.Value / 12)</code>
<b>Log</b>	Возвращает натуральный логарифм числа.		<code>=Log(33.5)</code> <code>=Log(Fields!NumberValue.Value)</code>
<b>Log10</b>	Возвращает десятичный логарифм числа.		<code>=Log10(33.5)</code> <code>=Log10(Fields!NumberValue.Value)</code>
<b>Max</b>	Возвращает большее из двух 8 битных чисел.		<code>=Max(Fields!YearlyIncome.Value)</code> <code>=Max(Fields!YearlyIncome.Value, "AdventureWorks")</code> <code>=Max(Fields!YearlyIncome.Value, "AdventureWorks", Recursive)</code>
<b>Min</b>	Возвращает меньшее из двух 8 битных чисел.		<code>=Min(Fields!YearlyIncome.Value)</code> <code>=Min(Fields!YearlyIncome.Value, "AdventureWorks")</code> <code>=Min(Fields!YearlyIncome.Value, "AdventureWorks", Recursive)</code>
<b>Pow</b>	Возвращает результат возведения числа в степень.		<code>=Pow(Fields!YearlyIncome.Value,2)</code>
<b>Rnd</b>	Возвращает случайное число типа Single.		<code>=Rnd()</code> <code>=Rnd(0)</code> <code>=Rnd(-1)</code>
<b>Round</b>	Округляет число с плавающей точкой к ближайшему целому.		<code>=Round(12.456)</code> <code>=Round(12.453,2)</code> <code>=Round(Fields!YearlyIncome.Value /12,2)</code> <code>=Round(2.5, System.MidpointRounding.AwayFromZero)</code> <code>=Round(2.5, System.MidpointRounding.ToEven)</code>
<b>Sign</b>	Возвращает знак числа.		<code>=Sign(Fields!YearlyIncome.Value - 60000)</code>
<b>Sin</b>	Возвращает синус заданного угла.		<code>=Sin(90)</code> <code>=Sin(Fields!Angle.Value)</code>
<b>Sinh</b>	Возвращает гиперболический синус заданного угла.		<code>=Sinh(90)</code> <code>=Sinh(Fields!Angle.Value)</code>
<b>Sqrt</b>	Возвращает квадрат заданного числа.		<code>=Sqrt(144)</code> <code>=Sqrt(Fields!Area.Value)</code>
<b>Tan</b>	Возвращает тангенс		<code>=Tan(135)</code>

	заданного угла.	=Tan(Fields!Angle.Value)
Функция	Описание	Пример использования
<b>Tanh</b>	Возвращает гиперболический тангенс заданного угла.	=Tanh(135) =Tanh(Fields!Angle.Value)

## 7.6.5.4 Функции проверки типов

Функция	Описание	Пример использования
<b>IsArray</b>	Возвращает true, если переменная это массив.	=IsArray(Parameters!Initials.Value)
<b>IsDate</b>	Возвращает true, если выражение представляет собой правильное значение в формате Date.	=IsDate(Fields!BirthDate.Value) =IsDate("31/12/2000")
<b>IsNothing</b>	Возвращает true, если выражение не ссылается на объект.	=IsNothing(Fields!MiddleInitial.Value)
<b>IsNumeric</b>	Возвращает true, если выражение может быть представлено как число.	=IsNumeric(Fields!YearlyIncome.Value)

## 7.6.5.5 Функции выполнения условия

Функция	Описание	Пример использования
<b>Choose</b>	Выбирает и возвращает значение из списка аргументов.	=Choose(2,"13","15","21") =Choose(Datepart("w",Fields!BirthDate.Value), "First","Second","Third","Fourth","Fifth","Sixth", "Seventh")
<b>IIf</b>	Возвращает один из двух объектов, в зависимости от значения условия.	=IIf(Fields!YearlyIncome.Value >= 60000, "High","Low")
<b>Switch</b>	Возвращает Object из первого истинного условного выражения.	=Switch(Fields!State.Value = "OR", "Oregon", Fields!State.Value = "WA", "Washington") =Switch(Fields!FirstName.Value = "Sue", "Susan", Fields!FirstName.Value = "Bob", "Robert")

## 7.6.5.6 Функции преобразования типов

Функция	Описание	Пример использования
<b>CBool</b>	Конвертирует в формат Boolean	=CBool(Fields!HouseOwnerFlag.Value)
<b>CByte</b>	Конвертирует в формат Byte	=CByte(Fields!NumberCarsOwned.Value) =CByte(Fields!Number.Value)
<b>CChar</b>	Конвертирует в формат Char	=CChar(Fields!MaritalStatus.Value) =CChar("ABC"))



Функция	Описание	Пример использования
<b>CDate</b>	Конвертирует в формат Date	=CDate(Fields!BirthDate.Value) =CDate("October 2, 2001")
<b>CDBl</b>	Конвертирует в формат Double	=CDBl(Fields!YearlyIncome.Value) =CDBl(573.45 * .4287 * 82561)
<b>CDec</b>	Конвертирует в формат Decimal	=CDec(Fields!YearlyIncome.Value) =CDec(573.4503)
<b>CInt</b>	Конвертирует в формат Integer	=CInt(Fields!YearlyIncome.Value) =CInt(734.62)
<b>CLng</b>	Конвертирует в формат Long	=CLng(Fields!YearlyIncome.Value) =CInt(73462.23)
<b>CObj</b>	Конвертирует в формат Object	=CObj(Fields!YearlyIncome.Value) =CObj(73462.23)
<b>CShort</b>	Конвертирует в формат Short	=CShort(Fields!NumberCarsOwned.Value) =CShort(200)
<b>CSng</b>	Конвертирует в формат Single	=CSng(Fields!YearlyIncome.Value) =CSng(12.746522945)
<b>CStr</b>	Конвертирует в формат String	=CStr(Fields!YearlyIncome.Value) =CStr(12.746)
<b>Hex</b>	Возвращает строку, содержащую шестнадцатеричное представление числа	=Hex(Fields!CellColor.Value) =Hex(46)
<b>Oct</b>	Возвращает строку, содержащую восьмеричное представление числа	=Oct(Fields!BitString.Value) =Oct(8)
<b>Str</b>	Возвращает строковое представление числа	=Str(Fields!YearlyIncome.Value) =Str(352.75)
<b>Val</b>	Функция Val прекращает считывание строки, встретив первый знак, не распознаваемый как часть числа. При этом не распознаются знаки, которые могут быть частью числового значения, например знак доллара и запятая. Тем не менее, данная функция распознает префиксы оснований системы счисления &O (в восьмеричной системе) и &H (в шестнадцатеричной). Пробелы, знаки табуляции и перевода строки из аргумента исключаются	=Val(Fields!AddressLine1.Value) =Val("6706 Ridgeview Dr")

## 7.6.5.7 Функции агрегирования

## Параметры

- **Expression** – выражение, результат которого вычисляется по указанной статистической функции над заданными числовыми данными. Выражение не может содержать статистические функции
- **Scope** – Имя (**String**) набора данных, группирования или области данных, содержащих элементы отчета, к которым применяется статистическая функция. Если указан аргумент **Scope**, статистическая функция применяется ко всем данным в наборе данных, группировании или области данных. Данные в наборе, указанном в выражении, должны иметь один и тот же тип. Воспользуйтесь для преобразования различных числовых типов к единому типу данных функциями преобразования **CInt**, **CDbl**, **CDec** и другими.

## Область действия

Каждая функция агрегирования использует параметр **Scope**, определяющий область действия, в которой выполняется функция. Допустимой областью действия может служить имя группирования, набора данных или области данных. В качестве области действия могут использоваться только группирования или области данных, которые прямо или косвенно содержат выражение. Если не указать параметр **Scope**, то областью действия функции агрегирования будет самая внутренняя область данных или группирование, к которому принадлежит элемент отчета. Задание области **Nothing** устанавливает областью действия внешнюю область данных, к которой принадлежит элемент отчета.

Для выражений за пределами областей данных параметр **Scope** ссылается на набор данных. Если отчет содержит больше одного набора данных, то параметр **Scope** обязателен. Если отчет содержит лишь один набор данных и параметр **Scope** пропущен, то областью действия устанавливается этот набор данных. Нельзя указывать ключевое слово **Nothing** для элементов отчета за пределами области данных.

**ВНИМАНИЕ!**

Нельзя использовать параметр **Scope** в колонтитулах страницы.

Функция	Описание	Пример использования
<b>Avg</b>	Возвращает среднюю величину всех значений указанного выражения, не являющихся значениями NULL. Тип результата: <b>Decimal</b> для десятичных выражений и <b>Double</b> – для всех остальных выражений	<code>=Avg(Fields!Cost.Value, Nothing)</code>
<b>Count</b>	Возвращает число значений указанного выражения, отличных от NULL. Тип <b>Expression</b> : <b>Variant</b> или <b>Binary</b> Тип результата: <b>Integer</b>	<code>=Count(Fields!EmployeeID.Value, Nothing)</code> <code>=Count(Fields!OrderID.Value, "Orders")</code>

Функция	Описание	Пример использования
<b>CountDistinct</b>	Возвращает число различных значений указанного выражения, отличных от NULL. Тип <b>Expression: Variant</b> Тип результата: <b>Integer</b>	<code>=CountDistinct(Fields!ManagerID.Value, "Department")</code>
<b>CountRows</b>	Возвращает количество строк в указанной области. Тип результата: <b>Integer</b>	<code>=CountRows("Department")</code>
<b>First</b>	Возвращает первое значение указанного выражения. Тип <b>Expression: Variant</b> или <b>Binary</b> Тип результата: определяется типом выражения	<code>=First(Fields!ProductNumber.Value, "Category")</code>
<b>Last</b>	Возвращает последнее значение указанного выражения. Тип <b>Expression: Variant</b> или <b>Binary</b> Тип результата: определяется типом выражения	<code>=Last(Fields!ProductNumber.Value, "Category")</code>
<b>Max</b>	Возвращает максимальное значение для всех отличных от NULL значений указанного выражения. Тип результата: определяется типом выражения	<code>=Max(Fields!OrderTotal.Value, "Year")</code>
<b>Min</b>	Возвращает минимальное значение для всех отличных от NULL значений указанного выражения. Тип результата: определяется типом выражения	<code>=Min(Fields!OrderTotal.Value, "Year")</code>
<b>RowNumber</b>	Возвращает текущее количество строк в указанной области. Тип результата: <b>Integer</b>	<code>=RowNumber(Nothing)</code>
<b>RunningValue</b>	Использует задаваемую функцию <i>Function</i> , чтобы вернуть текущее статистическое значение для указанного выражения. <i>Function (Enum)</i> – имя (в примере <b>Sum</b> ) статистической функции (не может быть <b>RunningValue</b> , <b>RowNumber</b> или <b>Aggregate</b> ), которая будет применяться к выражению Тип результата: определяется статистической функцией, указанной в параметре <i>Function</i>	<code>=RunningValue(Fields!Cost.Value, Sum, Nothing)</code>

Функция	Описание	Пример использования
<b>StDev</b>	Возвращает среднеквадратичное отклонение всех отличных от NULL значений заданного выражения. Тип результата: <b>Decimal</b> для десятичных выражений и <b>Double</b> – для всех остальных выражений	<code>=StDev(Fields!LineTotal.Value, "Order")</code>
<b>StDevP</b>	Возвращает среднеквадратичное отклонение наполнения для совокупности всех отличных от NULL значений указанного выражения. Тип результата: <b>Decimal</b> для десятичных выражений и <b>Double</b> – для всех остальных выражений	<code>=StDevP(Fields!LineTotal.Value, "Order")</code>
<b>Sum</b>	Возвращает сумму значений указанного выражения. Тип результата: <b>Decimal</b> для десятичных выражений и <b>Double</b> – для всех остальных выражений	<code>=Sum(Fields!LineTotal.Value, "Order")</code>
<b>Var</b>	Возвращает дисперсию всех значений указанного выражения, отличных от NULL. Тип результата: <b>Decimal</b> для десятичных выражений и <b>Double</b> – для всех остальных выражений	<code>=Var(Fields!LineTotal.Value, "Order")</code>
<b>VarP</b>	Возвращает для указанного выражения дисперсию совокупности всех значений, отличных от NULL. Тип результата: <b>Decimal</b> для десятичных выражений и <b>Double</b> – для всех остальных выражений	<code>=VarP(Fields!LineTotal.Value, "Order")</code>
<b>InScope</b>	Возвращает true, если данный объект находится внутри заданных границ	<code>=Iif(InScope("Month"), "ProductDetail", "ProductSummary") =Not(InScope("Product")) =InScope("Product")</code>
<b>Level</b>	Возвращает целое число (отсчитываемое от нуля), представляющее текущий уровень иерархии	<code>=Level("Employees")</code>

Функция	Описание	Пример использования
<b>Previous</b>	<p>Возвращает значение выражения для предыдущего экземпляра элемента в указанной области (предыдущей ячейки данных). Тип результата: <b>Variant</b> или <b>Binary</b>.</p> <p>В группе подробных сведений с помощью Previous можно указать значение ссылки на поле в предыдущем экземпляре строки подробных сведений. Ссылки на поля поддерживаются только в группе подробных сведений. Например, для текстового поля в группе сведений выражение =Previous(Fields!Quantity.Value) возвращает данные для поля Quantity из предыдущей строки. В первой строке это выражение возвращает значение NULL (Nothing в Visual Basic).</p> <p>В группах, не являющихся группами подробных сведений, следует использовать Previous со статистической функцией, например First или Last, содержащей конкретный экземпляр группы. Например, в текстовом поле в группе строк на основе =Fields!Year.Value=</p> <p>Previous(First(Fields!Year.Value)) возвращает год для первого экземпляра группы «Year».</p>	<p>=Previous(Fields!Quantity.Value)</p> <p>=Fields!Year.Value=Previous(First(Fields!Year.Value))</p>

## 7.6.5.8 Пользовательские функции

Пользовательские функции можно использовать в выражениях для любого элемента отчета.

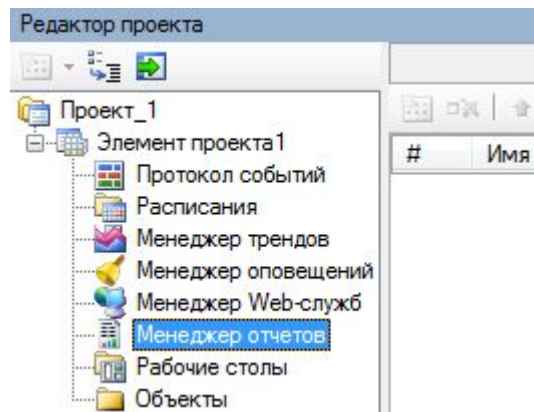
Функция	Описание	Пример использования
<b>RNaN(expression, value)</b>	Возвращает значение одного из двух параметров: "expression", если оно отлично от NaN, иначе - "value".	=RNaN(YearlyIncome.Value, 0.0)



## 7.7 Менеджер отчетов

**Менеджер отчетов** предназначен для управления хранением, печатью, рассылкой и публикацией отчетов.

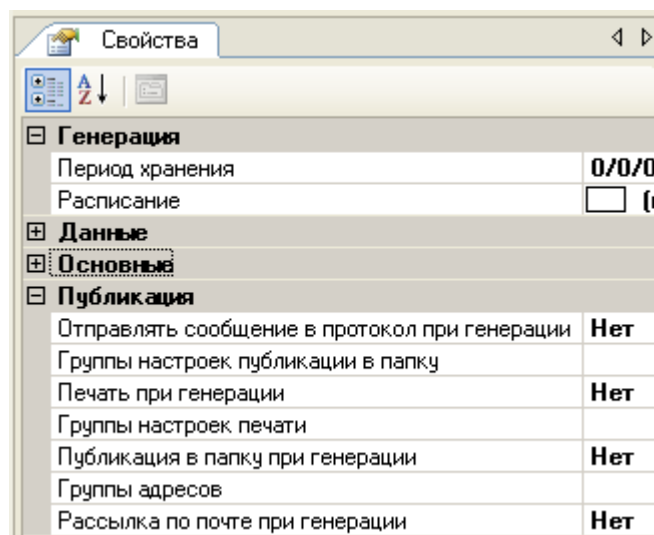
В Среде разработки **DataRate** **Менеджер отчетов** входит в состав элемента проекта.



### Используемые свойства шаблона отчета

В своей работе **Менеджер отчетов** использует свойства из следующих разделов шаблона отчета:

- **Генерация** : Период хранения
- **Публикация**: Отправлять сообщение в протокол при генерации, Группы настроек публикации в папку и другие свойства этого раздела



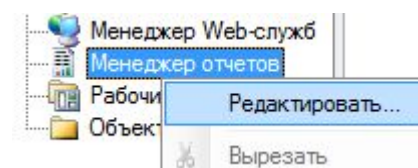
### Вызов менеджера отчетов

Вызвать **Менеджера отчетов** можно:

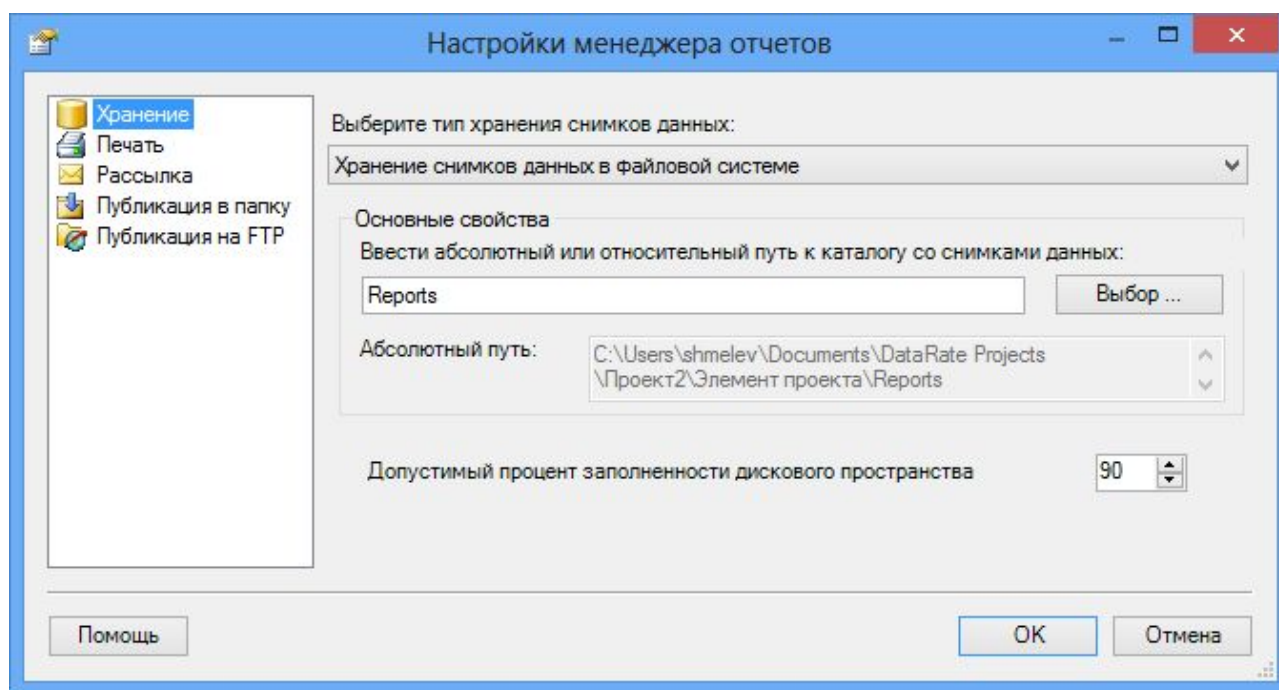
- Из контекстного меню элемента проекта командой **Редактировать...**

или

- Двойным щелчком мыши по иконке **Менеджера отчетов** в дереве элемента проекта.



После выполнения одного из этих действий появится окно **Настройки менеджера отчетов**.



Окно **Настройки менеджера отчетов** содержит следующие группы настроек:

- Хранение снимков данных
- Печать отчетов
- Рассылка отчетов
- Публикация в папку
- Публикация на FTP

### 7.7.1 Хранение снимков данных

Сгенерированный отчет храниться в виде снимков данных, которые можно просматривать.

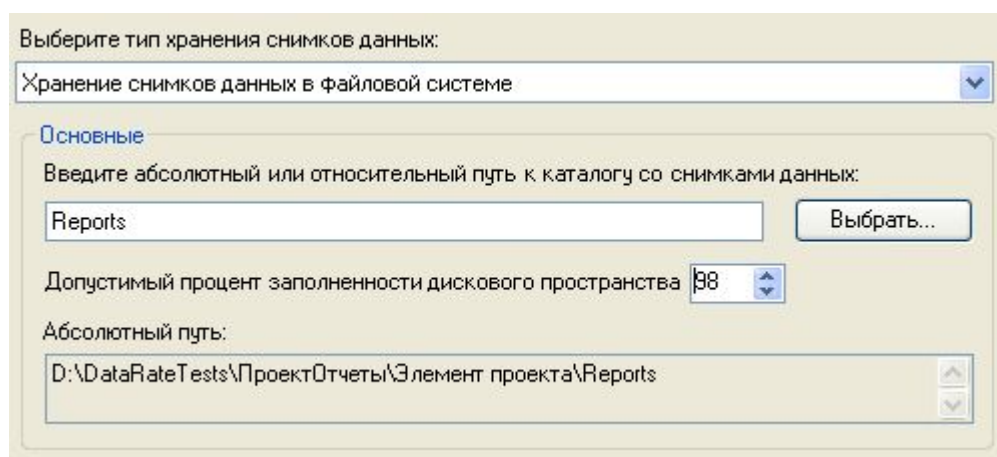
#### **ВНИМАНИЕ!!!**

При хранении снимков данных отчета рекомендуется:

- Обеспечивать достаточное дисковое пространство для хранения
- Избегать хранения снимков данных на том же дисковом носителе, который содержит файл подкачки Windows.

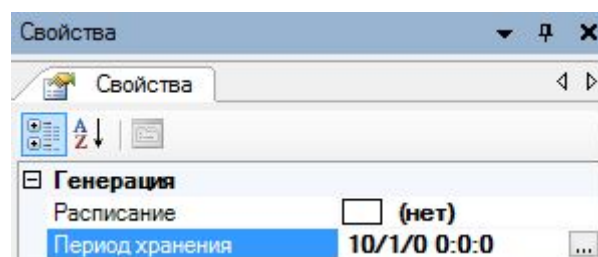
Группа настроек **Хранение** Менеджера отчетов предназначена для определения параметров сохранения снимков данных отчета и включает следующие параметры:

- **Допустимый процент заполненности дискового пространства.** Позволяет ограничить генерацию отчетов на диск. В случае превышения допустимого лимита отчеты не будут сгенерированы, о чем будет сформировано специальное сообщение в логах и в протоколе событий
- **Абсолютный или относительный путь к каталогу хранения снимков данных.** Относительный путь определяется относительно файла элемента проекта. Дерево файловой системы для хранения снимков данных формируется следующим образом:  
`<Каталог, заданный при настройке>\<Полное имя технологического объекта>\<Имя отчета>\<Метка времени формирования отчета> – ДД_ММ_ГГГГ_ЧЧ_ММ_СС_МС>.`



### Период хранения снимков данных

В поле свойства **Период хранения** шаблона отчета можно задать дату, после достижения которой снимок данных отчета будет удален.



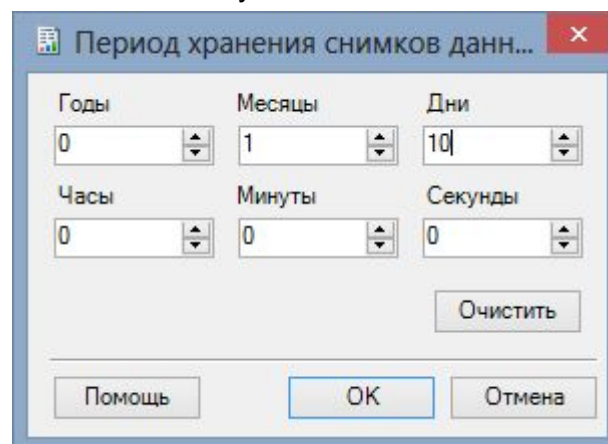
Для редактирования периода хранения снимков данных следует:

1. Выбрать свойство **Период хранения** и нажать на кнопку

2. В появившемся окне **Период хранения снимков данных** задать дату

При формировании периода хранения считается, что год **365** дней, а месяц 30 дней. Из-за этого при формировании периода хранения отчета может возникать погрешность.

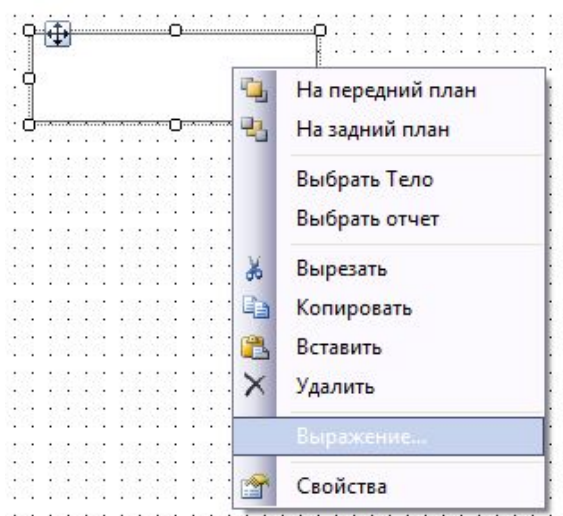
Например, если необходим период **18 месяцев**, пользователь может ввести **18** в поле **Месяцы**. После закрытия окна указанный период будет преобразован в формат **ДД/ММ/ГГ ЧЧ:мм:сс**. В окне свойств пользователь увидит в свойстве **Период хранения** значение **25/5/1 00:00:00**, т.е. **1 год, 5 месяцев и 25 дней**.



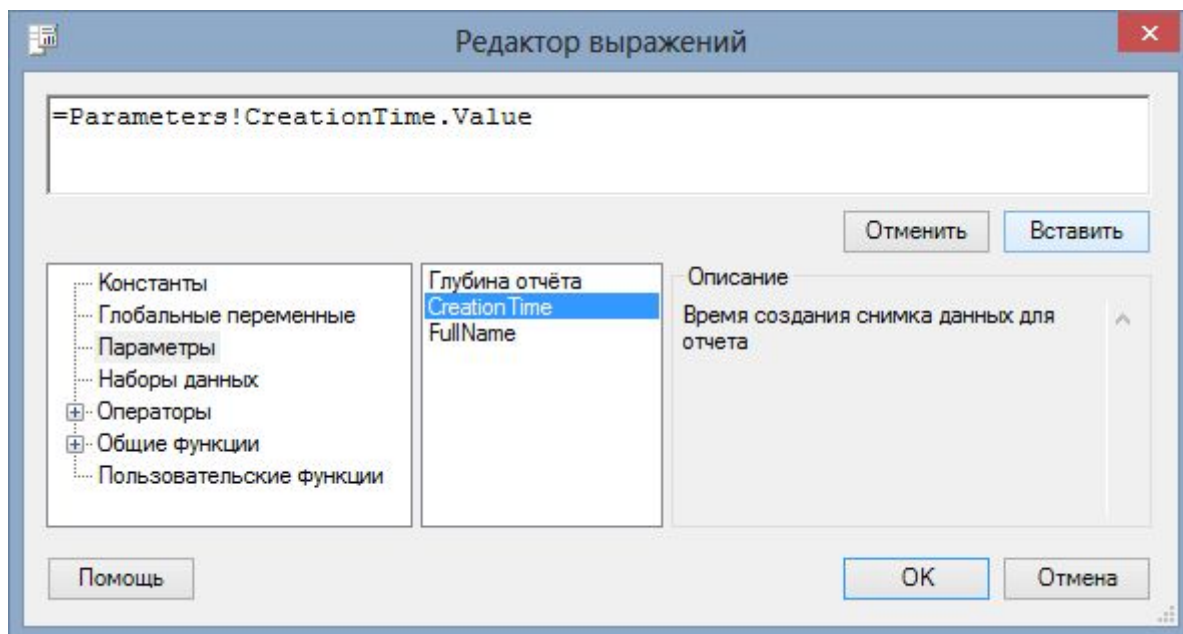
Кнопка **Очистить** позволяет обнулить все поля ввода. Если не задавать период хранения снимков данных - **0/0/0 00:00:00**, снимки данных не будут удаляться.

### Как задать время генерации снимка данных через параметр

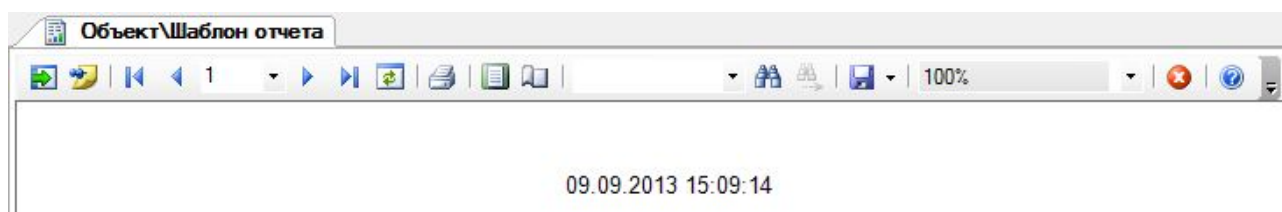
1. Откройте Шаблон отчёта
2. Нажмите на панели инструментов кнопку
3. Перетащите из открывшегося окна **Элементы отчета** элемент **Текстовое поле** в тело шаблона отчёта
4. Щелкните правой кнопкой мыши по добавленному элементу и в открывшемся контекстном меню выберите пункт **Выражение...**



- 5 В открывшемся окне **Редактор выражений** выберите в дереве узел **Параметры - > CreationTime**
- 6 Нажмите на кнопку **Вставить** или скопируйте в буфер выражение `=Parameters!CreationTime.Value`, а затем вставьте его в соответствующее поле ввода.



- 7 Нажмите на кнопку **ОК**
- 8 Запустите проект на имитацию и перейдите на вкладку **Предварительный просмотр**



### 7.7.2 Печать отчетов

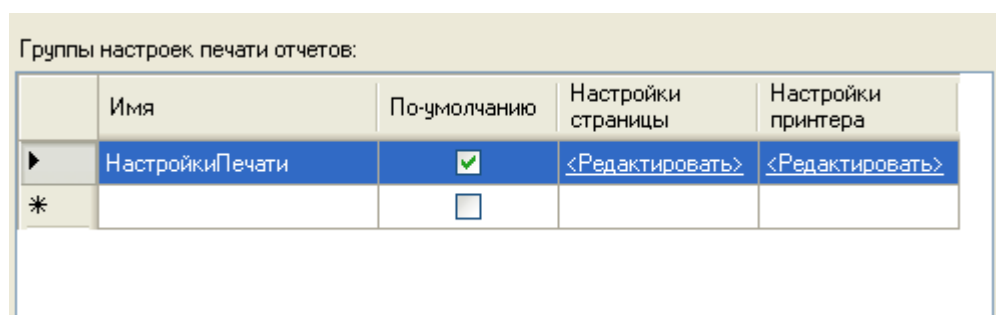
Менеджер отчетов позволяет при генерации отчета распечатать отчет по расписанию или из скриптов.

Для печати отчета необходимо создать группу настроек печати отчета.

#### Печать отчета при генерации

Для того чтобы печатать отчет при генерации следует:

- 1 Вызвать Менеджер отчетов
- 2 В появившемся окне **Настройки менеджера отчетов** выбрать настройки **Печать**
- 3 Создать группу настроек печати



В столбце **Имя** пользователь следует указать новое имя для группы настроек печати отчета.

Для изменения настроек страницы следует щелкнуть мышью на ячейку **Редактировать** в столбце **Настройки страницы**.

Для изменения настроек принтера следует щелкнуть мышью на ячейку **Редактировать** в столбце **Настройки принтер**.

Для настроек страницы и настроек принтера используются стандартные диалоговые окна.

- 4 Выбрать в свойстве **Группы настроек печати** шаблона отчета имя созданной группы настроек и установить свойство **Печать при генерации** в положение **Да**.

#### Печать отчета из скрипта

Скрипт для вывода отчета на печать настраивается с помощью редактора скриптов **DataRate** в скрипте отчета.

Печать отчета происходит путем вызова функции

```
Print(int historyId),
```

где **historyId** – идентификатор истории отчета:

**0** – последний сгенерированный отчет, **1** – предпоследний и так далее.

В редакторе скриптов эту функцию можно вызвать при выполнении некоторого условия, например, так:

```
If(Value > 0.5) Client.Print(0);
```

где **Value** – аргумент скрипта, **Client** – стандартная переменная (объект, для которого запускается скрипт).



## 7.7.3 Рассылка отчетов

Настройки **Рассылка** предназначены для отправления отчетов по указанным адресам электронной почты.

Сгенерированный отчет будет приложен к письму в соответствии с настройками группы рассылок. Группа рассылок включает в себя адреса получателей и типы отчетов.

Менеджер отчетов поддерживает следующие типы отчетов для рассылки: **PDF, Excel, PNG, JPEG, TIF, GIF, EMF и XML**. Если для рассылки в качестве формата файла отчета используется один из графических форматов **PNG, JPEG, TIF, GIF** или **EMF**, то количество приложенных к письму файлов будет соответствовать количеству страниц отчета.

### Как отправить отчет при генерации

Для того чтобы отправлять по электронной почте сгенерированный отчет следует:

- 1 Вызвать Менеджер отчетов
- 2 В появившемся окне **Настройки менеджера отчетов** выбрать настройки **Рассылка**
- 3 Создать группу настроек рассылки

Настройка отправителя

От: DataRate@reports.com Ответ:

Тема: {дата} - {время} {имя\_полн} ...

Имя письма: {дата} - {время} {имя\_полн} ...

Тело письма: ...

Пример темы и имени: 26.12.2011 - 16.34.25  
ЭлементПроекта.Папка.Объект.Отчет  
26.12.2011 - 16.34.25  
ЭлементПроекта.Папка.Объект.Отчет

Группы адресов:

	Имя	По-умолчанию	Адреса
*		<input type="checkbox"/>	

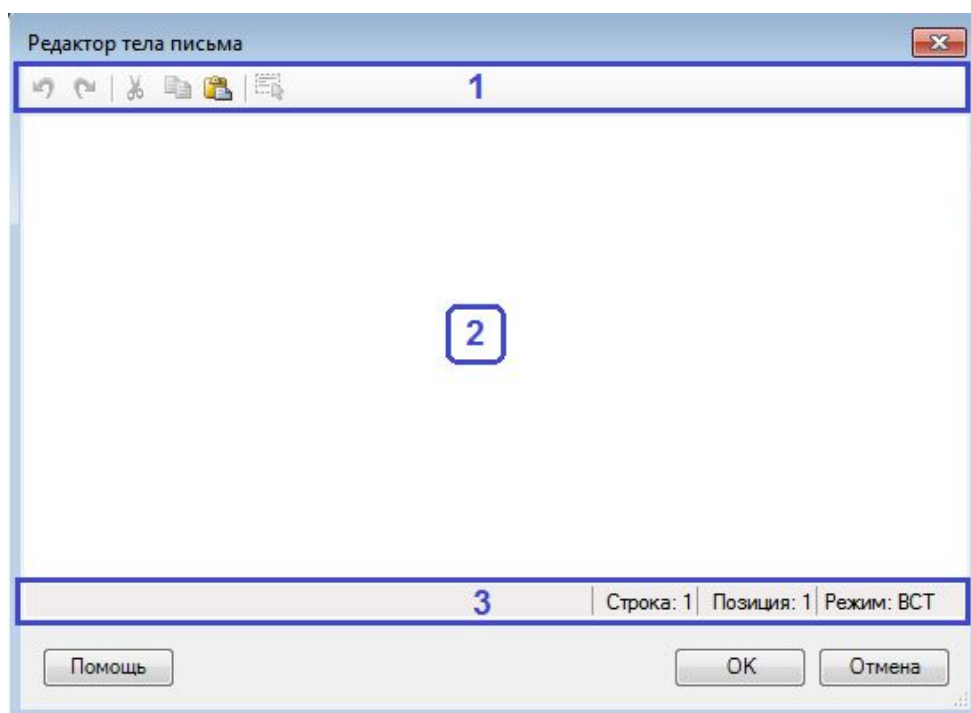
Пользователь должен указать:

- **Тему**. Для редактирования шаблона темы письма нажмите кнопку
- **Имя отчета**. Для редактирования шаблона имени письма нажмите кнопку
- **Тело письма**. Содержит текст письма.

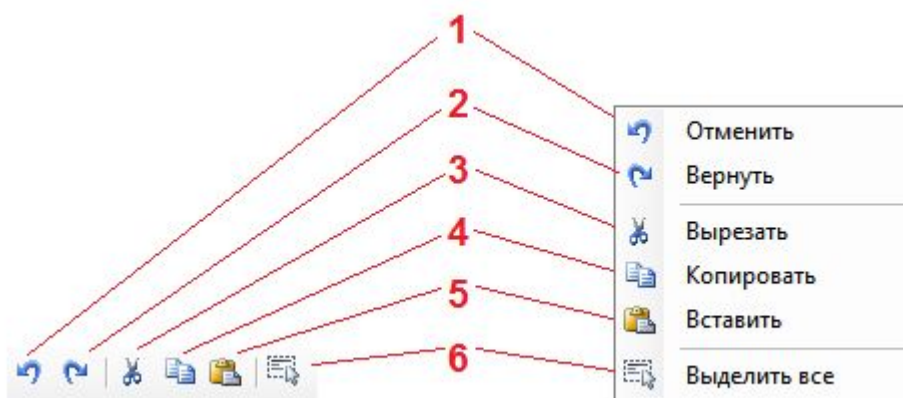
Для вызова редактора текста нажмите на кнопку .

Окно **Редактор тела письма** позволяет составить текст письма в редакторе с расширенными возможностями. В верхней части окна расположена Панель Инструментов (1). Рабочая зона представлена текстовым полем (2). В нижней части окна расположена строка состояния (3).

Панель инструментов предназначена для облегчения труда пользователей при вводе текста. Функции панели дублируются через контекстное меню. Чтобы вызвать меню необходимо правой клавишей мыши нажать по текстовому полю (2) рабочей зоны.



Панель инструментов и соответствующее ей контекстное меню:



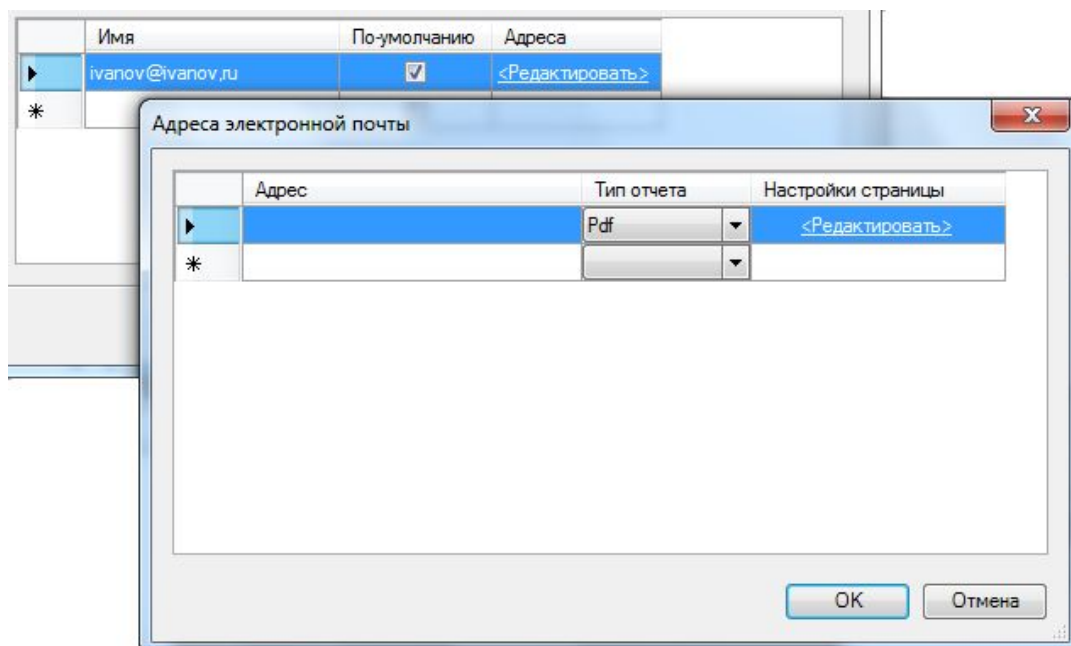
- 1 – Отменить - отменить последнее произведенное действие
- 2 – Вернуть - вернуть последнее отмененное действие
- 3 – Вырезать - удалить выделенный текст с копированием в буфер обмена
- 4 – Копировать - копировать выделенный текст в буфер обмена
- 5 – Вставить - вставить текст из буфера обмена
- 6 – Выделить все - выделить весь текст в поле

Строка состояния имеет 3 информационные секции, которые позволяют ориентироваться в редакторе:

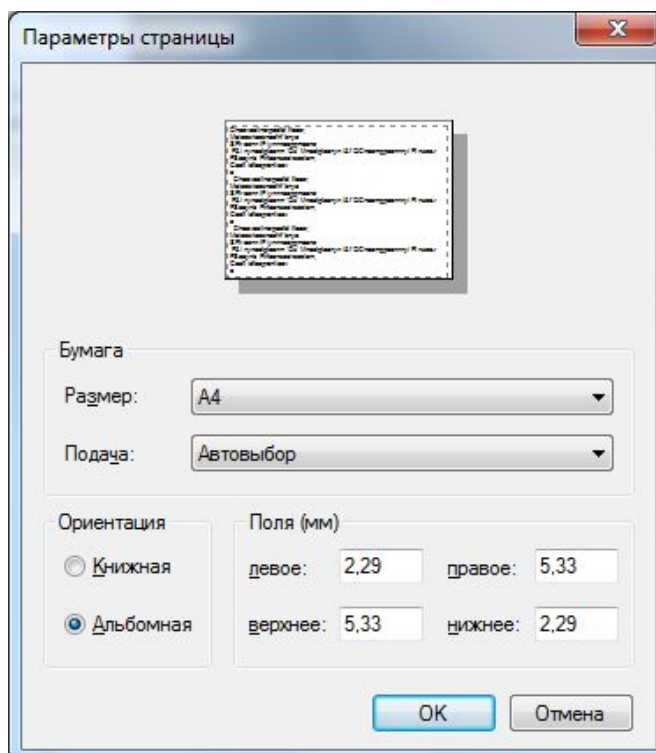
- Номер текущей строки
- Номер позиции в текущей строке
- Текущий режим работы (ВСТ - вставка, ЗАМ - замена).

Кнопка **Помощь** дает возможность пользователю раскрыть текущий раздел справки и ознакомиться с возможностями редактора. То же самое можно воспроизвести, нажав клавишу **F1** на клавиатуре. При нажатии на кнопку **ОК** набранный текст будет перенесен в окно **Тело письма** настройки **Рассылка**. Нажатие кнопки **Отмена** оставит текст в окне **Тело письма** без изменений.

- **Почтовый адрес.** Для задания адреса настройте параметры панели **Группа адресов**



Для изменения настроек страницы щелкните мышью по ячейке **Редактировать** столбца **Настройки страницы**



4. Выбрать в свойстве **Группы адресов** шаблона отчета имя созданной группы настроек и установить свойство **Рассылка по почте при генерации** в положение **Да**.

Настройка исходящего почтового сервера (SMTP) осуществляется в свойствах элемента проекта.

### Рассылка из скрипта отчета

Скрипт для рассылки отчета настраивается с помощью редактора скриптов **DataRate** в скрипте отчета.

Рассылка отчета происходит путем вызова функции

`Email(int historyId),`

где **historyId** – идентификатор истории отчета:

**0** – последний сгенерированный отчет, **1** – предпоследний и так далее.

В редакторе скриптов эту функцию можно вызвать при выполнении некоторого условия, например, так:

`If(Value > 0.5) Client.Email(0);`

где **Value** – аргумент скрипта, **Client** – стандартная переменная (объект, для которого запускается скрипт).

### 7.7.4 Публикация в папку

Менеджер отчетов при генерации отчета позволяет сохранять отчет по расписанию или из скриптов.

Для сохранения отчета необходимо задать группу настройки публикации в папку.

Настройки **Публикация в папку** предназначены для сохранения отчетов в указанной папке.

Менеджер отчетов поддерживает следующие типы файлов для сохранения отчетов: **PDF, Excel, PNG, JPEG, TIF, GIF, EMF и XML**. Если для рассылки в качестве формата файла отчета используется один из графических форматов **PNG, JPEG, TIF, GIF** или **EMF**, то количество приложенных к письму файлов будет соответствовать количеству страниц отчета.

#### ВНИМАНИЕ!!!

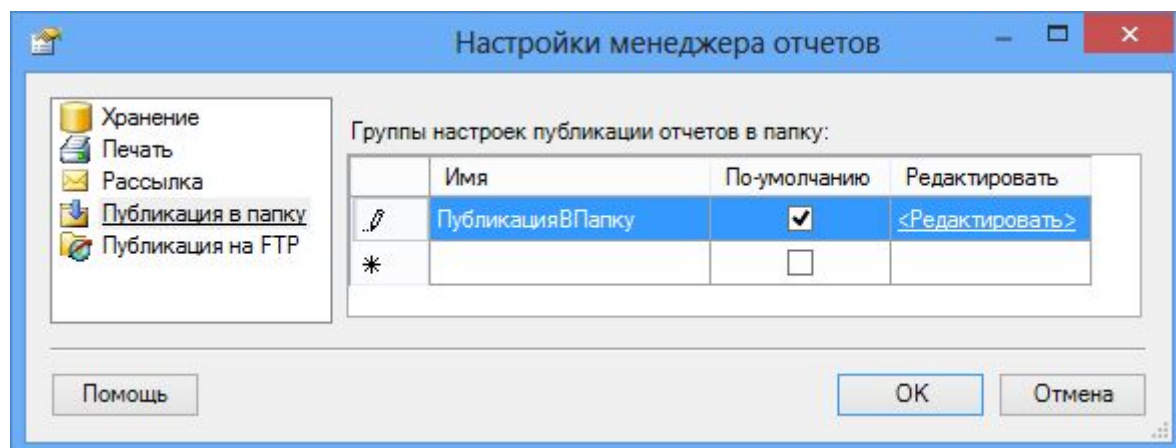
При публикации отчета в файловую систему рекомендуется:

- Обеспечивать достаточное дисковое пространство для хранения файлов отчетов
- Избегать публикации отчетов на тот же дисковый носитель, который содержит файл подкачки Windows.

### Сохранение отчета при генерации

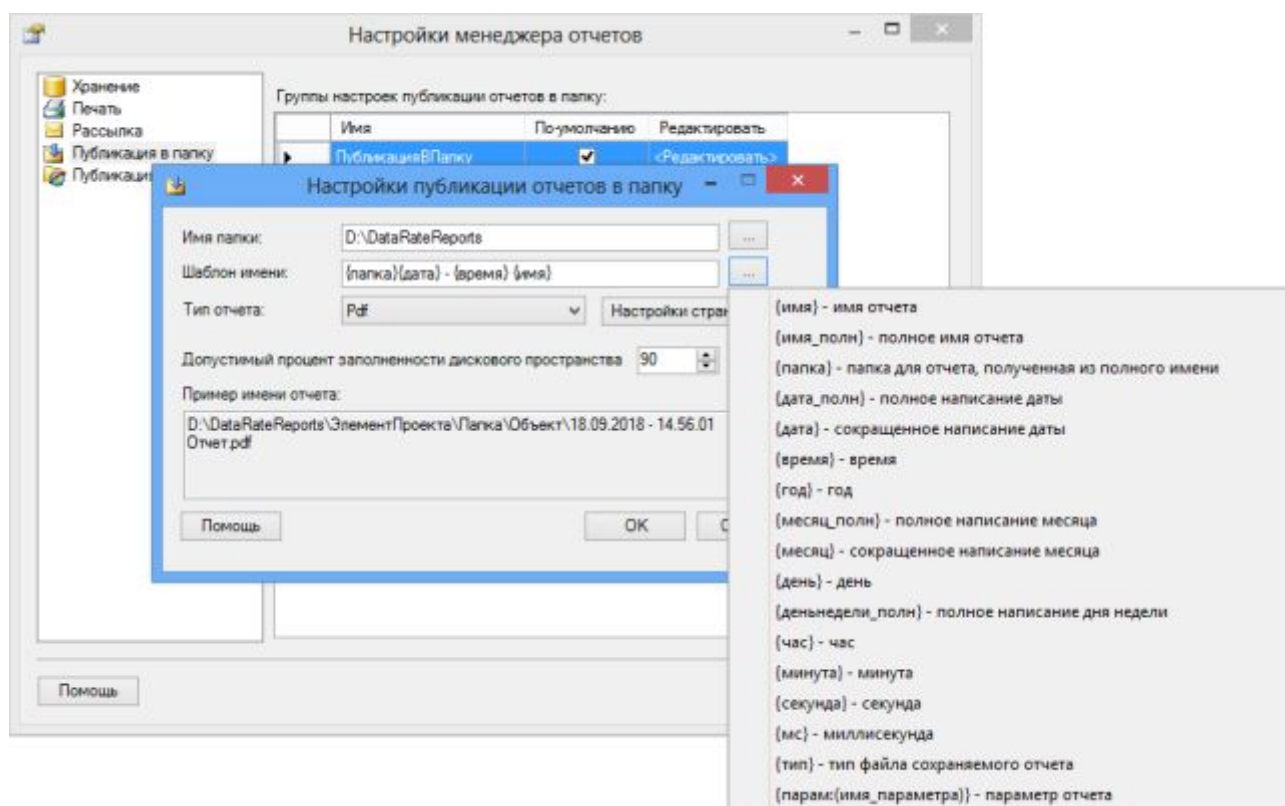
Для того чтобы сохранить отчет при генерации следует:

- 1 Вызвать **Менеджер отчетов** и в появившемся окне **Настройки менеджера отчетов** выбрать настройки **Публикация в папку**, создать группу настроек публикации

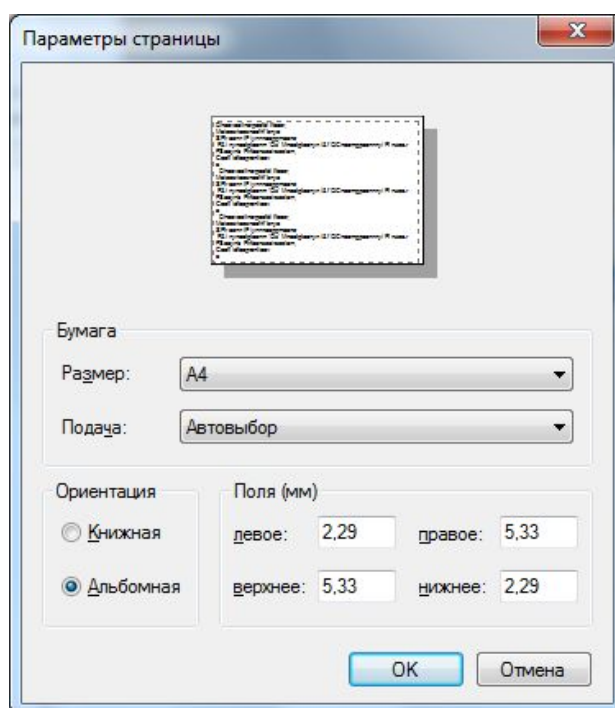


В столбце **Имя** при необходимости указать новое имя для группы.

- 2 Для изменения настроек щелкнуть мышью в ячейке **<Редактировать>**



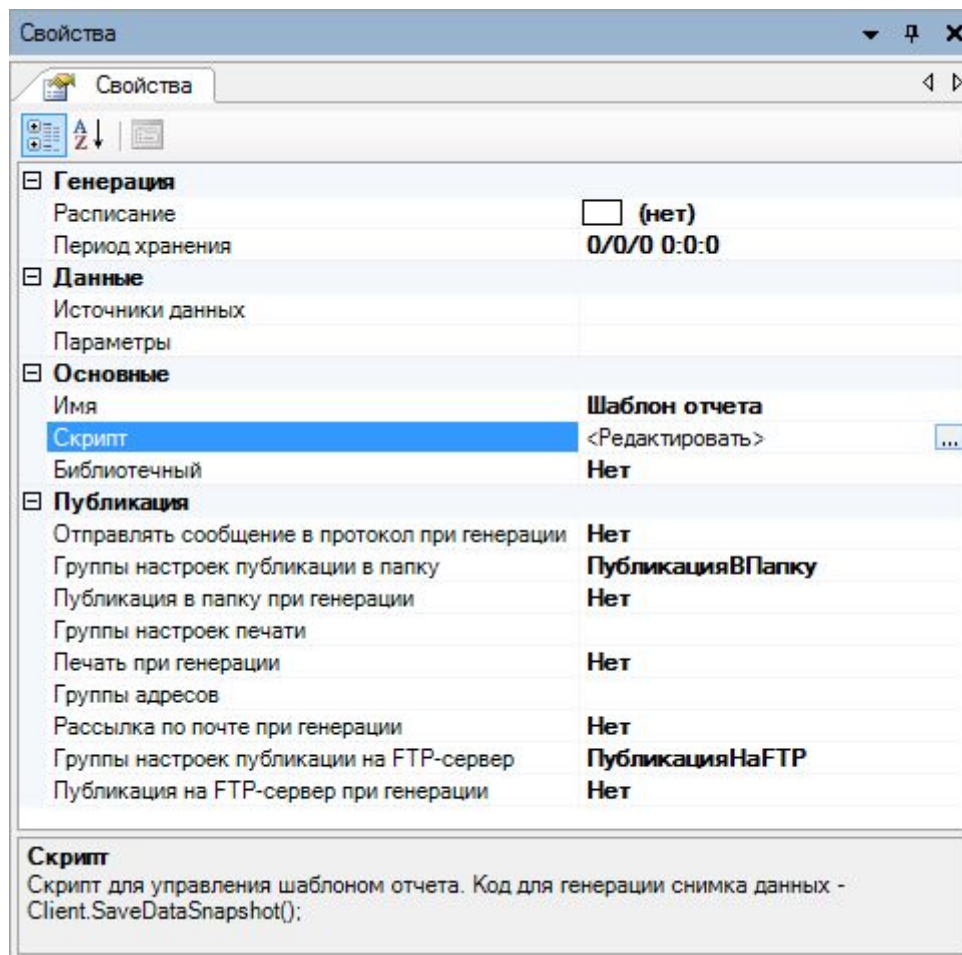
- 3 В появившемся окне **Настройки публикации отчетов в папку** задать: **Имя папки** публикации, **Шаблон имени** для файлов, **Тип отчета** для файла сохранения отчета, **Допустимый процент заполнения дискового пространства** (в случае превышения допустимого лимита отчеты не будут сгенерированы, а в логах и в протоколе событий появится специальное сообщение)
- 4 Для изменения настроек страницы нажать на кнопку **Настройки страницы**
- 5 Открыть шаблон отчета, выбрать в свойстве **Группы настроек публикации в папку** имя созданной группы настроек и установить свойство **Публикация в папку при генерации** в положение **Да**.





## Публикация отчета из скрипта

Скрипт для публикации отчета настраивается с помощью редактора скриптов **DataRate** в скрипте для генерации отчета.



Публикация отчета происходит путем вызова функции **FolderPublish**:

**FolderPublish(int historyId),**

где **historyId** – идентификатор истории отчета:

**0** – последний сгенерированный отчет, **1** – предпоследний и так далее.

**FolderPublish(DateTime snapshotDate),**

где **snapshotDate** - дата/время сформированного снимка данных.

**FolderPublish(ReportDataSnapshot snapshot),**

где **snapshot** - снимок данных.

В редакторе скриптов эту функцию можно вызвать при выполнении некоторого условия. Например, если значение аргумента скрипта превысит 0.5:

**If(Value > 0.5) Client.FolderPublish(0);**

где **Value** – аргумент скрипта, **Client** – стандартная переменная (объект, для которого запускается скрипт).

Существуют также методы для публикации в папку с непосредственным указанием папки для публикации, шаблона имени файлов и типа файла для сохранения отчета:

**FolderPublish(DateTime snapshotDate, string folderName, string nameTemplate, string exportType)**

```
FolderPublish(int historyIndex, string folderName, string nameTemplate,
string exportType)
FolderPublish(ReportDataSnapshot snapshot, string folderName, string
nameTemplate, string exportType),
```

где **folderName** – имя папки для публикации, **nameTemplate** – шаблон имени для файлов, **exportType** – тип файла публикации.

### Пример

Опубликовать последний сгенерированный отчет в формате **.pdf** в папку «D:\ПапкаПубликации», применив шаблон имени {имя}{время}:

```
Client.FolderPublish(0,@"D:\ПапкаПубликации","{имя}{время}","Pdf");
```

## 7.7.5 Публикация на FTP сервер

**Менеджер отчетов** позволяет сохранять отчет при генерации отчета по расписанию или из скриптов.

Для сохранения отчета необходимо задать группу настройки публикации на FTP-сервер. Настройки **Публикация на FTP** предназначены для сохранения отчетов на указанный FTP-сервер.

**Менеджер отчетов** поддерживает следующие типы файлов для публикации отчетов: **PDF, Excel, PNG, JPEG, TIFF, GIF, EMF** и **XML**. Если для публикации в качестве формата файла отчета используется один из графических форматов **PNG, JPEG, TIFF, GIF** или **EMF**, то количество сохранённых файлов будет соответствовать количеству страниц отчета.

### Сохранение отчета при генерации

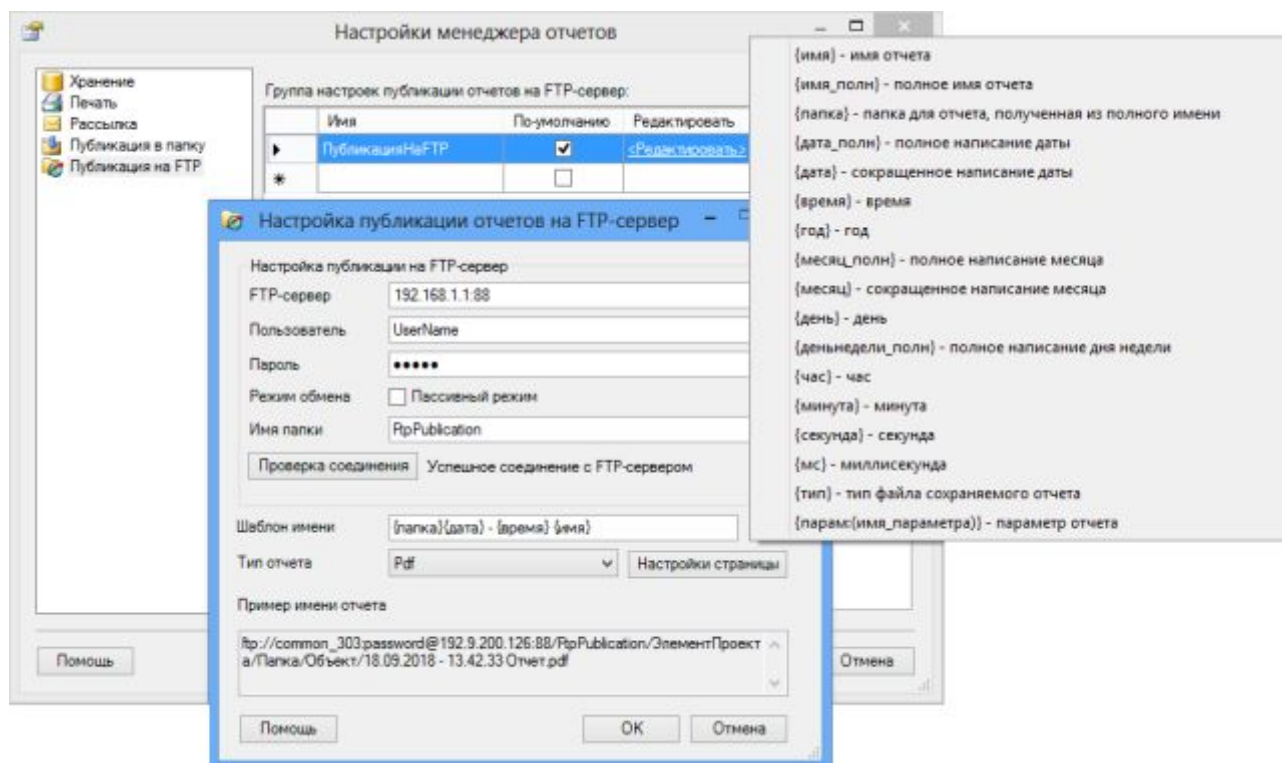
Для того чтобы сохранить отчет при генерации следует:


- 1 Вызвать **Менеджер отчетов** и в появившемся окне **Настройки менеджера отчетов** выбрать настройки **Публикация на FTP**, создать группу настроек публикации

Группа настроек публикации отчетов на FTP-сервер:			
	Имя	По-умолчанию	Редактировать
▶	ПубликацияНаFTP	<input checked="" type="checkbox"/>	<Редактировать>
*		<input type="checkbox"/>	

В столбце **Имя** при необходимости указать новое имя для группы.

- 2 Для изменения настроек щелкнуть мышью в ячейке **<Редактировать>**
- 3 В появившемся окне **Настройки публикации отчетов на FTP-сервер** задать:
  - **FTP-сервер** – обязательный параметр, который содержит IP-адрес и порт FTP-сервера для сохранения отчетов. Путь к FTP-серверу должен соответствовать шаблону: IP-адрес:порт
  - **Пользователя** – обязательный параметр, логин для подключения к FTP-серверу
  - **Пароль** – пароль для подключения к FTP-серверу



- **Режим обмена** – пассивный или активный режим обмена с FTP-сервером. Главное отличие между активным и пассивным режимом – это сторона, которая открывает соединение для передачи данных. В активном режиме FTP-клиент должен принять соединение для передачи данных от FTP-сервера. В пассивном режиме такое соединение иницирует клиент
  - **Имя папки** – путь для сохранения отчетов на FTP-сервере. Имя папки задается относительно корневого каталога FTP-сервера. Путь может содержать вложенные папки, которые разделяются символом "/"
  - **Шаблон имени** для файлов. Для этого щелкните мышью по кнопке 
  - **Тип отчета** для сохранения отчета в нужном формате.
- 4 Для проверки соединения с FTP-сервером нажмите на кнопку **Проверка соединения**.

### ВНИМАНИЕ!

Не желательно в имени папки указывать русские символы, лучше использовать символы латинского алфавита и арабские цифры.

В случае если проверка соединения возвращает ошибку:

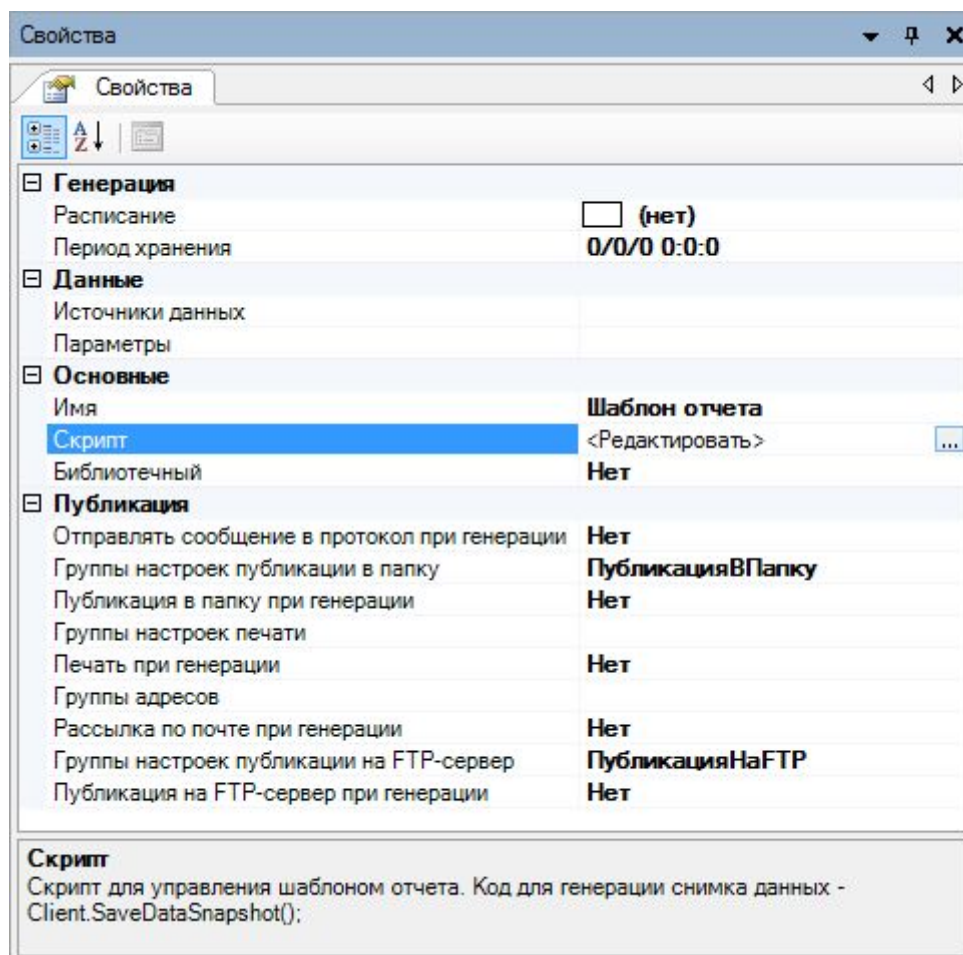
**«Удаленный сервер возвратил ошибку: (550) Файл недоступен (например, не найден или к нему нет доступа)»**, то это означает, что соединение установлено, но папка которая указана в параметре «Имя папки на FTP-сервере» не найдена (не создана).

Папка указанная в параметре Имя папки будет автоматически создана на FTP-сервере в среде исполнения проекта.

- 5 Для изменения настроек страницы нажать на кнопку **Настройки страницы**
- 6 Открыть шаблон отчета, выбрать в свойстве **Группы настроек публикации на FTP-сервер** имя созданной группы настроек и установить свойство **Публикация на FTP-сервер при генерации** в положение **Да**.

### Публикация отчета из скрипта

Скрипт для публикации отчета настраивается с помощью редактора скриптов **DataRate** в скрипте для генерации отчета.



Публикация отчета происходит путем вызова функции **FtpPublish**:

**FtpPublish(int historyId),**

где **historyId** – идентификатор истории отчета:

**0** – последний сгенерированный отчет, **1** – предпоследний и так далее.

**FtpPublish(DateTime snapshotDate),**

где **snapshotDate** - дата/время сформированного снимка данных.

**FtpPublish(ReportDataSnapshot snapshot),**

где **snapshot** - снимок данных.

В редакторе скриптов эту функцию можно вызвать при выполнении некоторого условия. Например, если значение аргумента скрипта превысит 0.5:

**If(Value > 0.5) Client. FtpPublish(0);**

где **Value** – аргумент скрипта, **Client** – стандартная переменная (объект, для которого запускается скрипт).

Существуют также методы для публикации на FTP-сервер с непосредственным указанием настроек подключения к FTP-серверу, шаблона имени файлов и типа файла для сохранения отчета:

**FtpPublish(DateTime snapshotDate, bool ftpPassiveMode, string ftpServer, string ftpLogin, string ftpPassword, string ftpFolder, string nameTemplate, string exportType);**

```
FtpPublish(int historyIndex, bool ftpPassiveMode, string ftpServer, string
ftpLogin, string ftpPassword, string ftpFolder, string nameTemplate, string
exportType);
```

```
FtpPublish(ReportDataSnapshot snapshot, bool ftpPassiveMode, string ftpServer,
string ftpLogin, string ftpPassword, string ftpFolder, string nameTemplate, string
exportType);
```

где:

**ftpPassiveMode** – режим обмена (true - пассивный режим, false - активный режим);

**ftpServer** – FTP-сервер задается по шаблону: IP-адрес:порт;

**ftpLogin** – имя пользователя для подключения к FTP-серверу;

**ftpPassword** – пароль пользователя для подключения к FTP-серверу;

**ftpFolder** – имя папки на FTP-сервере для публикации. Имя папки может содержать вложенные папки, которые разделяются символом "/";

**nameTemplate** – шаблон имени для файлов;

**exportType** – тип файла публикации (возможные варианты Pdf, Excel, Png, Jpeg, Tiff, Gif, Emf и Xml).

### Пример

Опубликовать последний сгенерированный отчет в формате .pdf на FTP-сервер в папку «ПапкаПубликации/ ВложеннаяПапкаПубликации», применив шаблон имени {имя}{время}:

```
Client.FtpPublish(0, false, "IP-адрес:порт", "логин", "пароль",
"ПапкаПубликации/ВложеннаяПапкаПубликации","{имя}{время}","Pdf");
```

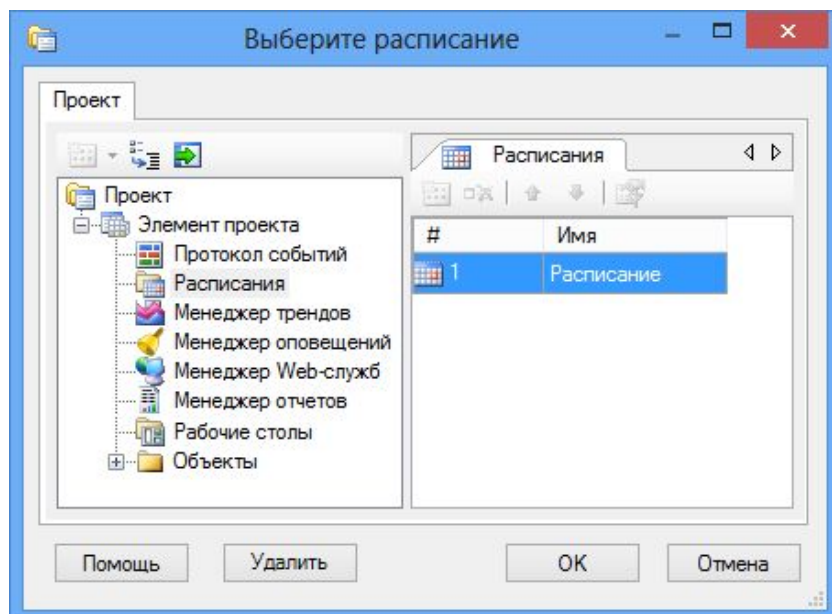
## 7.8 Генерация отчета

### Генерация отчета по расписанию

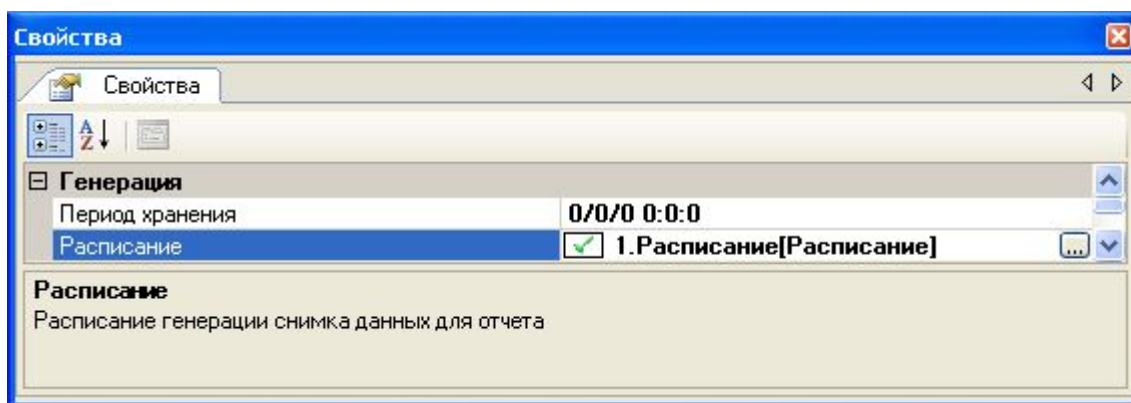
Отчет, на основе шаблона отчета, может быть сгенерирован по расписанию.

Для генерации отчета по расписанию следует

- 1 Создать расписание, по срабатыванию которого будет происходить генерация снимка данных
- 2 Выбрать расписание в свойстве **Расписание** шаблона отчета.



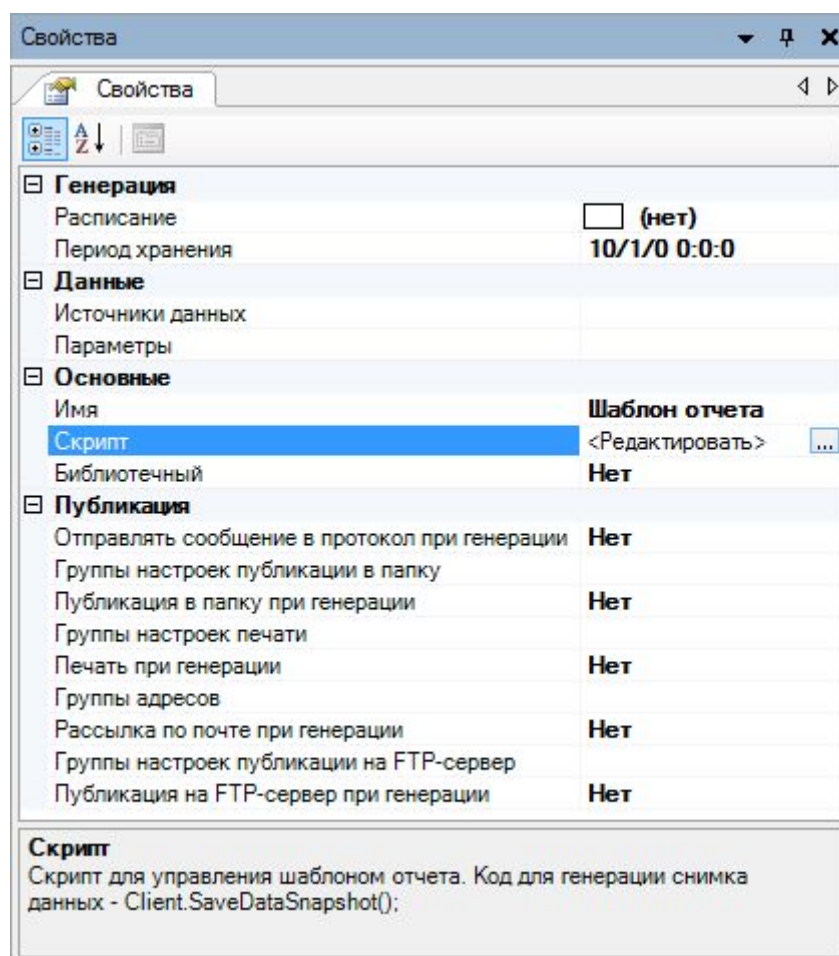




## Генерация отчета в скрипте

Для того чтобы сформировались снимок данных их скрипта необходимо:

- 1 Выбрать свойство **Скрипт** шаблона отчета и открыть редактор скрипта отчета
- 2 Создать **Аргумент**, по изменению которого будет происходить генерация, установить флажок **Запускать скрипт**
- 3 В коде скрипта задать логику, по которой будет выполняться генерация отчета



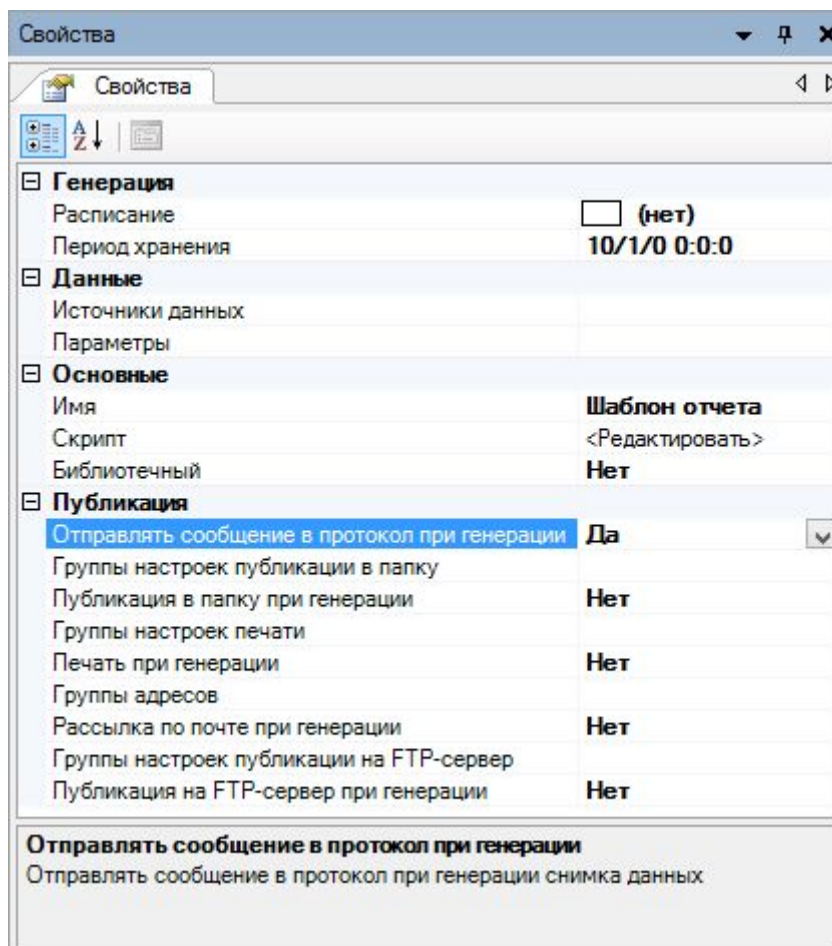
Пример кода скрипта для генерации снимка данных:

```
Client.SaveDataSnapshot();
```

Из кода скрипта можно не только генерировать снимки данных, но и печатать отчет, рассылать отчет по электронной почте и публиковать отчет в папку.

### Как послать сообщение в протокол событий при генерации отчета

Для отправления сообщения в лог при генерации отчета следует в свойстве **Отправлять сообщение в протокол при генерации** выбрать – **Да**.



Для проверки формирования сообщения в протоколе событий следует:

- 1 Создать элемент **Мнемосхема**
- 2 Разместить на мнемосхеме графический примитив **Протокол событий**
- 3 Запустить проект на исполнение.

