



Версия 4.2

Руководство Пользователя

© ООО «Энергокруг», 2018. 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, 440028, г. Пенза, ул. Титова 1

Тел. +7 (8412) 55-64-95, 55-64-97

Общие вопросы: info@scadadatarate.ru, info@energokrug.ru

Техническая поддержка: 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
SCADA DataRate™ Руководство Пользователя Обозначение документа: ИГТЯ.20500W- 04.20-И2.3.1	<u>Часть 1</u> ОСНОВЫ DataRate Руководство Пользователя ИГТЯ. 20500W-04.20-И2.3.2	1. Интегрированная среда разработки 2. Объектная модель 3. Динамическая визуализация данных
	<u>Часть 2</u> ОБЪЕКТЫ И СЕРВИСЫ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ Руководство Пользователя ИГТЯ. 20500W-04.20-И2.3.3	4. События и тревоги (алармы) 5. Расписания 6. История процесса. Тренды 7. Модуль отчетов
	<u>Часть 3</u> ОБМЕН ДАННЫМИ И СРЕДСТВА ИНТЕГРАЦИИ Руководство Пользователя ИГТЯ. 20500W-04.20-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. 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 4.2

ЧАСТЬ 1

ОСНОВЫ DataRate™

Руководство Пользователя

Основы **DataRate™**: Руководство Пользователя./3-е изд.

© ООО «ЭнергоКруг», 2018. 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, 440028, г. Пенза, ул. Титова 1

Тел. +7 (8412) 55-64-95, 55-64-97

Общие вопросы: info@scadadatarate.ru, info@energokrug.ru

Техническая поддержка: support@krugsoft.ru, support@scadadatarate.ru

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

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

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ	1
В.1. Системные требования	3
В.2. Технические характеристики	4
В.3. Инсталляция и начало работы	5
В.3.1 Установка DataRate	5
В.3.2 Установка .Net Framework 3.5 для Windows Server 2008 R2 Standart	11
В.3.3 Установка .Net Framework 3.5 для Windows 8 или Windows Server 2012t	15
В.3.4 Начало работы с DataRate	15
В.4. Деинсталляция	15
В.5. Регистрация и защита программными и аппаратными ключами	16
В.5.1 Программный ключ	16
В.5.2 Аппаратный ключ и защита	19
В.5.3 Работа в ознакомительном и ограниченном режимах	21
В.6. Совместимость версий	22
В.7. Смена операционной системы	22
В.7.1 Подготовка	22
В.7.2 Установка или обновление Windows	26
В.7.3 Установка SCADA DataRate	26
1 ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ	1-1
1.1 Окно Добро пожаловать	1-1
1.2 Рабочее пространство	1-2
1.2.1 Окно Настройки	1-3
1.2.2 Окно Аргументы	1-10
1.2.3 Окно Верификация	1-11
1.2.4 Окно Отладка	1-11
1.2.5 Окна вывода сообщений	1-13
1.2.6 Окно браузера скриптов	1-15
1.2.7 Окно выбора элементов	1-16
1.3 Работа с большим количеством объектов	1-16
1.3.1 Расширенный список колонок	1-17
1.3.2 Групповая привязка	1-17
1.3.3 Множественная вставка из буфера	1-17
1.4 Мышь и Drag-n-Drop	1-17
1.5 Горячие клавиши	1-20
2 ОБЪЕКТНАЯ МОДЕЛЬ	2-1
2.1 Проект	2-3
2.1.1 Свойства проекта	2-4
2.1.2 Функции редактирования проекта	2-5
2.1.3 Редактор проекта	2-5
2.1.4 Верификация проекта	2-16
2.1.5 Сохранение проекта	2-17
2.1.6 Отладка проекта	2-17
2.1.7 Лицензируемые параметры проекта	2-18
2.2 Элемент проекта	2-19

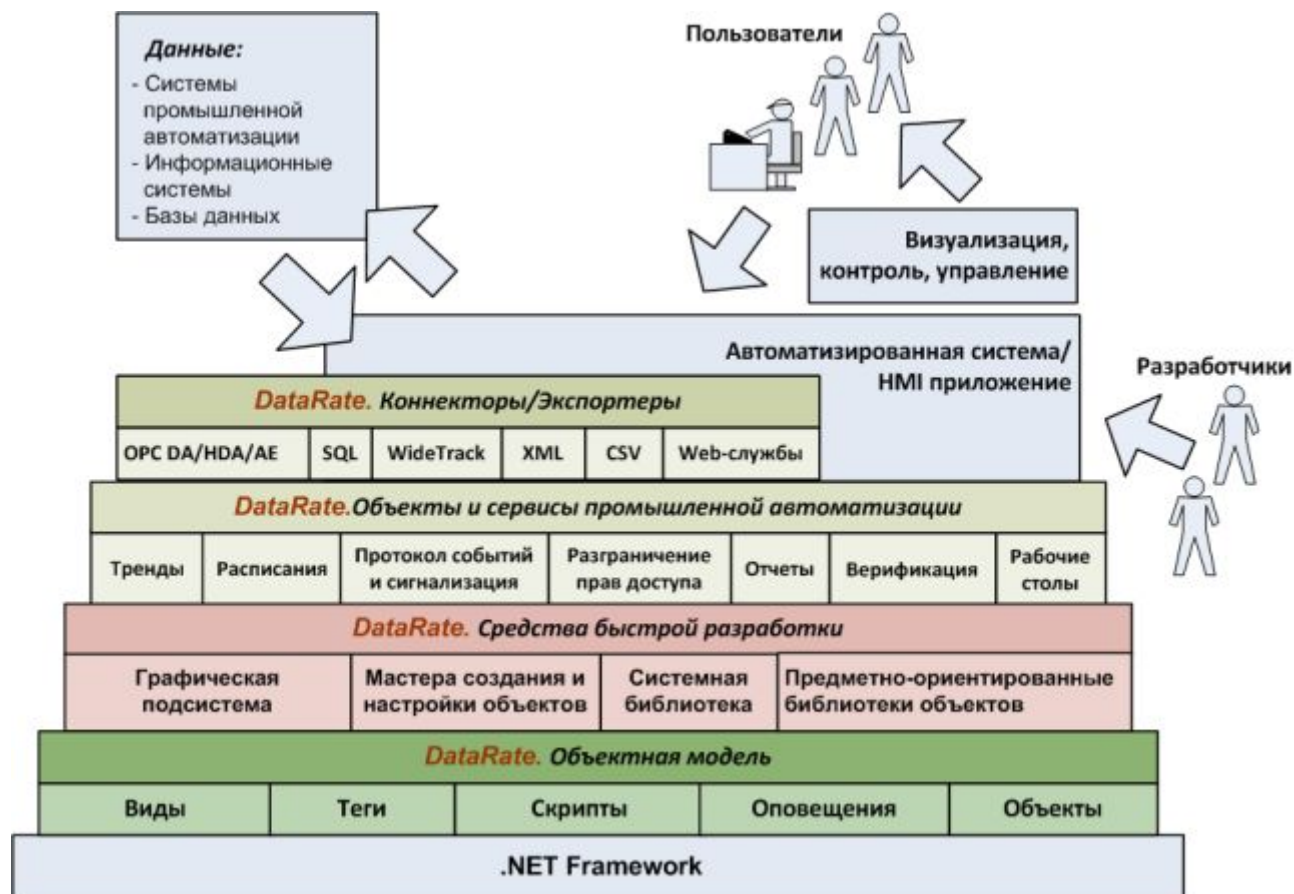
СОДЕРЖАНИЕ

	Стр.
2.2.1 Свойства элемента проекта _____	2-21
2.2.2 Защита элемента проекта _____	2-21
2.2.3 Сжатие элемента проекта _____	2-25
2.2.4 Импорт и экспорт базы данных тегов _____	2-25
2.3 Библиотеки _____	2-26
2.3.1 Редактор библиотек _____	2-26
2.3.2 Библиотека объектов _____	2-27
2.3.3 Библиотека типов тегов _____	2-28
2.3.4 Библиотека скриптов _____	2-31
2.3.5 Графическая библиотека _____	2-32
2.3.6 Управление библиотеками _____	2-32
2.4 Объекты _____	2-35
2.4.1 Теги _____	2-37
2.4.2 Виды _____	2-58
2.4.3 Скрипты _____	2-61
2.4.4 Аргументы _____	2-81
2.4.5 Оповещение _____	2-88
2.4.6 Поиск объектов _____	2-88
2.4.7 Множественная вставка _____	2-90
2.4.8 Формирование шаблона имени _____	2-91
2.4.9 Мастер настройки технологических объектов _____	2-92
2.4.10 Верификация объектов _____	2-107
3 ДИНАМИЧЕСКАЯ ВИЗУАЛИЗАЦИЯ ДАННЫХ _____	3-1
3.1 Графическая подсистема _____	3-1
3.1.1 Рабочие столы _____	3-1
3.1.2 Мнемосхема _____	3-10
3.1.3 Графические примитивы _____	3-18
3.2 Анимация _____	3-51
3.2.1 Простое преобразование _____	3-52
3.2.2 Преобразование скриптом _____	3-61
3.2.3 Вход пользователя проекта _____	3-61
3.3 Реакции _____	3-63
3.3.1 Редактирование событий _____	3-64
3.3.2 Присвоить значение _____	3-66
3.3.3 Переход _____	3-67
3.3.4 Выполнить скрипт _____	3-73
3.3.5 Поле ввода _____	3-74
3.3.6 Ползунок _____	3-76
3.4 Копирование реакций и анимаций _____	3-77
3.5 Примеры использования реакций и анимаций _____	3-79
3.5.1 Видимость/мигание _____	3-80
3.5.2 Управление открытием/закрытием мнемосхем _____	3-81
3.5.3 Цвет _____	3-85
3.5.4 Заливка _____	3-87

ВВЕДЕНИЕ

SCADA *DataRate* – современный программный продукт для быстрого создания систем автоматизации с расширенными функциональными возможностями.

DataRate предоставляет не только функции операторского интерфейса, но и функции опроса данных, управления, мониторинга и диагностики процессов промышленной автоматизации.



SCADA *DataRate* – это:

Интегрированная среда разработки

- Легкое освоение и быстрая разработка
- Объектная модель
- Повторное использование объектов
- Встроенный механизм верификации проекта

Создание решений промышленной автоматизации

- Специализированные объекты – тег, тренд, протокол событий и другие
- Ведение тренда для любого входа или выхода объекта
- База данных трендов
- Подсистема событий и тревог
- Пользовательские функции на C#
- Технология клиент-сервер
- Управление через Web-интерфейс
- Подсистема отчетов

Динамическая визуализация данных

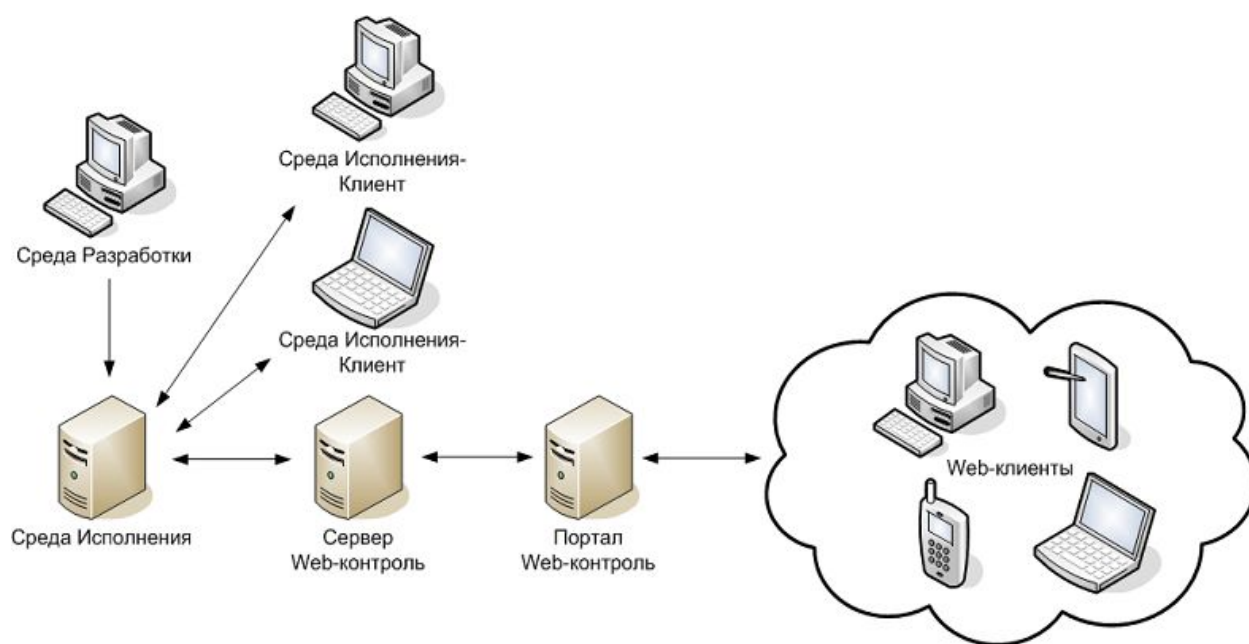
- Анимация любого свойства графического примитива
- Визуально настраиваемые алгоритмы анимации
- C#-скрипты для обработки и визуализации данных
- Библиотеки шаблонов и скриптов
- Динамика примитивов, управляемая событиями

Обмен данными

- Источники данных – УСО, информационные системы, базы данных
- OPC DA/HDA/AE коннекторы (обмен информацией с OPC DA/HDA/AE серверами)
- SQL-коннектор (обмен информацией с реляционными базами данных)
- WideTrack коннектор (получения текущих значений и истории – трендов из базы данных сервера консолидации данных WideTrack)
- Одновременное подключение до 50 OPC DA серверов
- Экспорт в формате XML, CSV, 80020
- Microsoft .NET – гибкость, надежность и безопасность.

Комплекс программного обеспечения

- Интегрированная среда разработки
- Сервер приложений – Среда Исполнения
- "Толстый" клиент – Среда Исполнения-Клиент
- Сервер Web-контроль
- Портал Web-контроль



В.1. Системные требования

DataRate является 32-разрядным приложением и функционирует под управлением как 32-разрядных, так и 64-разрядных операционных систем Microsoft Windows

Среда Разработки/ Среда Исполнения

Компонент	Минимальные требования	Рекомендуемые
Операционная система	MS Windows 7 SP1	MS Windows 7 SP1, 8, 8.1, 10/ MS Windows Server 2008, 2012 R1/R2, MS Windows Server 2016
.NET Framework	Версия 3.5	Версия 3.5 (русская локализация)
Процессор	Intel® Core 2 Duo E7500 2,93GHz	Intel® Core™ i5 2400 3,1GHz и выше
Память	2Gb	4Gb и выше

Среда Исполнения — Клиент

Компонент	Минимальные требования	Рекомендуемые
Операционная система	MS Windows 7 SP1	MS Windows 7 SP1, 8, 8.1, 10/ MS Windows Server 2008, 2012 R1/R2, MS Windows Server 2016
.NET Framework	Версия 3.5	Версия 3.5 (русская локализация)
Процессор	Intel® Pentium™ IV 2GHz	Intel® Core 2 Duo E7500 2,93GHz и выше
Память	1Gb	2Gb

Web-клиент

Компонент	Минимальные требования	Рекомендуемые
Операционная система	MS Windows 7 SP1 Internet Explorer 10.0	MS Windows 7 SP1, 8, 8.1 / MS Windows Server 2008, 2012 R1/R2, Internet Explorer 10.0 и выше
Память	512Mb	1Gb

Сервер Web-контроль

Компонент	Минимальные требования	Рекомендуемые
Операционная система	MS Windows 7 SP1 / MS Windows Server 2008 R1/R2	MS Windows 7 SP1, 8, 8.1, 10 / MS Windows Server 2008, 2012 R1/R2, MS Windows Server 2016
.NET Framework	Версия 3.5	Версия 3.5 (русская локализация)
Процессор	Intel® Core 2 Duo E7500 2,93GHz	Intel® Core™ i7-4770K 3,9GHz /Intel® Xeon E5-2600 v2 3,5GHz
Память	4Gb*	16Gb
IIS (Internet Information Service)	7.5	7.5

Версии IIS

Номер версии IIS	Выпущена в составе ОС**
7.5	Windows 7; Windows Server 2008 R2
8.0	Windows 8; Windows Server 2012
8.5	Windows 8.1; Windows Server 2012 R2
10	Windows 10; Windows Server 2016

* – примерно 100Mb на 1 клиента + 500Mb.

** – имеется поддержка IIS Express 8.0.Рекомендуем использовать версию IIS не ниже 7.5, т.к. Web-контроль использует расширенные возможности последних версий IIS.

ВНИМАНИЕ!!!

Требования к компьютеру определяются совокупностью требований запускаемых на нем программ и объемом исполняемого проекта *DataRate*

В.2. Технические характеристики

DataRate:

- Система клиент-сервер с возможностью выполнения на локальном компьютере одного или нескольких элементов проекта в одном процессе, а также подключение к серверу *DataRate* до **100** клиентов
- Обмен данными с любыми устройствами связи с объектом (УСО, контроллеры, приборы и другие), для которых существует OPC DA (OPC Data Access) сервер (версии 1.0 или 2.0), OPC HDA сервер или OPC AE (OPC Alarm & Events) сервер
- Количество OPC DA серверов, данные которых одновременно визуализирует *DataRate* – до **50**
- Обмен данными с реляционными базами данных (MS SQLServer и другими) и офисными приложениями (MS Excel)
- Обмен данными через скрипты с Web-службами (Web-сервисом *WideTrack* и другими WCF-сервисами) – начиная с версии 4.0
- Создание и публикация отчетов
- Управление с помощью Web-интерфейса
- Система защиты от несанкционированного доступа
- Количество тегов, поступающих от OPC DA сервера – до **50000**
- Количество технологических объектов – до **10000**
- Количество графических динамических элементов в одном окне – до **1000**
- Количество графических динамических элементов в графическом проекте в целом – до **50000**
- Количество одновременно открываемых окон – до **20**
- Количество трендов – до **64000**
- Минимальный период заполнения тренда не превышает **0.1 сек.**
- Период времени обновления информации в окне – не более **250 мсек.**
- Количество скриптов – до **5000**
- Количество рабочих столов - до **200**
- Системная библиотека шаблонов и изображений (легко дополняется и модифицируется Пользователем).

В.3. Установка и начало работы

В.3.1 Установка DataRate

ВНИМАНИЕ!!!

Пользователь должен иметь права Администратора!

Для инсталляции DataRate обязательна установка:

- Microsoft .NET Framework 3.5,
- .NET Framework 3.5 русский языковой пакет,
- .NET Framework 3.5 Пакет обновления 1.

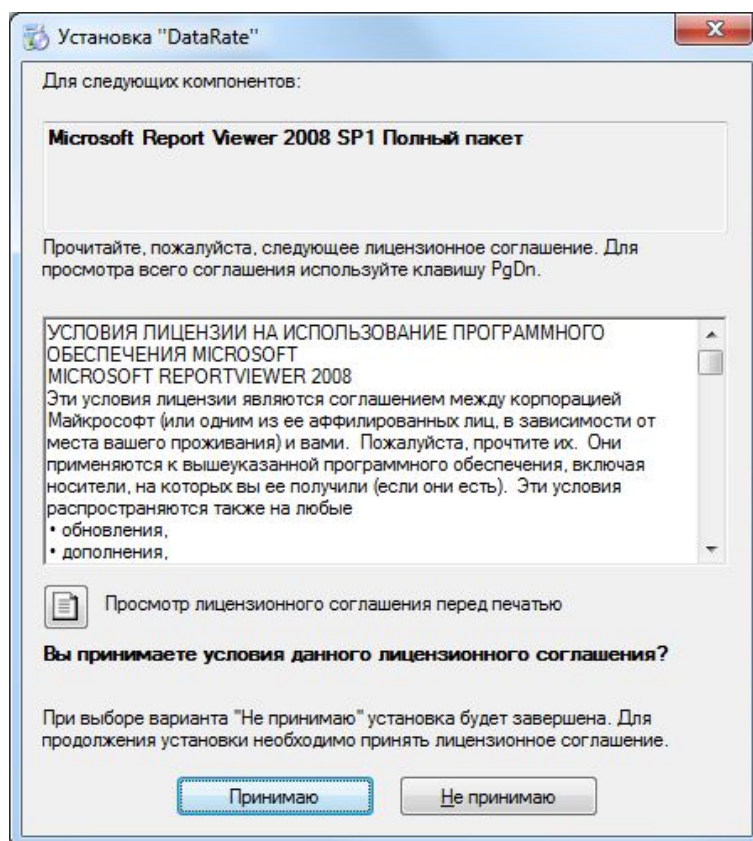
Для установки DataRate запустите инсталлятор:

<CD_drive>:\DataRate\Install\DataRate.exe

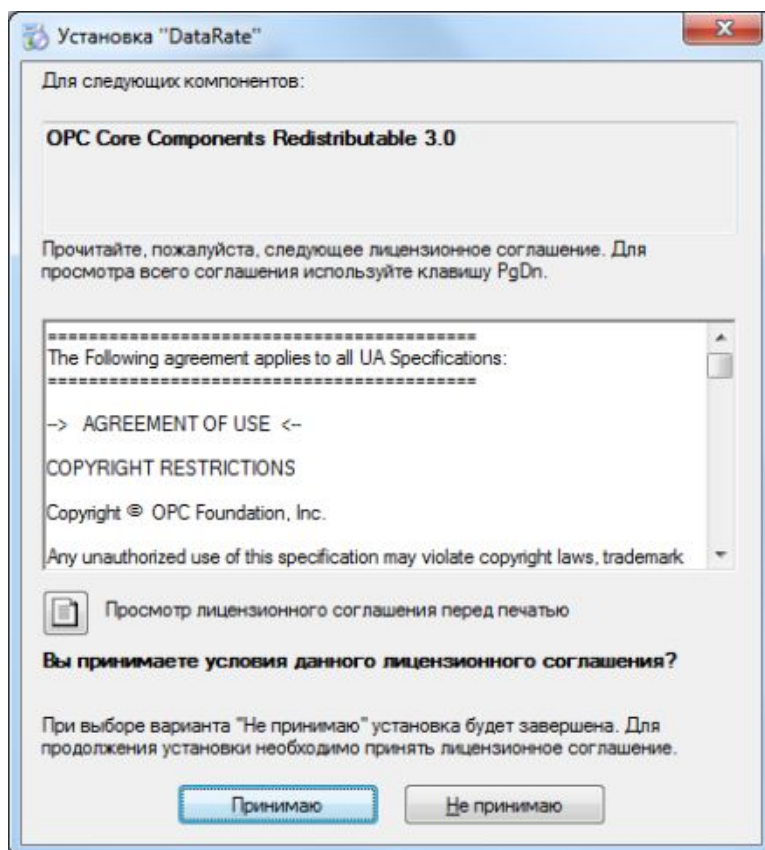
Далее следуйте указаниям инсталлятора.

Если на Вашем компьютере не установлены требуемые программные компоненты, инсталлятор предложит установить их. Для установки этих компонент следует нажать кнопку **Принимаю** в соответствующем диалоговом окне:

- **Microsoft Report Viewer** – средство просмотра отчетов

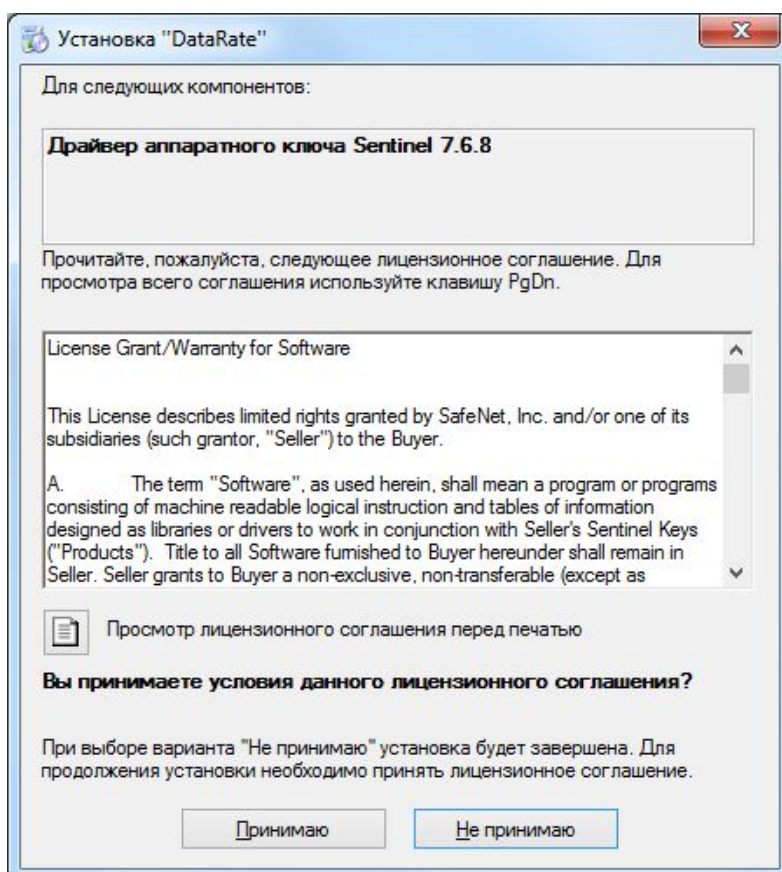


- **OPC Core Components** – компоненты для работы с OPC серверами



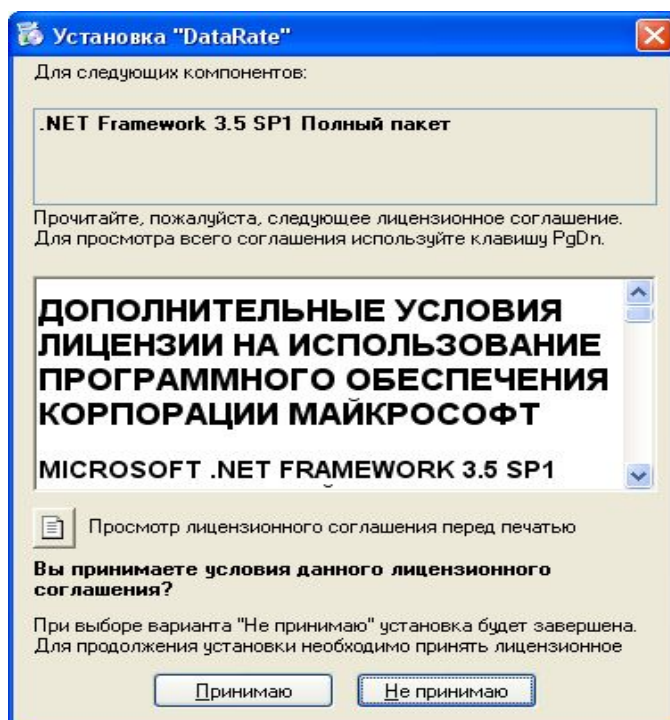
Драйвер аппаратного ключа Sentinel

После завершения установки драйвера возможно потребуется перезагрузить компьютер.

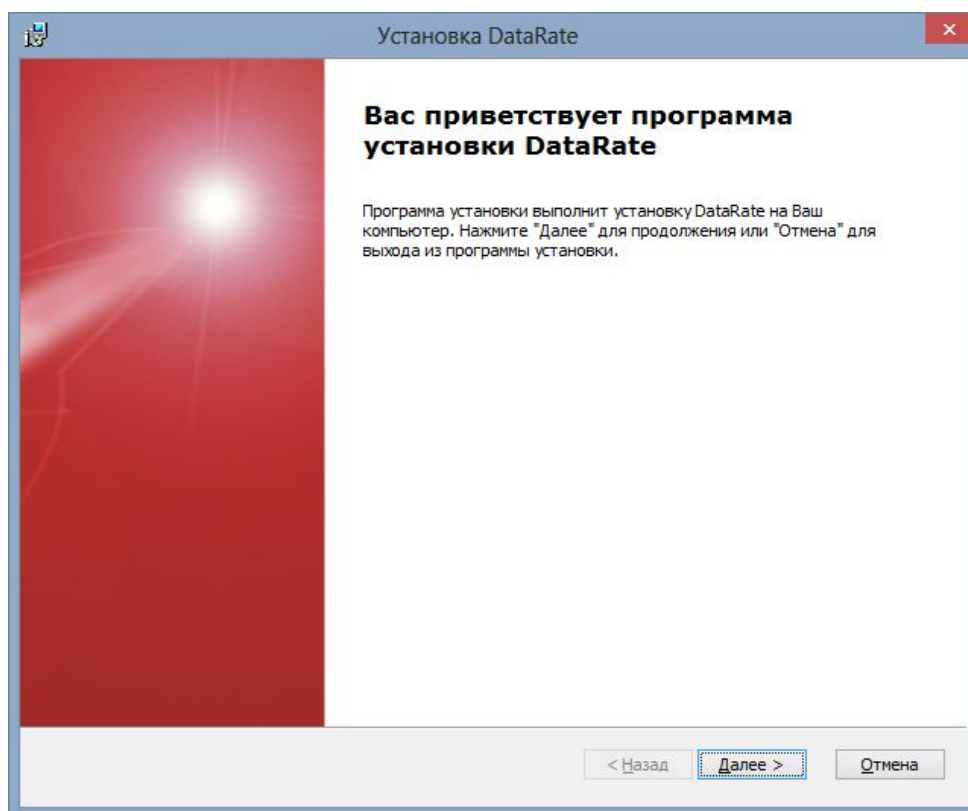


- Microsoft .NET Framework 3.5 и русский языковой пакет

После завершения установки необходимых компонентов может потребоваться перезагрузка, и в этом случае Ваш компьютер будет автоматически перезагружен.

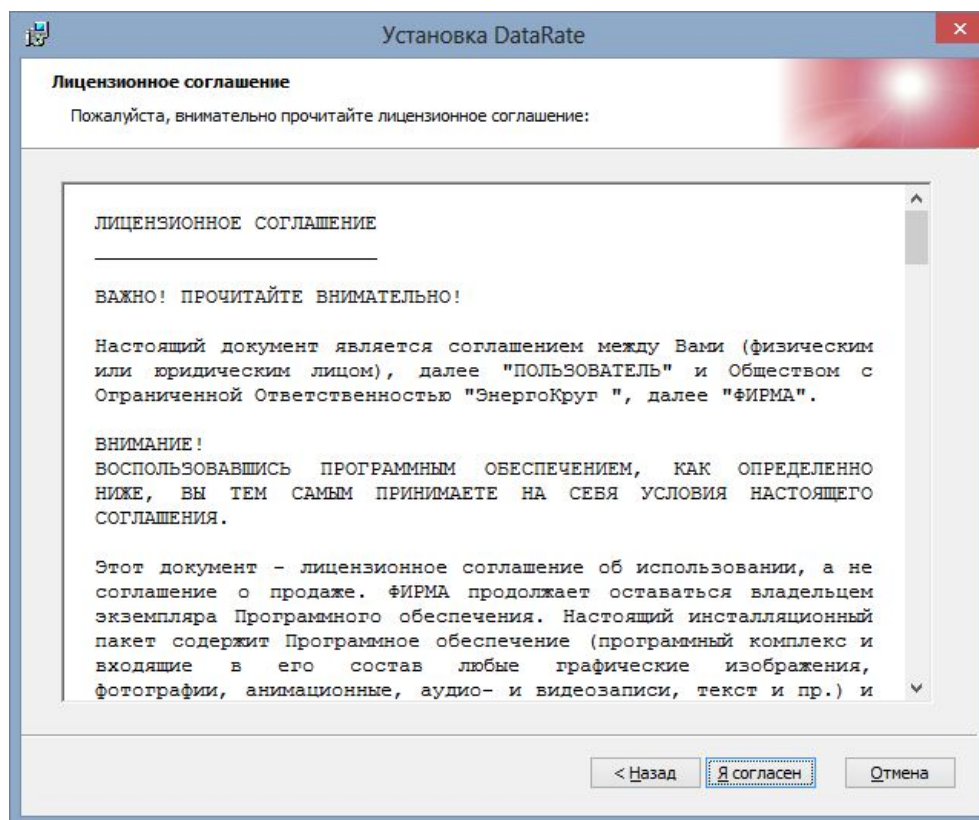


После перезагрузки инсталлятор автоматически продолжит установку **DataRate**.

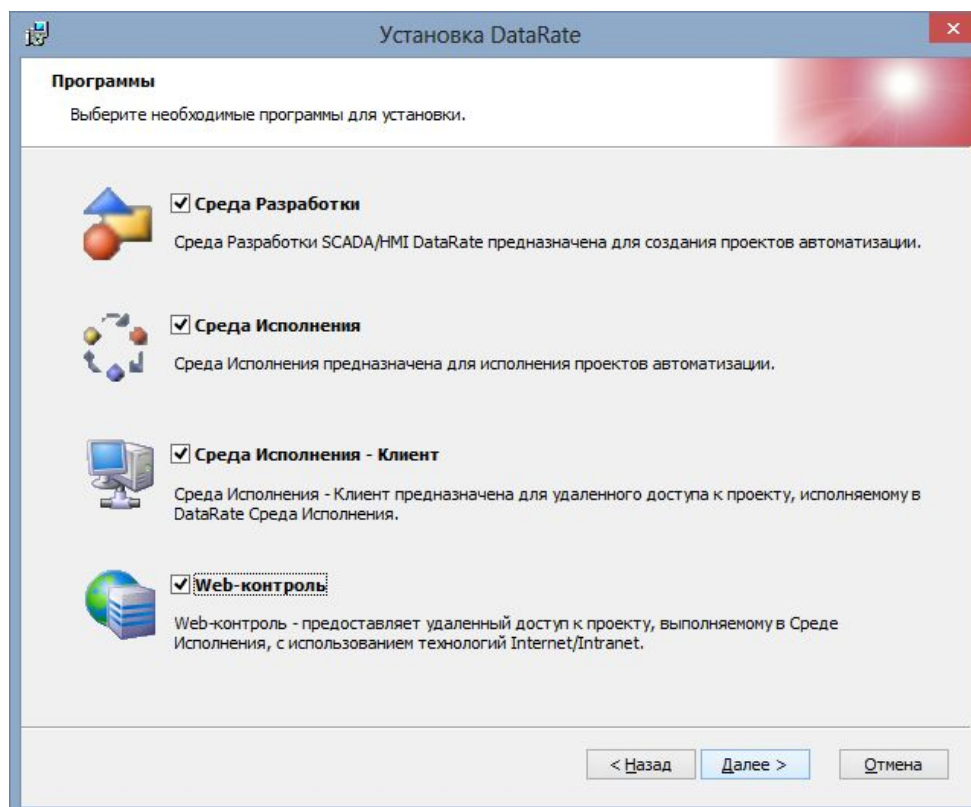


Для перехода к следующему шагу установки **DataRate** нажмите кнопку **Далее>**. В случае отказа от установки **DataRate** нажмите кнопку **Отмена**, работа инсталлятора в этом случае будет завершена.

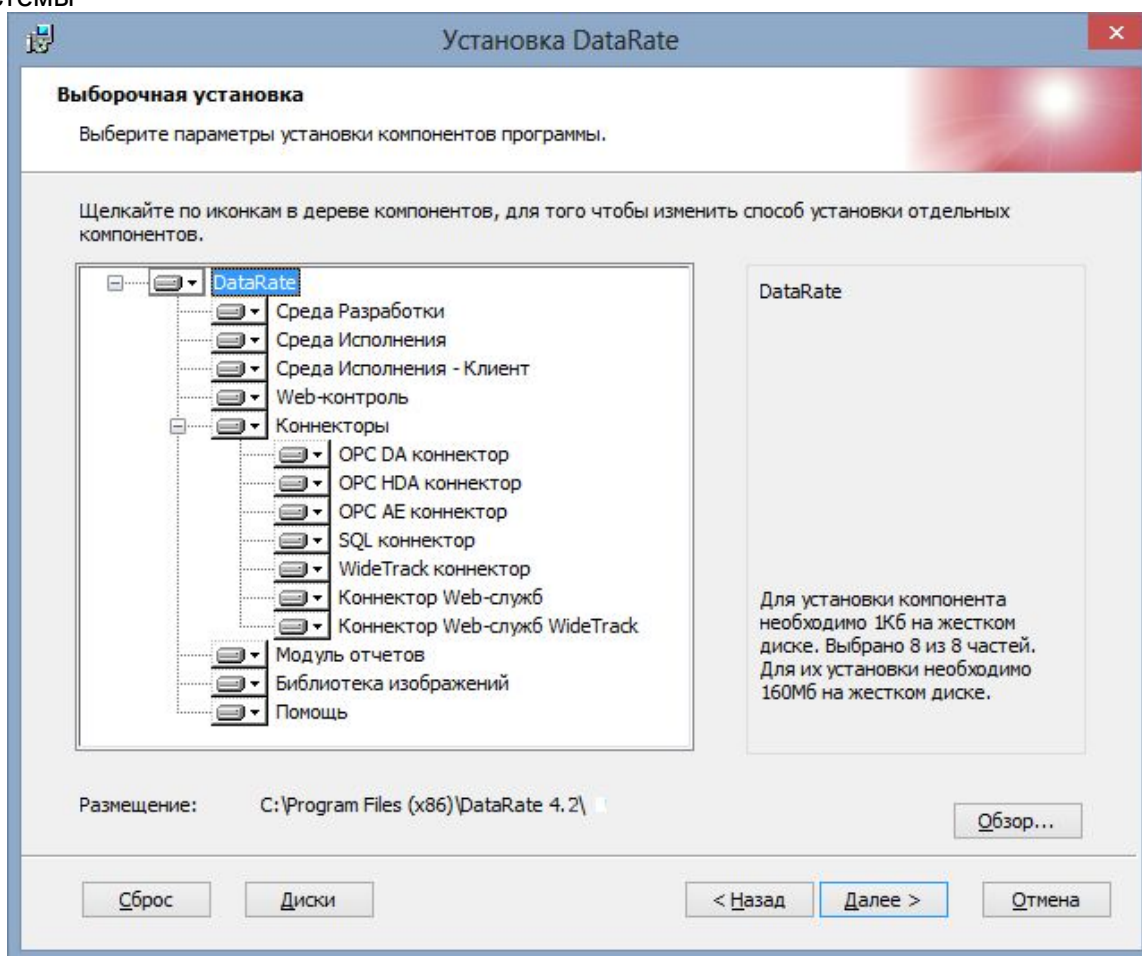
Внимательно ознакомьтесь с условиями лицензионного соглашения и, если изложенные условия Вас устраивают, нажмите кнопку **Я согласен**.



На следующем шаге выберите, какие именно подсистемы **DataRate** Вы хотите установить на свой компьютер. Для продолжения установки необходимо выбрать, по крайней мере, одну из подсистем **DataRate** и нажать кнопку **Далее>**.



Укажите компоненты **DataRate**, необходимые для создания и работы автоматизированной системы



Обозначения в дереве компонентов:

- Прямоугольный значок с белой заливкой – компонент будет устанавливаться
- Прямоугольный значок с серой заливкой – вложенные элементы будут устанавливаться, но лишь частично
- Красный крест внутри прямоугольника – данный компонент не будет устанавливаться.

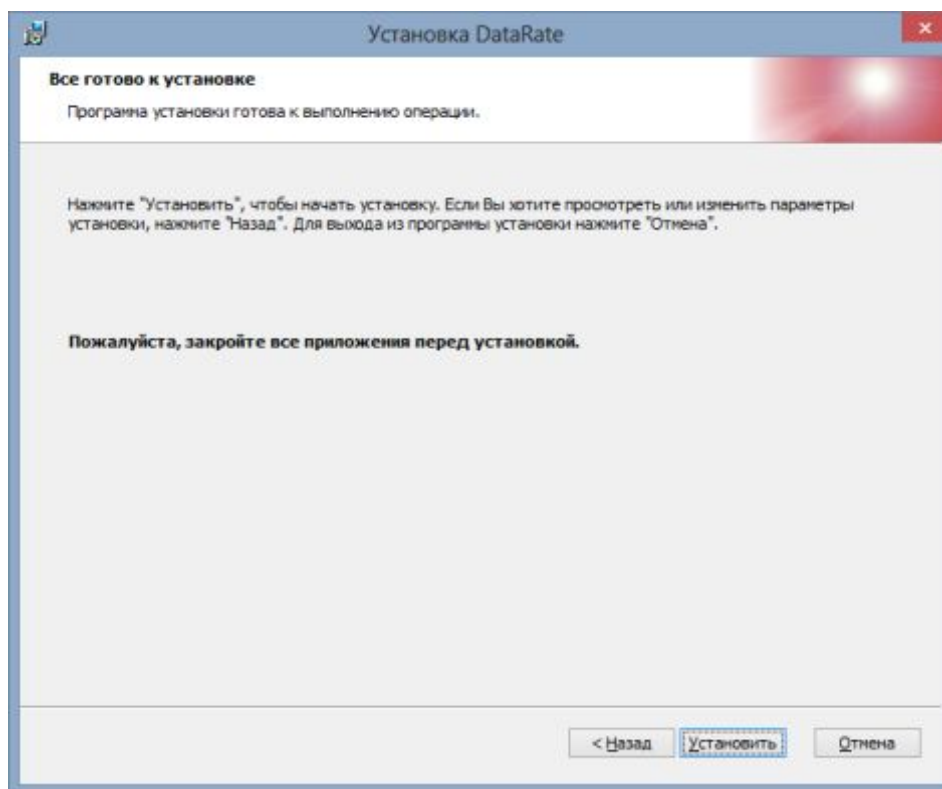
Элементы управления:

- **Обзор...** – выбор пути установки **DataRate**. Выбранный путь отображается в строке **Размещение**
- **Диски** – вывод на экран информации о наличии свободного места на дисках
- **Сброс** – возврат параметров установки к начальным значениям.

Для продолжения инсталляции нажмите кнопку **Далее>**.

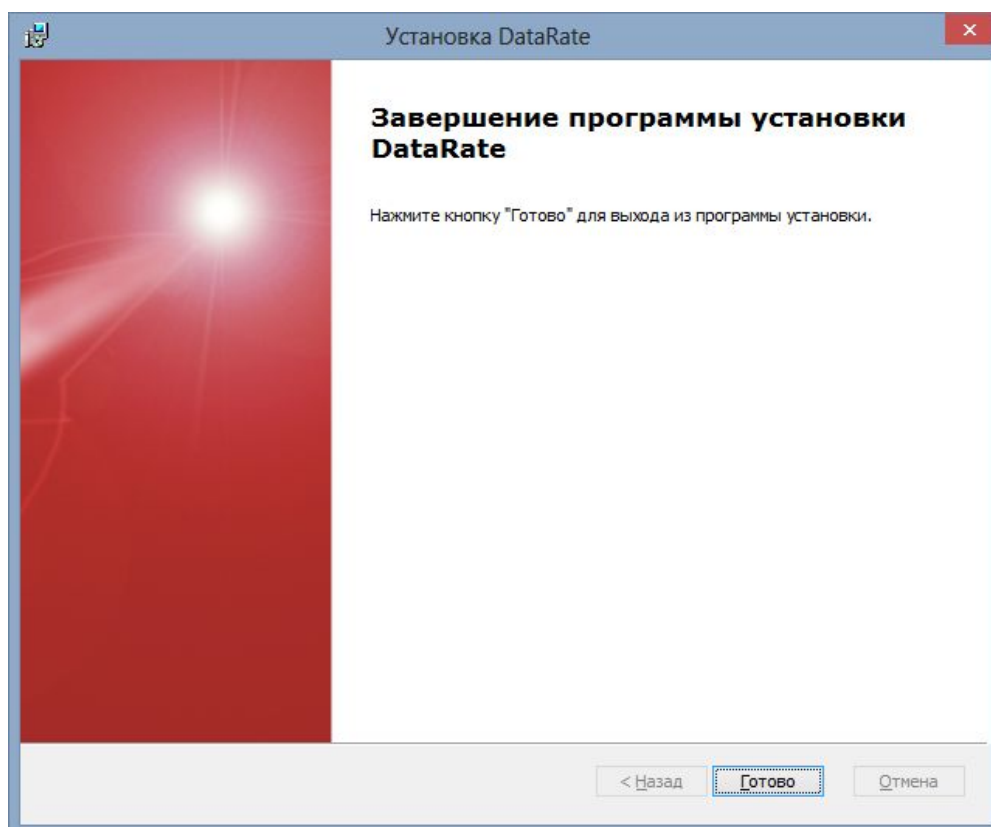
Так как на данном этапе все необходимые настройки произведены, инсталлятор отобразит на экране уведомление о готовности к установке **DataRate**.

Рекомендуется закрыть все приложения, после чего нажать кнопку **Установить** для выполнения установки **DataRate**.



В процессе установки на экране будет отображаться индикатор выполнения. После завершения установки на экране будет отображено окно с уведомлением об успешном завершении инсталляции.

Для завершения работы с инсталлятором нажмите кнопку **Готово**.



Если в процессе инсталляции возникают проблемы, то необходимо сформировать лог-файл установки [log.txt](#) и переслать его по адресу support@ScadaDataRate.ru с описанием ваших действий и результата.

Для создания лог-файла наберите в командной строке и выполните:

```
msiexec /i <пакет>.msi /lx log.txt
```

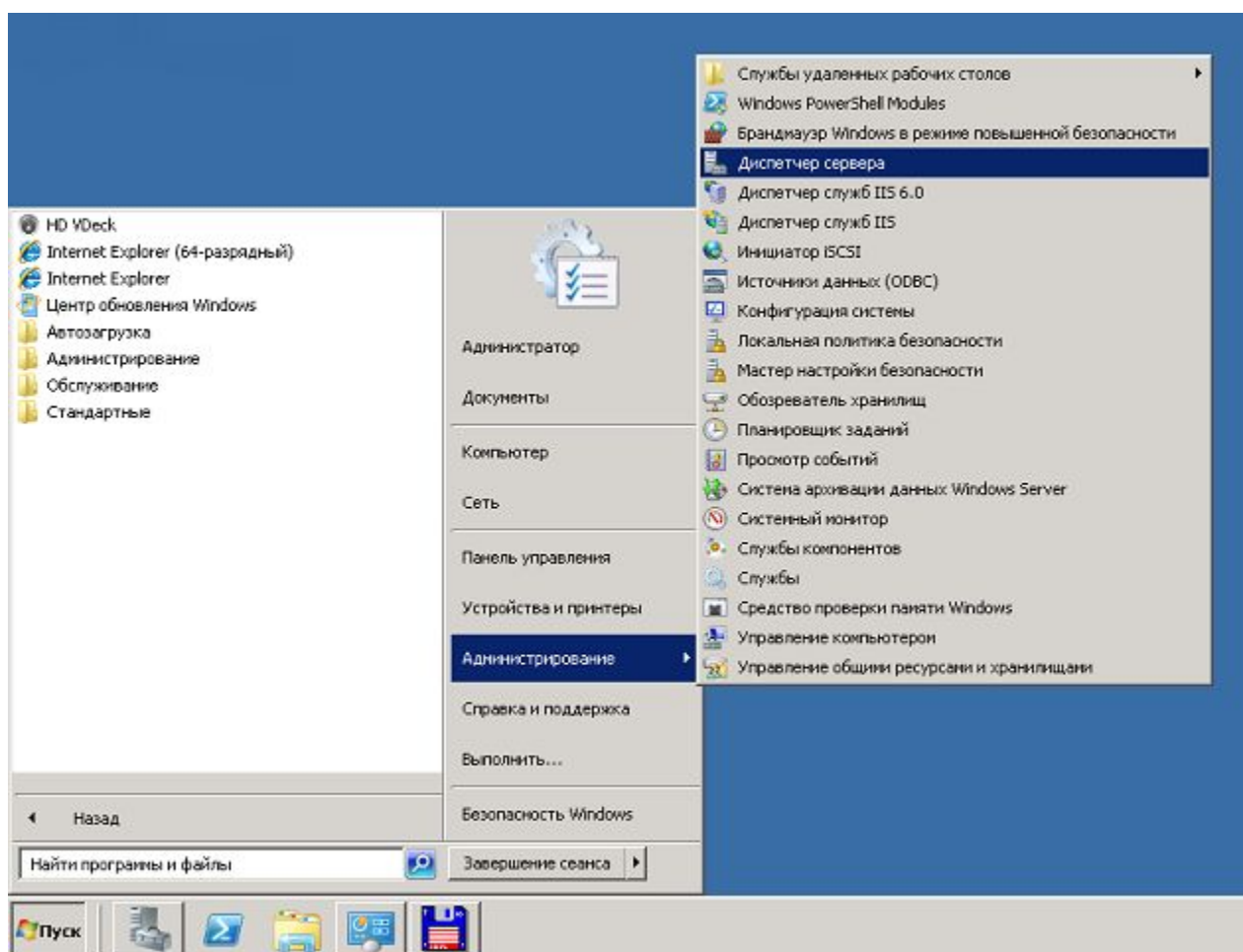
где **<пакет>** – название инсталляционного пакета,
например: [DataRate](#), [DataRate-WebControlSite](#), [DemoProjects](#).

В.3.2 Установка .Net Framework 3.5 для Windows Server 2008 R2 Standart

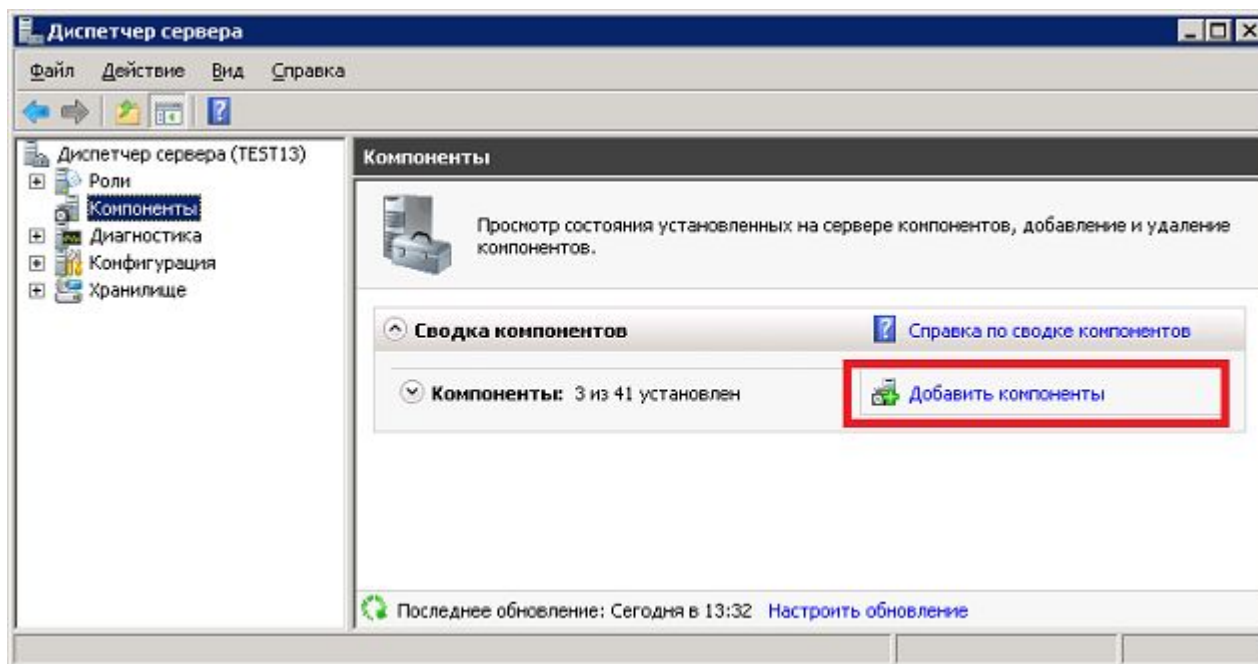
Если не удалась установка **DataRate** на машину под **ОС Windows Server 2008 R2 Standart**, убедитесь, что установлен компонент **.Net Framework 3.5**.

Порядок установки **.Net Framework 3.5** следующей:

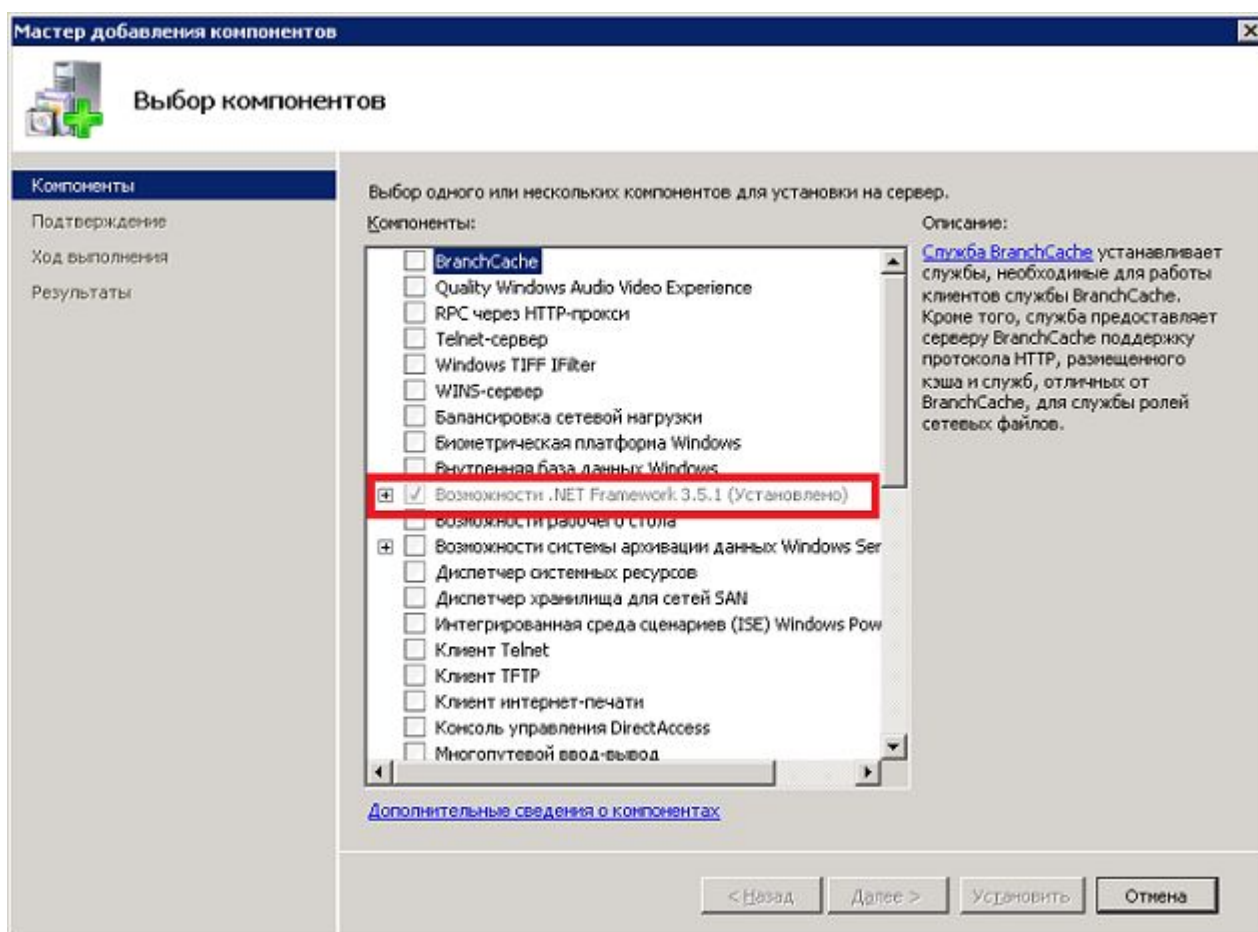
- 1 Откройте **Пуск**, выберите пункт меню **Администрирование/Диспетчер сервера**



- 2 В появившемся окне нажмите на кнопку **Добавить компоненты**

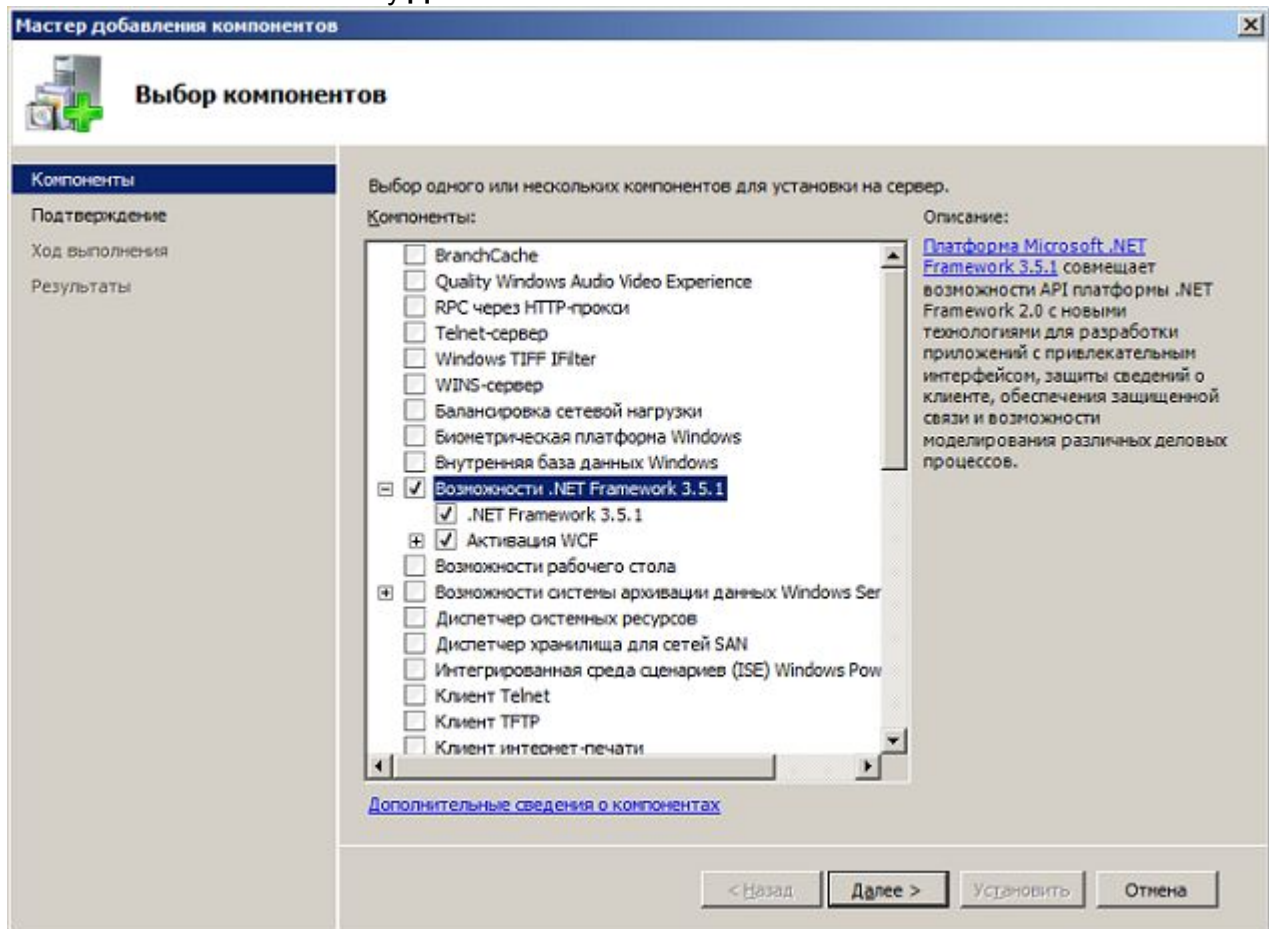


- 3 Нажмите на значок "плюс" рядом с названием компонента **Возможности .NET Framework 3.5**

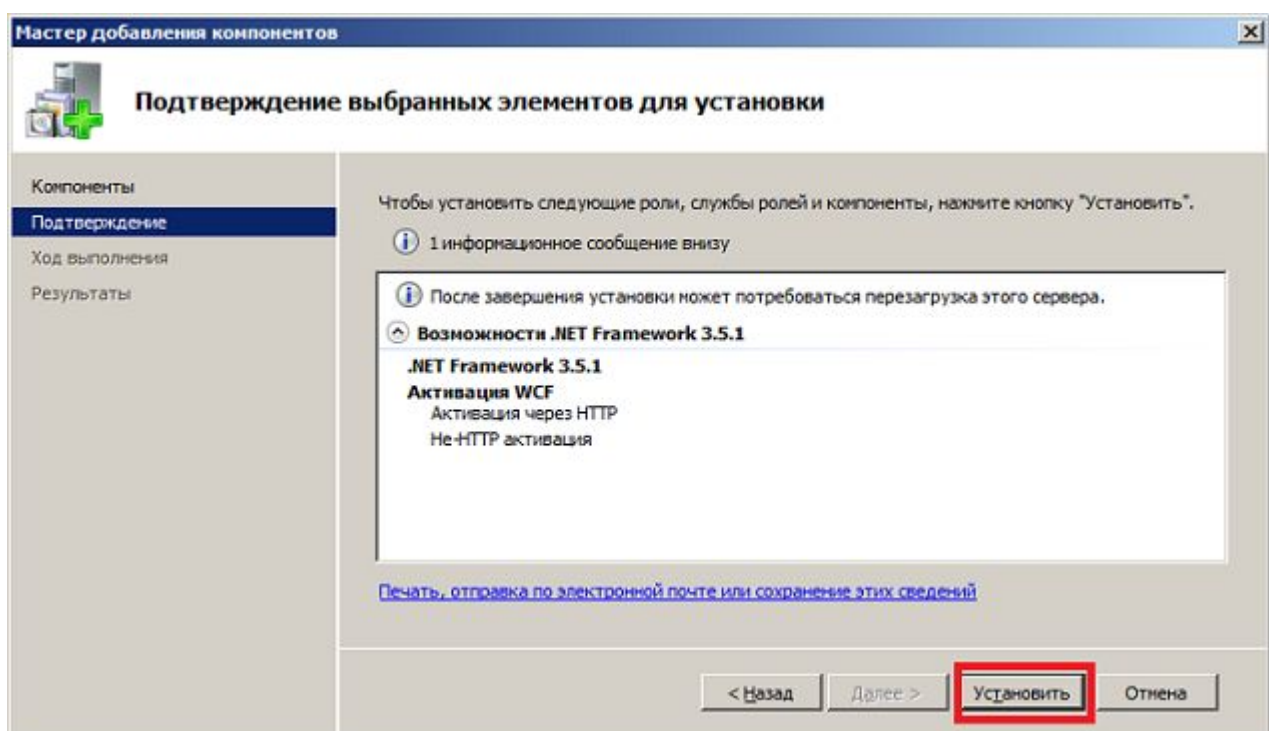


4 Установите галочку для элемента **.NET Framework 3.5**

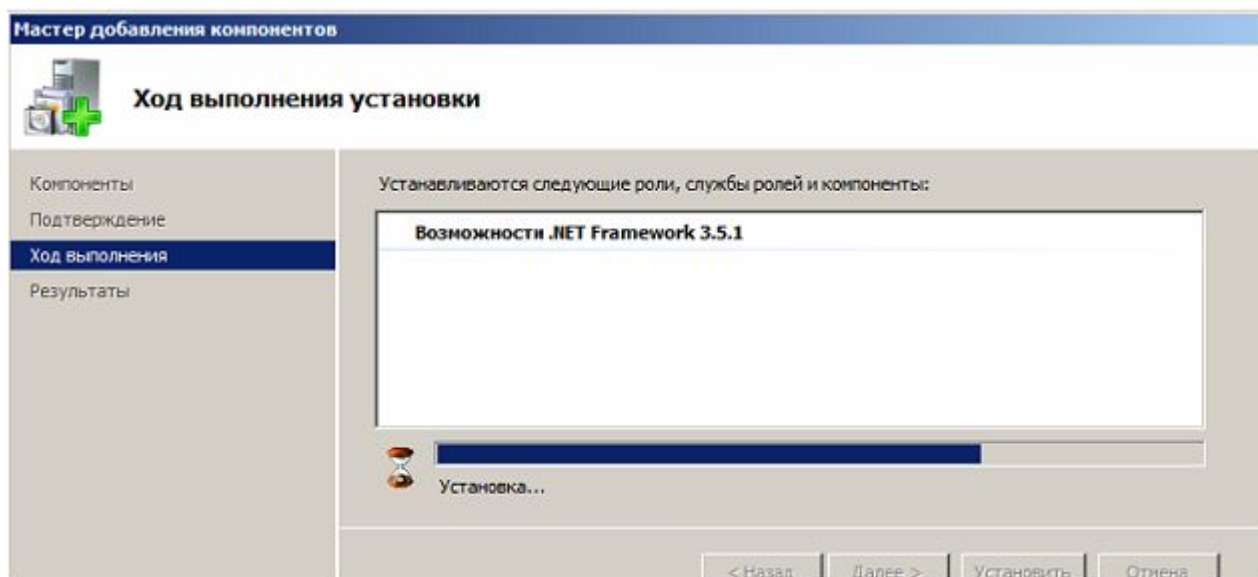
5 Нажмите на кнопку **Далее>**



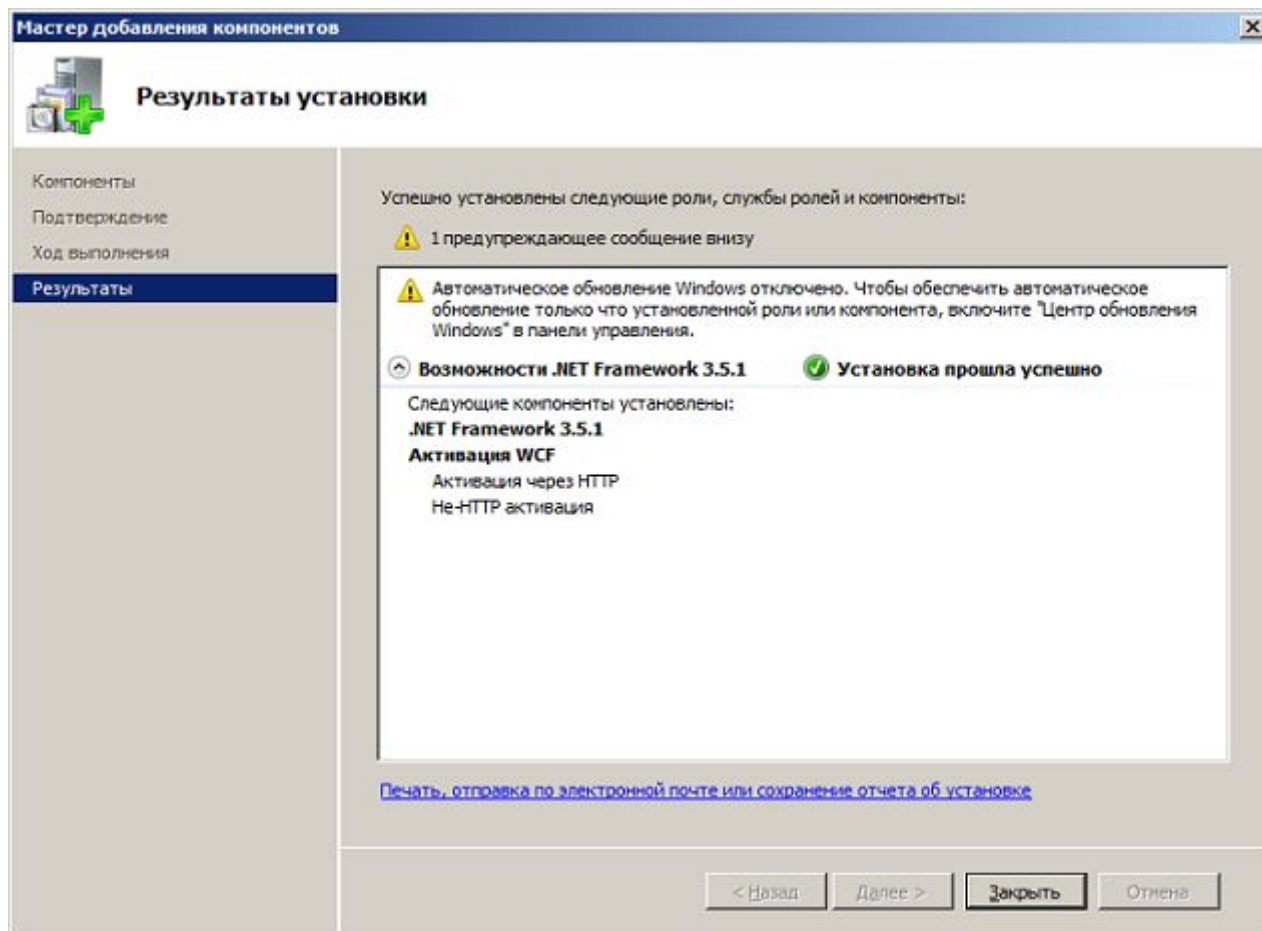
6 Нажмите на кнопку **Установить**



7 Дождитесь окончания установки



8 Закройте окно **Мастер добавления компонентов**, перезагрузите сервер и повторно запустите инсталлятор *DataRate*.



В.3.3 Установка .Net Framework 3.5 для Windows 8 или Windows Server 2012

Если не удалась установка **DataRate** на машину под **ОС Windows 8** или **ОС Windows Server 2012** по причине невозможности установки **.Net Framework 3.5**, то следуйте описанному ниже алгоритму. Если не удаётся подключить компьютер к Интернету, можно установить **.Net Framework 3.5** с помощью установочного носителя (образ ISO или DVD-диск), с которого производилась установка **ОС Windows 8** или **ОС Windows Server 2012**. Подключите установочный носитель **ОС Windows 8** и **ОС Windows Server 2012** введите следующую команду в командной строке с повышенными привилегиями:

```
Dism /online /enable-feature /featurename:NetFx3 /All /Source:x:\sources\sxs /LimitAccess
```

где

x – буква диска, для образа ISO или диска DVD с **ОС Windows 8** или **ОС Windows Server 2012**

/Online – задает запущенную операционную систему (вместо автономного образа Windows).

/Enable-Feature /FeatureName: NetFx3 – указывает, что необходимо включить платформу **.Net Framework 3.5**

/All – включает все родительские функции платформы **.Net Framework 3.5**

/LimitAccess – предотвращает контакт DISM с **Windows Update**.

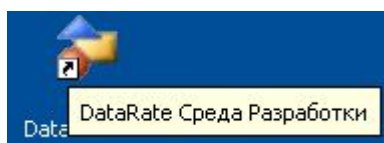
/Source – указывает расположение файлов, необходимых для восстановления функции (в этом примере каталог **x:\sources\sxs**).

Дополнительные сведения о параметрах и возможностях DISM см. в разделе «Как включать или отключать функции Windows» справочной системы ОС.

В.3.4 Начало работы с DataRate

Для того, чтобы начать разработку системы автоматизации в **DataRate**, следует:

- Щелкнуть на иконке **DataRate Среда разработки** на рабочем столе Вашего компьютера



или

- Нажать кнопку **ПУСК** и в системном меню **Все программы** выбрать **DataRate -> Среда разработки**.

В.4. Деинсталляция

Для деинсталляции **DataRate** следует:

- 1 Открыть **Панель управления\Все элементы панели управления\Программы и компоненты** для Windows Server/ 7/ 8.x/10
- 2 Выбрать название **DataRate 4.x** и нажать на кнопку **Удалить**.

В.5. Регистрация и защита программными и аппаратными ключами

Для защиты *DataRate* используются программные и аппаратные ключи.

Параметры программного/аппаратного ключа:

На использование:

- WideTrack коннектора
- SQL коннектора
- OPC AE коннектор
- *DataRate* Среда Исполнения – Клиент с управлением
- Web-клиент с управлением
- SMS оповещений
- Web служб, коннекторов Web служб, WideTrack коннектора Web служб.

Количественные:

- Общее количество лицензируемых точек ввода/вывода (OPC DA + OPC HDA + WideTrack + Web служб + клиентских коннекторов)
- Количество сессий **Web-контроль**
- Количество терминальных сеансов *DataRate* Среда Исполнения
- Количество терминальных сеансов *DataRate* Среда Исполнения – Клиент

В.5.1 Программный ключ

Программный ключ – файл, содержащий персональный регистрационный ключ, предназначенный для защиты автоматизированной системы от нелегального использования и несанкционированного распространения.

ВНИМАНИЕ!!!

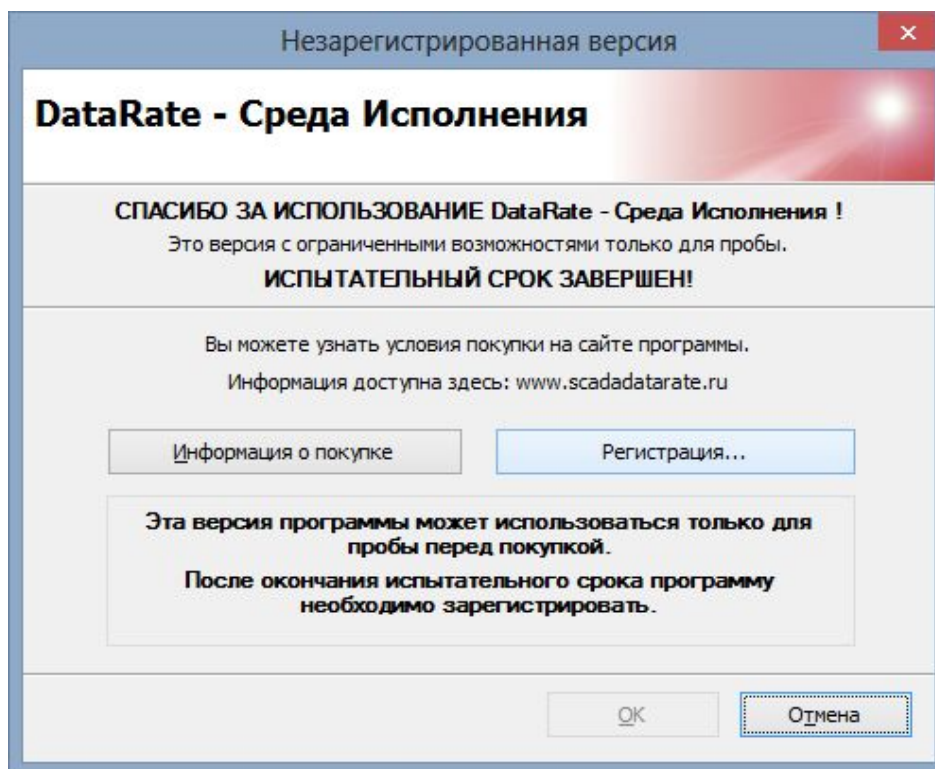
Аппаратный ключ имеет приоритет над программным ключом (при одновременном использовании аппаратного и программного ключей, учитываются только параметры аппаратного ключа).

Для регистрации *DataRate* с помощью программного ключа следует:

- 1 Получить уникальный номер компьютера. Уникальный номер компьютера отображается в окне **Регистрация**. Это окно можно вызвать двумя способами:

Способ 1

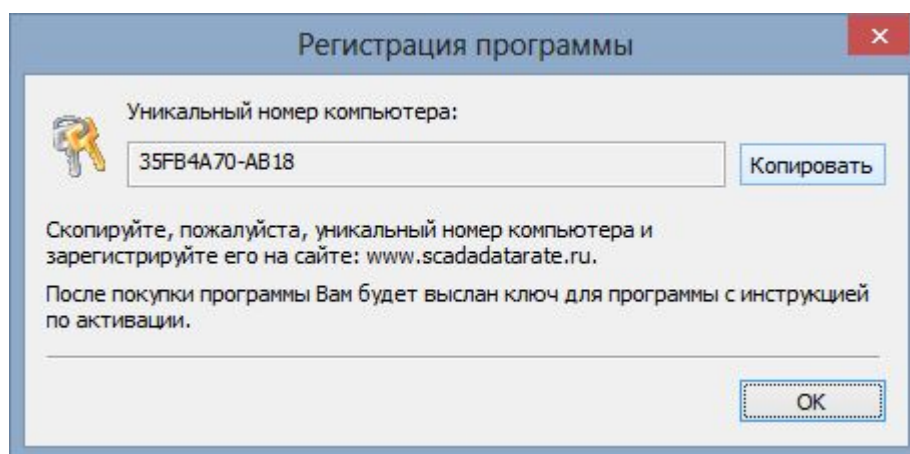
- Запустите **Среду Исполнения**
- В появившемся окне **Незарегистрированная версия** нажмите кнопку **Регистрация**



Способ 2

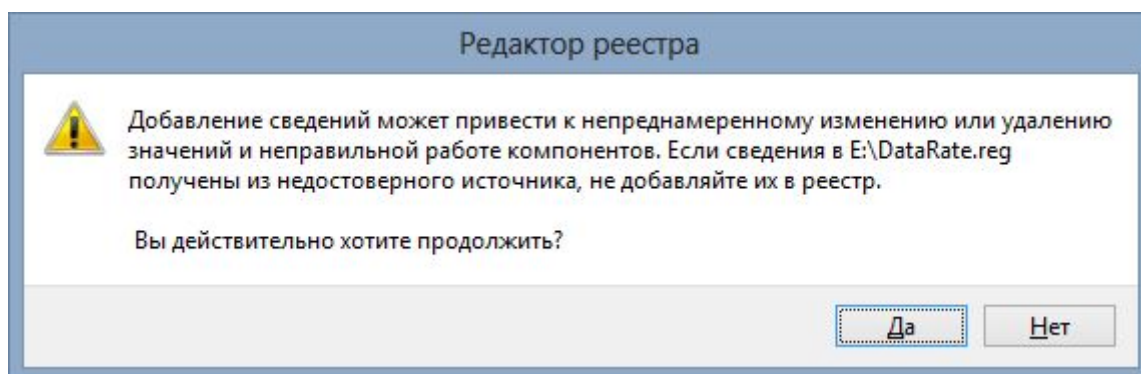
- Запустите **Среду Разработки** и выберите в меню **Помощь** пункт **О Программе**
- В появившемся окне **О программе** нажмите кнопку **Регистрация**

- 2 В появившемся окне **Регистрация программы** для сохранения уникального номера компьютера (отображается в одноименном поле) нажать на кнопку **Копировать**, а потом – на кнопку **ОК**

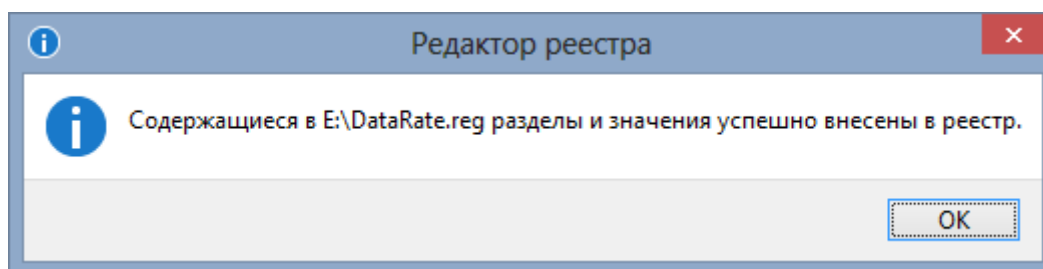


- 3 Отправьте на электронный адрес info@scadadatarate.ru или на почтовый адрес: **Россия, 440028, г.Пенза, ул.Титова, 1** следующие данные:
- **Уникальный номер компьютера**
 - **Номер регистрационной карты** (регистрационная карта входит в состав документов при приобретении **DataRate**)

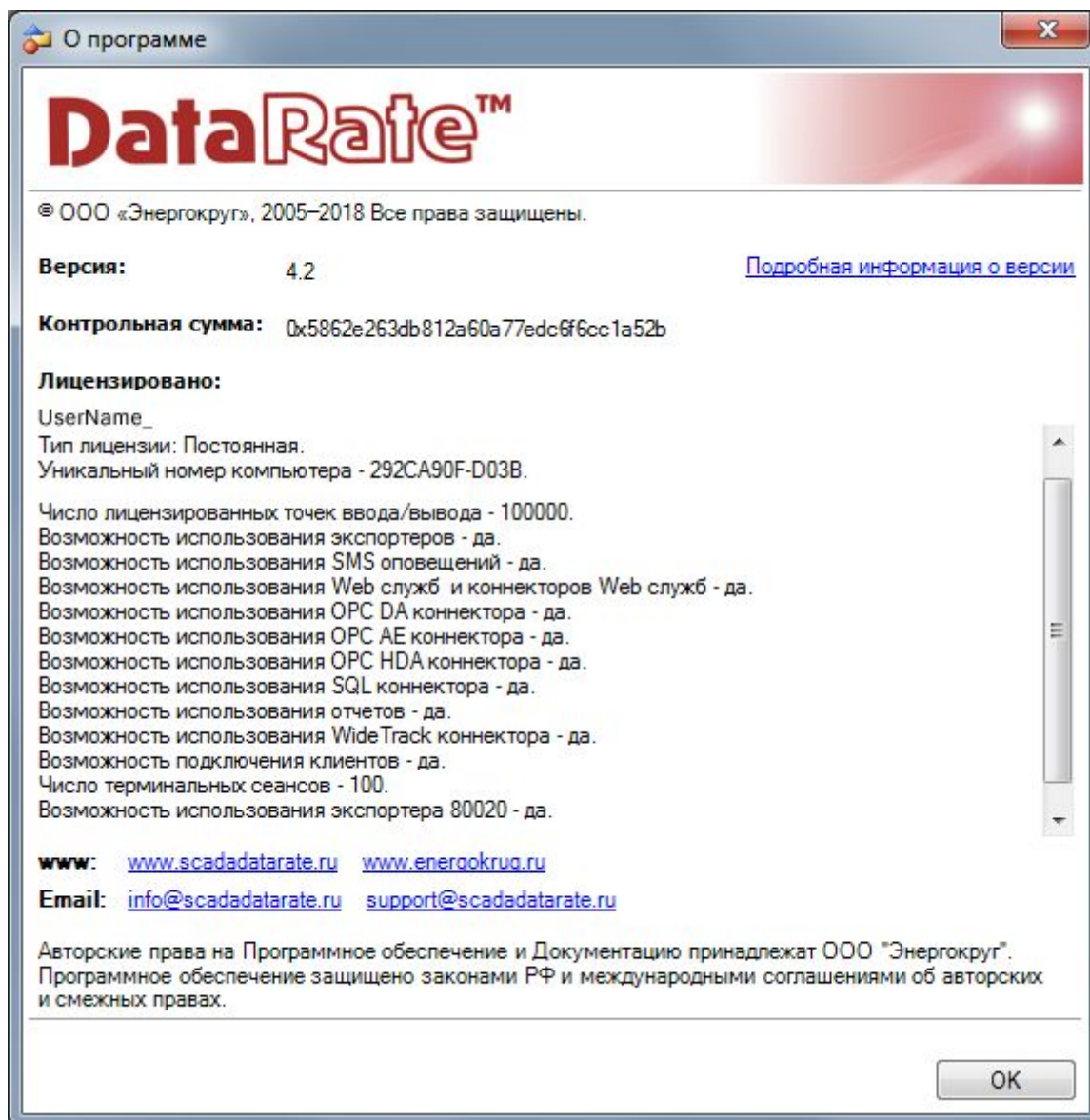
- 4 В ответ Вам придёт письмо с программным ключом. Разархивируйте файл приложения к письму и сохраните в надёжном месте – **это ваш персональный регистрационный ключ**.
- 5 Скопируйте файл с регистрационным ключом на компьютер, где установлен *DataRate*, и запустите его двойным щелчком мыши в проводнике. В ответ на вопрос – желаете ли вы добавить сведения в реестр? – нажмите кнопку **Да**.



- 6 Информация об успешной установке программного ключа отображается в окне **Редактор реестра**



- 7 Регистрация *DataRate* завершена. Подробную информацию о зарегистрированной версии можно увидеть в окне **О программе**.



В.5.2 Аппаратный ключ и защита

Аппаратный ключ – аппаратное средство (USB, LPT), предназначенное для защиты автоматизированной системы от нелегального использования и несанкционированного распространения.

Установка драйвера аппаратного ключа

В состав дистрибутива **DataRate** входит драйвер аппаратного ключа **Sentinel**. Процесс установки описан в разделе В.3 **Инсталляция и начало работы**.

ВНИМАНИЕ!!!

Если Вы используете автоматическую загрузку проекта, ознакомьтесь с разделом **Среда Исполнения/Запуск проекта**

Каскадирование аппаратных ключей

Функция «Каскадирование ключей» предназначена для обеспечения **DataRate** возможностью использовать несколько своих аппаратных ключей, как единый ключ.

В этом случае происходит слияние значений ячеек нескольких аппаратных ключей: если в ячейке одного ключа компонент разрешён, то он имеет приоритет над этим же,

но запрещённым компонентом в другом ключе. При сравнении численных параметров, выигрывает наибольшее значение параметра.

Например:

Аппаратный ключ №1	Аппаратный ключ №2	Результат
Компонент разрешен	Компонент запрещен	Компонент разрешен
100 точек ввода/вывода	1000 точек ввода/вывода	1000 точек ввода/вывода

При использовании каскадирования ключей на клиенте значение количества точек ввода/вывода рассчитывается по стандартному алгоритму каскадирования ключей, используемому на сервере *DataRate* **Среда Исполнения**.

Защита при работе с терминальным сервером

Терминальный сервер – сервер, предоставляющий клиентам вычислительные ресурсы (процессорное время, память, дисковое пространство) для решения задач. Терминальный сервер служит для удалённого обслуживания пользователя по протоколу RDP (Remote Desktop Protocol) с предоставлением рабочего стола. Количество терминальных сессий определяется конфигурацией программного/аппаратного ключа. Если количество запущенных процессов превышает разрешённое, то выдаётся сообщение об ошибке «Запуск *DataRate* **Среды Исполнения – Клиент** не возможен. Превышено разрешённое количество используемых сессий терминального сервера».

Внимание!

К терминальным сессиям относятся все RDP-соединения к серверу, на котором запущен сервис терминальных подключений, в том числе и соединения через обычный удаленный рабочий стол Windows.

Защита на клиенте

Ключ защиты для *DataRate* **Среда Исполнения** содержит разрешения на использование отдельных лицензируемых модулей и количественные параметры для отдельных модулей.

Для *DataRate* **Среда Исполнения – Клиент** учитывается только количество точек ввода/вывода.

Параметр на клиенте должен быть больше или равен соответствующему параметру ключа на сервере. Иначе сработает предупреждение системы защиты, пользователь увидит соответствующее окно с сообщением. При этом дальнейшая работа с сервером будет прекращена.

При использовании каскадирования ключей на клиенте значение количества точек ввода/вывода рассчитывается по стандартному алгоритму каскадирования ключей, используемому на сервере *DataRate* **Среда Исполнения**.

Пример использования ключей защиты:

Ключ <i>DataRate</i> Среда Исполнения	Ключ <i>DataRate</i> Среда Исполнения – Клиент	Результат
100 точек ввода/вывода	1000 точек ввода/вывода	Разрешено
1000 точек ввода/вывода	1000 точек ввода/вывода	Разрешено
1000 точек ввода/вывода	100 точек ввода/вывода	Запрещено

В.5.3 Работа в ознакомительном и ограниченном режимах

Работа в ознакомительном режиме

При отсутствии и программного и аппаратного ключей, **Среда исполнения DataRate** функционирует в ознакомительном режиме.

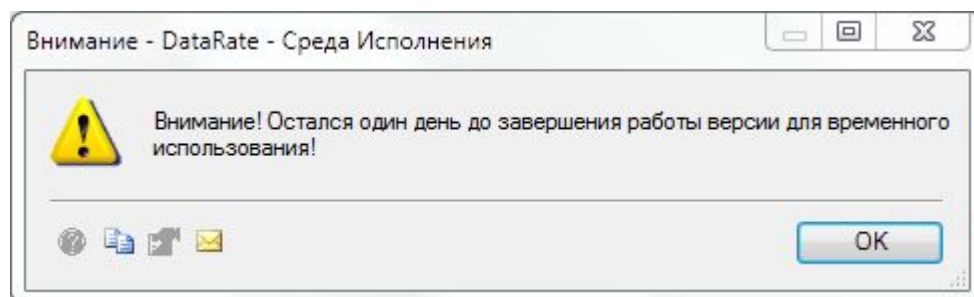
ВНИМАНИЕ!!!

В ознакомительном режиме период использования DataRate ограничен 30 днями со дня установки.

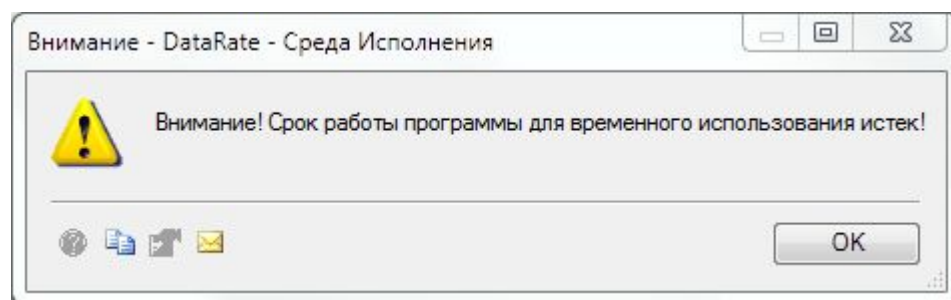
В ознакомительном режиме разрешается использование:

- Экспорта в файл
- Модуля отчетов
- OPC DA коннектора
- OPC HDA коннектора
- OPC AE коннектора
- WideTrack коннектора
- SQL коннектора
- SMS оповещений
- Клиент-серверной конфигурации – сетевая Среда Исполнения
- Среды Исполнения-Клиент с управлением
- Web-клиента с управлением
- Web-служб и коннекторов к Web-службам
- 10000 точек ввода/вывода
- 2 одновременно работающих с Web-контроль пользователей (сессий)
- 1 терминального сеанса Среды Исполнения
- 1 терминального сеанса Среды Исполнения-Клиент.

После завершения ознакомительного периода работа **DataRate** будет остановлена. За **1 сутки до остановки DataRate** откроет окно с предупреждением об оставшемся времени.



На следующий день будет открыто окно с предупреждением о завершении ознакомительного периода и в течение следующих **30 минут** приложение **DataRate** завершит свою работу.



При последующих запусках исполняемая часть *DataRate* будет функционировать в ограниченном режиме.

Работа в ограниченном режиме

В *DataRate* реализована возможность функционирования в ограниченном режиме.

При отсутствии программного и аппаратного ключей **ограниченный режим** (режим «30 точек») включается автоматически при завершении ознакомительного периода. В этом режиме разрешено использование:

- OPC DA коннектора
- OPC HDA коннектора
- Модуля отчетов
- Экспорта в файл
- 30 точек ввода/вывода.

В.6. Совместимость версий

Форматы файлов проекта *DataRate* (.krproj) совместимы следующим образом:

- Проект младшей версии может быть открыт в версиях старше. Например, проект версии **3.3** может быть открыт в версии **4.2**
- Проект старшей версии не может быть открыт в версиях младше. Например, проект версии **4.2** нельзя открыть в версии **4.0** и младше!

Имеются некоторые **исключения в совместимости** с версиями **1.x**:

- Рабочие столы не будут импортированы, их придется создать заново
- В связи с изменениями в системе трендирования (начиная с версий **2.x** отсутствует объект **Самописец**), тренды нужно будет настроить заново, как в проекте, так и в мнемосхемах
- Если в проекте используется протокол событий, то его тоже нужно будет настроить заново.

В.7. Смена операционной системы

В.7.1 Подготовка

Перед установкой или обновлением ОС сохраните следующие файлы настроек *DataRate* и графического проекта, запомните их расположение.

ВНИМАНИЕ!!!

Все пути к файлам, а также названия директорий указаны со значениями по умолчанию. Если Вы меняли эти параметры при установке или настройке проекта, то, пожалуйста, учитывайте их в дальнейшей работе.

Пути различаются в зависимости от операционной системы, поэтому приведены пути по умолчанию для старых и новых операционных систем.

В.7.1.1 Файлы, располагающиеся в папке с проектом

Файлы Графического проекта

- *.krproj – файл Графического проекта
- *.kitem – файл Элемента проекта
- *.events – файл Протокола событий
- *.backup – резервная копия файла.

Подробная информация о хранении истории Протокола событий содержится в руководстве пользователя «2 часть _DataRate_4.2_Объекты автоматизации техпроцессов» в разделе «События и тревоги (алармы)/Протокол событий/Настройки Протокола событий/Настройки хранения истории».

Файлы с данными трендов

Если Вы храните данные трендов в виде **Структурированного хранилища**, то сохраните файл ***.history**.

Если Вы используете тип **Файловое хранилище**, то сохраните целиком директорию, содержащую файлы с данными трендов.

Подробная информация о типах хранения данных трендов содержится в руководстве пользователя «2 часть _DataRate_4.2_Объекты автоматизации техпроцессов» в разделе «История процесса. Тренды/Модуль Менеджер трендов/Настройки менеджера трендов/Группы настроек истории/Редактирование группы настроек».

Файлы снимков данных отчётов

Сохраните целиком папку **Reports** с файлами снимков данных отчётов ***.xml**.

Подробная информация о хранении снимков данных отчётов содержится в руководстве пользователя «2 часть _DataRate_4.2_Объекты автоматизации техпроцессов» в разделе «Модуль отчётов/Менеджер отчётов/Хранение снимков данных» документации к программе.

Файлы настроек сортировок и фильтров

- **SortingFiltering.sfcfg** – файл настроек сортировок и фильтров.

Подробная информация об использовании сортировок и фильтров содержится в руководстве пользователя «1 часть _DataRate_4.2_Основы» в разделе «Объектная модель/Проект/Расширенный список колонок».

ВНИМАНИЕ!

Сохраняется в папку с проектом, если в настройках установлен флаг «Для текущего проекта».

Файлы настроек шаблонов имени

- **NameTemplates.tmpls** – файл настроек шаблонов имени.

Подробная информация об использовании шаблонов имени содержится в руководстве пользователя «1 часть _DataRate_4.2_Основы» в разделе «Объектная модель/Объекты/Формирование шаблона имени».

ВНИМАНИЕ!!!

Сохраняется в папку с проектом, если в настройках установлен флаг «Видим только в этом проекте».

Файлы шаблонов привязки

- ***.rbmpls** – файл шаблонов привязки

Подробная информация о шаблонах привязки тегов содержится в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Объектная модель/Объекты/Теги/Групповая привязка тегов».

ВНИМАНИЕ!!!

Сохраняется в папку с проектом; содержит шаблоны, сохраненные для проекта.

В.7.1.2 Файлы, доступные для всех пользователей

Файлы словарей Протокола событий

<Системный диск>:\ProgramData\Energokrug\DataRate 4.x\Dictionaryes\ru-RU\

<Системный диск>:\Documents and Settings\All Users\Application Data\Energokrug\DataRate 4.x\Dictionaryes\ru-RU\

- *.eclrdic – файл словаря категорий
- *.sdic – файл словаря серьезности
- *.edic – файл словаря событий.

Подробная информация о словарях Протокола событий содержится в руководстве пользователя «2 часть _DataRate_4.2_Объекты автоматизации техпроцессов» в подразделах раздела «События и тревоги (алармы)/Протокол событий/Настройки Протокола событий».

Файлы используемых общих и отраслевых библиотек

<Системный диск>:\ProgramData\Energokrug\DataRate 4.x\SharedLibraries\

<Системный диск>:\Documents and Settings\All Users\Application Data\Energokrug\DataRate 4.x\SharedLibraries\

- Системная.kritem – файл системной библиотеки *DataRate* 4.0
- Системная библиотека.kritem – файл системной библиотеки *DataRate* 4.2
- EnergyAccountingLib.kritem – файл библиотеки технологических объектов учёта энергоресурсов.
- и другие.

ВНИМАНИЕ!!!

Если Вы не вносили изменений в несистемные библиотеки, рекомендуем вместо копирования файлов библиотек установить их заново.

Файлы шаблонов скриптов

<Системный диск>:\ProgramData\Energokrug\DataRate 4.x\Templates\ru-RU\

<Системный диск>:\Documents and Settings\All Users\Application Data\Energokrug\DataRate 4.x\Templates\ru-RU\

- Krug.Tools.Scripting.dll.config – файл шаблонов скриптов.

Подробная информация о пользовательских шаблонах скриптов содержится в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Объектная модель/Объекты/Скрипты/Редактор скриптов/Настройки редактора скриптов/Настройки Шаблонов пользователей».

В.7.1.3 Файлы, доступные только конкретному пользователю

Файлы настроек

<Системный диск>:\Users\<User>\AppData\Local\Energokrug\DataRate 4.x\Settings\
<Системный диск>:\Documents and Settings\<Users>\Local Settings\
Application Data\Energokrug\DataRate 4.x\Settings\

Сохраните целиком папку **Settings**, содержащую файлы настроек компонентов Среда Разработки, Среда Исполнения, Среда Исполнения - Клиент, Web-контроль.

Файлы шаблонов привязки

<Системный диск>:\Users\<User>\AppData\Local\Energokrug\DataRate 4.x\Settings\
<Системный диск>:\Documents and Settings\<Users>\Local Settings\
Application Data\Energokrug\DataRate 4.x\Settings\

- *.rbmpls – файл шаблонов привязки.

Подробная информация о шаблонах привязки тегов содержится в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Объектная модель/Объекты/Теги/Групповая привязка тегов».

ВНИМАНИЕ!!!

Сохраняется в указанную папку; содержит шаблоны, сохраненные для Среда Разработки.

Файлы настроек сортировок и фильтров

<Системный диск>:\Users\<User>\AppData\Local\Energokrug\DataRate 4.x\Settings\
<Системный диск>:\Documents and Settings\<Users>\Local Settings\
Application Data\Energokrug\DataRate 4.x\Settings\

- SortingFiltering.sfcfg - файл настроек сортировок и фильтров.

Подробная информация об использовании сортировок и фильтров содержится в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Объектная модель/Проект/Расширенный список колонок».

ВНИМАНИЕ!!!

Сохраняется по указанному пути, если в настройках не установлен флаг Для текущего проекта».

Файлы настроек шаблонов имени

<Системный диск>:\Users\<User>\AppData\Local\Energokrug\DataRate 4.x\Settings\
<Системный диск>:\Documents and Settings\<Users>\Local Settings\
Application Data\Energokrug\DataRate 4.x\Settings\

- NameTemplates.tmpls – файл настроек шаблонов имени.

Подробная информация об использовании шаблонов имени содержится в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Объектная модель/Объекты/Формирование шаблона имени».

ВНИМАНИЕ!!!

Сохраняется по указанному пути, если в настройках не установлен флаг Видим только в этом проекте.

Файлы пользовательской палитры

<Системный диск>:\Users\<User>\AppData\Local\Energokrug\DataRate 4.x\Palettes\
<Системный диск>:\Documents and Settings\<User>\Local Settings\
Application Data\Energokrug\DataRate 4.x\Palettes\

- *.palette – файл пользовательской палитры.

Подробная информация об использовании пользовательской палитры содержится в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Динамическая визуализация данных/Графическая подсистема/Графические примитивы/Свойства графических примитивов/Цвет».

В.7.1.4 Дополнительные файлы

Файлы конфигурации OPC-серверов

<Папка с конфигурацией OPC-серверов>

ВНИМАНИЕ!!!

Если Вы используете OPC-сервера в своём проекте, то сохраните файлы конфигурации OPC-серверов.

В.7.2 Установка или обновление Windows

Обновление версии Windows, возможно, не повлияет на работоспособность SCADA *DataRate*. Но в связи с тем, что полностью проверить корректную работу SCADA за короткое время невозможно, мы рекомендуем Вам переустановить *DataRate*.

ВНИМАНИЕ!!!

Если производилась полная установка Windows, то необходимо заново установить SCADA *DataRate*. В этом случае рекомендуем Вам предварительно установить обновления для ОС.

В.7.3 Установка SCADA *DataRate*

Процесс установки SCADA *DataRate* описан в руководстве пользователя «1 часть_DataRate_4.2_Основы» в разделе «Введение/Инсталляция и начало работы».

После успешной инсталляции основной программы запустите установку сервисных пакетов обновлений (service pack) и оперативных обновлений (hotfix). Порядок их установки описан в прилагаемых к ним файлах [readme.txt](#).

Если необходимо, установите отраслевые библиотеки.

Разместите сохранённые перед установкой/обновлением Windows файлы в соответствующих директориях.

1 ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ

Интегрированная среда разработки (ИСР) предоставляет рабочее пространство и необходимые средства для быстрого создания, конфигурирования и отладки проекта системы автоматизации.

Интегрированная среда разработки позволяет создавать проекты, исполняемые **Средой Исполнения DataRate**.

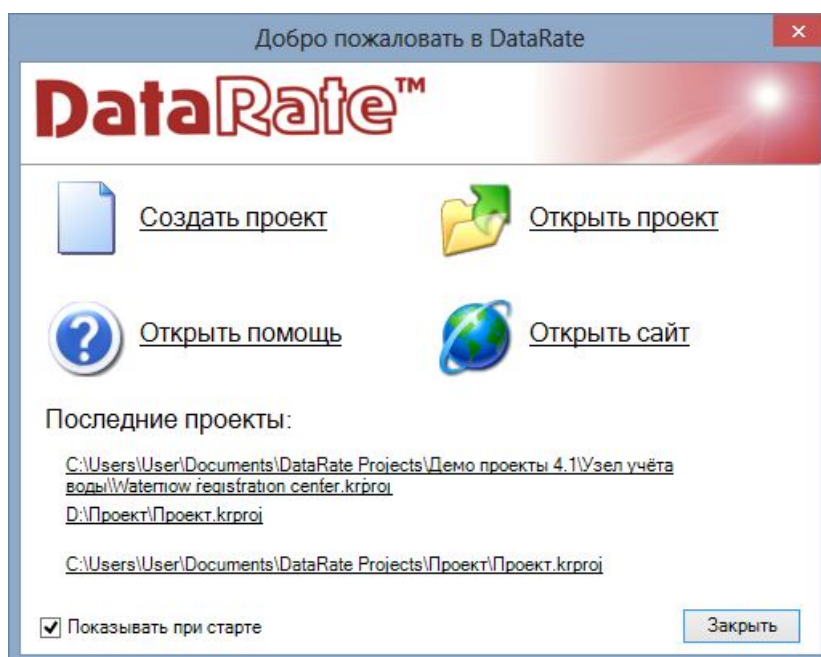
Основными возможностями ИСР являются:

- Разработка пользовательского интерфейса (мнемосхемы, отчёты, тренды)
- Разработка логики работы проекта автоматизированной системы (алгоритмы управления датчиками, устройствами, исполнительными механизмами и другие алгоритмы)
- Привязка внешних источников данных (ОПС, БД, информационные системы)
- Разработка шаблонов технологических объектов.

1.1 Окно Добро пожаловать

Окно **Добро пожаловать в DataRate** появляется при запуске **DataRate** и содержит список операций для быстрого доступа к функциям:

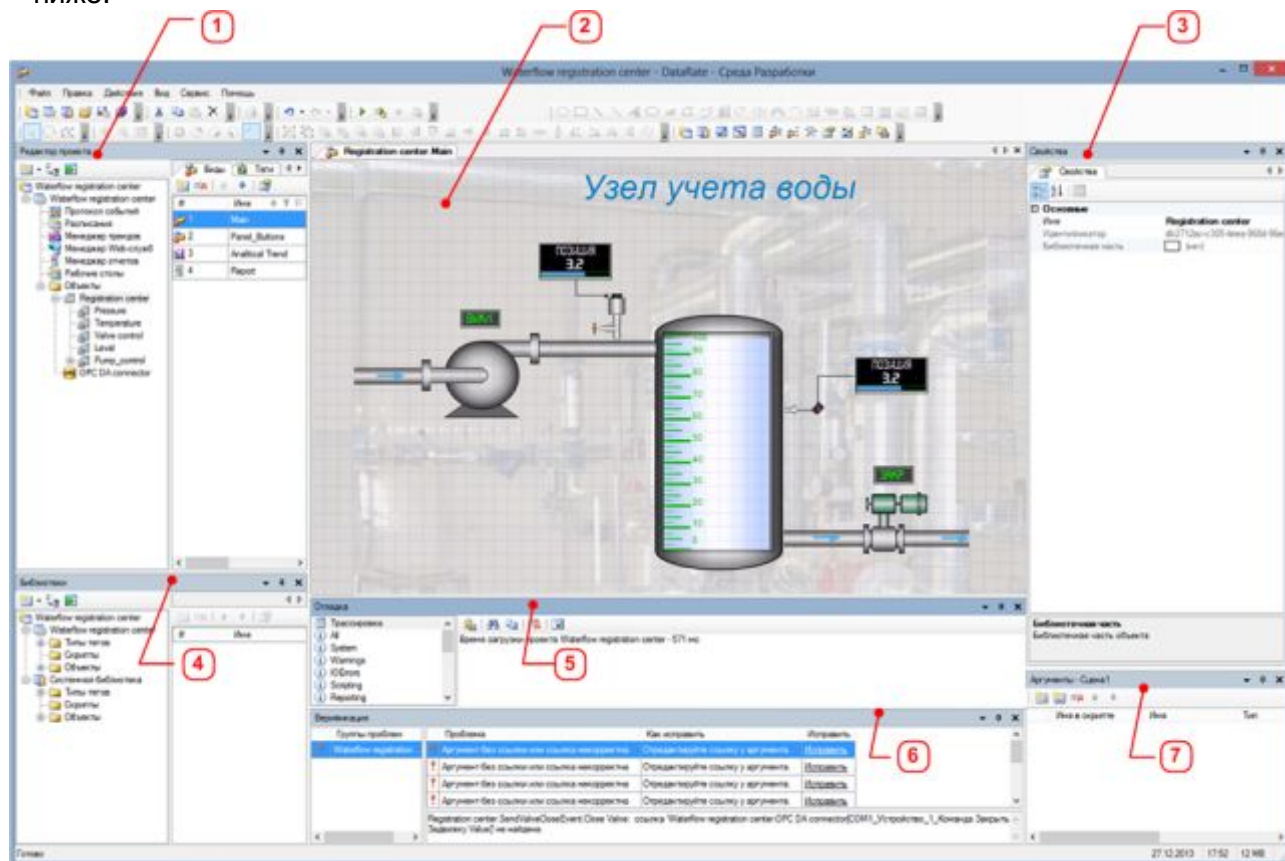
- **Создать проект** – откроется окно создания нового проекта
- **Открыть проект** – откроется окно для загрузки сохраненного ранее проекта
- **Открыть помощь** – откроется справка
- **Открыть сайт** – откроется сайт фирмы производителя (необходимо подключение к Интернет)
- **Последние проекты** – позволяет быстро открыть проект из отображаемого списка проектов.



- **Показывать при старте** – если установлена галочка, то при каждом старте **DataRate** будет показываться данное окно. Данная опция также доступна в окне **Параметры** на странице свойств Среды разработки.

1.2 Рабочее пространство

Общий вид рабочего пространства Интегрированной среды разработки **DataRate** приведен ниже.



Главное окно ИСР содержит:

- **Окно редактора проекта (1).** Показывает дерево проекта и позволяет редактировать его элементы. Для сохранения объекта в библиотеке достаточно перетащить объект из дерева проекта в окно библиотеки
- **Область редактирования и отладки (2).** Находится в центре главного окна ИСР. Каждая закладка окна содержит открытый для редактирования и отладки элемент проекта (например: мнемосхема, атрибуты тега, скрипт). Чтобы открыть элемент для редактирования и отладки, выполните двойной щелчок на элементе или команду **Редактировать** в контекстном меню элемента
- **Окно свойств (3).** Отображает свойства выделенного элемента
- **Окно библиотек (4).** Предназначено для добавления/удаления библиотек и их редактирования. Объекты библиотеки (шаблоны) используются для создания объектов проекта. Для этого достаточно перетащить объекты из библиотеки в окно проекта
- **Окно аргументов (7).** Отображает аргументы скрипта или графического элемента.

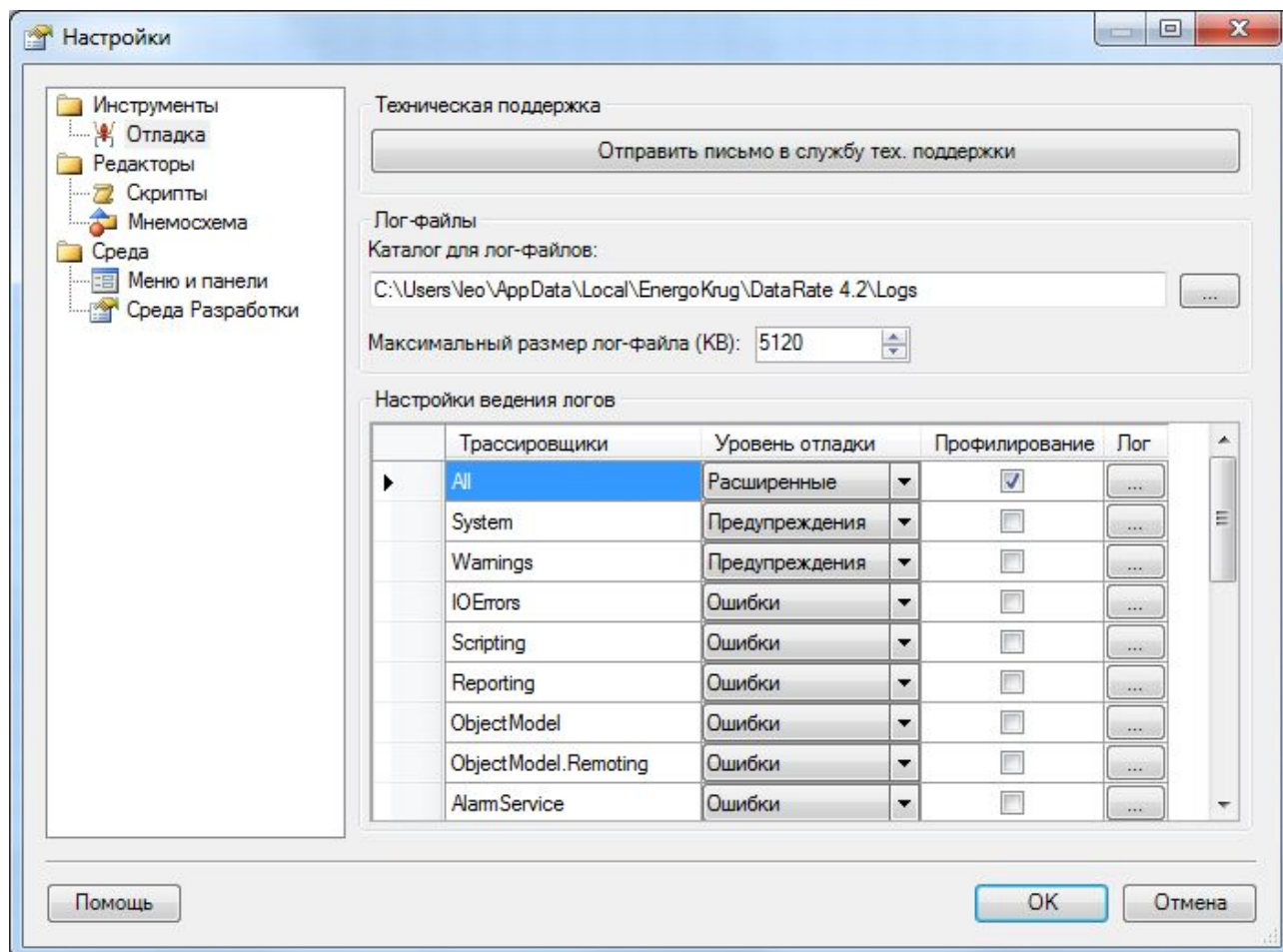
В главном окне ИСР могут быть открыты и другие окна, например:


- **Окно отладки (5).** Предназначено для отображения сообщений, попадающих в лог-файлы во время работы системы
- **Окно верификации (6).** Отображает информацию обо всех найденных ошибках в процессе верификации проекта

1.2.1 Окно Настройки

Окно **Настройки** предназначено для настройки параметров:

- Лог-файлов
- Редактора скриптов
- Отображения мнемосхемы
- Меню и панелей инструментов
- Среды разработки (каталога сохранения проектов).



Для вызова окна **Настройки** нажмите кнопку  на панели инструментов или комбинацию клавиш **Ctrl+Alt+Shift+S**.

Также окно **Настройки** можно вызвать из меню **Сервис/Настройки**.


1.2.1.1 Лог-файлы

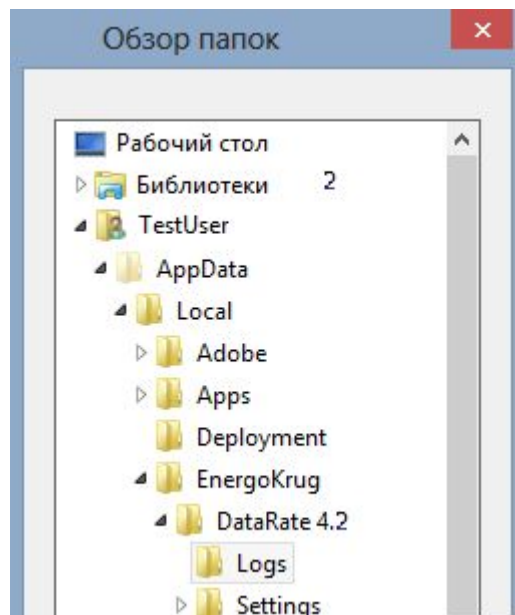
Настройки **Инструменты/Отладка** используются для задания параметров лог-файлов, содержащих сообщения об ошибках и событиях, связанных с работой приложения, созданного в **DataRate**, а также для отправки писем с информацией о проблемах в службу технической поддержки.

Максимальный размер лог-файла установлен **по умолчанию** в **5120 Кб**. Максимальное значение размера лог-файла – **51200 Кб**, минимальное – **10 Кб**.

По умолчанию лог-файлы хранятся в следующей папке:

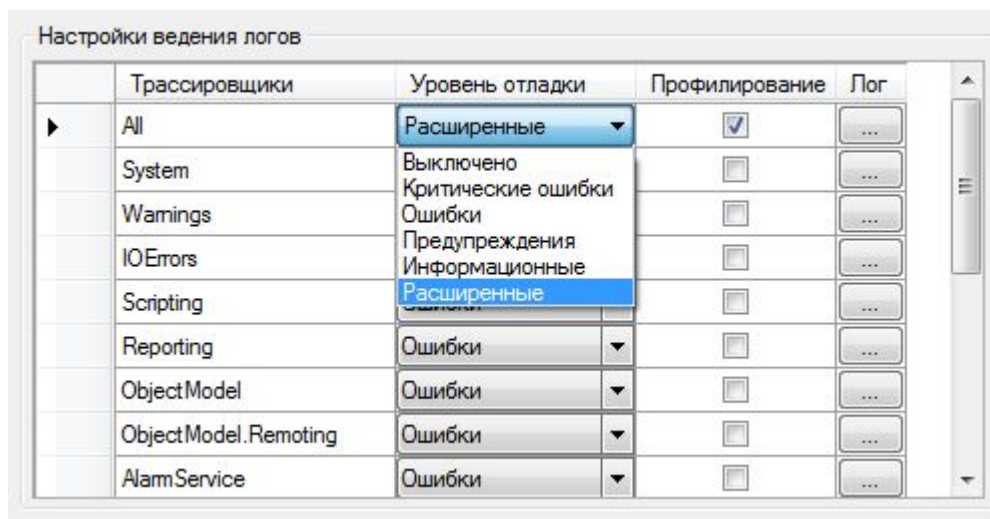
<Системный диск>:\Users\
<Имя пользователя>\AppData\Local\
Energokrug\DataRate 4.2\Logs\

Чтобы изменить папку, в которую будут сохраняться лог-файлы, следует в окне **Настройки** в строке **Каталог для лог-файлов** нажать на кнопку  и в открывшемся окне **Обзор папок** выбрать необходимую папку.



В нижней части окна **Настройки** расположен список **трассировщиков** – журналов регистрации сообщений, определенных в **DataRate**. Каждому трассировщику можно установить **уровень отладки**, который определяет тип сообщений, записываемых в соответствующий трассировщику лог-файл. Например, уровень **Выключено** означает, что никакие сообщения в лог-файл не попадут.

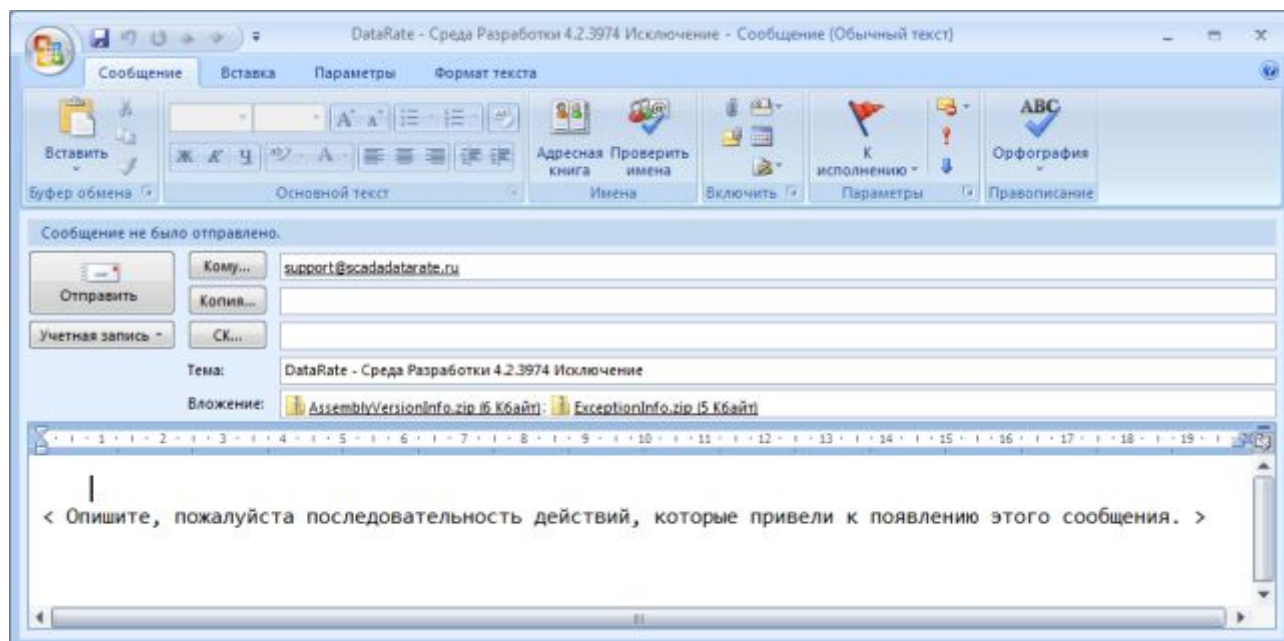
Для просмотра лог-файла соответствующего трассировщика нужно нажать на кнопку в столбце **Лог**.



Как отправить письмо в службу технической поддержки

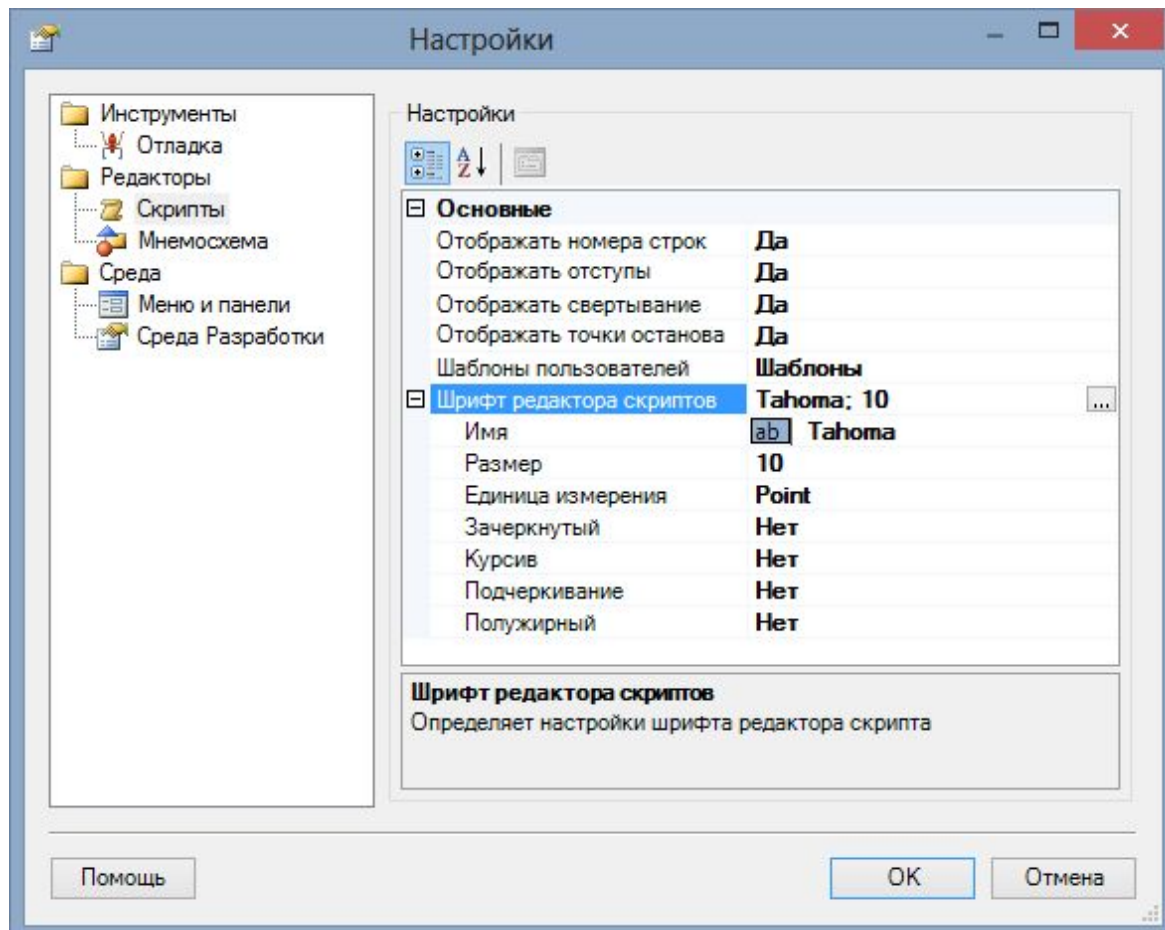
Для отправки письма в окне **Настройки** нажмите на кнопку **Отправить письмо в службу тех. поддержки**.

Появится окно **Outlook**, которое содержит вложенные лог-файлы (сообщения, зарегистрированные в журналах **DataRate** в процессе работы), сведения об установленных на Вашем компьютере программных средствах, а также предложение описать последовательность действий, которые привели к возникновению исключительной ситуации.



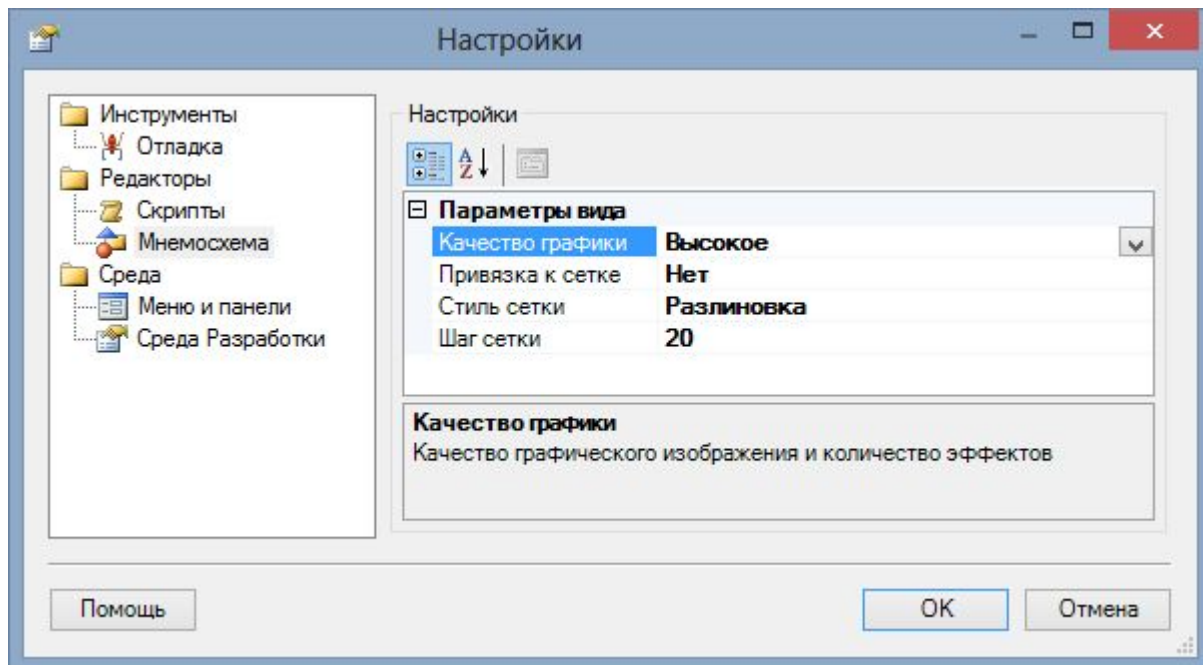
1.2.1.2 Редактор скриптов

Настройки **Редакторы/Скрипты** используются для настройки параметров, используемых в редакторе скриптов.



1.2.1.3 Редактор мнемосхем

Настройки **Редакторы/Мнемосхема** используются для настройки отображения мнемосхемы.



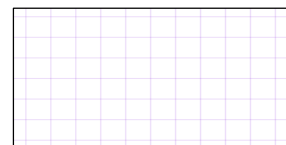
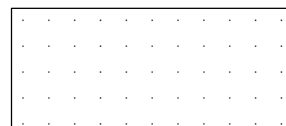
Параметры мнемосхемы:

- **Качество графики** – качество отображения графических элементов (определяет вид объектов *DataRate*): **Высокое**, **Среднее**, **Низкое**
- **Привязка к сетке:**
 - **Да** – графический примитив перемещается строго по узлам сетки
 - **Нет** – графический примитив перемещается произвольно.
Значение **Нет** устанавливается по умолчанию.

Данная настройка является общей для **Среды Разработки** и поэтому распространяется на все открываемые мнемосхемы. Если пользователю необходимо во время редактирования мнемосхемы временно изменить эту настройку, то это возможно сделать через окно **Свойства**. Изменения свойств самой мнемосхемы не сохраняются.

Если необходимо, чтобы всегда при открытии мнемосхем свойство **Привязка к сетке** было выставлено в **Да**, следует изменить эту настройку через окно **Настройки Среды Разработки DataRate**.

- **Стиль сетки:**
 - **Нет** – сетка отображаться не будет
 - **Точечная сетка** – сетка будет представлена в виде точек в узлах сетки
 - **Разлиновка** – сетка в виде линий
- **Шаг сетки** – размер ячеек сетки: от **3** до **200** включительно.

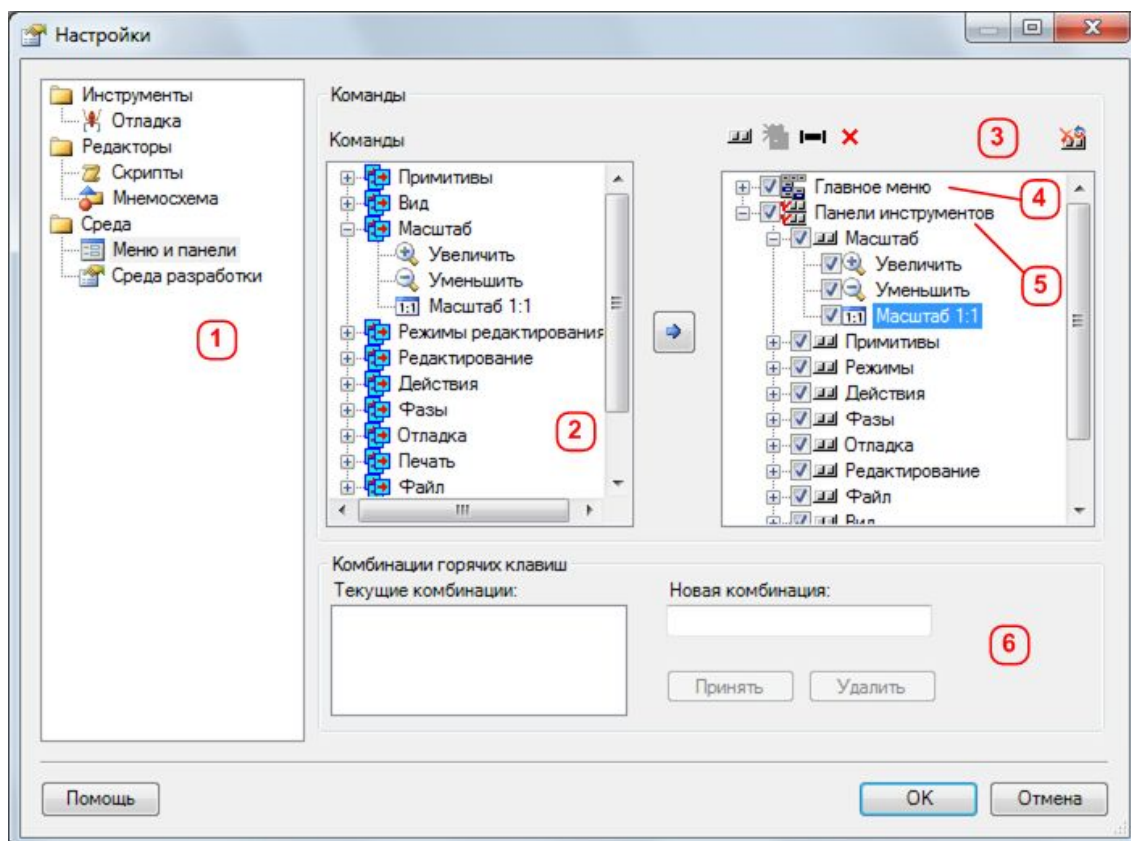


1.2.1.4 Меню и панели инструментов

Для настройки **Меню** и **Панелей инструментов** в окне **Настройки** выберите в дереве настроек (1) элемент **Среда/Меню и панели**.

ВНИМАНИЕ!!!

Вы можете не только настроить состав меню и панелей инструментов, но и дать свои наименования и назначить «горячие клавиши» для их элементов.




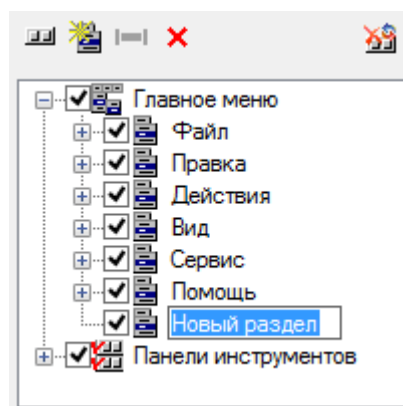
Окно настроек меню и панелей инструментов содержит:


- Дерево настроек (1)
- Дерево команд (2), которые можно добавить в меню и панели инструментов
- Кнопки (3) формирования меню и панелей инструментов:
 - добавить панель инструментов
 - добавить раздел меню
 - добавить разделитель
 - удалить элемент меню или панели инструментов
 - сбросить настройки меню и панелей инструментов в состояние по умолчанию
- Структура создаваемых меню и панелей инструментов (дерево 4, 5)
- Группа **Комбинации горячих клавиш** (6) .

Добавление подменю

Для добавления Подменю в Главное меню (или подменю) следует:

- 1 Выделить в дереве (4) элемент Главное меню (или подменю)
- 2 Нажать на кнопку  (3)
- 3 Для редактирования имени выделить добавленный элемент и нажать F2.

**Добавление команды в меню**


- 1 Выделите меню (4), в которое добавляется команда
- 2 Выделите в дереве команд (2) команду, которая добавляется в меню
- 3 Нажмите на кнопку 

Используйте операцию Drag&Drop для редактирования меню:


- Можно перетащить отдельную команду или группу команд из дерева команд (2) в меню (4)
- Можно изменить порядок следования команд в меню (4).

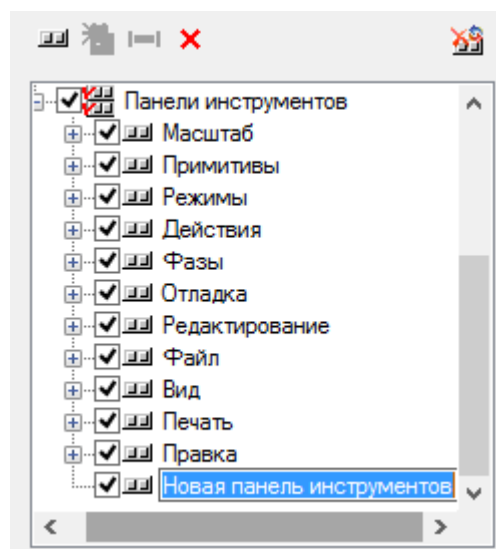
Добавление панели инструментов

Для добавления панели инструментов следует:

- 1 Выделить в дереве (5) элемент Панели инструментов
- 2 Нажать на кнопку  на панели инструментов (3)
- 3 Для редактирования имени выделить добавленный элемент и нажать F2.


Добавление команды на панель инструментов

- 1 Выделите панель инструментов (5), на которую добавляется команда
- 2 Выделите в дереве (2) команду, которая добавляется на панель
- 3 Нажмите на кнопку 




Используйте операцию Drag&Drop для редактирования панели инструментов

Добавление разделителя

- 1 В меню (4) или на панели инструментов (5) выделите команду, за которой нужно добавить разделитель
- 2 Нажмите на кнопку  (3)

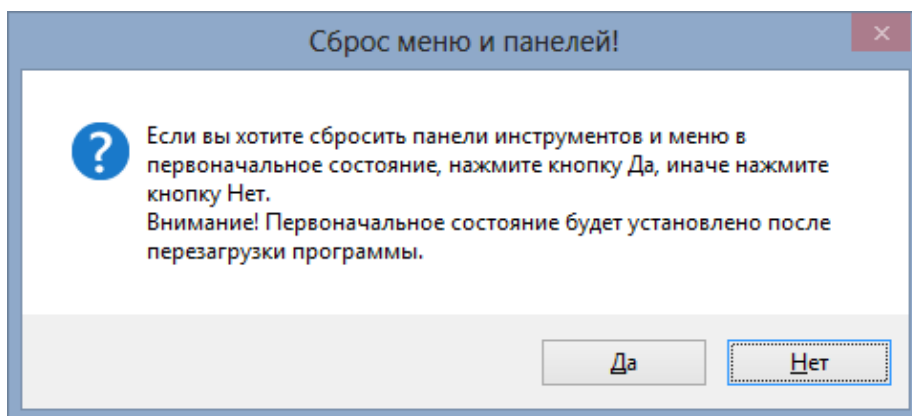
Для изменения положения разделителя используйте операцию Drag&Drop.

Удаление элемента меню или панели инструментов

- 1 Выделите удаляемый элемент
- 2 Нажмите на кнопку .

Сброс меню и панелей инструментов в состояние по умолчанию

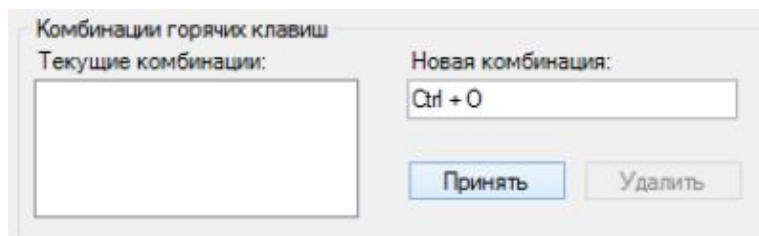
- 1 Нажмите на кнопку 
- 2 Нажмите кнопку **Да** в появившемся диалоговом окне для сброса



- 3 Перезапустите программу.

Добавление комбинации горячих клавиш

- 1 Выделите команду (в дереве **2, 4** или **5**)
- 2 Введите в поле **Новая комбинация** (**6**) комбинацию клавиш, которая будет соответствовать выделенной команде и нажмите на кнопку **Принять**.




Удаление комбинации горячих клавиш

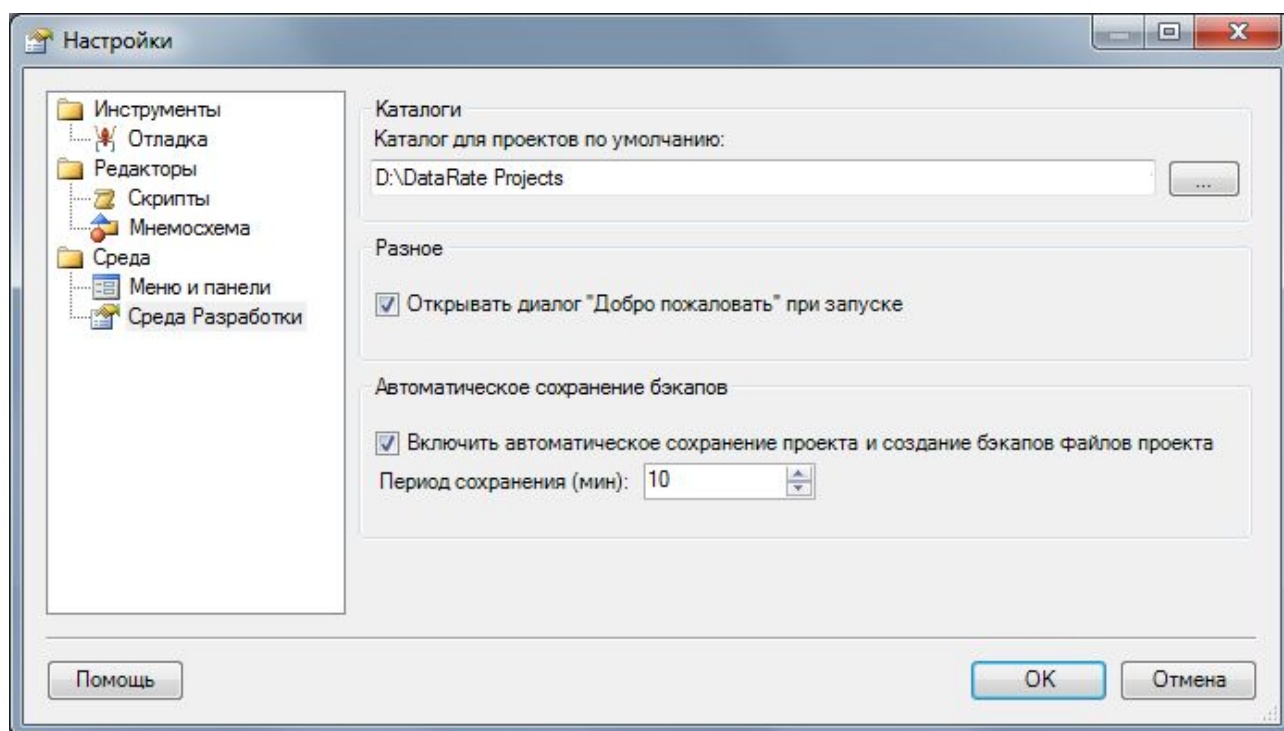
- 1 Выделите команду (в дереве **2, 4** или **5**)
- 2 Выделите в поле **Текущие комбинации** (**6**) удаляемую комбинацию клавиш и нажмите на кнопку **Удалить**.

1.2.1.5 Среда разработки

Настройки Среда/Среда разработки используются для настройки каталога проектов среды разработки **DataRate**.

Управляющие элементы:

- **Каталог для проектов по умолчанию** – каталог, в котором по умолчанию будут сохраняться проекты. Чтобы выбрать каталог, нужно нажать на кнопку  и в появившемся диалоговом окне выбрать необходимую папку.
- **Открывать диалог "Добро пожаловать"** – если галочка поставлена, то окно приветствия будет появляться при запуске **DataRate**.
- **Автоматическое сохранение бэкапов** – установите признак автоматического сохранения и укажите период сохранения проектов (**10 - 120 мин.**)



1.2.2 Окно Аргументы

Аргумент – это ссылка на данные, значения которых используются в объекте.

Окно **Аргументы** предназначено для управления аргументами объекта.

Чтобы показать окно **Аргументы**, нажмите на кнопку  панели инструментов.

Аргументы - Sensor Value Text				
	Имя в скрипте	Имя	Тип	Привязка
✓	Quality	Quality	String	Waterflow registration center.Registration center.Temperature[Value.Quality]
✓	Value	Current Value	Double	Waterflow registration center.Registration center.Temperature[Value.Value]

Когда пользователь открывает мнемосхему, в окне аргументов отображается список аргументов всех примитивов мнемосхемы. Добавлять и удалять аргументы в этом случае нельзя.



В режиме просмотра аргументов мнемосхемы доступна сортировка по столбцам списка аргументов.

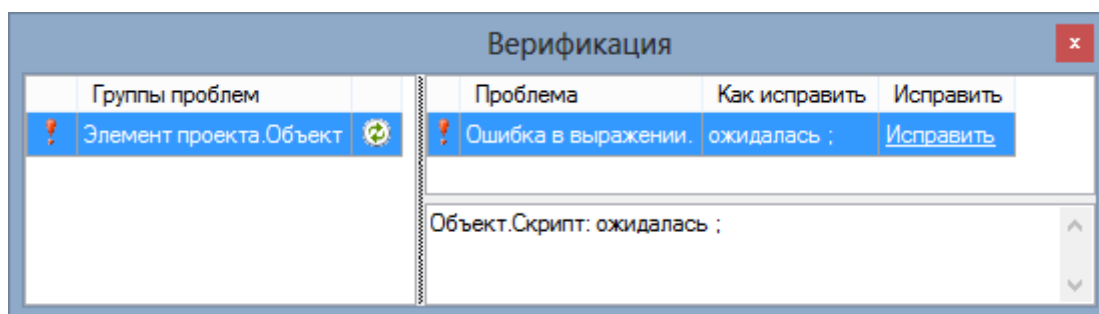
Аргументы - Sensor					
	Имя в скрипте	Имя	Тип	Привязка	Родитель
	Name	Name	Object	NA	Sensor Name Text
✓	Quality	Quality	String	Waterflow registration center.Registration center.Temperature[Value.Quality]	Sensor Value Text
✓	Value	Current Value	Double	Waterflow registration center.Registration center.Temperature[Value.Value]	Sensor Value Text

1.2.3 Окно Верификация

Для того чтобы выявить ошибки в проекте, не запуская его на исполнение, необходимо выполнить **верификацию** проекта.

Для того чтобы посмотреть результаты верификации следует:

- 1 Нажать на кнопку  панели инструментов или в главном меню **Вид**. В рабочей области **DataRate** появится вкладка  Верификация
- 2 Щелчком мыши по вкладке открыть окно **Верификация**



Окно **Верификация** отображает информацию обо всех найденных ошибках:

- **Проблема** – название проблемы
- **Как исправить** – возможное решение проблемы
- **Исправить** – ссылка на место ошибки
- Описание ошибки приводится в нижней части окна.

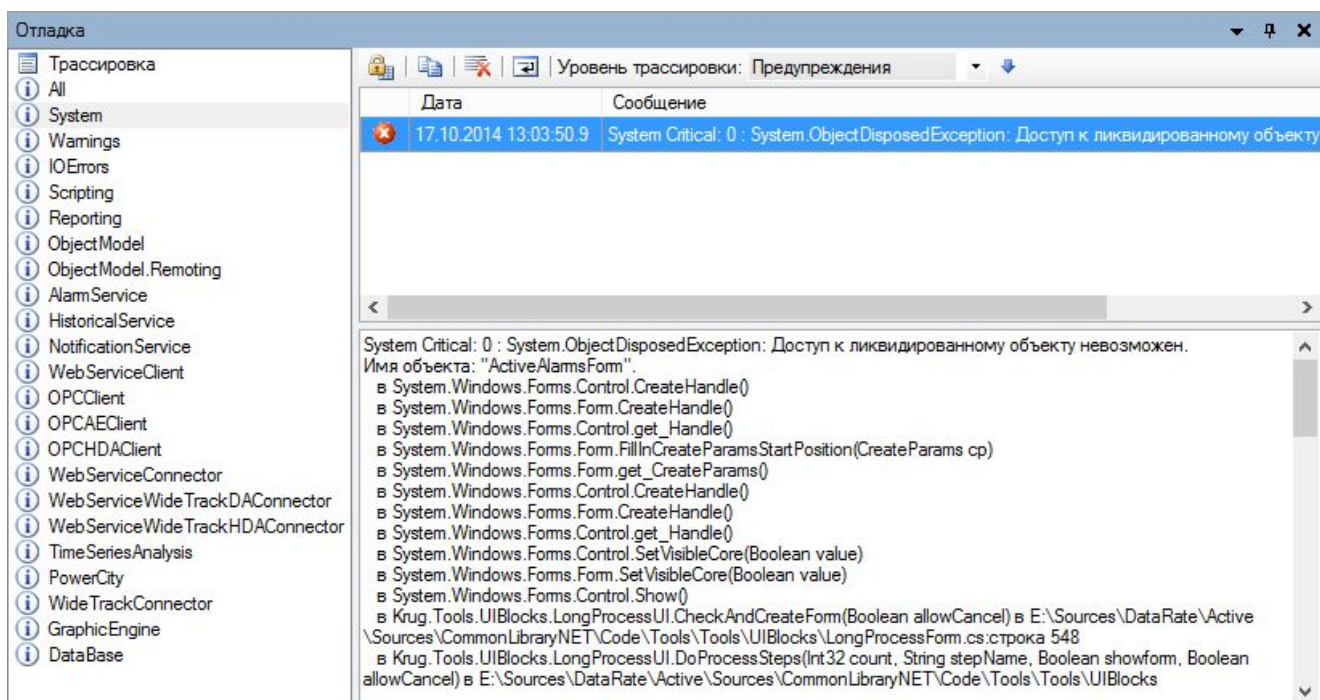
1.2.4 Окно Отладка

Окно **Отладка** предназначено для отображения сообщений, попадающих в лог-файлы во время работы системы.






Чтобы открыть окно отладки, необходимо нажать на кнопку .

В рабочей области **DataRate** появится вкладка  Отладка.

В левой части окна расположен список журналов регистрации сообщений. При выборе одного из журналов в правой части окна отображаются сообщения из соответствующего лог-файла.

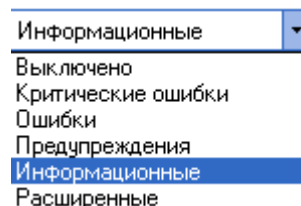


Панель инструментов

-  – **Разрешить/запретить обновление.** Разрешает или запрещает прием сообщений.
-  – **Копировать выделенный текст в буфер обмена**
-  – **Очистить окно с сообщениями**
-  – **Включить/выключить перенос строк.** Если перенос строк включен, то переносы будут расставлены так, чтобы текст был отформатирован по ширине окна
-  – **Включить/выключить профилирование.** Включение/отключение профилирования также будет отражено и в Настройках

Уровень трассировки. Устанавливает уровень сообщений, которые будут попадать в лог-файл. Уровни сообщений идентичны тем, что устанавливаются в настройках **Сервис/Параметры/Отладка**:

Выключено – означает, что сообщения приниматься не будут.



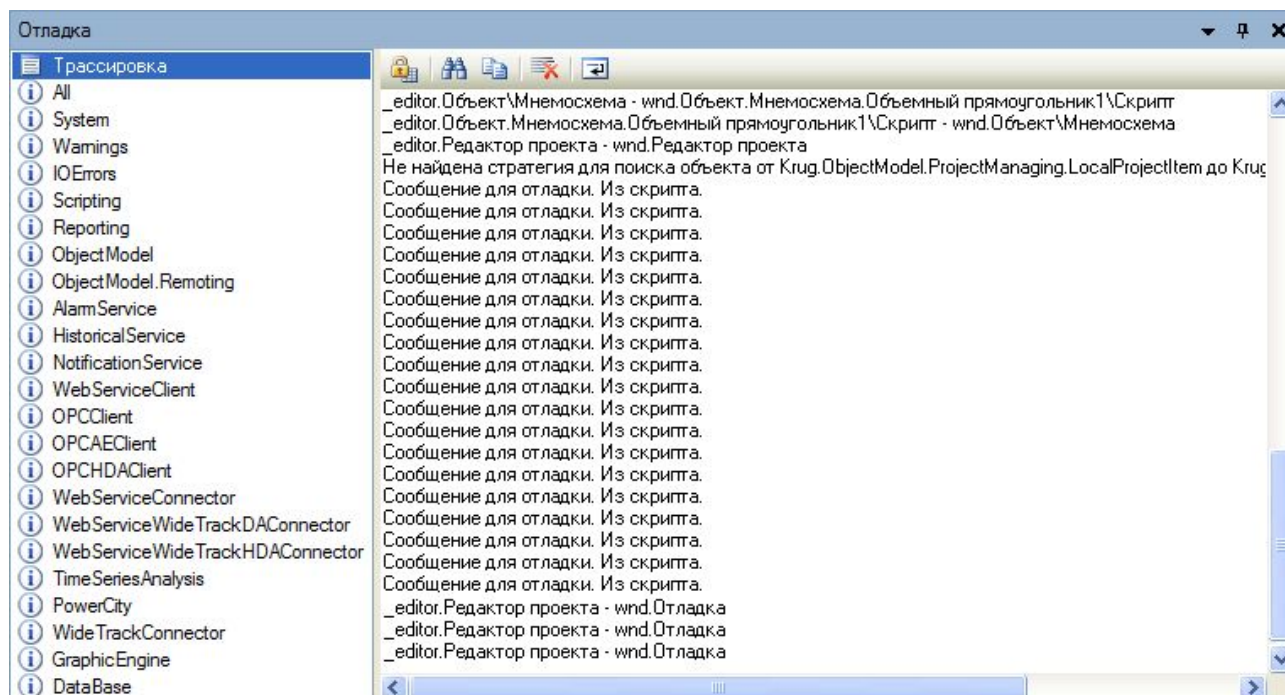
Трассировка

Трассировка – это журнал регистрации отладочных сообщений, формируемых с помощью функции ***Trace.WriteLine()***:

Например, в скрипт, который запускается при щелчке левой кнопкой мыши по графическому примитиву, был добавлен следующий код:

Trace.WriteLine("Сообщение для отладки. Из скрипта.");

После этого каждый щелчок мыши по примитиву будет зарегистрирован в журнале **Трассировка** в виде сообщения ***"Сообщение для отладки. Из скрипта."***



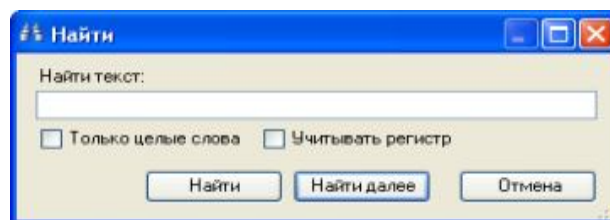
Панель инструментов



– **Разрешить/запретить обновление.**
Разрешает или запрещает прием сообщений



– **Поиск.** Вызывает окно **Найти**



– **Копировать выделенный текст в буфер обмена.**



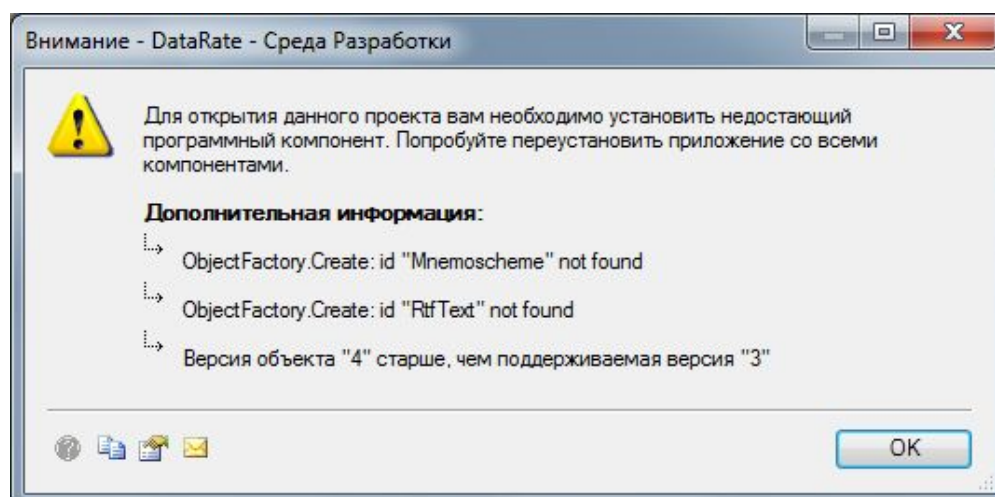
– **Очистить окно с сообщениями.**





– **Включить/выключить перенос строк.** Если перенос строк включен, то переносы будут расставлены так, чтобы текст был отформатирован по ширине окна.

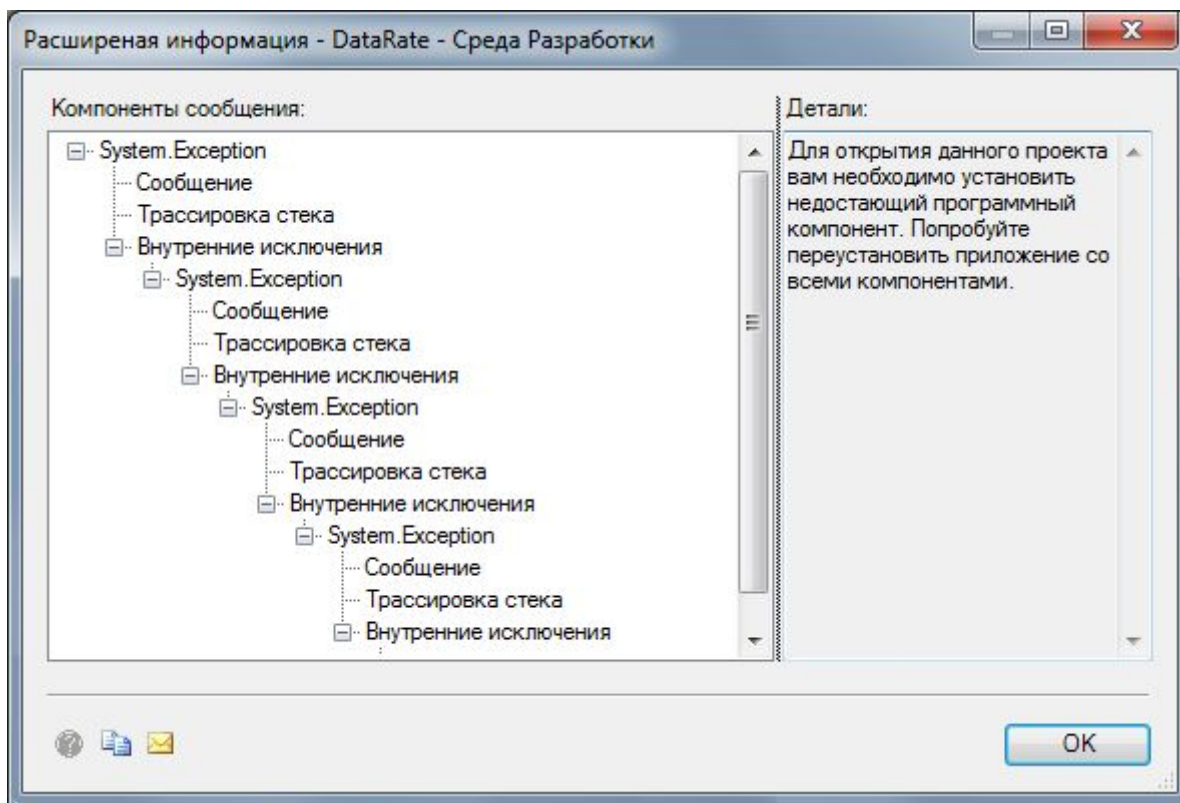
1.2.5 Окна вывода сообщений


Окно **Внимание** служит для вывода сообщений для Пользователя.



Элементы управления:

-  – поместить информацию о возникшей ситуации в буфер обмена
-  – отобразить расширенную информацию. При щелчке мышью по этой кнопке открывается окно **Расширенная информация**



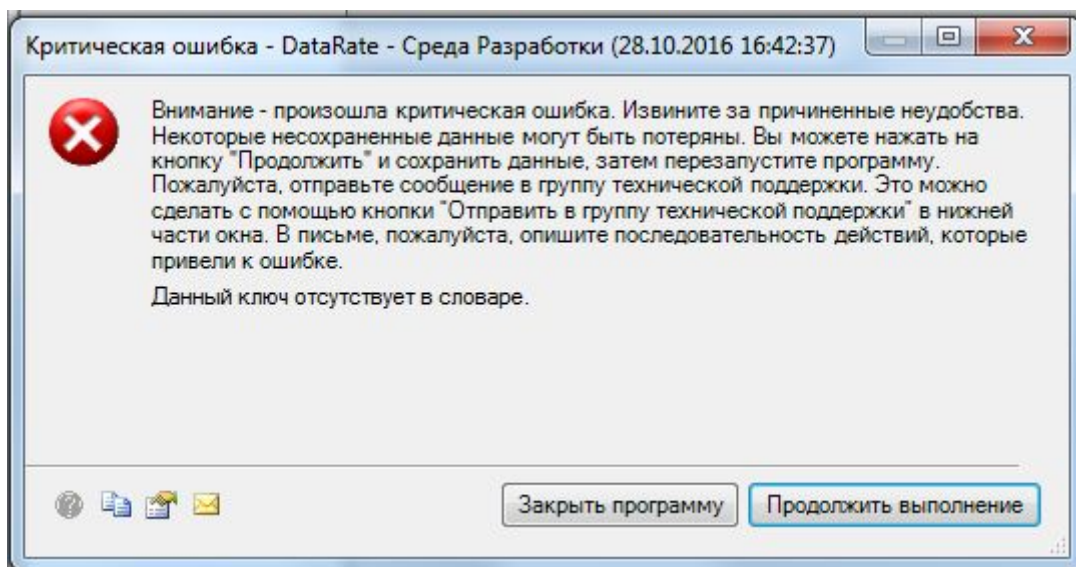
-  – отправить сообщение в группу технической поддержки. При щелчке мышью по этой кнопке будет сформировано письмо для отправки в группу технической поддержки. В письме следует описать последовательность действий, которые привели к возникновению данной ситуации и отправить письмо.

ВНИМАНИЕ!!!


Чтобы письмо было автоматически сформировано, нужно установить на компьютере почтовую программу (например, Outlook Express). Если почтовая программа не установлена, то система предложит отправить лог-файлы вручную (например, перенеся их на компьютер, где установлена почтовая программа).

Сообщение о критической ошибке

Окно **Критическая ошибка** появляется при возникновении в программе исключительной ситуации и содержит описание ошибки.



Выбрать вариант продолжения работы после ошибки можно с помощью кнопок:

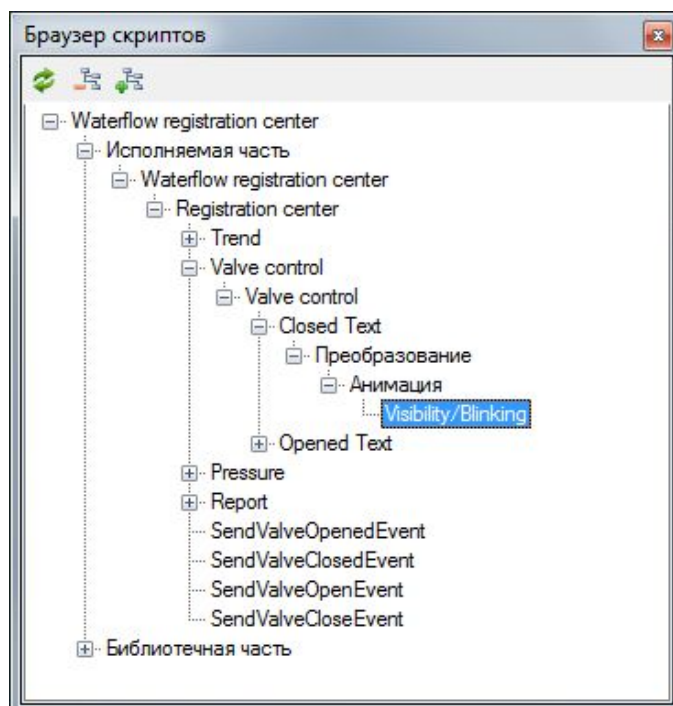
- **Закрыть программу** – равносильно нажатию на кнопку , в этом случае программа будет закрыта, а все несохраненные данные будут утеряны
- **Продолжить выполнение** – программа попытается восстановить работу и продолжить выполнение.

1.2.6 Окно браузера скриптов

Скрипт – небольшая программа на языке **C#**.

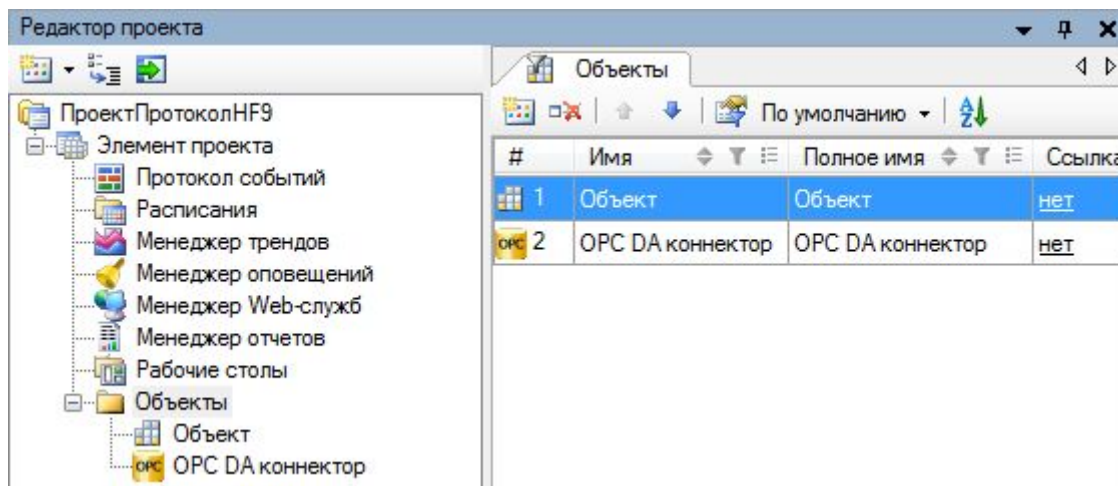
Используя скрипт, можно реализовать необходимые алгоритмы обработки данных.

Окно браузера скриптов предназначено для быстрой навигации по скриптам в проекте.



1.2.7 Окно выбора элементов

Окно выбора элементов – служит для выбора элементов. В зависимости от конкретной задачи могут отображаться только элементы определенного типа. Например, на рисунке ниже отображаются только объекты и **SQL коннекторы** с их группами данных.



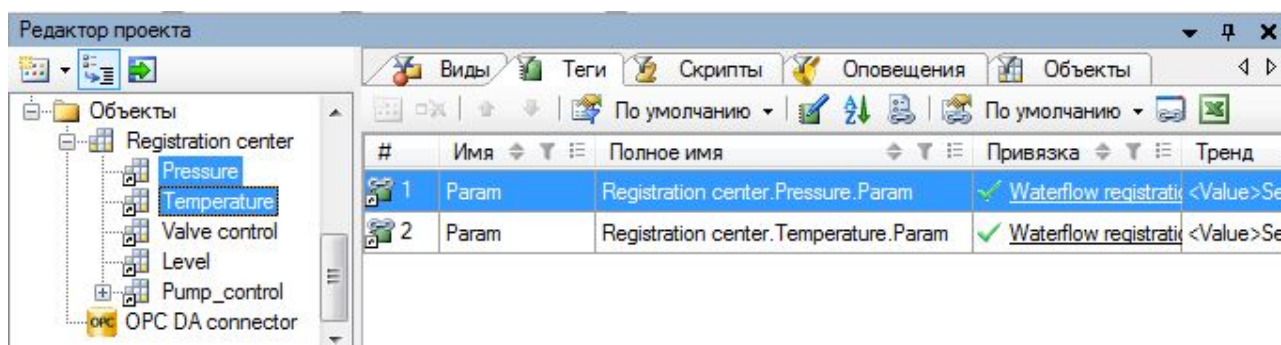
ВНИМАНИЕ !!!


Выбранным считается элемент в правой части диалога (Объект), а не элемент, выбранный слева в дереве проекта (коннектор).

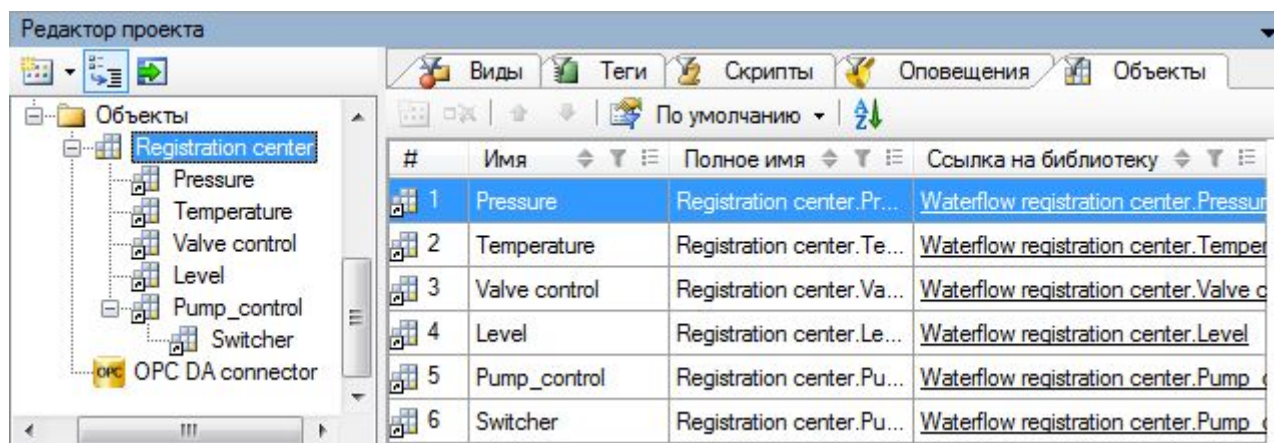
1.3 Работа с большим количеством объектов



Редактор проекта обеспечивает работу с большим количеством объектов. Для того чтобы сделать такую работу простой и "прозрачной", редактор предоставляет следующие возможности:

- При выделении нескольких объектов все их теги/виды/скрипты показываются в списке тегов/видов/скриптов.



- При нажатой кнопке  во вкладках отображаются все теги/виды/скрипты вложенных объектов. При этом узел дерева «раскрывается», и показываются все вложенные узлы.



- Список колонок включает операции сортировки, выделения и фильтрация тегов/видов/скриптов по заданным критериям
- Для каждого типа элемента на панели инструментов присутствует свой набор дополнительных команд.
Например, для тегов доступны команды:
 – групповое переименование
 – групповая перепривязка.

1.3.1 Расширенный список колонок

DataRate позволяет в процессе редактирования проекта задать набор колонок и свойств, по которым можно сортировать, фильтровать и выделять требуемые элементы проекта.

1.3.2 Групповая привязка

Групповая привязка – дополнительная команда для тегов, результатом выполнения которой является установка новых свойств привязки тегов.

1.3.3 Множественная вставка из буфера

Множественная вставка работает аналогично простой вставке объектов. Отличие заключается в том, что вставляется не один экземпляр, скопированного в буфер обмена элемента(ов), а столько, сколько укажет Пользователь.

1.4 Мышь и Drag-n-Drop

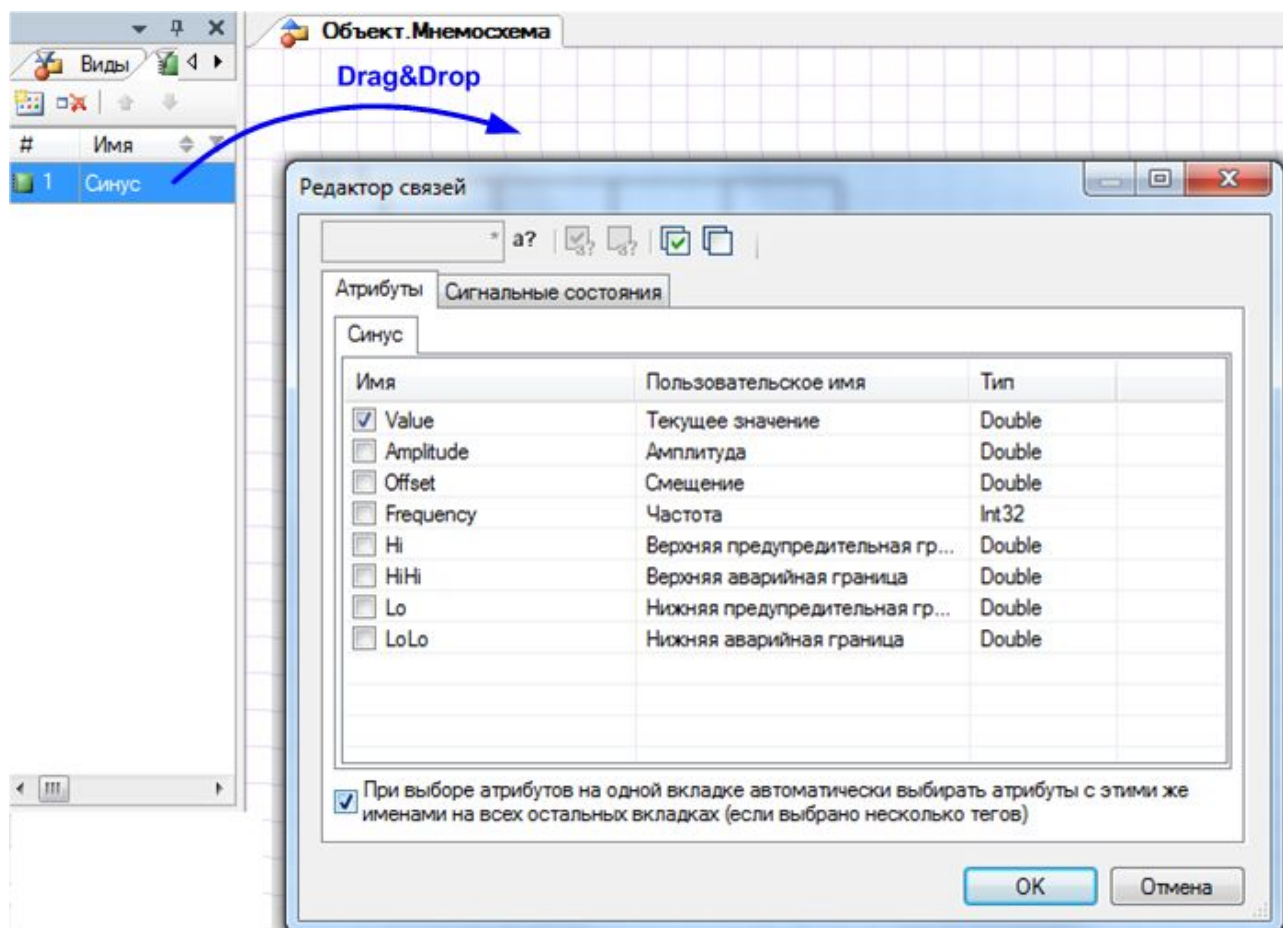
Для создания автоматизированной системы в рабочем пространстве ИСР используйте меню, кнопки на панели инструментов, а также:

- **Двойной щелчок левой кнопкой мыши** для отображения элемента проекта или его атрибутов в окне редактирования
- **Щелчок правой кнопкой мыши** для вызова контекстного меню
- **Операцию «перетащить и оставить» (Drag&Drop)** совместно с нажатой во время операции клавишей клавиатуры для выполнения следующих действий:
 - **Перемещение элемента** проекта – нажатых клавиш **нет**
 - **Связывание элементов** проекта – нажата клавиша **Alt**
 - **Копирование элемента** проекта – нажата клавиша **Ctrl**.

Пример

Рассмотрим, как с помощью операции **Drag&Drop** отобразить значение атрибута тега на мнемосхеме.

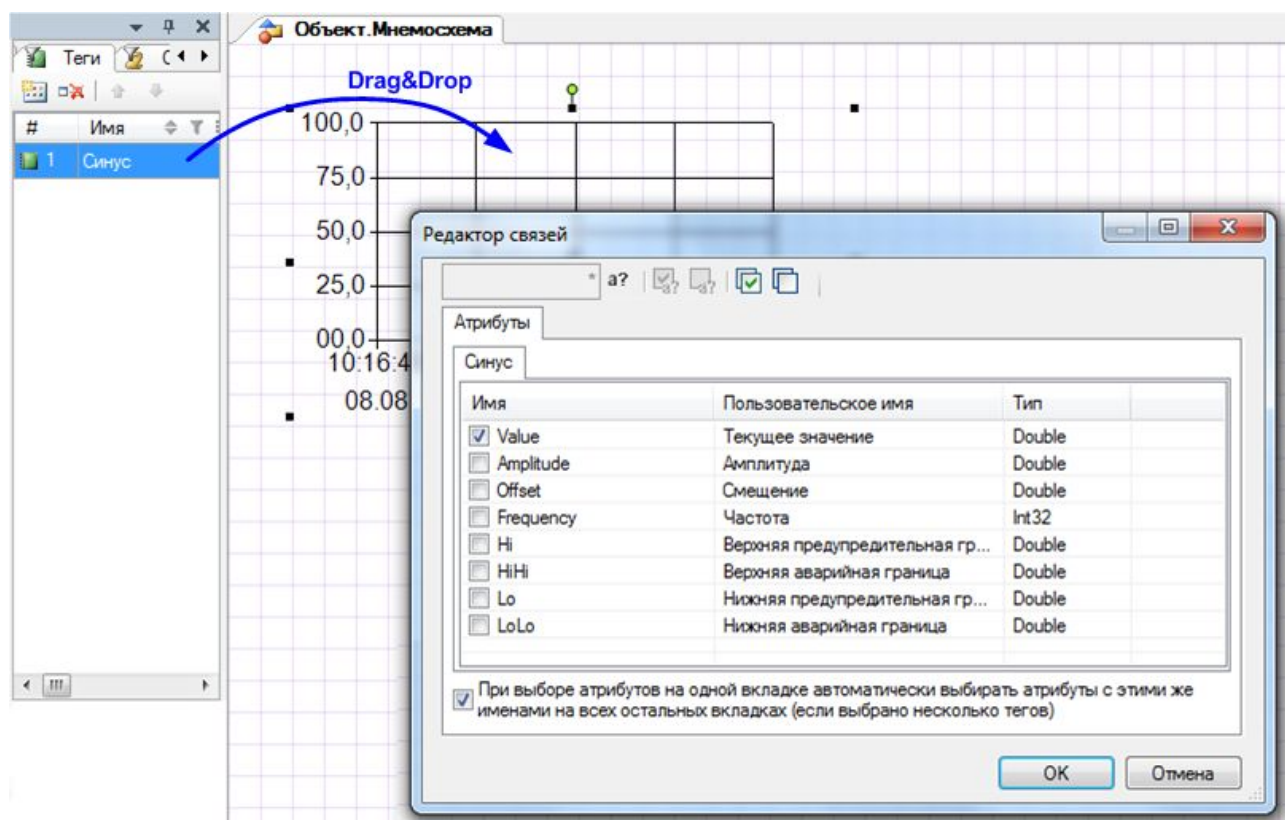
Сначала просто перетащим тег (в примере **Синус**) на мнемосхему. Автоматически открывается окно Редактор связей, в котором следует указать атрибут тега **Value** (значение).



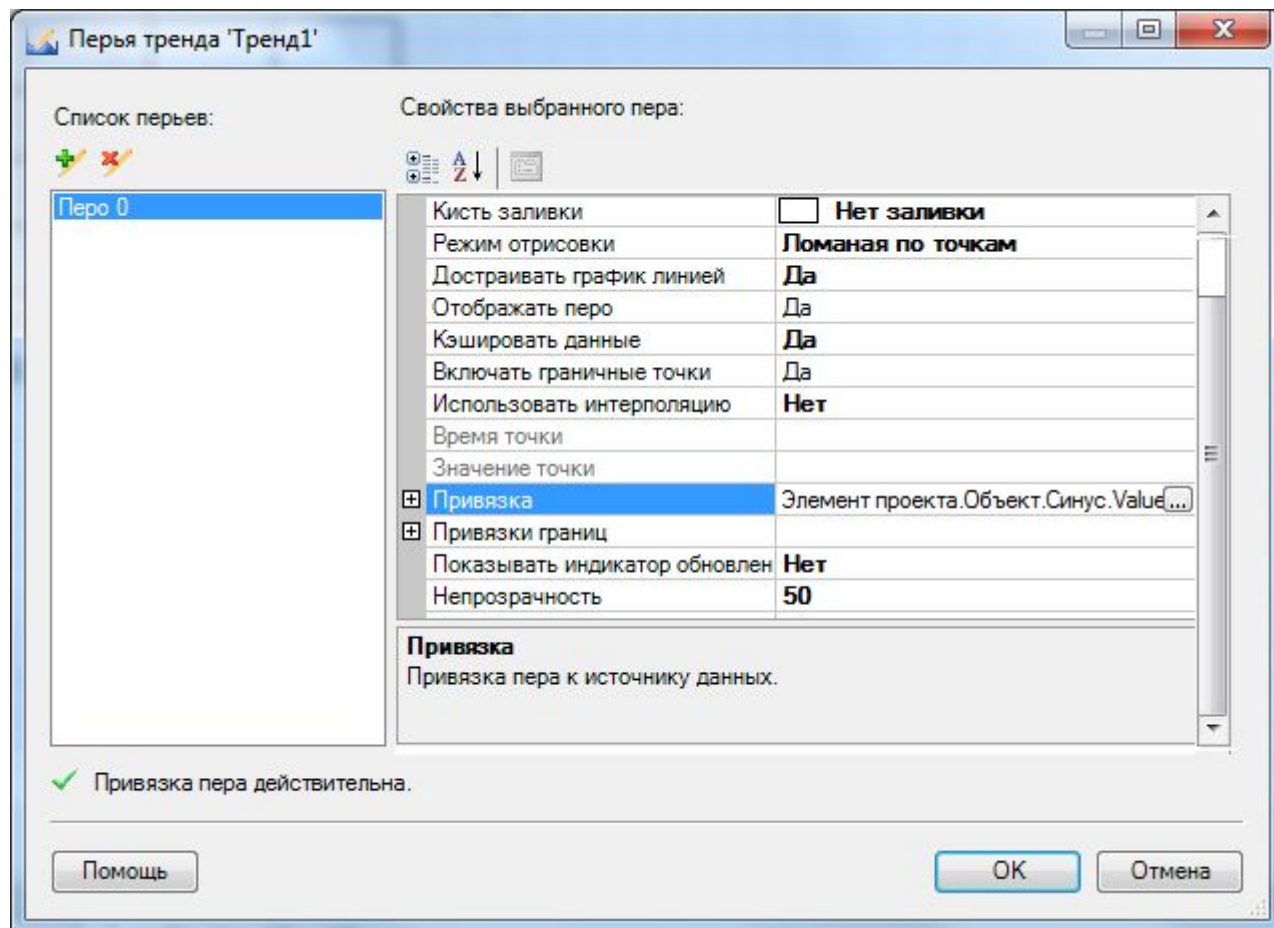
В результате на мнемосхему добавлен графический примитив Текст, значение которого «привязано» к указанному атрибуту тега.



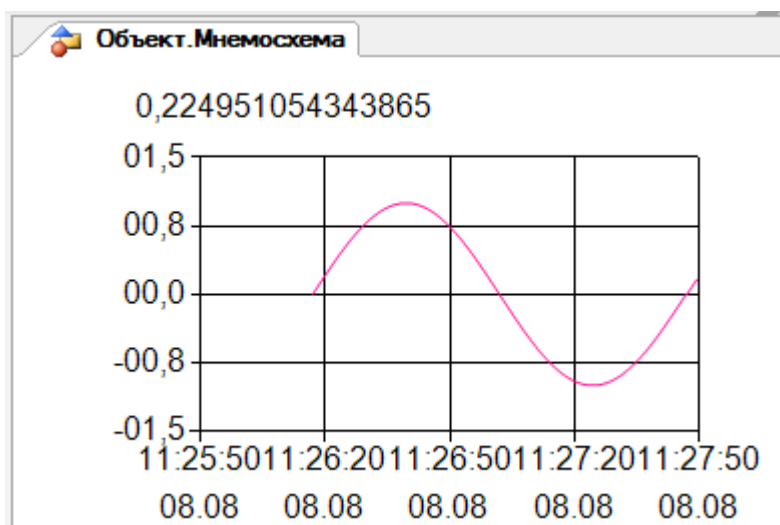
Аналогично можно перетянуть тег на графический примитив **Тренд**.



В этом случае автоматически добавиться перо тренда.



В результате на мнемосхеме значение атрибута тега отображается, как текст, так и в виде графика.



1.5 Горячие клавиши

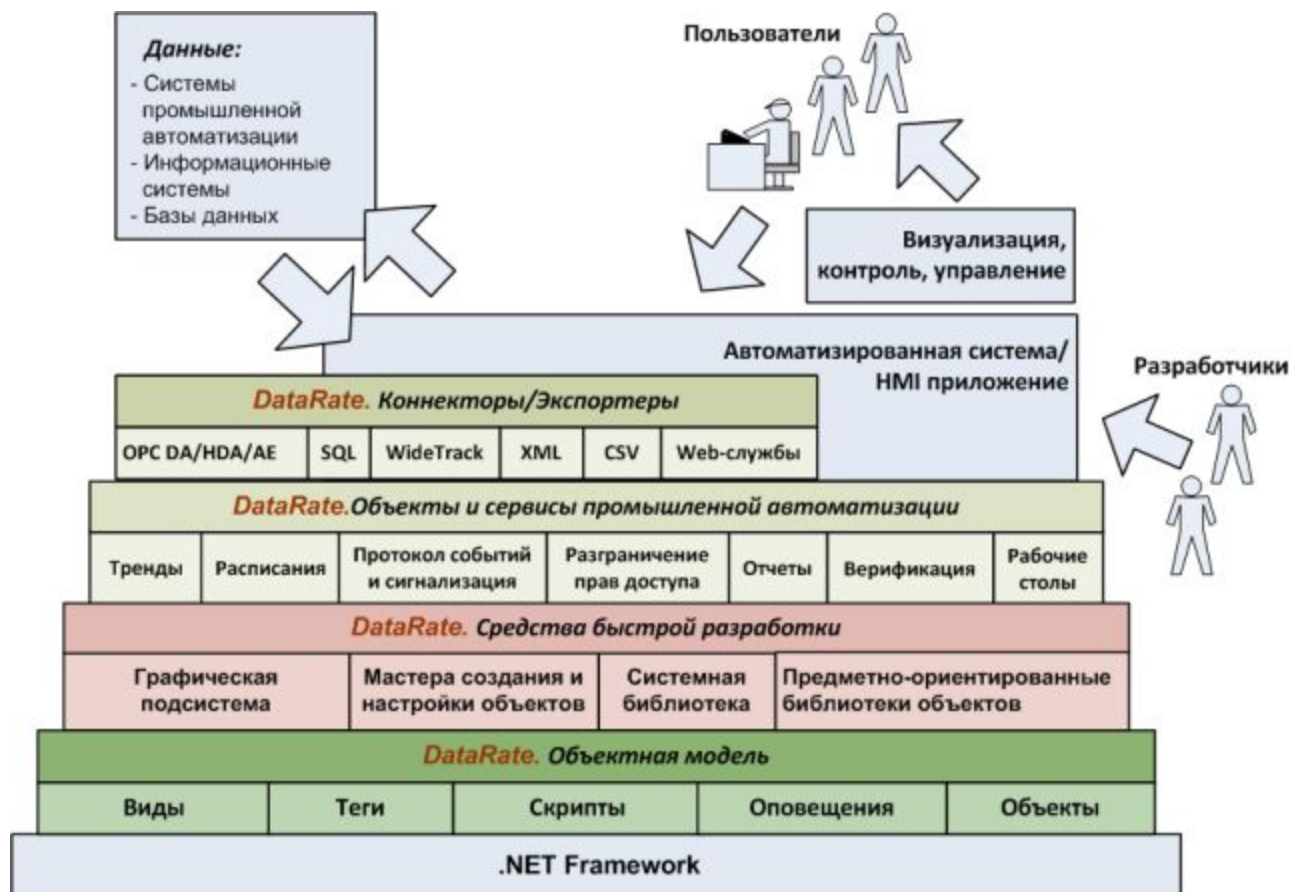
Горячие клавиши – специальные комбинации клавиш, призванные упростить и ускорить работу с проектом DataRate.

Комбинация	Назначение	Комбинация	Назначение
Редактирование			
<Ctrl>+<X>	Вырезать	<Ctrl>+<V> <Shift>+<Insert>	Вставить из буфера обмена
<Ctrl>+<C> <Ctrl>+<Insert>	Копировать в буфер обмена	<Ctrl>+<Delete>	Удалить
Файл			
<Ctrl>+<N>	Создать новый проект	<Ctrl>+<S>	Сохранить проект
<Ctrl>+<O>	Открыть проект	<Ctrl>+<Shift>+<S>	Сохранить проект как...
Отладка			
<F5>	Запустить проект на исполнение	<Shift>+<F5>	Остановить исполнение проекта
<Ctrl>+<F5>	Запустить проект с отладкой скриптов		
Помощь			
<F1>	Открыть справку	<Ctrl>+<F1>	Открыть окно "О программе"
Правка			
<Ctrl>+<Z>	Отменить действие	<Ctrl>+<Y>	Вернуть действие

О добавлении/удалении горячих клавиш смотрите в разделе **Меню и панели инструментов**.

2 ОБЪЕКТНАЯ МОДЕЛЬ

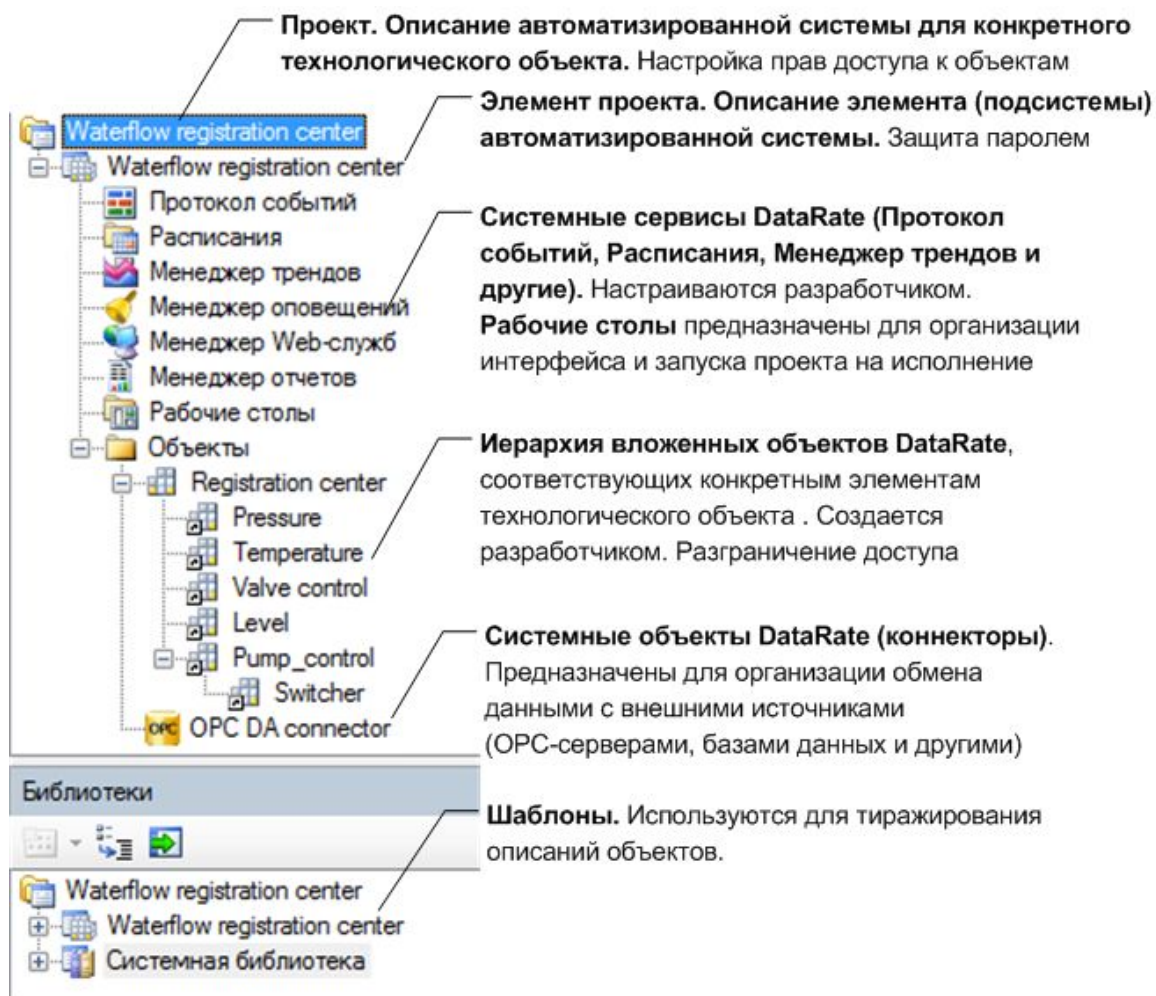
SCADA DataRate предоставляет разработчику визуальные и программные средства для создания автоматизированных систем из готовых и библиотечных объектов, а также ресурсы для создания собственных объектов и интерактивного взаимодействия с созданной автоматизированной системой и ее окружением.



Объектно-ориентированная архитектура **DataRate** позволяет реализовать типовой подход к разработке автоматизированной системы контроля и управления, при котором функциональные возможности системы (например, измерение и отображение параметров процессов и оборудования, ведение истории процесса и другие) можно встраивать в объекты, на основе дублирования и объединения которых формируется законченная система.

Проект автоматизированной системы в **DataRate** состоит из объектной модели технологического объекта и набора системных сервисов: **Протокол событий, Расписание, Менеджер трендов, Менеджер отчетов и Рабочие столы.**

Проект автоматизированной системы, созданный в **DataRate**, базируется на таких понятиях, как **проект, элемент проекта, объект, библиотека.**



Объект содержит виды, теги, скрипты, оповещения и вложенные объекты

Редактор проекта

#	Имя	Полное имя	Библиотечный
1	Circle_device	Registration center.Pr...	Да
2	Trend	Registration center.Pr...	Да
3	Sensor with scale...	Registration center.Pr...	Да

- Виды** – графические изображения (графические примитивы, мнемосхемы, шаблоны отчетов и другие)
- Теги** – входы/выходы объекта (данные с датчиков, сигналы управляющих механизмов, команды оператора и другие)
- Скрипты** – поведение объекта, алгоритмы работы на языке C# (с использованием .NET Framework). Высокая скорость выполнения алгоритма, реализованного скриптом, достигается за счет предварительной компиляции скриптов в машинные коды
- Оповещения** – действие, выполняемое для оповещения о достижении объектом определенного (например, аварийного) состояния.

Объекты можно помещать друг в друга, формируя при этом дерево, а также соединять друг с другом – просто «перетаскивая» мышью выходы одного объекта на вход другого.

Одним из главных преимуществ объектно-ориентированного подхода к разработке автоматизированных систем контроля и управления является повторное использование объектов.

Повторное использование объектов

Любой созданный объект можно поместить в библиотеку объектов. После этого он станет доступен для тиражирования в любом количестве в любом проекте. При создании копии объекта из библиотеки связь с библиотекой не теряется. Любые изменения в библиотеке будут отражаться и на самом объекте (при необходимости эту возможность можно отключить для конкретного объекта). Объект, взятый из библиотеки, «не статичен» – ему можно добавлять новые свойства и переопределять доставшиеся в наследство. Эти изменения никак не отразятся на других объектах.

2.1 Проект

Проект – корневой элемент объектной модели автоматизированной системы.


Проект включает **элементы проекта** и **библиотеки**. Структура проекта отображается в интегрированной среде в виде двух деревьев: **дерево проекта** и **дерево библиотек**.

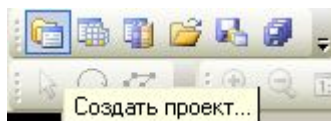
Создание, редактирование и отладка проекта осуществляется в **Интегрированной среде разработки**. Выполняется проект в **Среде исполнения**.

ВНИМАНИЕ!!!

При создании проекта в **Интегрированной среде разработки DataRate** необходимо учитывать ряд ограничений накладываемых на функциональность проекта при исполнении.

Функциональность	Среда Исполнения	Среда Исполнения-Клиент	Веб-клиент (браузер)
Исполнение скриптов «на примитивах»	+	+	+
Исполнение скриптов объектов	+	-	-
Реакции примитивов по правой кнопке мыши	+	+	-
Визуализация видов	+	+	+
Визуализация окна протокола событий	+	+	+
Формирование снимков данных в отчетах	+	+	+
Доступ из скриптов к объектной модели проекта	+	-	-
Доступ к источникам данных и параметрам видов	+	+	+
Доступ к файловой системе сервера Среды исполнения	+	-	-
Использование контролов WinForms на мнемосхемах	+	+	-
Использование примитива Элемент управления	+	+	-

Для создания проекта щелкните на кнопку  **Создать проект** на панели инструментов или в меню **Файл**.

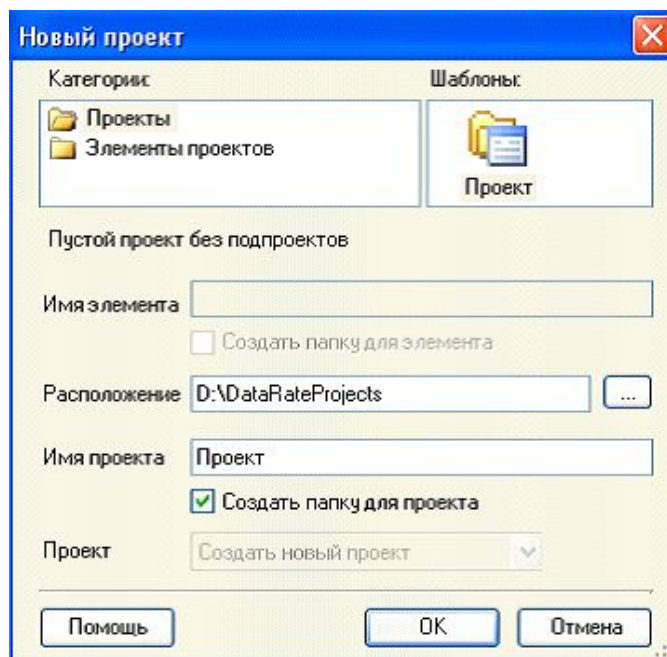


В появившемся окне **Новый проект** следует:

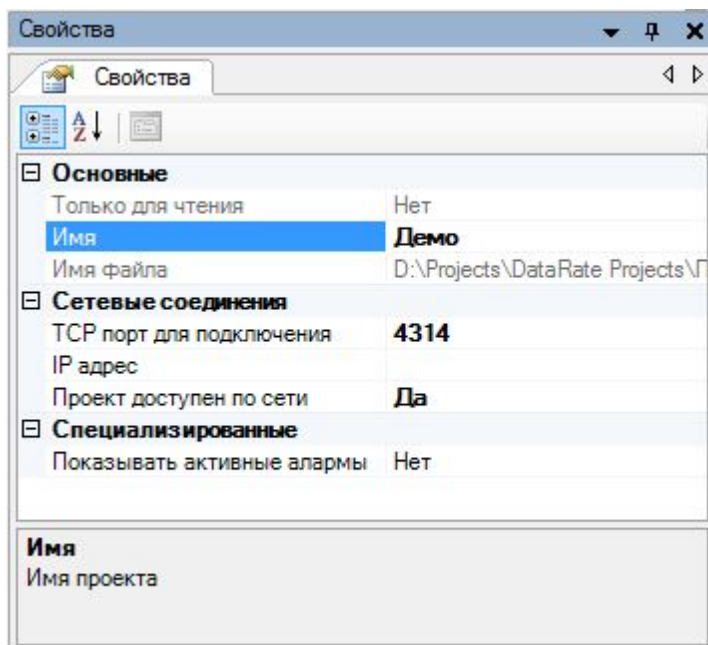
- 1 Выбрать категорию **Проекты** и шаблон **Проект**
- 2 Задать путь к файлу проекта **Расположение**
- 3 Ввести **Имя проекта**
- 4 Если галочка **Создать папку для проекта** установлена, файл проекта будет создан внутри одноименной папки

В дальнейшем Вы можете добавить в проект новые или существующие элементы проекта и библиотеки.

Описание проекта хранится в файле с расширением **.krproj**.



2.1.1 Свойства проекта



Секция Основные

- **Только для чтения** – признак возможности редактирования проекта: **Нет** – проект можно редактировать; **Да** – нельзя
- **Имя** – имя проекта
- **Имя файла** – имя файла, в который будет сохранен проект.

Секция Сетевые соединения

- **Проект доступен по сети** – если установить **Да**, то проект доступен по сети, **Нет** – проект не доступен
- **IP-адрес** – IP-адрес сетевого интерфейса для подключения сетевых клиентов. Если в компьютере стоит одна сетевая карта, интегрированная либо отдельный модуль (узнать эту информацию можно, выбрав в **компьютер/Управление/Диспетчер**

контекстном меню Windows **Мой компьютер/Управление/Диспетчер устройств/Сетевые платы**), то это поле можно оставить пустым. Если же в компьютере стоит две и более сетевых карты, то в этом поле надо ввести IP-адрес для

той сетевой карты, из сети которой будет вестись подключение к **DataRate Среда исполнения**

- **ТСР порт для подключения** – порт, на котором проект будет доступен из сети. Порт может быть задан, например, в **Среде исполнения-Клиент** в окне настройки сетевого проекта.

Секция Специализированные

Показывать активные алармы – свойство отвечает за автоматическое отображение окна активной сигнализации. По умолчанию – значение **Нет**. Для включения автоматического отображения окна активной сигнализации необходимо установить – **Да**.

2.1.2 Функции редактирования проекта

С помощью редактора проекта можно выполнить:

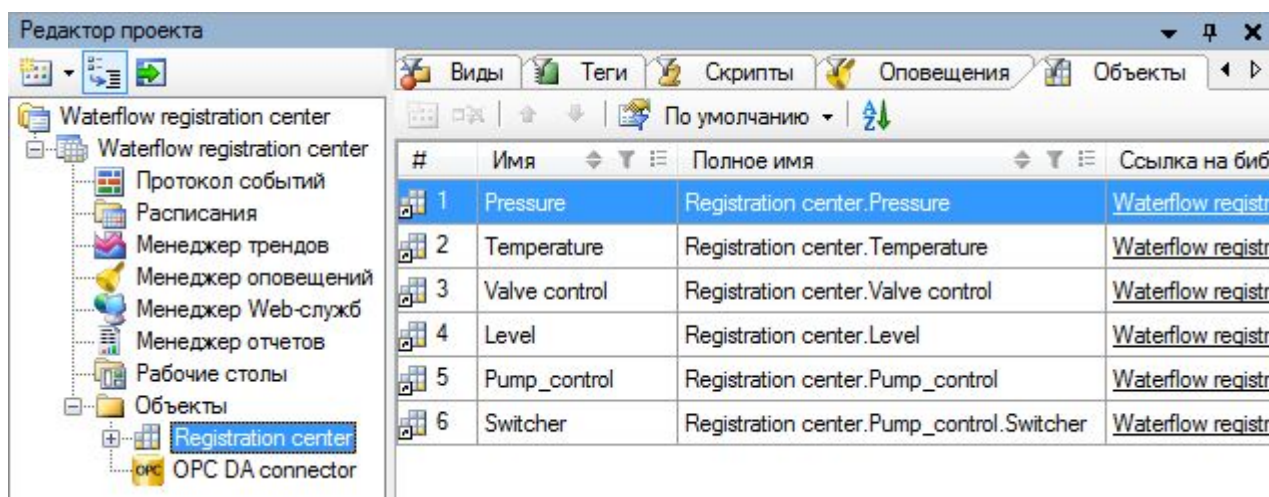
- Добавление/удаление элементов проекта, рабочих столов, папок
- Добавление/удаление объектов, их тегов, видов, скриптов и оповещений
- Помещение объекта в библиотеку
- Создание объекта из библиотеки.

Используйте операцию Drag&Drop и контекстное меню для редактирования структуры проекта.

2.1.3 Редактор проекта

Окно редактора проекта состоит из двух взаимодополняющих частей:

- дерево проекта
- список дочерних элементов узла дерева.



2.1.3.1 Дерево проекта

В дереве проекта отображаются только те элементы проекта, которые могут содержать дочерние элементы, например, **Протокол событий**, **Расписания**, **Менеджер трендов** и другие элементы.

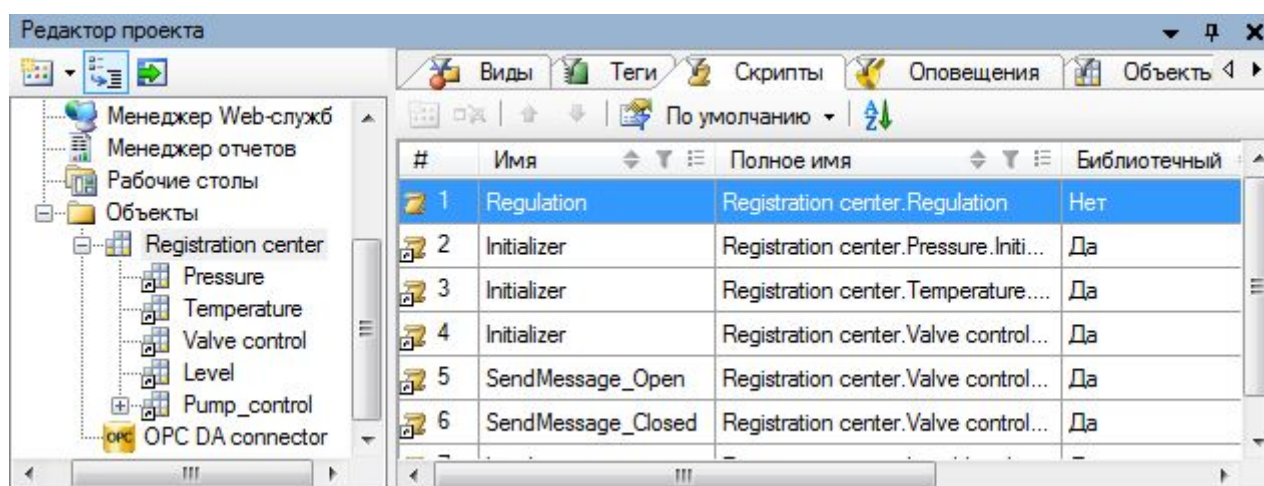
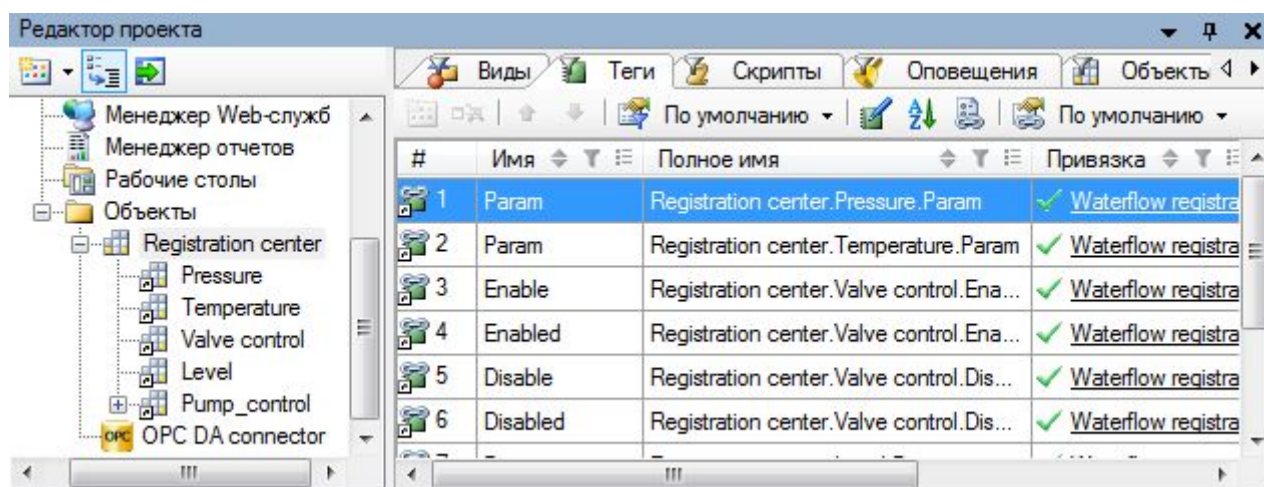
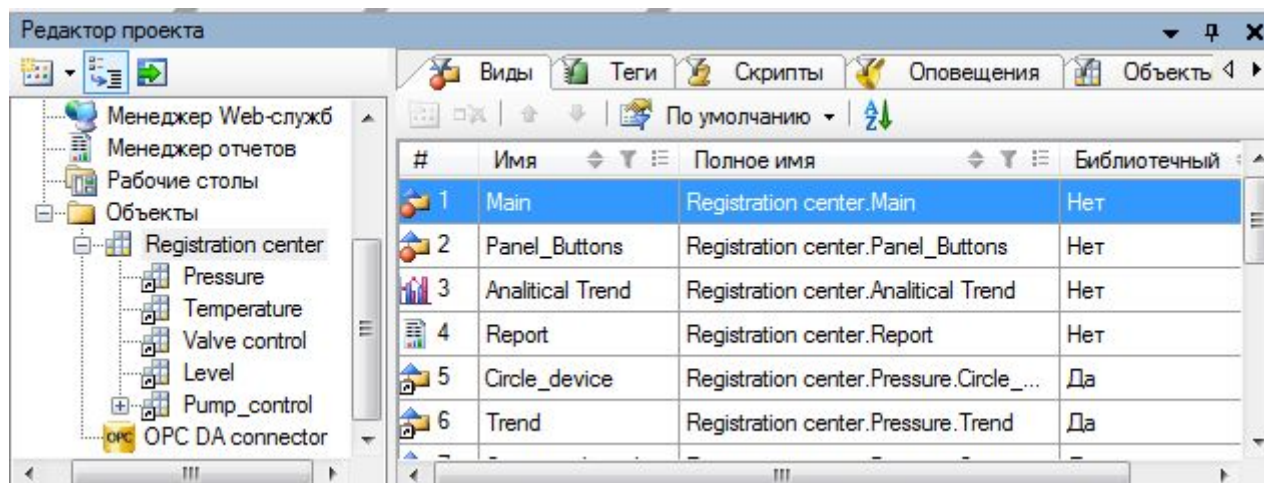
Объекты, содержащие вложенные объекты, также отображаются в дереве проекта. Например, объект **Registration center** содержит объекты **Pressure**, **Temperature** и другие.

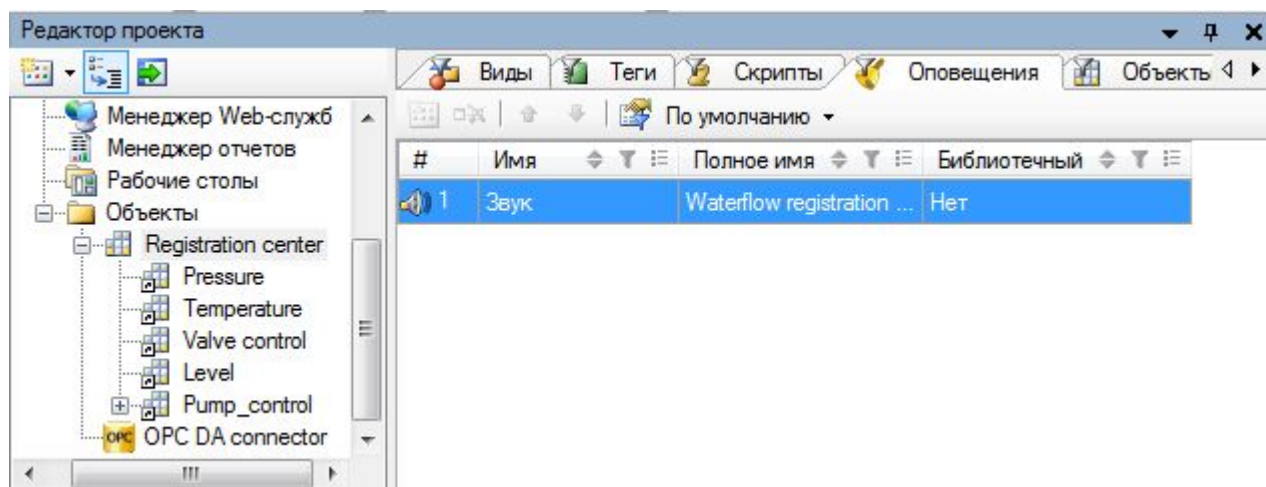
2.1.3.2 Список дочерних элементов


Список дочерних элементов узла дерева отображает элементы, являющиеся дочерними по отношению к выделенному узлу дерева.

Если курсор в дереве проекта установлен на папке объектов, то в списке дочерних элементов будут отображаться содержащиеся в этой папке объекты.

Если в дереве выделить объект, то в списке дочерних элементов появятся закладки с описанием выделенного объекта.



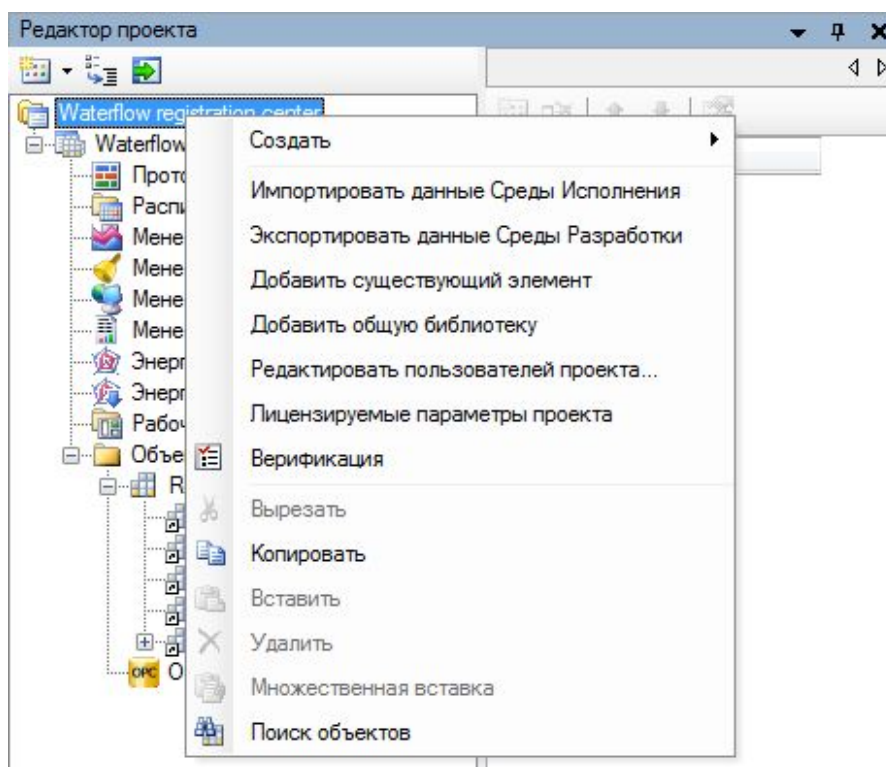


Кнопка  позволяет скрыть или повторно отобразить список дочерних элементов.

Кнопка  позволяет включить или отключить отображение всех дочерних элементов

2.1.3.3 Контекстное меню

У каждого элемента дерева проекта есть контекстное меню, которое вызывается щелчком правой клавиши мыши, с индивидуальными командами для каждого элемента.



Общими командами редактирования в контекстном меню являются: **Вырезать**, **Копировать**, **Вставить**, **Удалить**, **Множественная вставка** и **Поиск объектов**.

2.1.3.4 Расширенный список колонок

Редактор проекта **DataRate** позволяет задать набор колонок и свойств, по которым можно сортировать, фильтровать и выделять требуемые элементы проекта.

Виды Теги Скрипты Оповещения Объекты			
По умолчанию			
#	Имя	Полное имя	Библиотечный
1	Мнемосхема	Главный узел.Мнемосхема	Нет
2	Мнемосхема1	Главный узел.Мнемосхема1	Нет
3	Отчет	Главный узел.Отчет	Нет
4	Протокол событий	Главный узел.Протокол событий	Нет
5	Тренд	Главный узел.Тренд	Нет

2.1.3.5 Операции

Каждое свойство элемента (вида, тега, скрипта и объекта) имеет следующий набор операций.

Иконка



Операция

Сортировка элементов
Фильтрация элементов
Выделение элементов

Операция может быть активна или не активна.

Если операция активна, то соответствующая иконка будет иметь **красный цвет**.

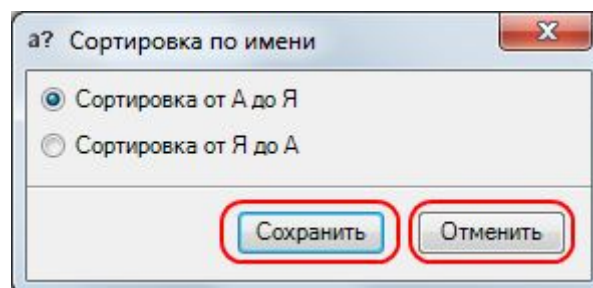
Если для свойства элемента не задано операций, то кнопки в соответствующей колонке будут отсутствовать.

Свойства операции

Для задания свойств операции следует щелкнуть правой кнопкой мыши на иконке, соответствующей операции, и в появившемся окне задать значения свойств операции.

При нажатии на кнопку **Сохранить** свойства операции будут сохранены и будет выполнена операция с заданными свойствами.

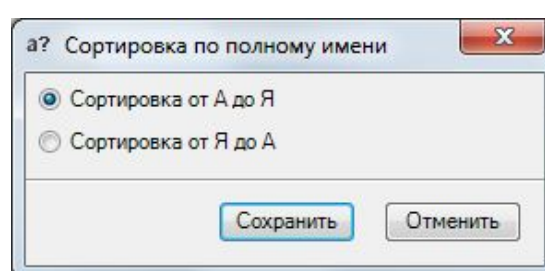
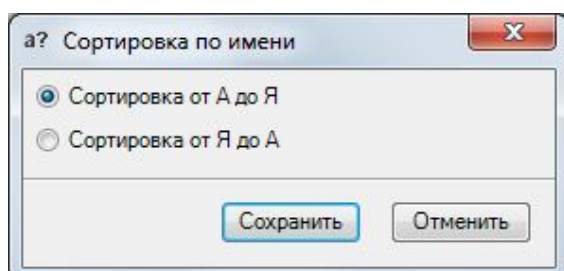
Кнопка **Отменить** «сбрасывает» свойства операции и не выполняет операцию.



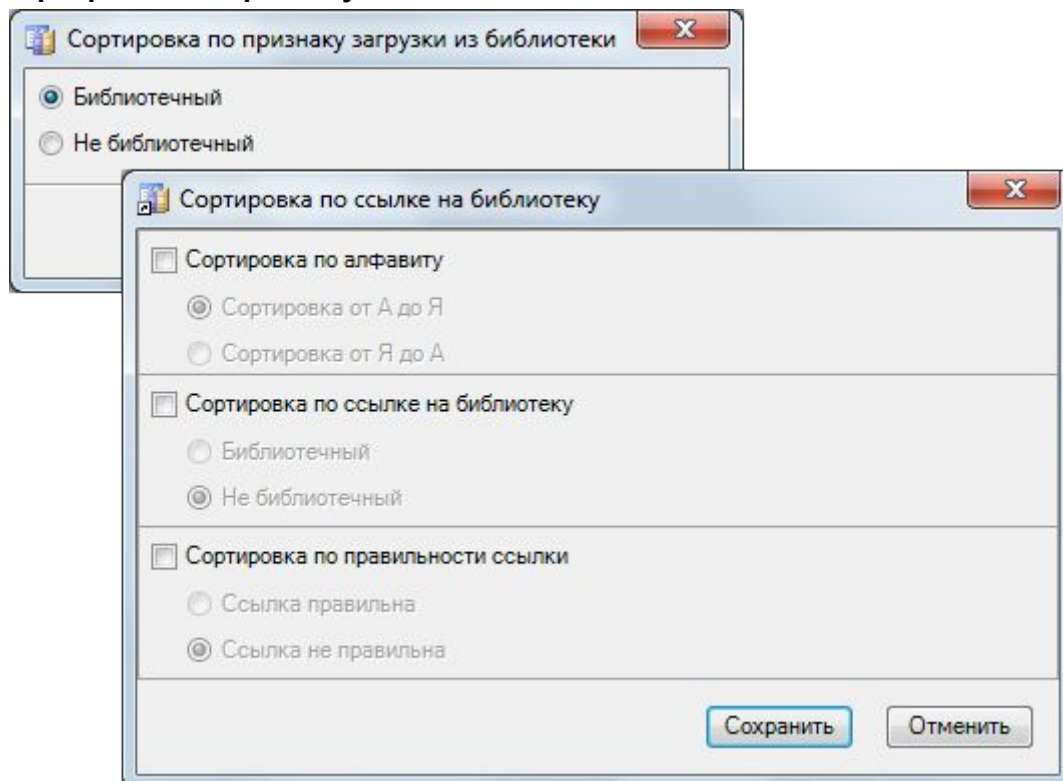
Сортировка элементов

Сортировка элементов доступна в каждой колонке расширенного списка редактора проекта. В зависимости от вкладки редактора проекта (Объекты, Виды, Теги и другие) доступны разные колонки и разные виды сортировки:

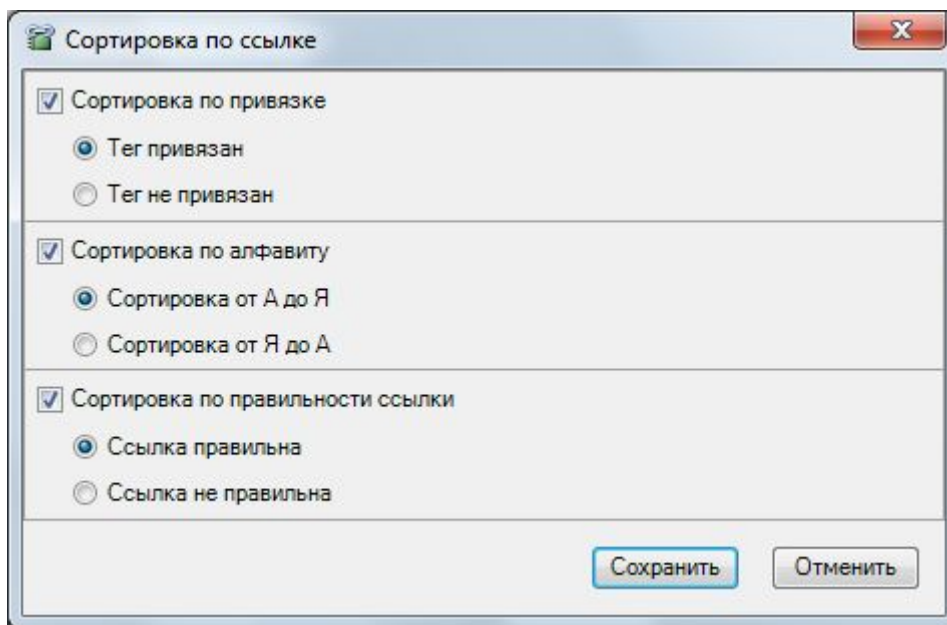
- **Сортировка по имени**



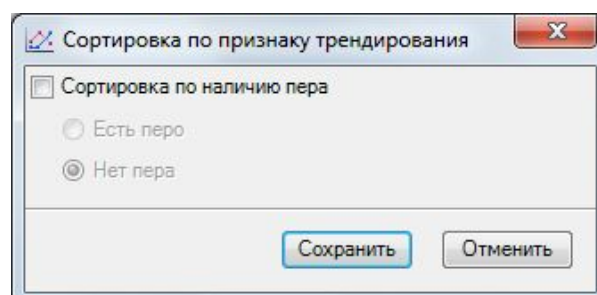
- **Сортировка по признаку связи с библиотекой**



- **Сортировка для тегов по колонке Привязка**



- **Сортировка для тегов по наличию пера Менеджера трендов**

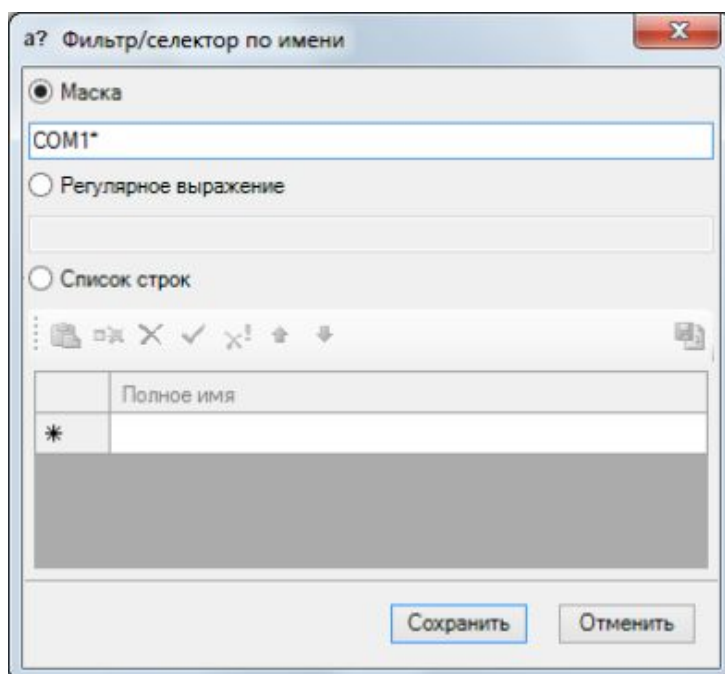


Фильтрация

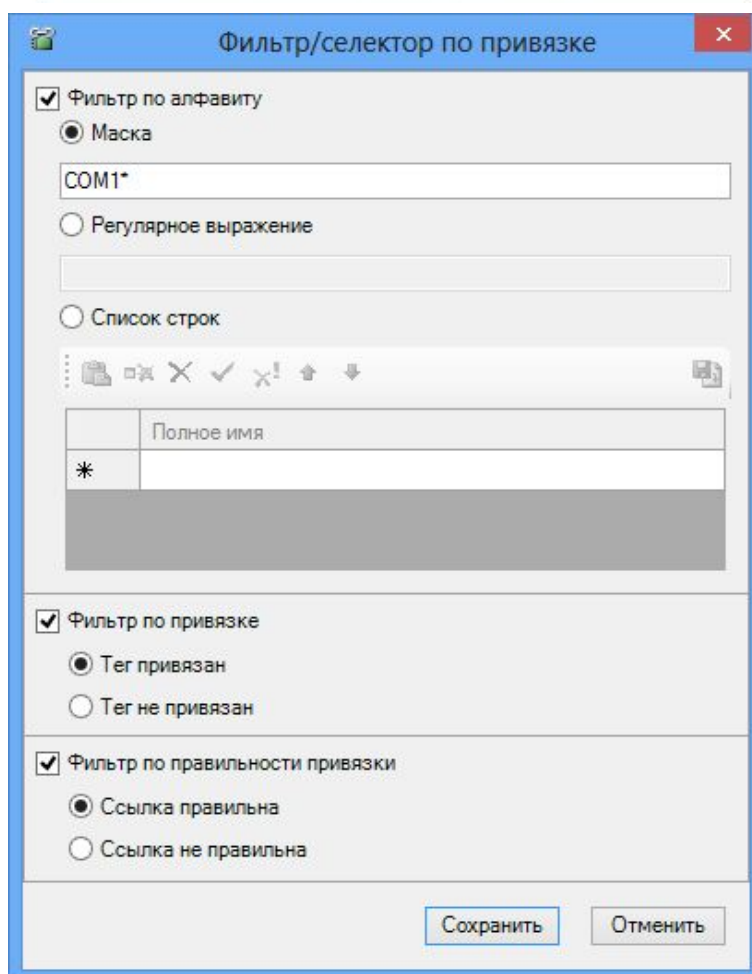
Фильтрация позволяет оставить в списке редактора проекта только те элементы, которые удовлетворяют критериям выбора. Фильтр/селектор элементов доступен в каждой колонке расширенного списка редактора проекта.

В зависимости от вкладки редактора проекта (Объекты, Виды, Теги и т.д.) доступны разные колонки и разные виды фильтрации:

- **Фильтрация по имени**



- **Фильтрация по свойствам привязки тегов**



- Фильтрация по наличию пера и тега в разных группах настроек

Фильтр/селектор для тегов по признаку наличия трендирования

☒ Фильтр по наличию пера

☒ Есть перо

☐ Нет пера

☒ Фильтр по группам настроек

Имя группы	
(by default)	<input type="checkbox"/>
SettingsGroup	<input checked="" type="checkbox"/>

Сохранить Отменить

- Фильтрация по признаку связи с библиотекой

Фильтр/селектор по признаку загрузки из библиотеки

☐ Библиотечный

☒ Не библиотечный

Сохранить Отменить

Фильтр/селектор для ссылок на библиотеку

☒ Фильтр по алфавиту

☒ Маска

Задвижка*

☐ Регулярное выражение

☐ Список строк

Полное имя

*

☒ Фильтр по ссылке на библиотеку

☒ Библиотечный

☐ Не библиотечный

☒ Фильтр по правильности привязки

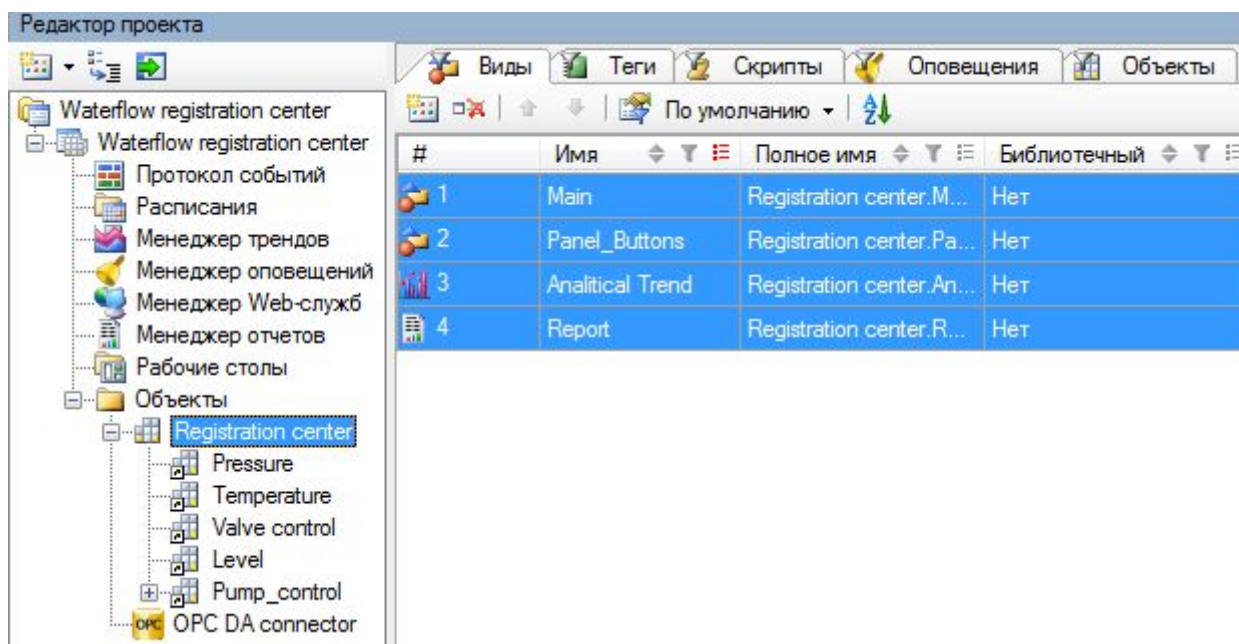
☒ Ссылка правильна

☐ Ссылка не правильна


Сохранить Отменить

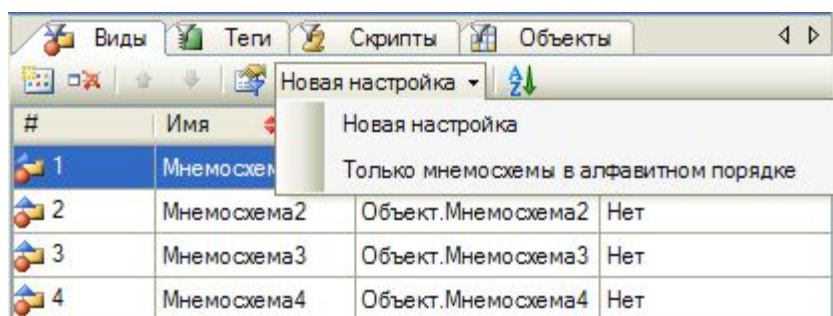
Выделение элементов

Выделение элементов доступно в каждой колонке расширенного списка редактора проекта. В зависимости от вкладки редактора проекта (Объекты, Виды, Теги и т.д.) доступны разные колонки и разные виды выделения элементов. Выделение элементов позволяет выделить группу элементов списка для выполнения групповых операций



Загрузка конфигурации

Если есть сохраненные значения свойств операций – **конфигурации**, то в выпадающем списке, справа от кнопки , будет показана текущая загруженная конфигурация свойств, а в списке – все доступные конфигурации.




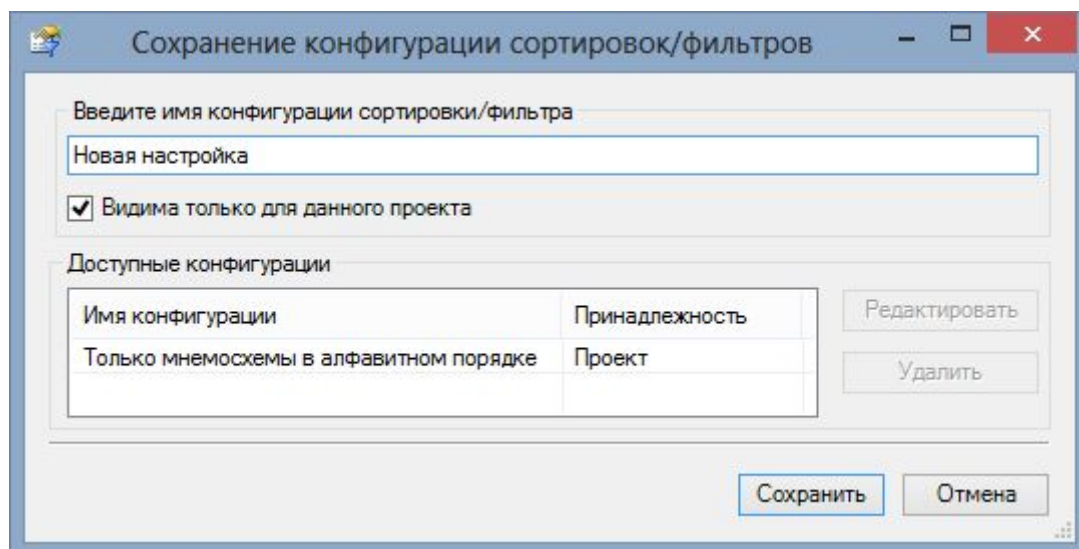
Для того чтобы загрузить ранее сохраненные конфигурации свойств достаточно в выпадающем списке выбрать нужную конфигурацию.

Если доступных настроек нет, то в списке будет выведена надпись **Нет**.

Сохранение конфигурации

Для того чтобы сохранить текущее состояние операций следует:

- 1 Щелкнуть по кнопке 
- 2 В появившемся окне задать **имя конфигурации** и ее **видимость**
- 3 Если признак **Для текущего проекта** установлен, то конфигурация видна только текущему проекту, в противном случае – всем проектам.



Редактирование и удаление

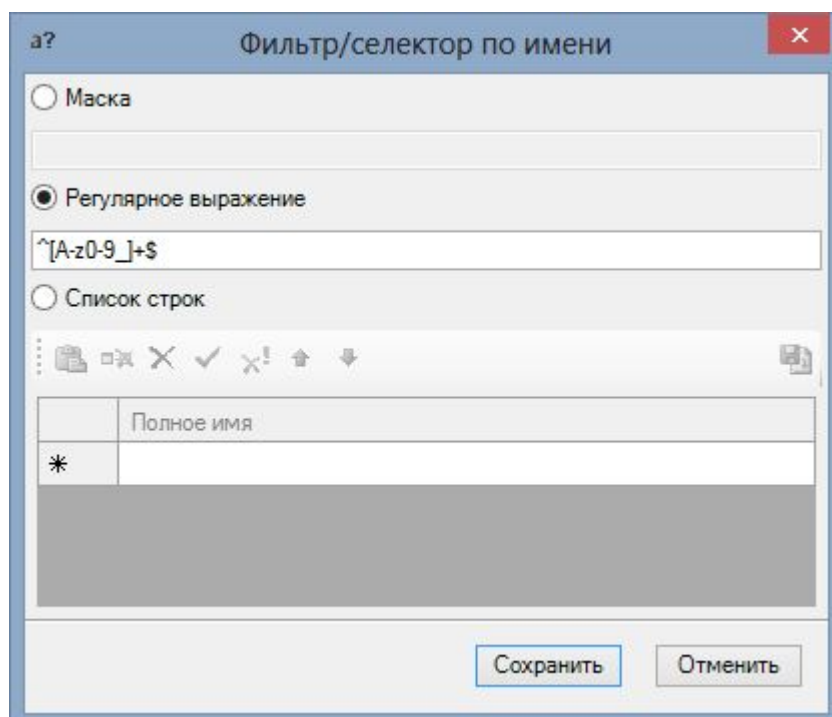
При нажатии на кнопку **Редактировать** откроется окно, в котором можно изменить имя и видимость конфигурации.

Кнопка **Удалить** удаляет выделенные конфигурации.

2.1.3.6 Использование регулярного выражения для фильтрации или выделения элементов

Для того, чтобы выбрать или выделить элементы с использованием регулярного выражения следует:

- 1 Открыть окно **Фильтр/селектор** соответствующего свойства
- 2 Установить флаг **Регулярное выражение** и ввести текст регулярного выражения в соответствующее поле
- 3 Нажать на кнопку **Сохранить**.



Пример 1. Фильтрация/выделение элементов по имени

Имя может включать:

- латинские большие и малые символы
- цифры
- символ нижнего подчёркивания

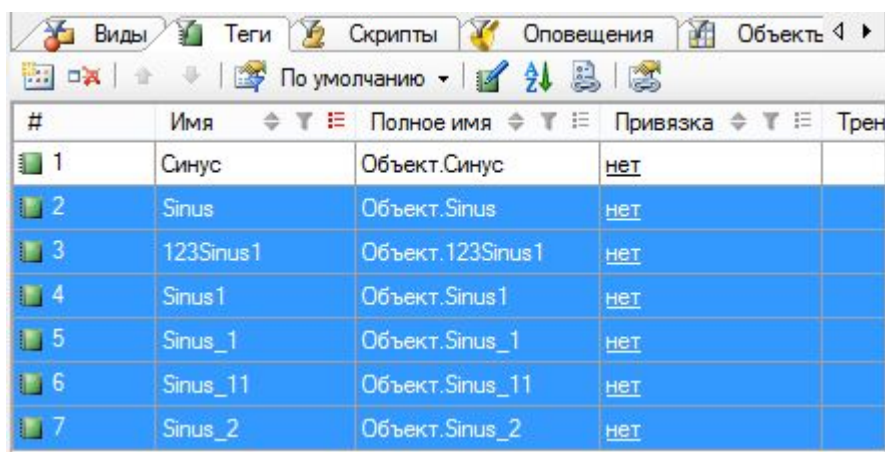
Регулярное выражение: **^[A-Za-z0-9_]+\$**

или более кратко: **^[A-z0-9_]+\$**

где:

- ^** – начало строки
- [A-Za-z0-9]** – диапазон символов
- A-Z** – большие латинские символы
- a-z** – малые латинские символы
- 0-9** – цифры от 0 до 9
- _** – символ нижнего подчёркивания
- +** – повторение указанного символа один или более раз (в данном случае, любого из диапазона символов)
- \$** – конец строки

Результат применения:



#	Имя	Полное имя	Привязка	Трен
1	Синус	Объект.Синус	нет	
2	Sinus	Объект.Sinus	нет	
3	123Sinus1	Объект.123Sinus1	нет	
4	Sinus1	Объект.Sinus1	нет	
5	Sinus_1	Объект.Sinus_1	нет	
6	Sinus_11	Объект.Sinus_11	нет	
7	Sinus_2	Объект.Sinus_2	нет	

Пример 2. Фильтрация/выделение элементов по имени

Имя может включать от 5 до 7 символов:

- русские большие и малые символы
- латинские большие и малые символы
- цифры
- символ нижнего подчёркивания

Регулярное выражение: **^[A-яA-z0-9_]{5,7}\$**

где:

- ^** – начало строки
- [A-яA-z]** – диапазон символов
- A-я** – большие и малые русские символы
- A-z** – большие и малые латинские символы
- 0-9** – цифры от 0 до 9
- _** – символ нижнего подчёркивания
- {5,7}** – длина строки от 5 до 7 символов
- \$** –конец строки

Результат применения:

#	Имя	Полное имя	Привязка	Трен
1	Синус	Объект.Синус	нет	
2	Sinus	Объект.Sinus	нет	
3	123Sinus1	Объект.123Sinus1	нет	
4	Sinus1	Объект.Sinus1	нет	
5	Sinus_1	Объект.Sinus_1	нет	
6	Sinus_11	Объект.Sinus_11	нет	
7	Sinus_2	Объект.Sinus_2	нет	

Пример 3. Фильтрация/выделение элементов по имени

Имя содержит:

- любое количество любых символов
- начинается с символов «Sinus.»
- заканчивается символом «1»

Регулярное выражение: **^Sinus.*1\$**

где:

- | | |
|--------------|---|
| ^ | – начало строки |
| Sinus | – подстрока «Sinus» |
| . | – любой символ |
| * | – повторение указанного символа любое количество раз (в данном случае, повторение любого символа) |
| 1 | – символ «1» |
| \$ | – конец строки |

Результат применения:

#	Имя	Полное имя	Привязка	Трен
1	Синус	Объект.Синус	нет	
2	Sinus	Объект.Sinus	нет	
3	123Sinus1	Объект.123Sinus1	нет	
4	Sinus1	Объект.Sinus1	нет	
5	Sinus_1	Объект.Sinus_1	нет	
6	Sinus_11	Объект.Sinus_11	нет	
7	Sinus_2	Объект.Sinus_2	нет	

ВНИМАНИЕ!!!

Если не указывать символы начало строки – ^ и конец строки – \$, то условия будут применяться не для целой строки, а для любой её части (подстроки).

Пример 4. Фильтрация/выделение элементов по имени

Имя содержит:

- любое количество любых символов
- подстроку «Sinus»
- подстроку «1»
- подстрока «Sinus» расположена ближе к началу, чем подстрока «1»

Регулярное выражение: **Sinus.*1**

где:

- Sinus** – подстрока «Sinus»
. – любой символ
***** – повторение указанного символа любое количество раз (в данном случае, повторение любого символа)
1 – символ «1»

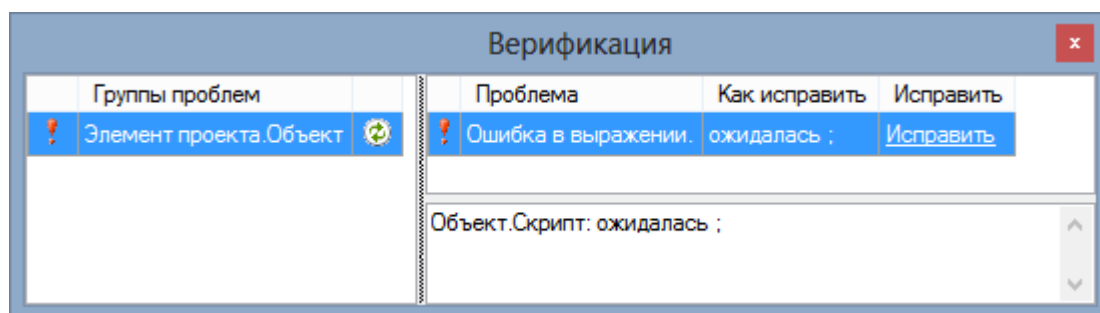
Результат применения:

#	Имя	Полное имя	Привязка	Трен
1	Синус	Объект.Синус	нет	
2	Sinus	Объект.Sinus	нет	
3	123Sinus1	Объект.123Sinus1	нет	
4	Sinus1	Объект.Sinus1	нет	
5	Sinus_1	Объект.Sinus_1	нет	
6	Sinus_11	Объект.Sinus_11	нет	
7	Sinus_2	Объект.Sinus_2	нет	
8	221Sinus33	Объект.221Sinus33	нет	

2.1.4 Верификация проекта

Для того чтобы выявить ошибки в проекте, не запуская его на исполнение, необходимо выполнить **верификацию** проекта.

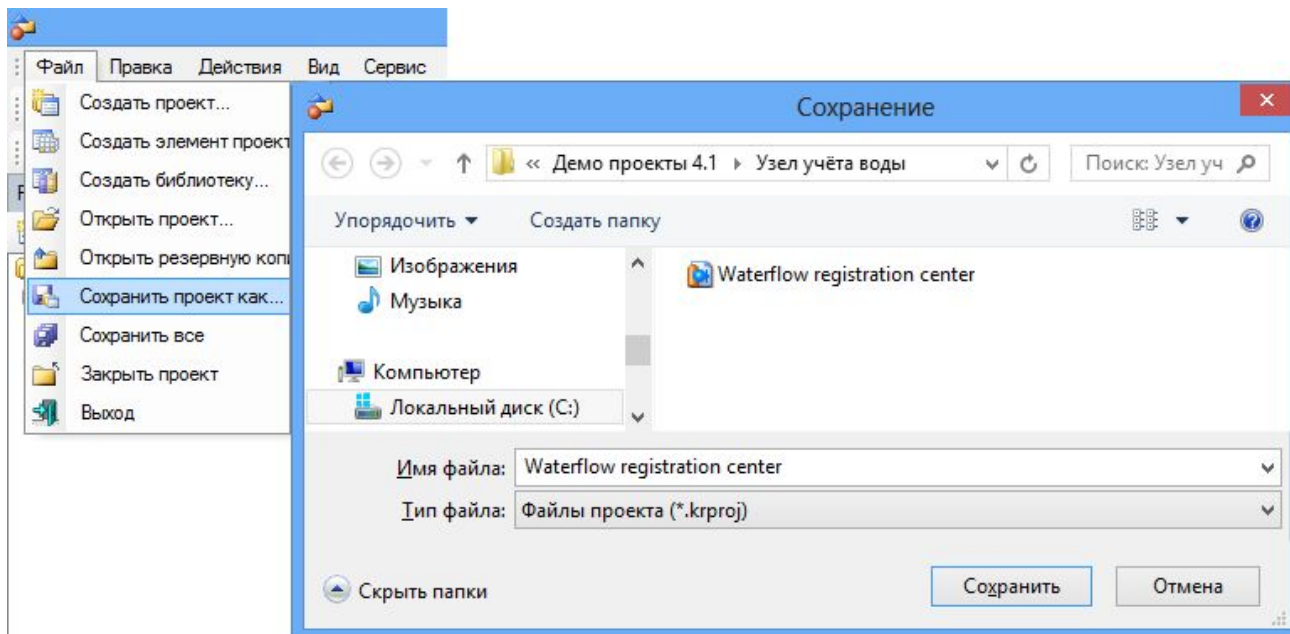
Для верификации выберите пункт **Верификация** из контекстного меню проекта. Просмотреть результаты верификации можно в окне **Верификация**.



Аналогично можно осуществить и верификацию объектов проекта.

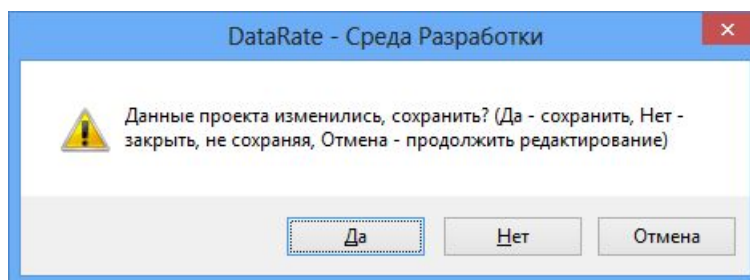
2.1.5 Сохранение проекта

Сохранить проект в **Среде Разработки DataRate** можно из меню **Файл/Сохранить проект как**



При закрытии **Среды Разработки DataRate** автоматически выводится окно с предложением сохранить редактируемый проект.

Если Пользователь отвечает **Да**, то проект сохраняется в соответствии с параметрами, заданными при его создании (**Расположение**, **Имя проекта** и другими).



ВНИМАНИЕ!!!

Рекомендуется, периодически в процессе работы и по окончании работы над проектом создавать резервную копию проекта.

Как создать резервную копию проекта

Для создания резервной копии проекта следует нажать кнопку **Сохранить все** на панели инструментов. При этом автоматически создается копия проекта в файле с расширением ***.backup**.

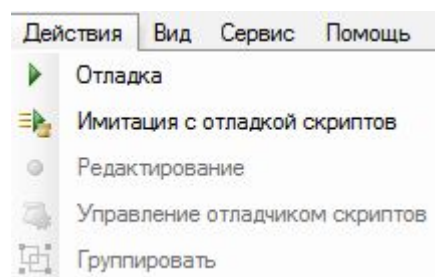
Копия расположена там же, где и основной проект.

Как открыть резервную копию проекта

Копию проекта можно открыть из меню **Файл/Открыть резервную копию**.

2.1.6 Отладка проекта

Используйте кнопки для начала отладки проекта и кнопку **Редактирование** – для завершения отладки (на панели инструментов или из меню **Действия**).

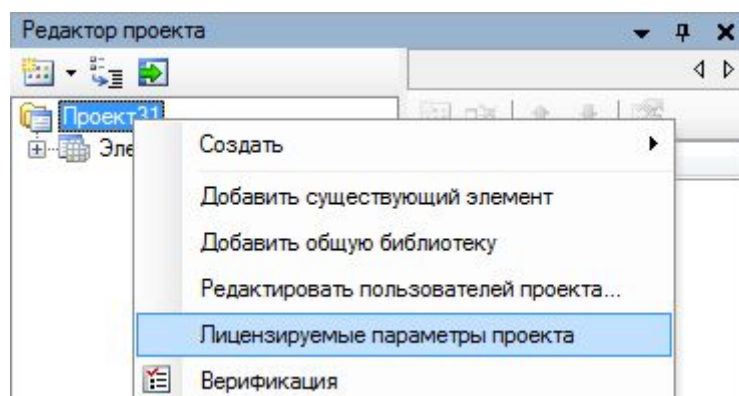


2.1.7 Лицензируемые параметры проекта




Лицензируемые параметры проекта содержат информацию о количестве и составе используемых компонентов.

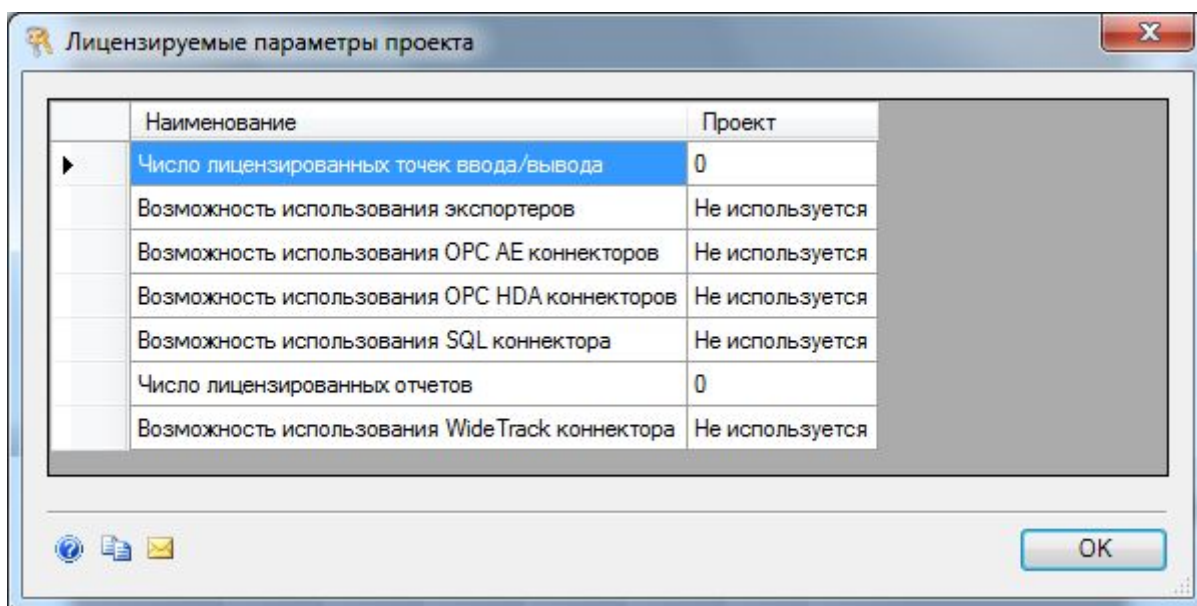
Чтобы открыть окно **Лицензируемые параметры проекта** следует:

- 1 Вызвать контекстное меню проекта (вызывается правой кнопкой мыши)



- 2 Выбрать пункт **Лицензируемые параметры проекта**
- 3 В появившемся окне отображается информация о лицензируемых параметрах проекта. В окне доступны следующие кнопки управления:

-  – Открыть окно справки
-  – Поместить информацию в буфер обмена
-  – Отправить сообщение в группу технической поддержки. При щелчке мышью по этой кнопке будет сформировано письмо для отправки в группу технической поддержки.

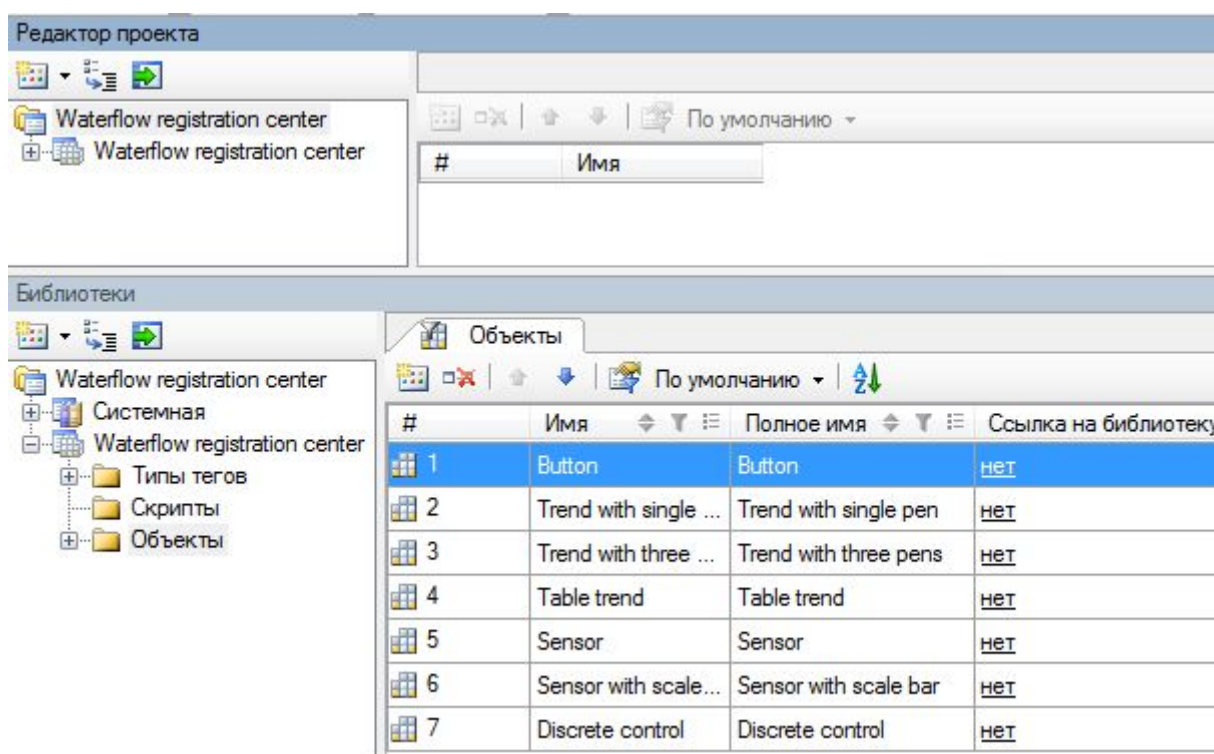


2.2 Элемент проекта

Элемент проекта – часть проекта, которая содержит рабочие столы и объекты. Элемент проекта может использоваться в проекте автоматизированной системы как подсистема, компонент подсистемы или отдельный объект.

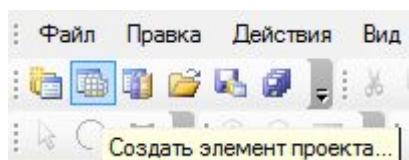
Структура элемента проекта включает две части:

- **Основная часть.** Отображается в дереве проекта. Содержит объекты, необходимые для реализации функций элемента проекта.
- **Библиотечная часть.** Отображается в дереве библиотек. Содержит те составляющие элемента проекта, которые предназначены для повторного использования (шаблоны объектов, типы тегов, библиотеки скриптов). Библиотечная часть используется для создания объектов основной части проекта.

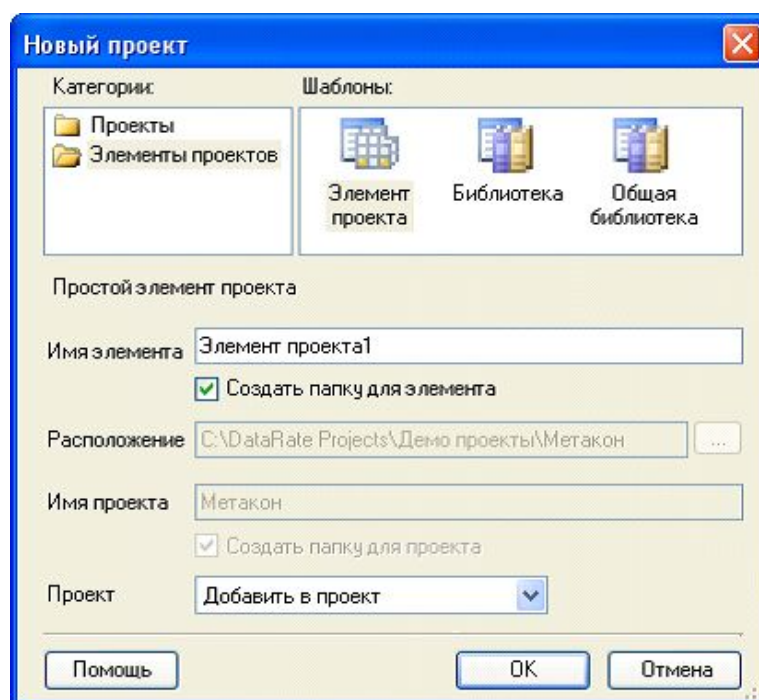


Для создания элемента проекта следует:

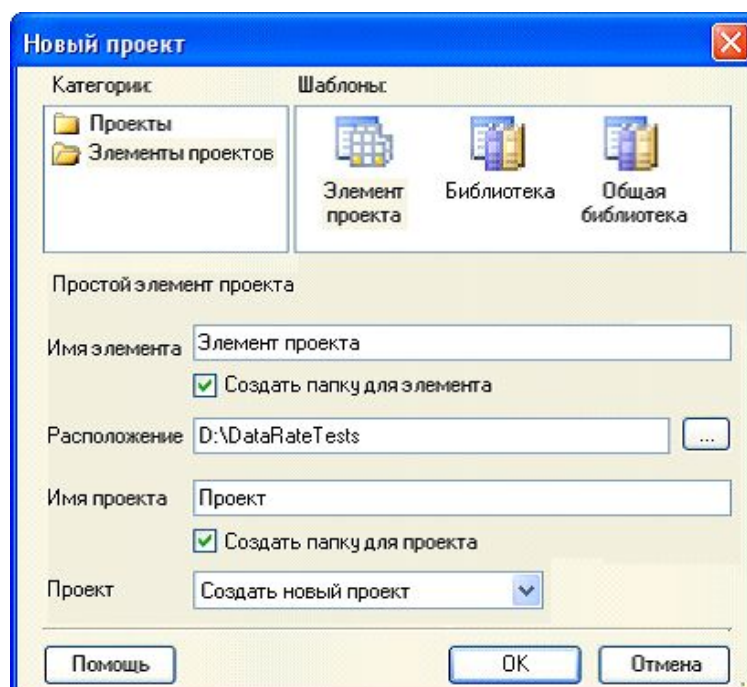
- 1 Нажать на кнопку  **Создать элемент проекта** на панели инструментов или в меню **Файл**



- 2 В появившемся окне **Новый проект** выбрать категорию **Элементы проекта** и шаблон **Элемент проекта**
- 3 Задать **Имя**
- 4 Для добавления элемента проекта к текущему проекту выбрать опцию **Добавить в проект**



Если установить опцию **Создать новый проект**, то в окне станут доступными для редактирования поля, задающие параметры создания нового проекта:



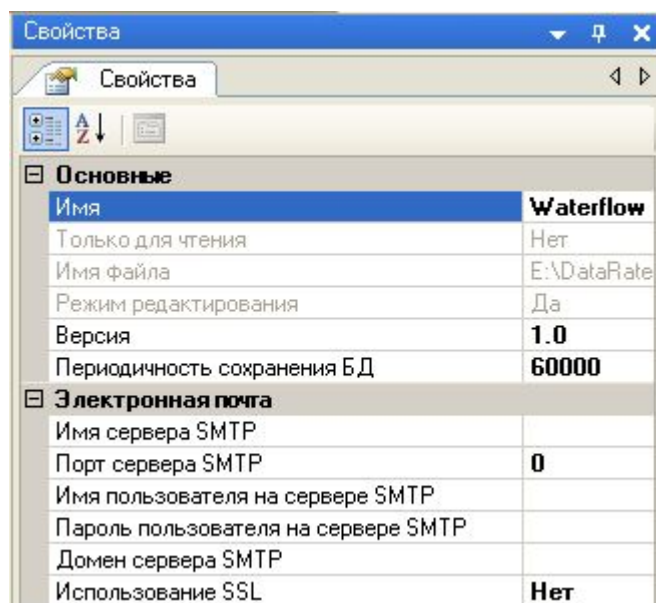
5 Нажать OK.

Если Вы выбрали опцию **Создать новый проект**, то после нажатия кнопки **OK** будет создан новый проект и один элемент проекта в его составе.

После создания элемента проекта Вы можете добавить в элемент проекта объекты, типы тегов и другие составляющие.

Описание элемента проекта хранится в файле с расширением **.kitem**.

2.2.1 Свойства элемента проекта



Секция Основные

- **Имя файла** – имя файла, в который будет сохранен элемент проекта
- **Имя** – имя элемента проекта
- **Только для чтения** – признак запрета редактирования элемента проекта – **Да**
- **Режим редактирования** – управление режимом редактирования для контейнера
- **Версия** – номер версии элемента проекта
- **Периодичность сохранения БД** – периодичность сохранения оперативной базы данных во время выполнения проекта

Секция Электронная почта

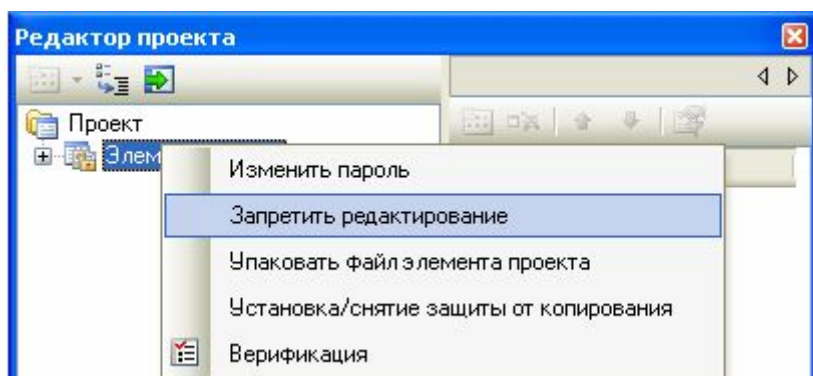
- **Имя сервера SMTP** – имя сервера SMTP для рассылки сообщений по электронной почте
- **Порт сервера SMTP** – порт сервера SMTP для рассылки сообщений (обычно **25**)
- **Имя пользователя на сервере SMTP** – имя пользователя на сервере SMTP для рассылки сообщений
- **Пароль пользователя на сервере SMTP** – пароль пользователя на сервере SMTP для рассылки сообщений
- **Домен сервера SMTP** – домен пользователя, от лица которого будут отправляться сообщения
- **Использование SSL** – признак использования защищенного соединения SSL для связи с сервером SMTP. Если задано **Да**, но почтовый сервер не поддерживает **SSL** соединение, то письмо не будет отправлено. Об отсутствии поддержки **SSL** пользователь может узнать, запустив проект на исполнение в Среде Разработки **DataRate**, при этом в окне **Отладка** в соответствующей вкладке будет отображено описание ошибки.

2.2.2 Защита элемента проекта

Подсистема защиты представляет собой совокупность средств, предназначенных для защиты от несанкционированного доступа к редактированию элементов проекта.

Запрет редактирования

Для того чтобы запретить редактирование элемента проекта, выполните в контекстном меню команду **Запретить редактирование**.

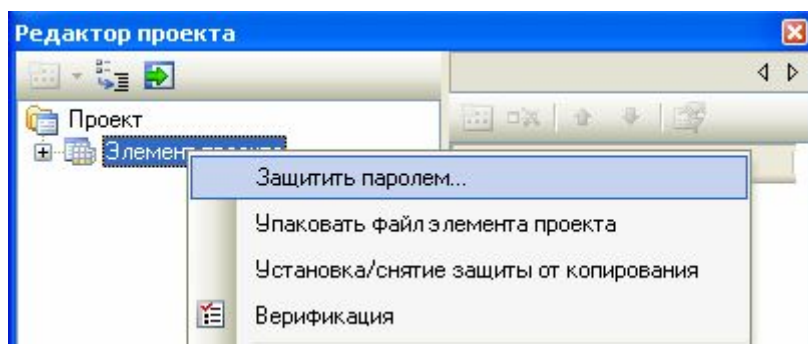


Защита паролем

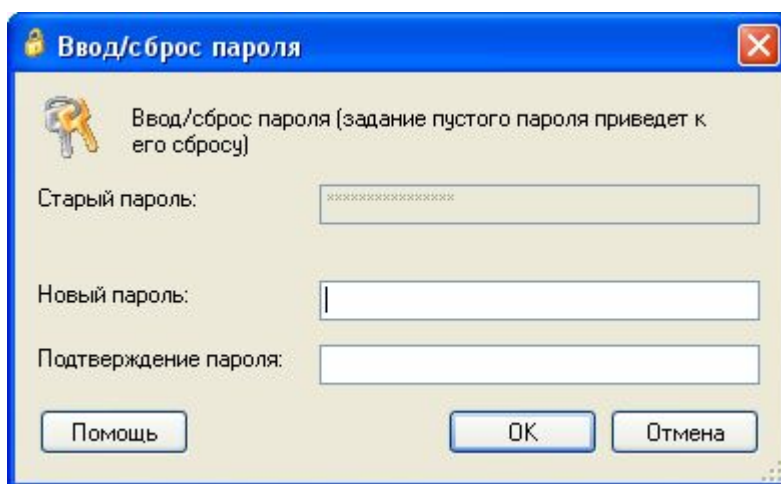
Установить пароль можно только на отдельные элементы проекта. При установке пароля на какой-либо элемент проекта этот пароль устанавливается на все его дочерние элементы, не защищенные другим паролем.

Для установки пароля элемента проекта следует:

- 1 В контекстном меню элемента проекта выполнить команду **Защитить паролем**

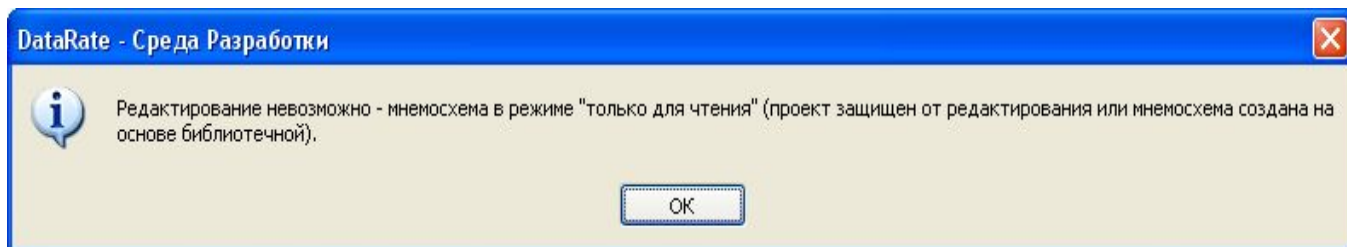


- 2 В появившемся окне **Ввод/сброс пароля** ввести **Новый пароль** и **Подтверждение пароля**
- 3 Нажать **ОК**



После выполнения рассмотренных действий защищенный проект или элемент проекта со всеми его дочерними элементами станет недоступным для редактирования.

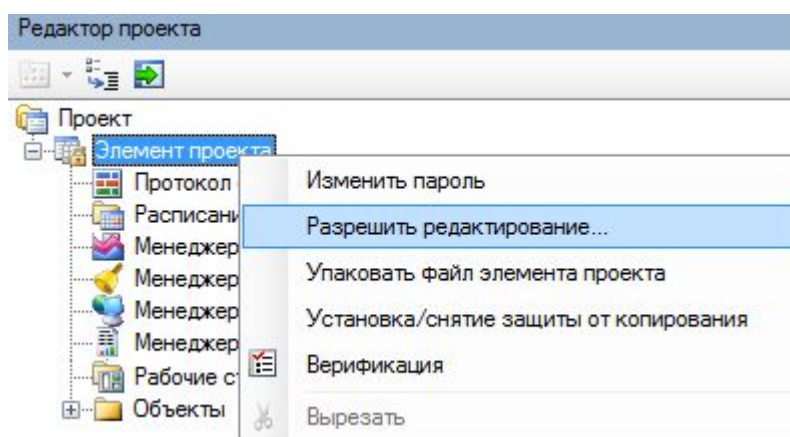
Если попытаться начать защищенный проект, то появится следующее сообщение:



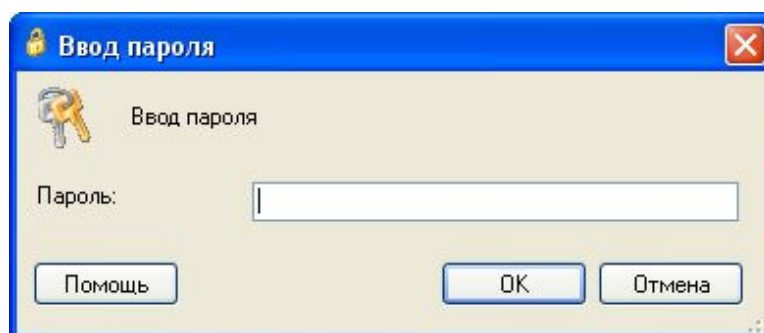
Редактирование защищенного паролем проекта

Для того чтобы получить возможность редактировать защищенный проект или элемента проекта следует:

- 1 Выполнить в контекстном меню команду **Разрешить редактирование**



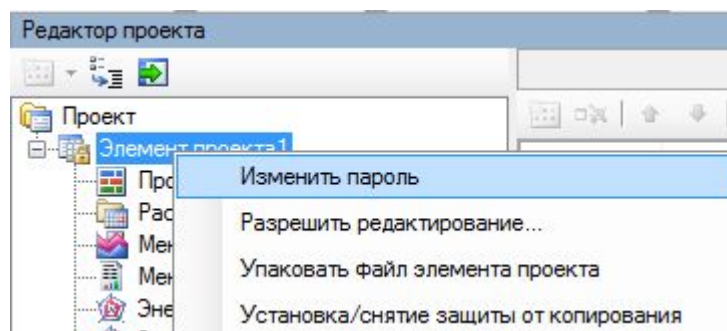
- 2 В появившемся окне **Ввод пароля** ввести пароль и нажать на кнопку **OK**



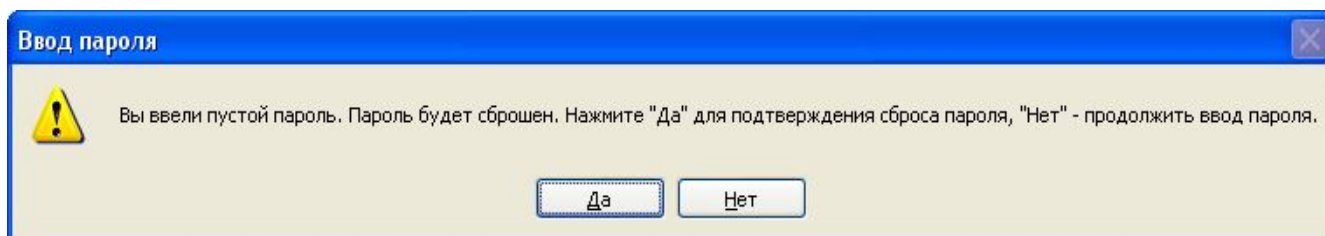
Изменение/сброс пароля

Для изменения пароля следует:

- 1 Выполнить в контекстном меню команду **Изменить пароль**



- 2 В появившемся окне **Ввод/сброс пароля** ввести **Старый пароль**, **Новый пароль** и **Подтверждение пароля**.
Для сброса пароля введите только **Старый пароль**, а поля **Новый пароль** и **Подтверждение пароля** оставьте пустыми. В этом случае появится следующее сообщение:



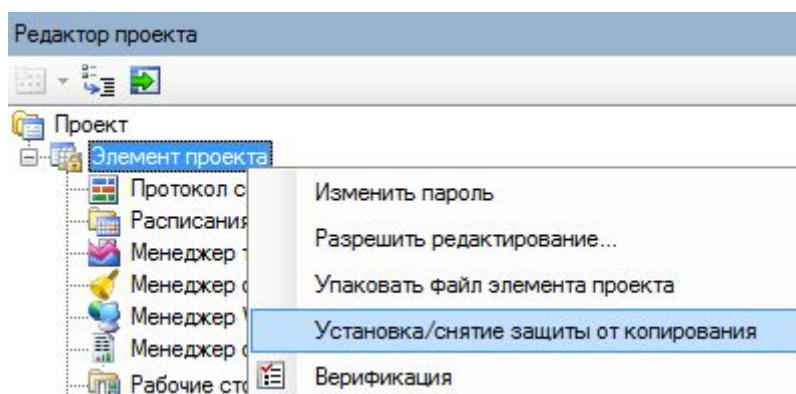
ВНИМАНИЕ!!!

Пароль изменяется или сбрасывается у всех дочерних элементов, защищенных таким же паролем или имеющих пустой пароль.

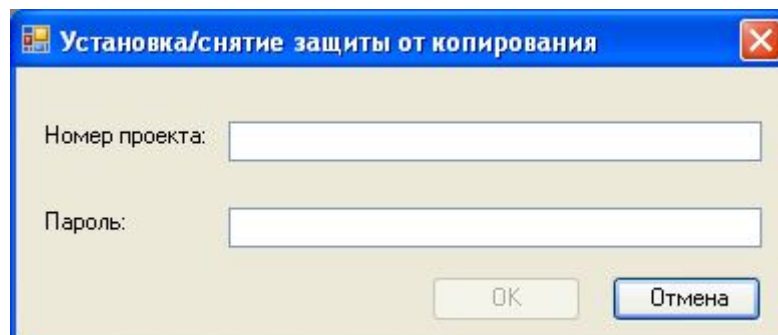
Установка/снятие защиты проекта от копирования

Для установки, снятия защиты от копирования следует:

- 1 Выполнить в контекстном меню команду **Установка/снятие защиты от копирования**



- 2 В появившемся окне **Установка/снятие защиты от копирования** ввести **Номер проекта** и **Пароль**.



После подтверждения ввода номера и пароля на экран будет выведен диалог с сообщением о результате операции. Если защита проекта от копирования установлена успешно, то будет выдано сообщение «**Защита установлена**». Если проект был защищён, и пользователь ввёл номер проекта и пароль, которые идентичны введенным ранее, то защита проекта от копирования снимается. При этом будет выдано сообщение «**Защита снята**».

В случае несовпадения пароля и номера проекта с введенными ранее, выдается сообщение «**Защита проекта от копирования не снята. Номер проекта или пароль неверный**».

ВНИМАНИЕ!!!

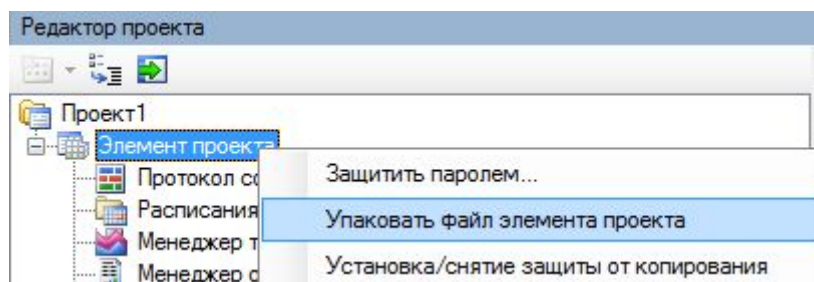
При защите проекта от копирования необходимо задавать одинаковые номера проектов и пароли для каждого элемента проекта.

2.2.3 Сжатие элемента проекта

Сжатие элемента проекта заключается в удалении из файла-хранилища (*.kitem) элемента проекта «недействительных» элементов объектной модели. Это такие элементы, которые были удалены из дерева редактора проекта. Учитывая тот факт, что процесс разработки является итерационным и подразумевает многократные операции по созданию, копированию, перемещению и удалению объектов, такая возможность является полезной, особенно когда структура проекта полностью определена.

Чтобы запустить процесс сжатия следует:

- 1 Выделить в редакторе проекта элемент проекта
- 2 В контекстном меню элемента проекта выбрать пункт **Упаковать файл элемента проекта**.



ВНИМАНИЕ!!!

Для элементов проекта, расположенных в библиотечной части проекта, в том числе и для библиотеки "Системная", сжатие не поддерживается.

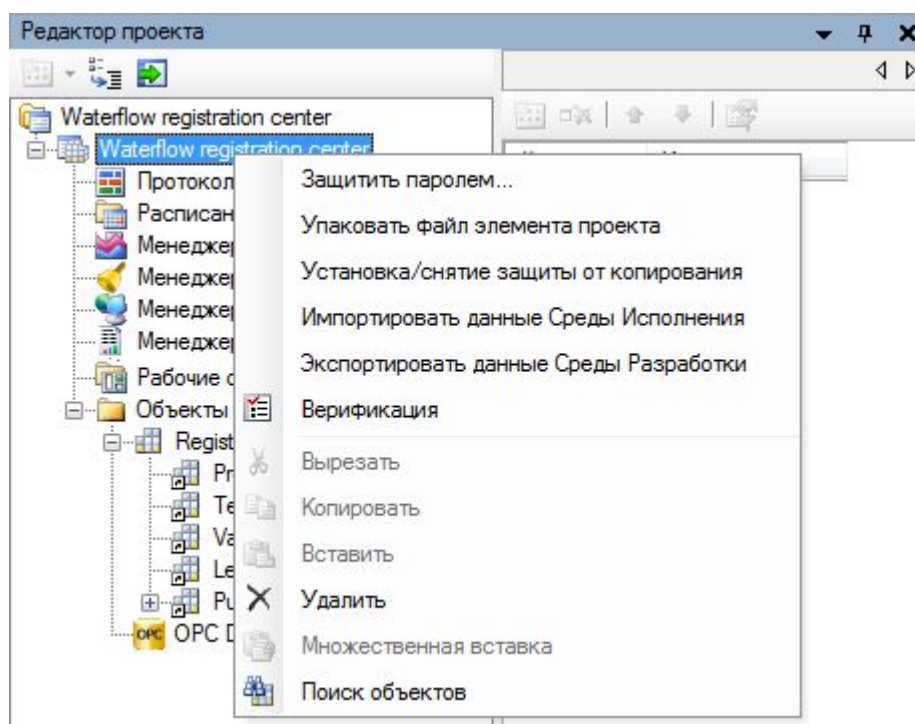
2.2.4 Импорт и экспорт базы данных тегов

Процесс импорта и экспорта атрибутов тегов описывается в разделе **Объекты.Теги.База данных тегов**.

Для элемента проекта импорт и экспорт осуществляются для всех объектов, существующих в рамках выбранного элемента проекта.

Импорт данных тегов для элемента проекта осуществляется путем выбора пункта меню **Импортировать данные Среды исполнения** из контекстного меню элемента проекта.

Экспорт данных тегов для элемента проекта осуществляется путем выбора пункта меню **Экспортировать данные Среды разработки** из контекстного меню элемента проекта.



2.3 Библиотеки

Библиотека – это набор повторно используемых элементов.

В проект могут быть включены несколько библиотек.

Любой элемент проекта имеет библиотечную часть.

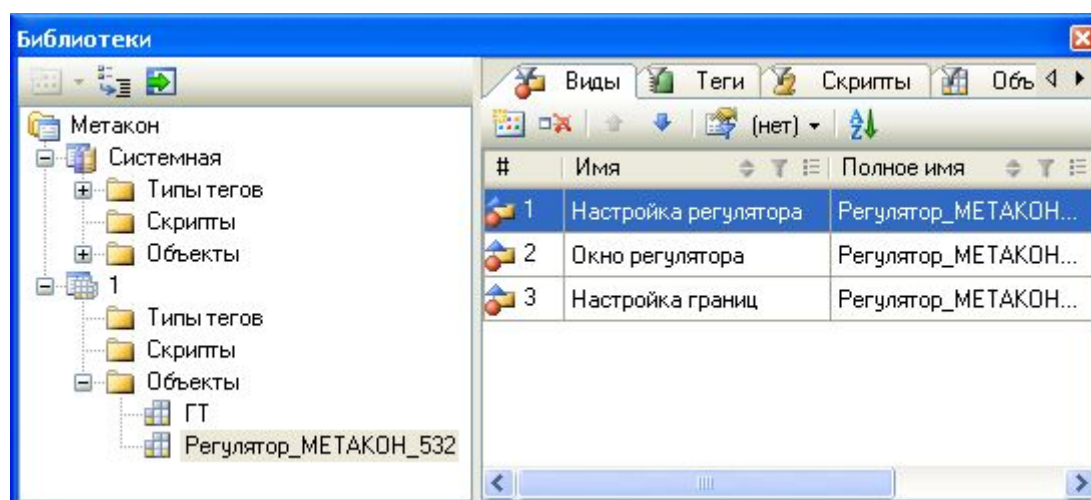
Структурно библиотеки организованы в виде дерева библиотек.

2.3.1 Редактор библиотек

Редактор библиотек имеет дизайн, аналогичный дизайну редактора проекта.

Окно редактора библиотек включает дерево библиотек и список дочерних элементов текущего узла дерева.

В дереве библиотек отображаются все библиотеки проекта, а также все библиотечные части элементов проекта.



2.3.2 Библиотека объектов

Библиотека объектов - это шаблоны объектов.

Объекты в библиотеке организованы в виде дерева.

В библиотеку можно помещать объекты из проекта (либо перетаскивая мышью, либо через выполнив команду **Поместить в библиотеку** в контекстном меню объекта), а также создавать объекты непосредственно в библиотеке.

Возможно, создавать в библиотеке объекты на основе объектов, которые в свою очередь находятся в библиотеке.

Создание объектов на основе объектов библиотеки

Создавать объекты на основе библиотечных можно двумя способами:

- **Копирование объектов.** Копировать объект можно, просто перетаскивая его из библиотеки в проект с помощью мыши, удерживая клавишу **Ctrl**, или через буфер с помощью стандартных операций «**Копировать/Вставить**»
- **Копирование объектов с сохранением связи с библиотекой.** В этом случае объект из библиотеки перетаскивается в проект с помощью мыши, удерживая клавишу **Alt**, или создается с помощью контекстного меню объекта, выбирая команду **Создать/Объект** и закладку **Из библиотеки** в появившемся окне **Добавление нового элемента**. Копирование с сохранением связи означает, что при изменении библиотечного объекта (добавление, удаление, изменение тегов, видов, скриптов) будет изменен и объект, который создан на основе библиотечного объекта.

ВНИМАНИЕ!!!

Для того, чтобы изменения библиотечного объекта стали доступны в порожденном из библиотеки объекте, необходимо в его контекстном меню выполнить команду **Обновить**.

Если объект создан на основе библиотечного объекта и после этого в библиотечный объект был добавлен дочерний объект, то по команде **Обновить** добавленные дочерние объекты библиотечного объекта не появятся в порожденном объекте.

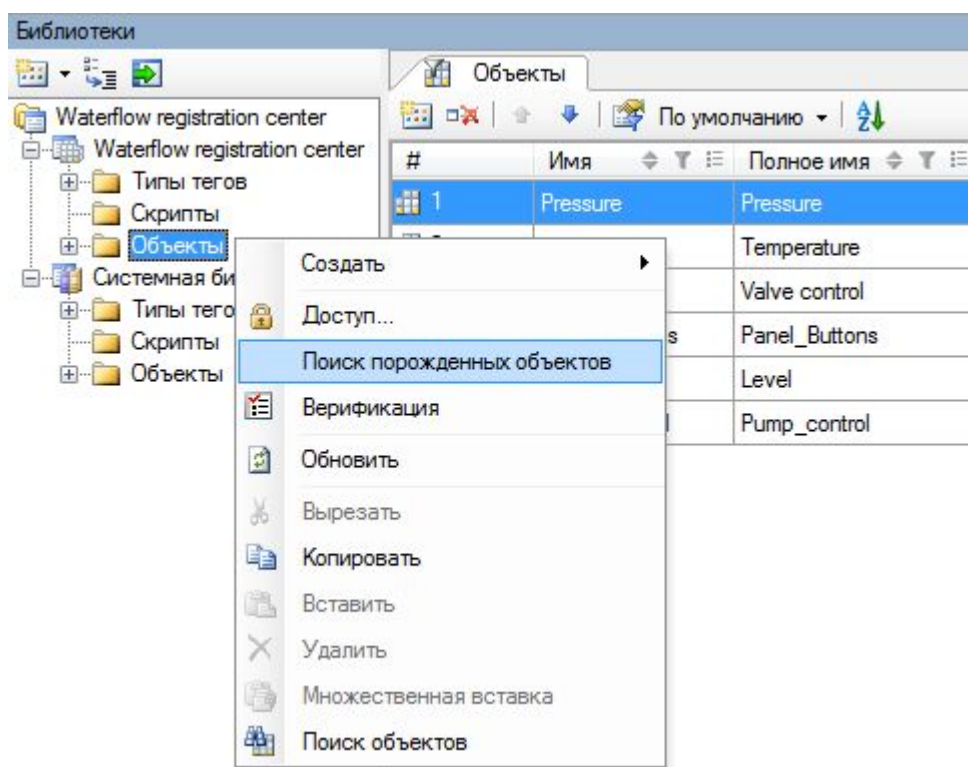
Индивидуальная настройка объекта

Если вы хотите индивидуально настроить тег (вид, скрипт) объекта, который был порожден от библиотечного, и этот тег (вид, скрипт) присутствует в библиотечном объекте, то свойство **Библиотечный** тега (вида, скрипта) настраиваемого объекта нужно установить в **Нет**.

В случае установки свойства **Библиотечный** в **Да**, и при этом объект не имеет соответствующей библиотечной части, будет выдано сообщение «**Элемент не может находиться в библиотеке**».

Поиск порожденных объектов

Для библиотечного объекта можно найти другие объекты проекта, которые от него порождены. Для этого выберите объект в библиотеке и в его контекстном меню вызовите команду **Поиск порожденных объектов**.



В окне поиска можно задать параметры поиска, как описано в разделе **Поиск объектов**.

2.3.3 Библиотека типов тегов

Тип тега - это коллекция атрибутов и скриптов.

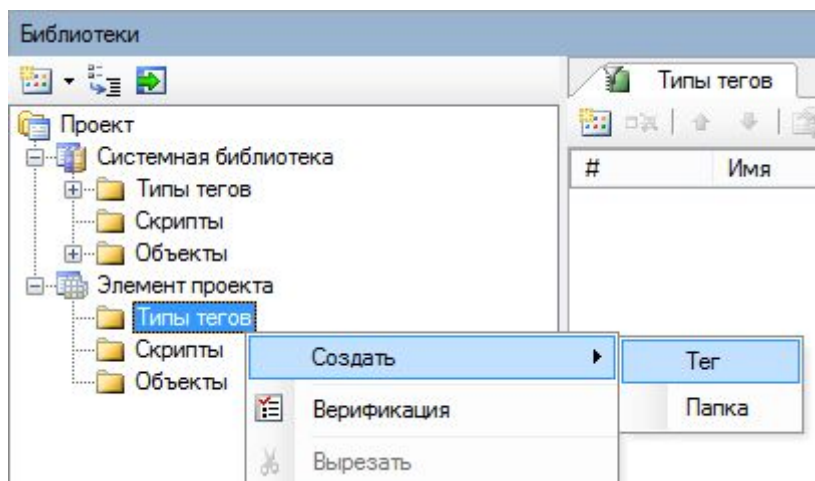
ВНИМАНИЕ!!!

Тип тега является основой для создания тега.

Тип тега должен быть определен в библиотеке типов тегов.

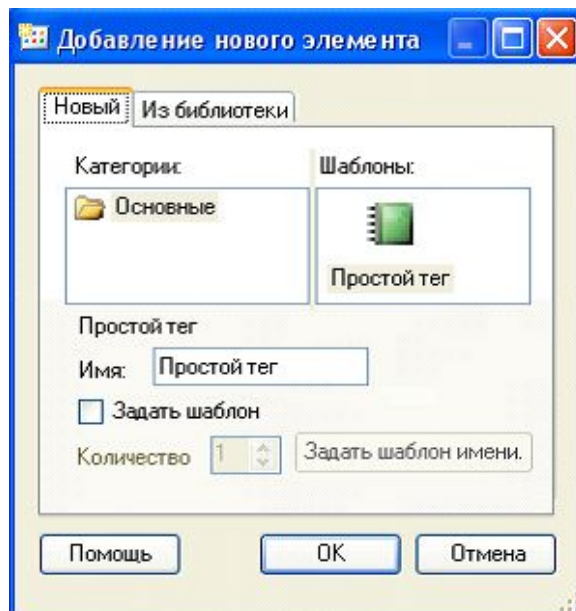
Создание типа тега

Для создания типа тега выберите в **дереве библиотеки** папку **Типы тегов** и выполните в контекстном меню команду **Создать/Тег**.

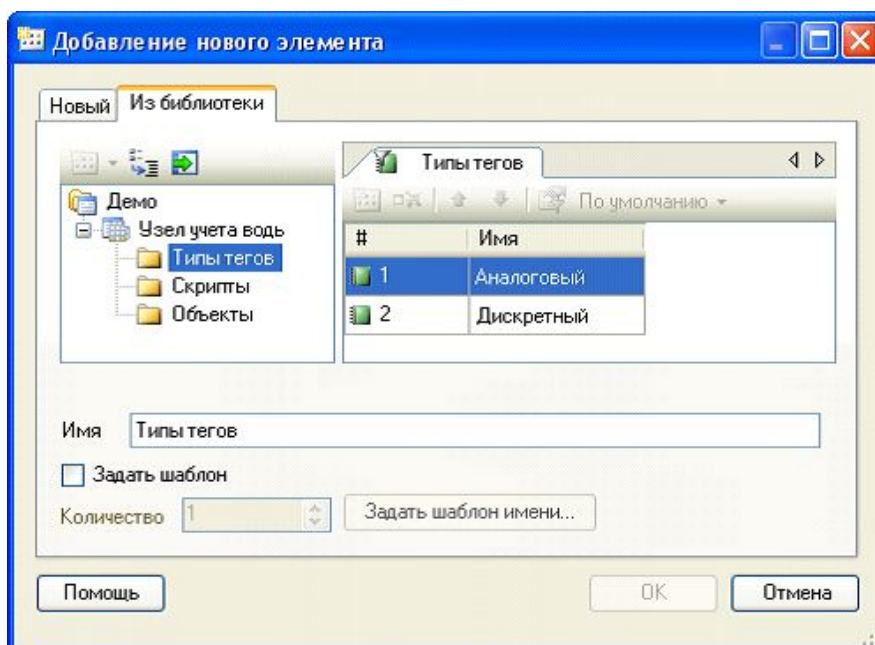


В появившемся окне **Добавление нового элемента** следует:

1. Выбрать категорию **Основные**
2. Выбрать шаблон **Простой тег**
3. Задать **Имя** типа тега



Для создания типа тега на основе другого типа перейдите на вкладку **Из библиотеки**, выберите тип тега и введите **Имя** нового типа тега.



Атрибуты

Любой тег включает набор атрибутов. Количество атрибутов произвольно. Значения атрибутов тегов можно читать и устанавливать в скриптах и видах с помощью механизмов анимации, а также использовать в трендах.

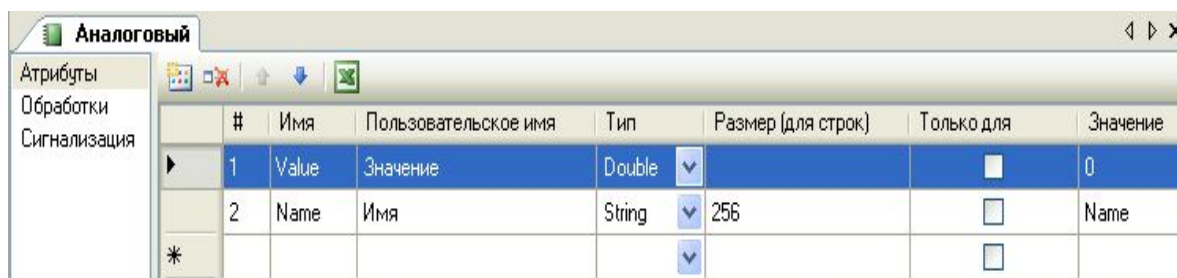
ВНИМАНИЕ!!!

При изменении значений атрибутов в одном теге новые значения копируются в атрибуты связанного с ним тега.

Редактирование атрибутов

Вызвать редактор атрибутов можно любым из следующих способов:


- Использовать окно **Свойства** типа тега
- Выполнить команду **Задачи/Редактировать** в контекстном меню типа тега
- Выполнить двойной щелчок на типе тега.



Характеристики атрибутов:

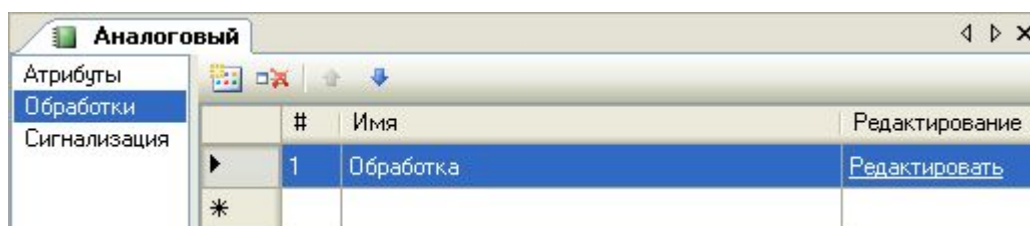
- **Имя** – уникальное имя атрибута. Используется в скриптах и других обработках. Например, **Value** – текущее значение (отображается при выполнении приложения)
- **Пользовательское имя** – дружественное имя
- **Тип** – тип значения атрибута (целый, вещественный, строка и т.д.)
- **Только для чтения** – установленная галочка запрещает запись в этот атрибут
- **Значение** – начальное значение атрибута

Используйте **для добавления** атрибутов кнопку , **для удаления** – кнопку .

Кнопка  позволяет экспортировать список атрибутов в формат **CSV**.

Обработки

При выборе элемента **Обработки** в списке, расположенном в левой части окна, на экране отображается список обработок, predeterminedенных для данного типа тега.




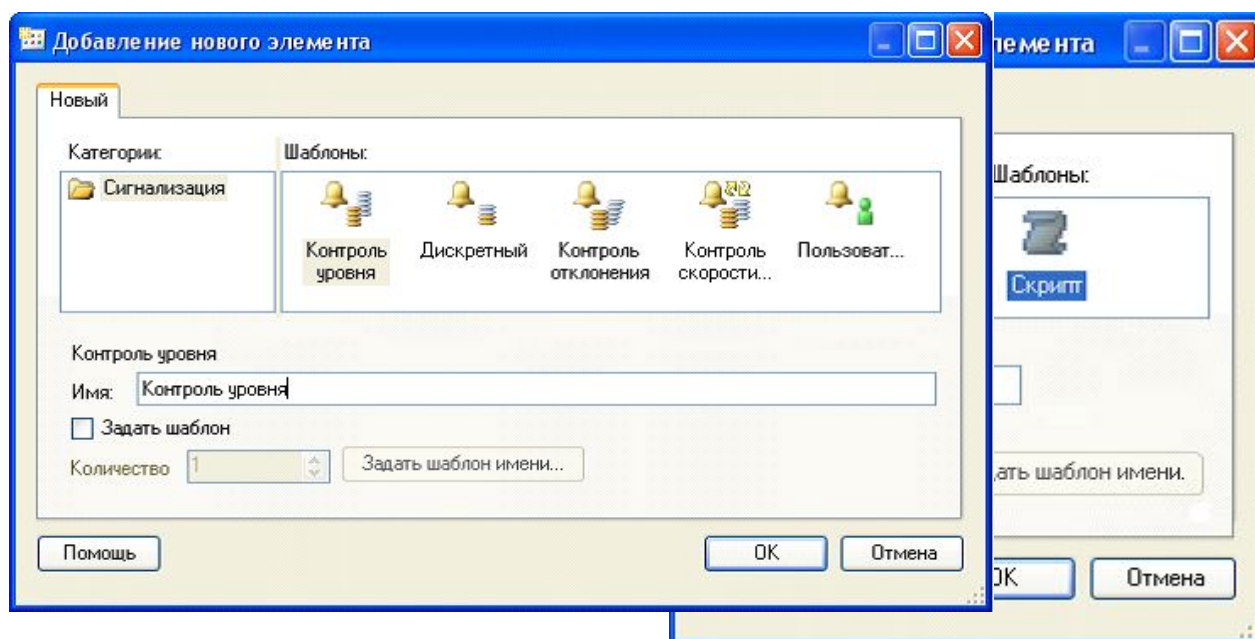
Обработка – это специальный скрипт, который активизируется всякий раз, когда происходит изменение значений атрибутов тега. Щелчок на ссылке **Редактировать** открывает окно редактора скриптов для определения программного кода скрипта.

Сигнализации

Каждый тип тега имеет свой набор доступных **Сигнализаций** (аварийных состояний).

Для добавления сигнализации к типу тега следует:

1. Выбрать элемент **Сигнализация** в списке редактора атрибутов тега
2. Нажать кнопку 
3. В появившемся окне **Добавление нового элемента** выбрать нужную сигнализацию.



2.3.4 Библиотека скриптов

Библиотека скриптов - это набор скриптов (шаблонов обработок), предназначенных для повторного использования в объектах проекта.

Создание

Для создания библиотечного скрипта следует:

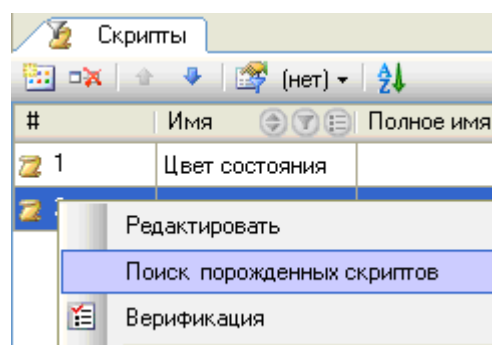
1. Выбрать в дереве библиотеки папку **Скрипты** и выполнить в контекстном меню команду **Создать/Скрипт** или в списке дочерних элементов нажать на кнопку **Добавить новый элемент**
2. В появившемся окне **Добавление нового элемента** выбрать категорию **Основные** и шаблон **Скрипт**
3. Задать **Имя** типа и нажать **OK**

Для создания библиотечного скрипта на основе скрипта из библиотеки перейдите на вкладку **Из библиотеки**, выберите **скрипт** и введите **Имя** нового скрипта. Создавать скрипт на основе библиотечного можно при помощи операции **Drag&Drop**.

Поиск порожденных скриптов

Для поиска скриптов порожденных от библиотечного скрипта выберите в его контекстном меню пункт **Поиск порожденных скриптов**.

В окне поиска можно задать параметры поиска, как описано в разделе **Поиск объектов**.



Удаление привязок аргументов

Если скрипт является библиотечным (помечен иконкой с ссылкой), то менять привязки его аргументов нельзя. Привязки аргументов скрипта можно удалить в случае, когда скрипт не является библиотечным или если скопировать скрипт с ссылкой в библиотеку (скрипт станет небиблиотечным).

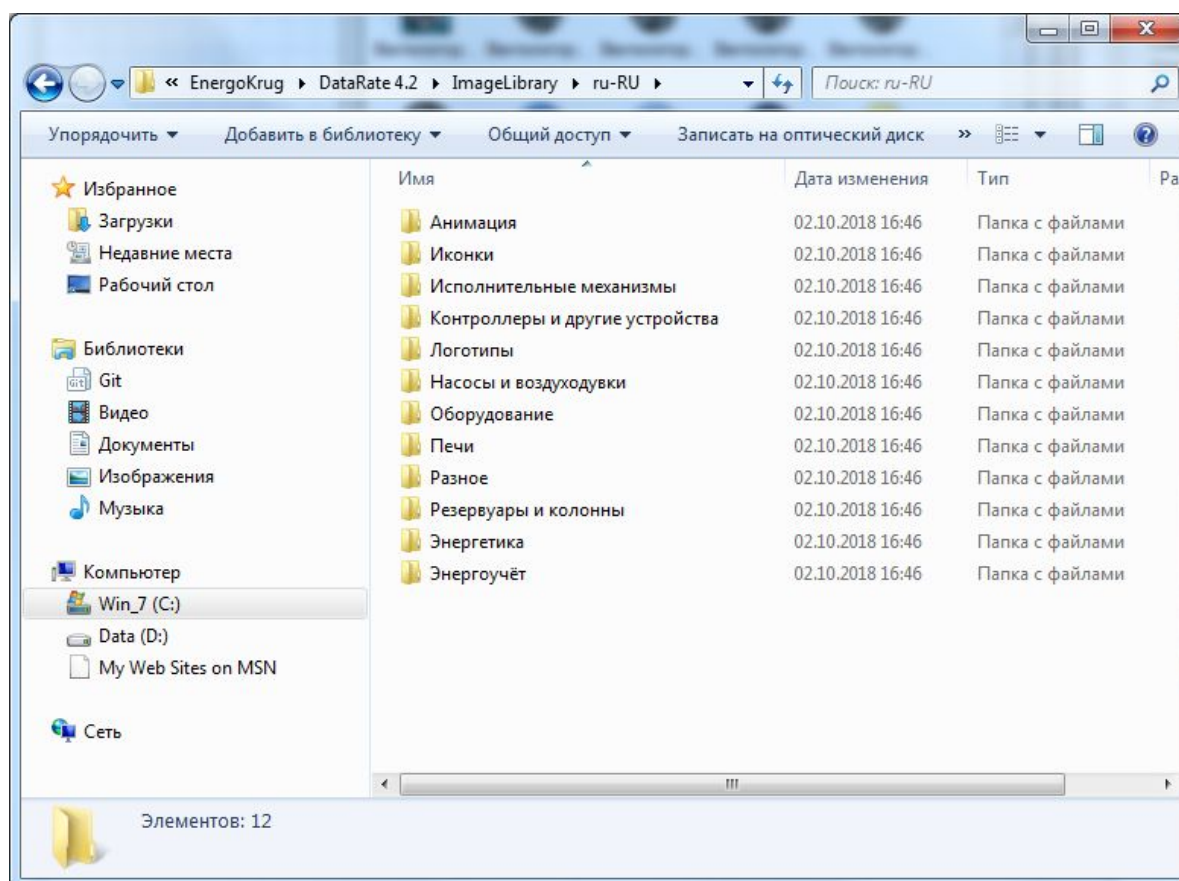
Для удаления привязок откройте окно **Выбор привязки**. Окно выбора привязки не доступно, если аргументы находятся в библиотечном скрипте (не путать со скриптом в библиотеке).

2.3.5 Графическая библиотека

Графическая библиотека (Image Library) – это множество изображений, используемых в проектах *DataRate* для анимации и отображения элементов технологических процессов (пример использования изображений из библиотеки приведен в разделе 3.1.3.9 «Рисунок»).

После инсталляции *DataRate* библиотека изображений расположена:

**<системный диск>:\Program Data\Energokrug\DataRate 4.2\
ImageLibrary\ru-RU**

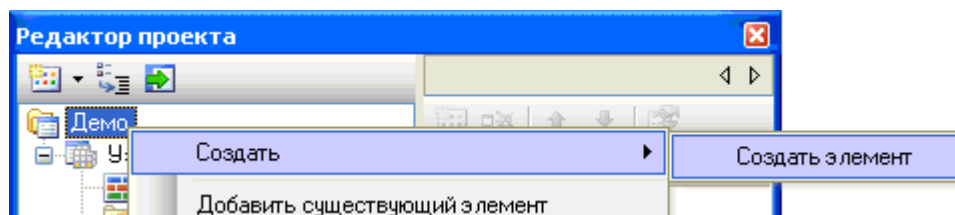


2.3.6 Управление библиотеками

Создание

Для создания в текущем проекте библиотеки следует:

- 1 В контекстном меню проекта выбрать пункт **Создать\Создать элемент**



- 2 В появившемся окне **Новый проект** выбрать шаблон **Библиотека** или **Общая библиотека**

Общая библиотека – это библиотека, предназначенная для использования во множестве проектов **DataRate**.

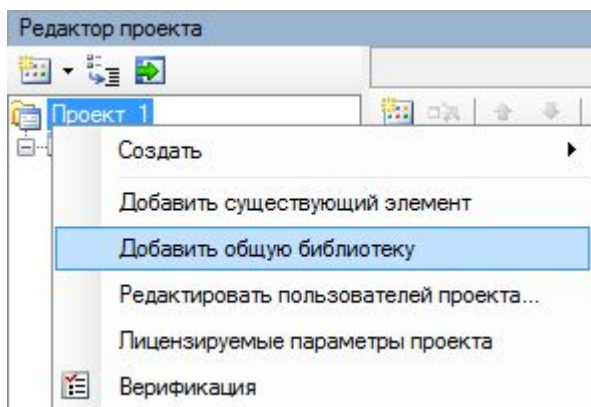
Располагаются общие библиотеки на системном диске в папке:

[Documents and Settings\All Users\Application Data\Energokrug\DataRate\SharedLibraries](#)

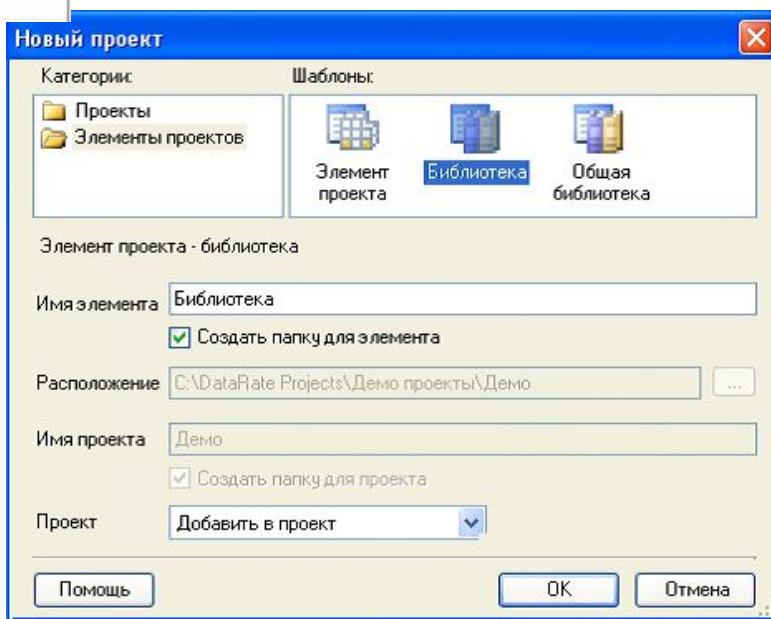
Добавление общей библиотеки в проект

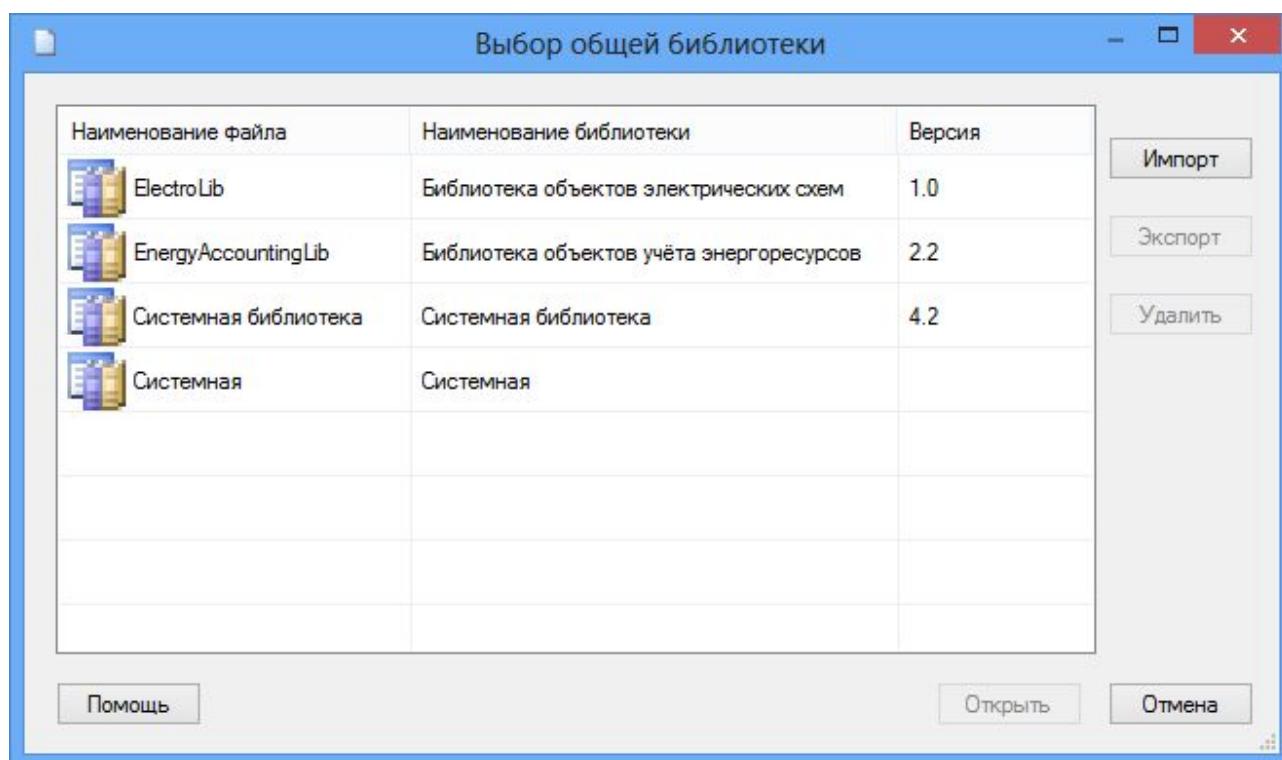
Для добавления общей библиотеки в проект следует:

- 1 Вызвать команду контекстного меню проекта **Добавить общую библиотеку**

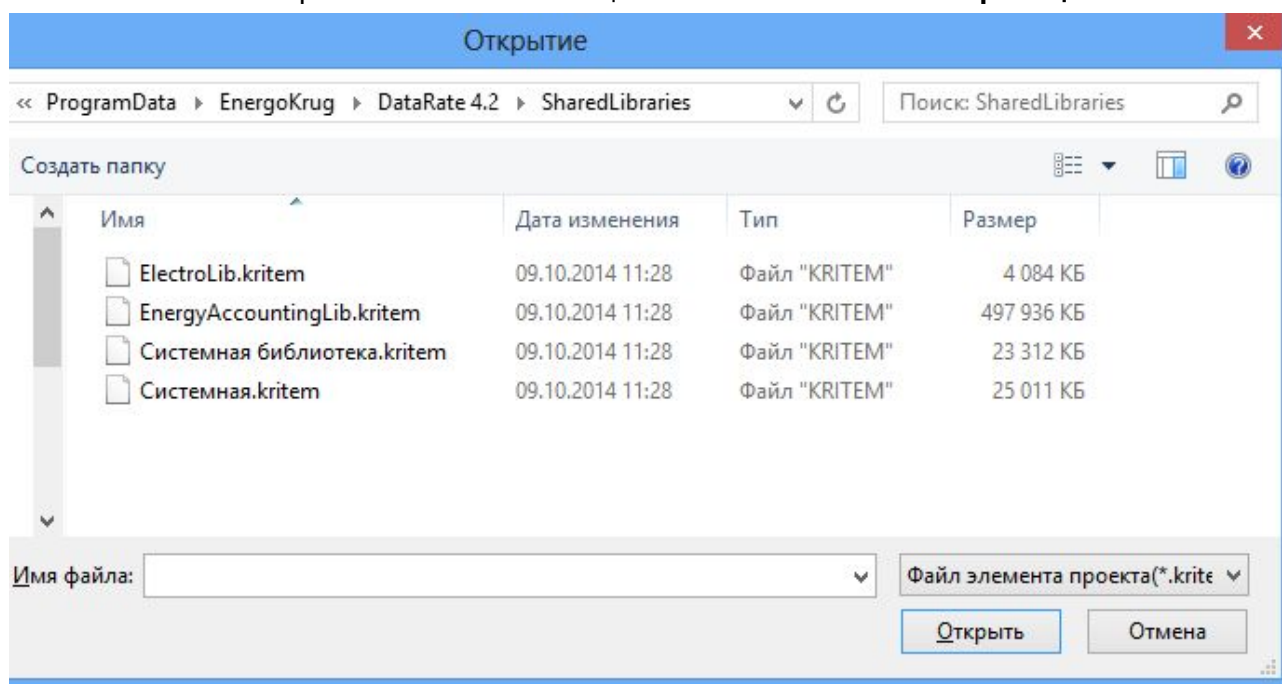


- 2 В появившемся окне **Выбор библиотеки** для работы использовать кнопки **Импорт**, **Экспорт**, **Удалить**:





- Кнопка **Импорт** позволяет импортировать файлы библиотек *DataRate*, расположенные вне папки общих библиотек. По нажатию кнопки **Импорт** отображается окно выбора файла библиотеки, в котором следует выбрать файл библиотеки и нажать **Открыть**. Выбранный файл будет помещен в папку общих библиотек и будет отображаться в списке общих библиотек в окне **Выбор общей библиотеки**



- Кнопка **Экспорт** позволяет экспортировать выбранную библиотеку из папки общих библиотек *DataRate* в заданный каталог

- Кнопка **Удалить** позволяет удалить выбранную библиотеку из папки общих библиотек.

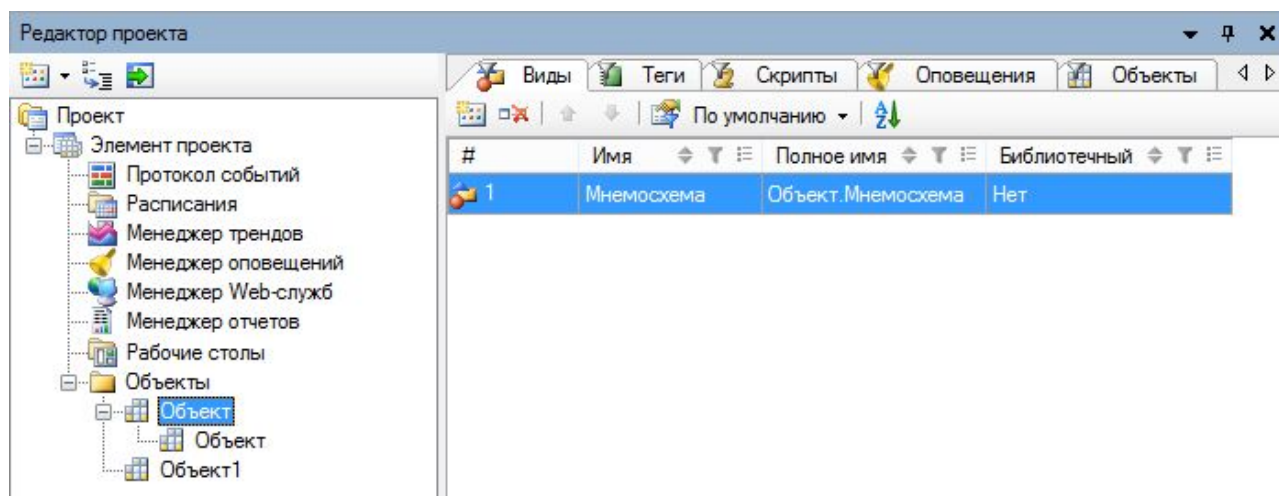
Пользователь может выбирать библиотеки по версии. Номер версии задается разработчиком библиотеки (когда с библиотеки снята защита от редактирования).

2.4 Объекты

Объект – это основной элемент любого приложения, создаваемого в **DataRate**.

Объект содержит набор видов, тегов, скриптов и оповещений.

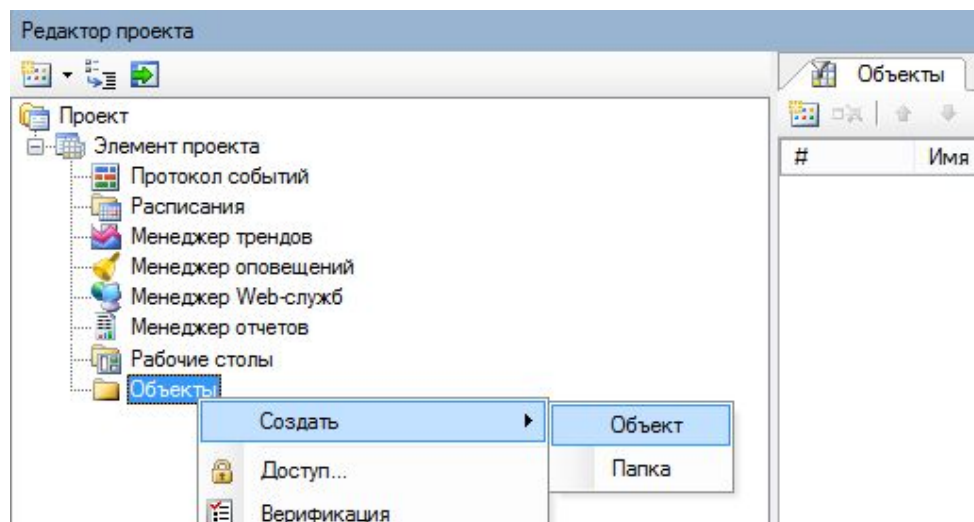
Объекты могут помещаться друг в друга, формируя при этом дерево.



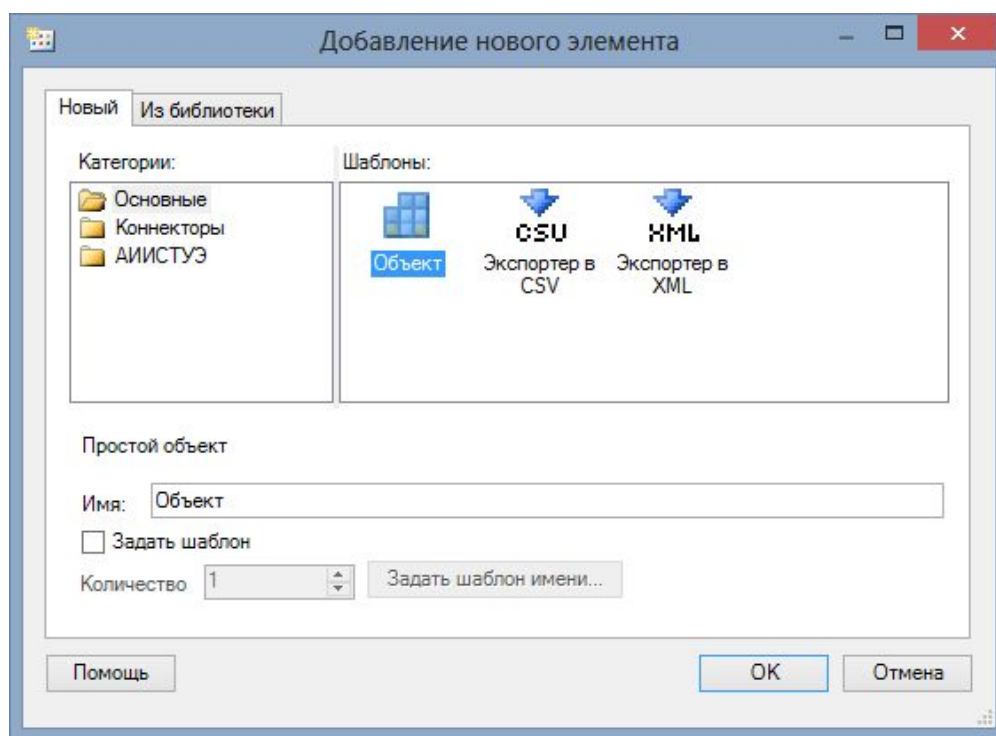
Создание

Для создания объекта следует:

- 1 В контекстном меню **Объекты** выполнить команду **Создать/Объект**

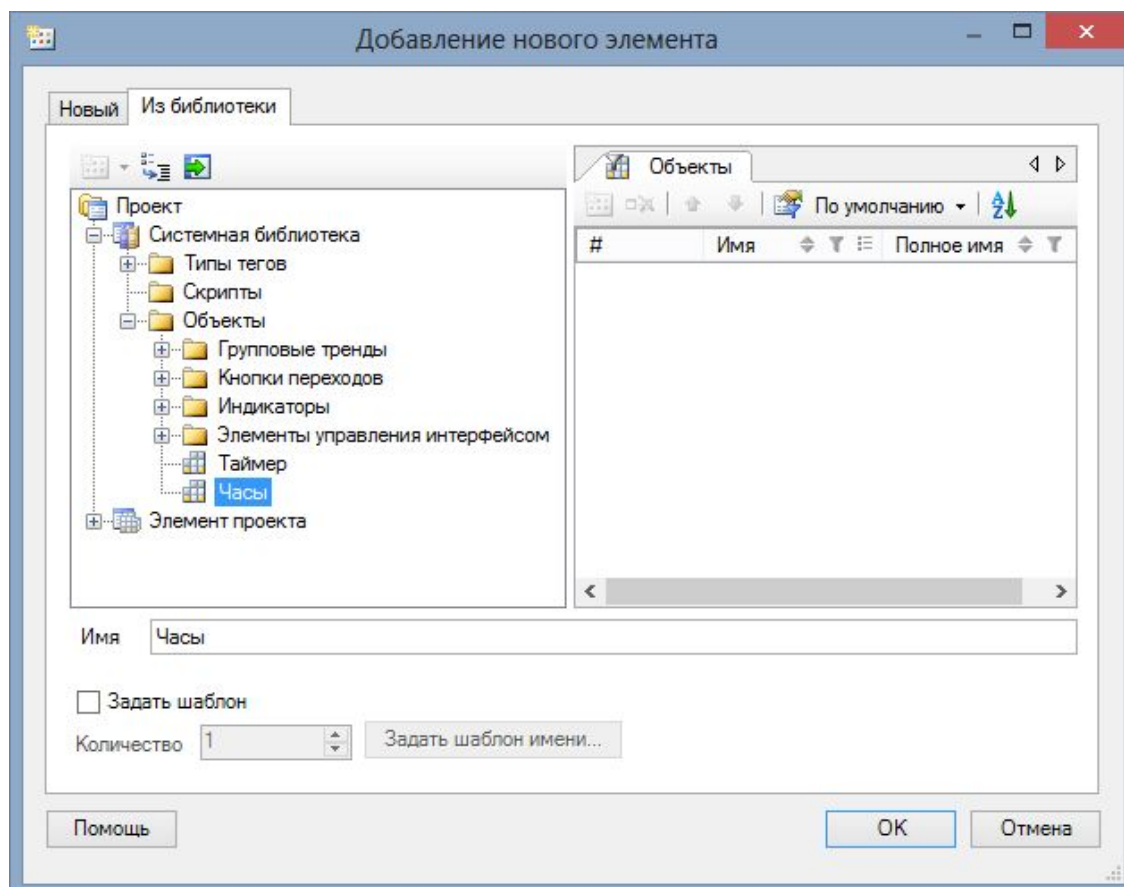


- 2 В появившемся окне **Добавление нового элемента** выбрать категорию **Основные** и нужный **Шаблон** объекта
- 3 Задать **Имя** объекта и нажать **ОК**



Для создания объекта из библиотеки следует:

- 1 Перейти на вкладку **Из библиотеки**
- 2 Выбрать библиотечный объект
- 3 Ввести **Имя** нового объекта.



Создать объект из библиотеки можно перетаскиванием его из библиотечного дерева в основное дерево проекта.

ВНИМАНИЕ!!!

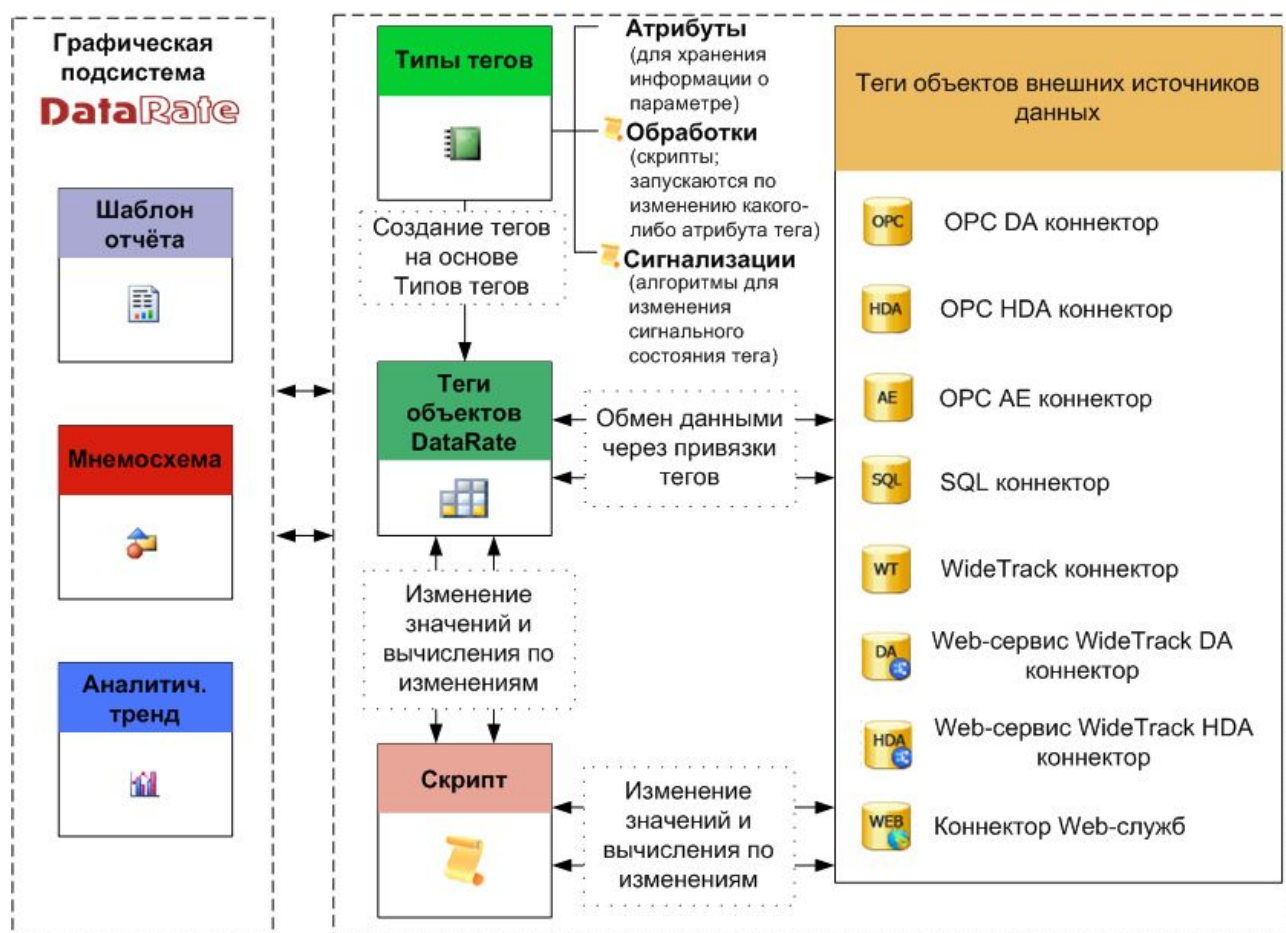
При переименовании вида объекта ссылки на объект, содержащий этот вид, становятся недействительными.

Запускайте верификацию проекта каждый раз, когда изменяете имена его компонентов.

Для создания нескольких однотипных объектов воспользуйтесь флажком **Задать шаблон**. Введите **Количество** объектов и задайте шаблон имени создаваемых объектов. Задать шаблон имени можно нажав на кнопку **Задать шаблон имени** (подробнее смотрите в разделе «Формирование шаблона имени»).

2.4.1 Теги

Теги – это входы, выходы объекта.



ВНИМАНИЕ!!!

Основой для создания тега является тип тега (класс тегов), определенный в библиотеке типов тегов.

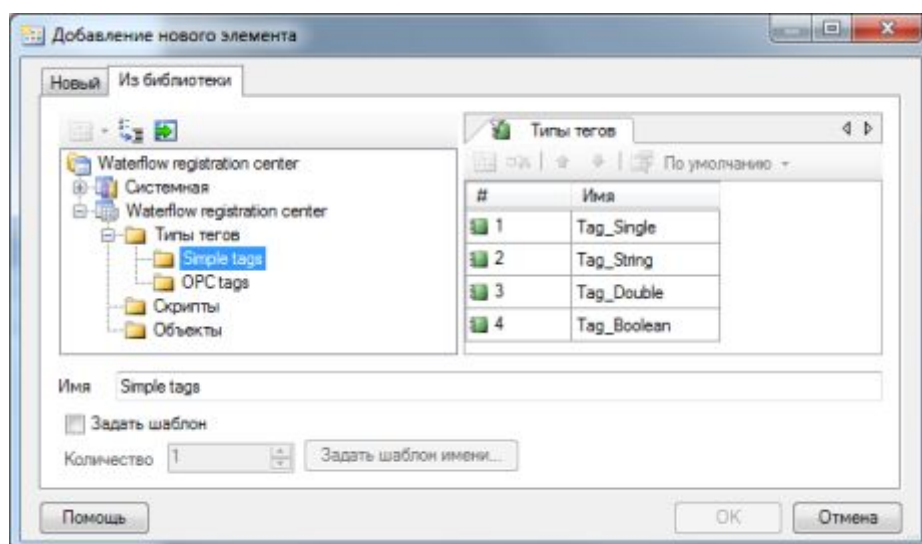
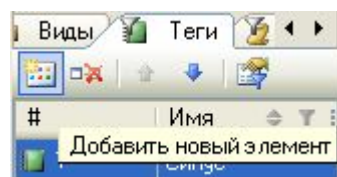
Тип тега – это коллекция атрибутов и «обработок» атрибутов (скриптов). Количество атрибутов произвольно.

Атрибуты тега – это свойства объекта, определяемые типом тега. Значения атрибутов тегов можно читать и устанавливать в скриптах и видах с помощью механизмов анимации, а также использовать в трендах. При изменении значений атрибутов в одном теге новые значения копируются в атрибуты связанного с ним тега.

Создание

Для создания тега следует:

- 1 В контекстном меню объекта выполнить команду **Создать/Тег** или на вкладке **Теги** для выбранного объекта в редакторе проекта нажать на кнопку **Добавить новый элемент**
- 2 В появившемся окне **Добавление нового элемента** открыть вкладку **Из библиотеки**, выбрать тип тега и ввести **Имя**



Создать тег можно перетаскиванием его типа из библиотечного дерева в объект дерева проекта.

Индивидуальная настройка тега объекта

Если вы хотите индивидуально настроить тег объекта, который был порожден от библиотечного, и этот тег присутствует в библиотечном объекте, то свойство **Библиотечный** тега настраиваемого объекта нужно установить в **Нет**.

Если объект не имеет соответствующей библиотечной части и свойство **Библиотечный** установлено в **ДА**, то будет выдано сообщение **"Элемент не может находиться в библиотеке"**.

2.4.1.1 Связи тегов

Теги одного объекта можно связать с тегами другого объекта. Связь тегов означает, что тег приемника (**владельца связи**) получает значение тега источника.

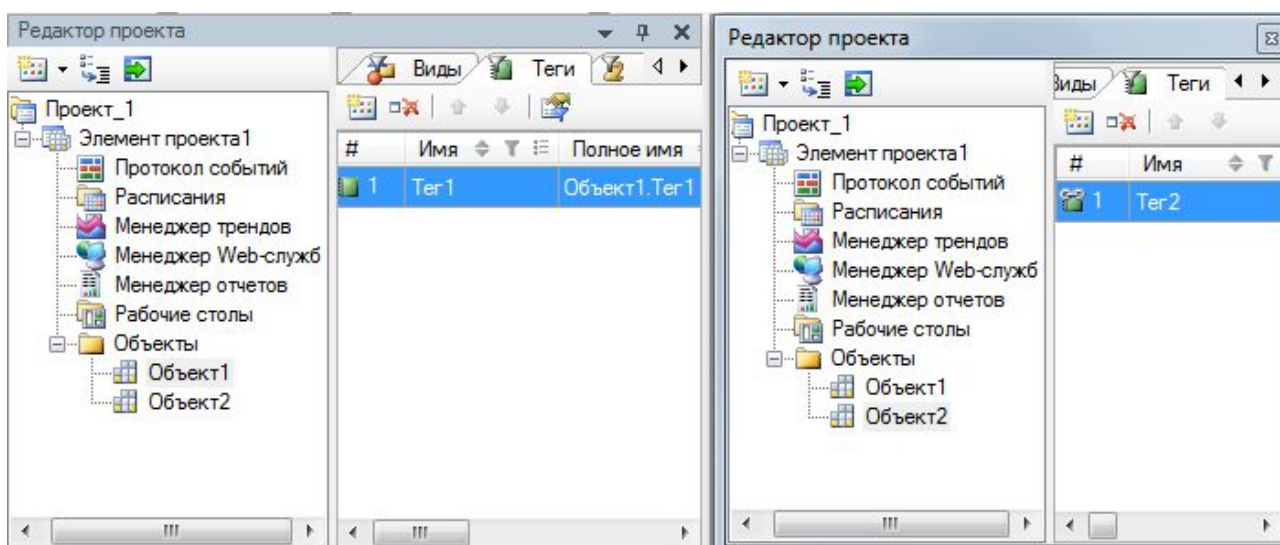
Как связать теги?

Например, необходимо связать тег **ТЕМП_1** (источник данных – объект **База данных**) с тегом **Вход** (приемник данных – объект **Температура**).

Связать теги можно следующими способами:

- **Способ 1:**

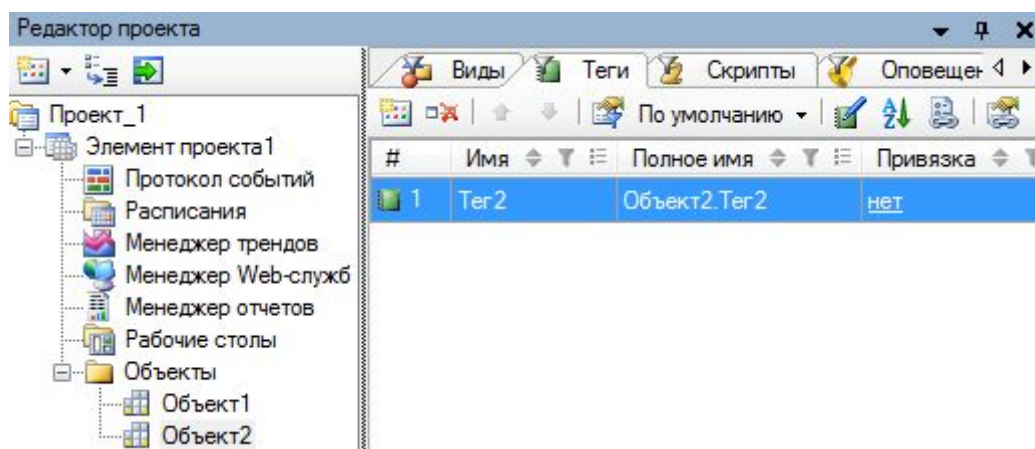
- 1 Открыть два окна **Редактор проекта** и сделать оба связываемых тега видимыми
- 2 «Перетащить» с нажатой клавишей **Alt** тег **Тег1** на тег **Тег2**.



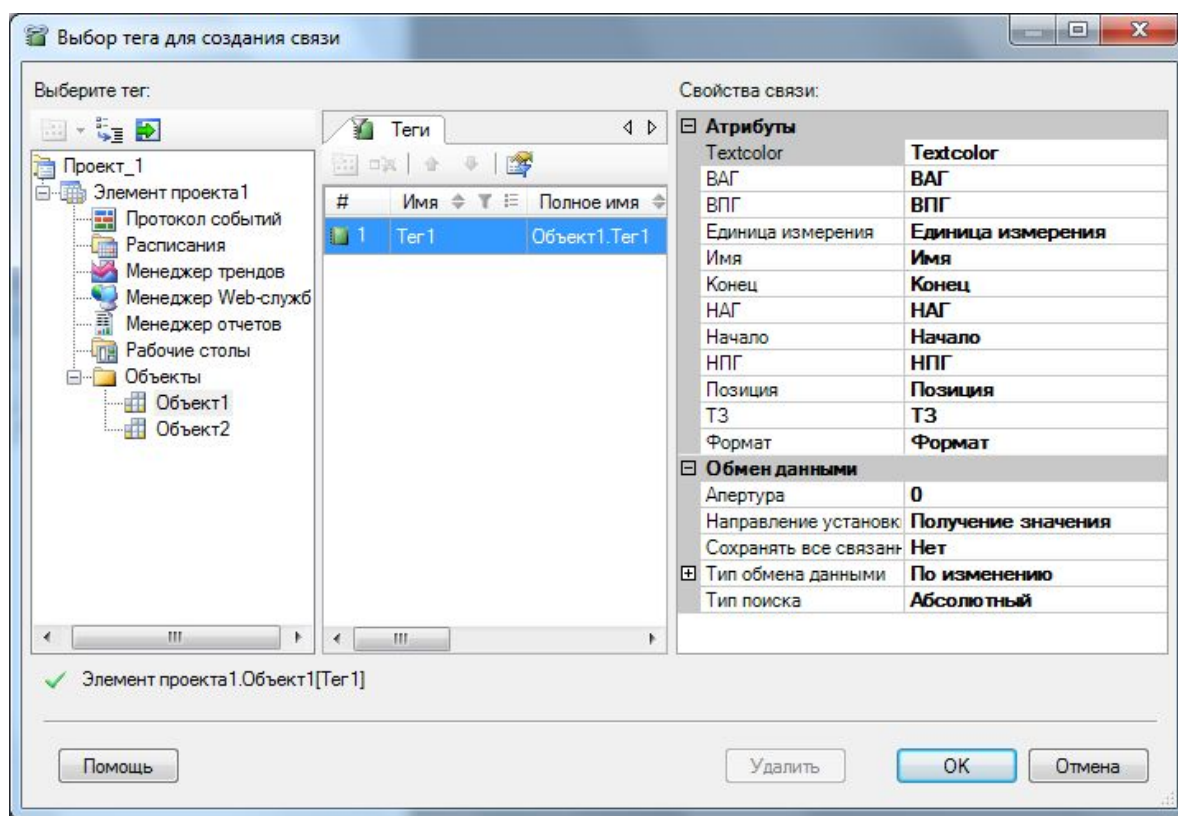
- **Способ 2:**

настроить у тега **Тег2** (приемника данных, владельца связи) привязку. Для этого следует:

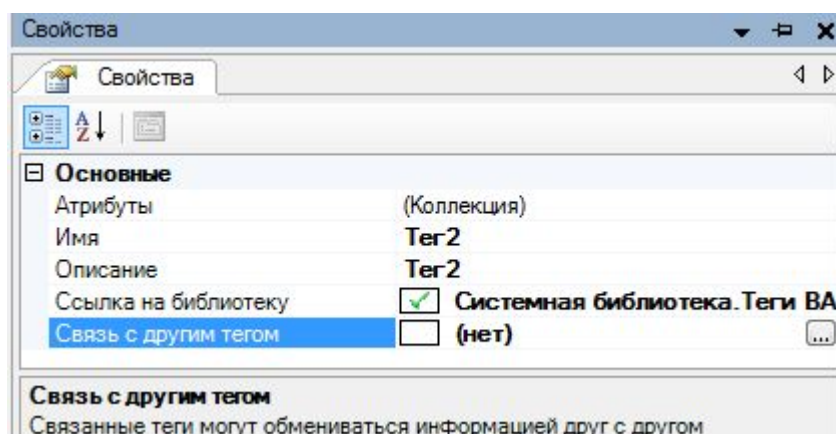
- 1 Выбрать в Редакторе проекта объект **Объект2** и открыть вкладку **Теги**
- 2 Выделить тег **Тег2**
- 3 В колонке **Привязка** нажать на нет



- 4 В появившемся окне **Выбор тега для создания связи** выбрать тег **Тег1** объекта **Объект1**
- 5 Если необходимо, настроить свойства связи **Атрибуты** и **Обмен данными**



- **Способ 3:**
настроить у тега **Тег2** (приемника данных, владельца связи) свойство **Связь с другим тегом**. Для этого следует:
 - 1 Выделить тег и в окне **Свойства** выделить строку **Связь с другим тегом**
 - 2 Нажать справа в строке на кнопку



- 3 В появившемся окне **Выбор тега для создания связи** выбрать тег **Тег1** объекта **Объект1** и выполнить настройку свойств связи **Атрибуты** и **Обмен данными**, если это необходимо.

Свойства связи

Атрибуты – пары связанных атрибутов, которые будут участвовать в обмене данными. В левой части находятся пользовательские имена атрибутов тега приемника данных.

В правой части находится выпадающие списки с пользовательскими именами атрибутов тега источника данных. Для связывания атрибутов выберите нужный атрибут из выпадающего списка.

ВНИМАНИЕ!

Автоматически связываются атрибуты с одинаковыми именами.

Обмен данными

- **Апертура** – определяет минимальную величину изменения значения атрибута связанного тега, в случае равенства или превышения которой изменяется текущее значение связываемого с ним атрибута тега
- **Направление установки значения** – указывает направление обмена данными:
 - **Получение значения** – значения связанных атрибутов будут копироваться в атрибуты тега приемника данных (владельца связи)
 - **Установка значения** – значения связанных атрибутов будут копироваться в атрибуты тега источника данных
 - **Получение и установка значения** – при изменении значений атрибутов тега приемника данных будет происходить установка значений атрибутов тега источника данных (установка значения); при изменении атрибутов тега источника данных значения связанных атрибутов будут копироваться в атрибуты тега приемника данных (получение значений).
- **Сохранять все связанные атрибуты тега** – определяет количество сохраняемых атрибутов:
 - **Да** – сохраняет значения всех связанных атрибутов тега
 - **Нет** – сохраняет только измененное значение атрибута тега (установлено по умолчанию)
- **Тип обмена данными** – определяет событие, которое приводит к обмену данными:
 - **По изменению** – при любом изменении связанных атрибутов будет происходить обмен данными
 - **По изменению не чаще чем** – обмен данными будет происходить по изменению, но не чаще чем указано в настройке Не чаще чем (мсек), где Вы можете задать частоту обмена в миллисекундах.
- **Тип поиска** – определяет каким образом будет происходить поиск тега источника данных:
 - **Абсолютный** – поиск будет осуществляться глобально во всем проекте
 - **Относительный** – поиск будет осуществляться относительно положения тега приемника данных (владельца связи).

ВНИМАНИЕ!!!

Абсолютный поиск не возможен в библиотеке. В библиотеке также невозможно организовать связь между двумя группами объектов (под группой понимается ветвь дерева объектов, корнем которой является объект, находящийся в папке).

Результат связывания тегов

В нижней части окна **Выбор тега для создания связи** расположена информационная строка – путь к связываемому тегу-источнику, которая содержит результат связывания тегов:









- связь установлена и будет работать нормально
- ошибка установления связи. Текст ошибки отображается рядом с восклицательным знаком.

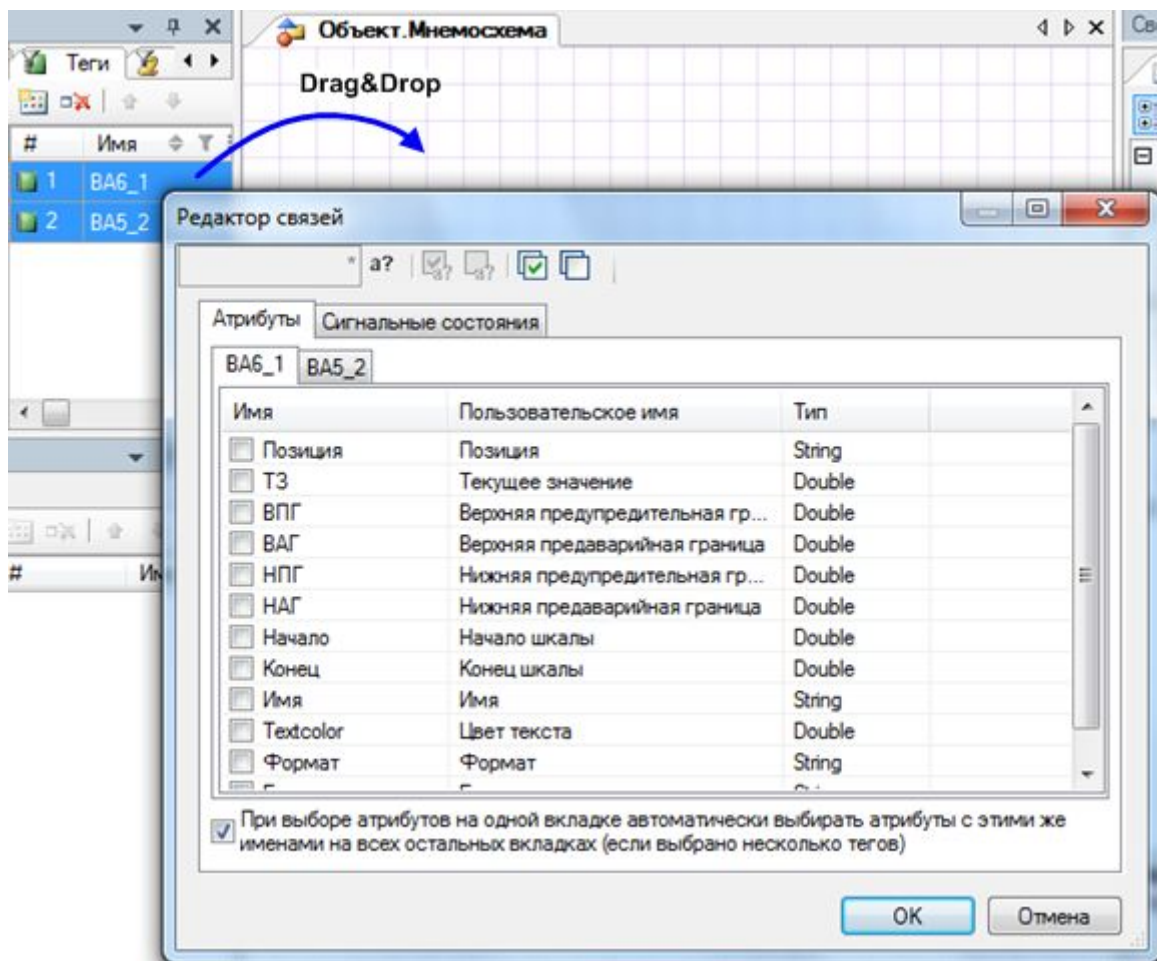
2.4.1.2 Окно редактора связей

Окно **Редактор связей** инициируется операцией **Drag&Drop** (при перетаскивании тегов из **Редактора проекта** на мнемосхему) и позволяет выбрать атрибуты и свойства тегов, к которым осуществляется привязка.



В верхней части окна **Редактор связей** расположена **Панель Инструментов**.

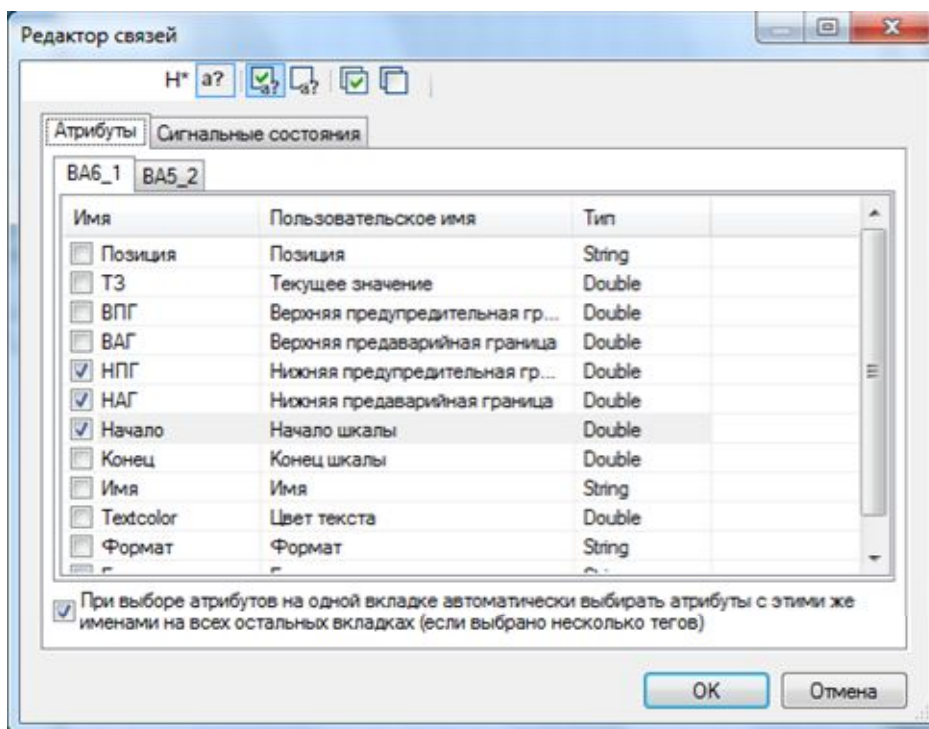
- | | |
|---|--|
|  | – Поле ручного ввода символов маски.
Разрешается использовать обычные текстовые знаки и метасимволы "?" и "*", представляющие одиночный знак и группу знаков соответственно |
|  | – Включить/выключить маскирование |
|  | – Отметить элементы в соответствии с маской |
|  | – Снять отметки с элементов в соответствии с маской |
|  | – Отметить все элементы |
|  | – Снять отметки со всех элементов |

Средняя, рабочая зона, включает управляющие вкладки верхнего уровня – **Атрибуты** и **Сигнальные состояния**, а также вкладки отображение тегов, выбранных в Редакторе проекта.



Для автоматического выбора атрибутов или сигнальных состояний тега по маске следует:

- 1 Включить маскирование – нажать на кнопку 
- 2 Ввести в поле ввода символы маски
- 3 Нажать на кнопку  – отметить элементы, соответствующие маске.



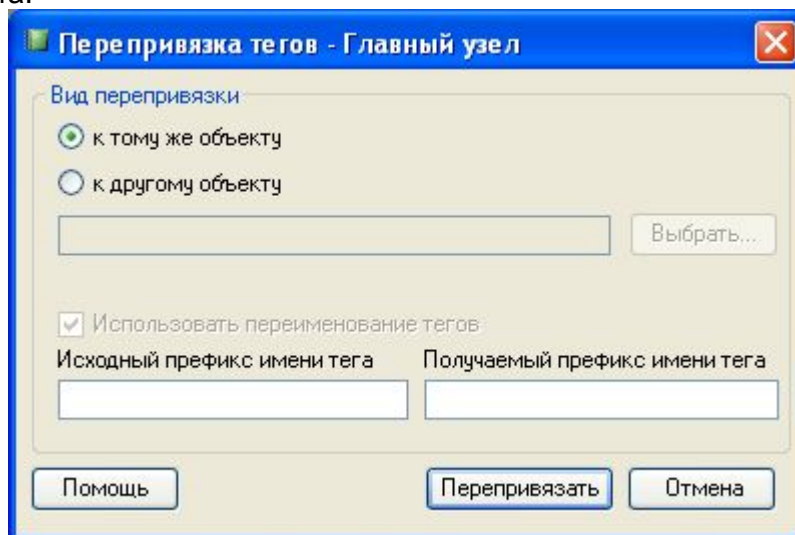
В нижней части окна **Редактор связей** расположены кнопки закрытия окна. При нажатии на кнопку **ОК** помеченные элементы будут сохранены в качестве привязок. Нажатие кнопки **Отмена** завершает операцию без сохранения привязок.

2.4.1.3 Перепривязка тегов

Перепривязка тегов позволяет пользователю в среде разработки **DataRate** изменять привязки тегов выбранного объекта.

Для перепривязки тегов объекта дерева проекта следует:

- 1 Выбрать в контекстном меню объекта пункт **Перепривязать теги**
- 2 В появившемся окне **Перепривязка тегов** задать параметры перепривязки тегов. В заголовке окна отображается имя выбранного в дереве проекта объекта.



Перепривязка тегов может осуществляться:

- к тегам того же объекта, к которому привязан выбранный объект дерева проекта
- к тегам другого объекта.

Как перепривязать теги к тегам того же объекта?

В этом случае каждый тег исходного объекта будет перепривязан к тегу того же объекта что и текущая привязка, но с измененным префиксом имени тега.

Пример:

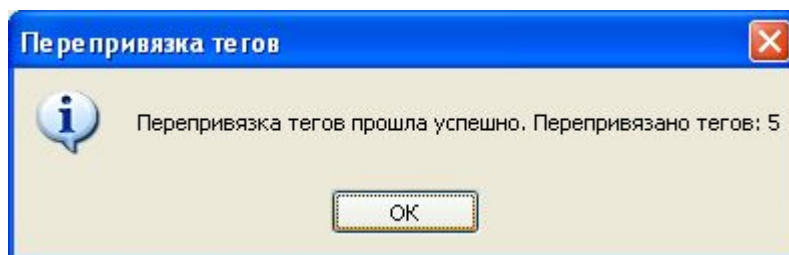
Пусть имеется объект **Объект1** с тегам **ВА***, которые привязаны к тегам **ОРСКоннектор1.Com1_Param***.

Необходимо перепривязать теги **Объект1.ВА*** к тегам **ОРСКоннектор1.Com2_Param***.

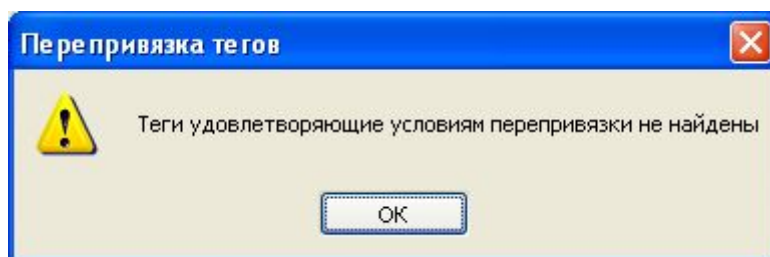
Для этого в окне **Перепривязка тегов** следует:

- 1 Ввести **Com1** в поле ввода исходного префикса и **Com2** – в поле получаемого префикса
- 2 Нажать кнопку **Перепривязать**.

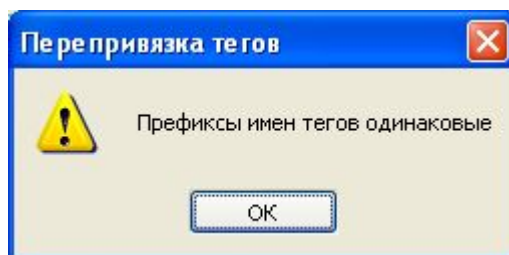
В случае удачной перепривязки будет выведено сообщение с указанием количества перепривязанных тегов.



Если тегов удовлетворяющих условиям нет, то будет выведено следующее сообщение:



Если введенные префиксы тегов будет одинаковые, т.е. нет смысла в перепривязке, будет выведено следующее сообщение:



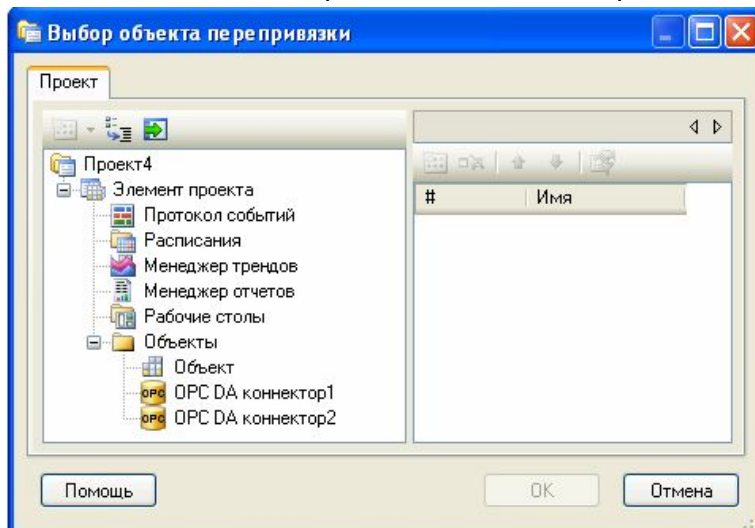
Если при перепривязке тегов к тому же объекту, теги перепривязки не будут найдены, пользователь увидит список этих тегов в окне **Ошибки при перепривязке тегов**.

Пользователь может задать теги, к которым необходимо осуществить перепривязку, но при этом могут появиться ссылки на несуществующие теги. В случае верификации исходного объекта такие ссылки будут выявлены.

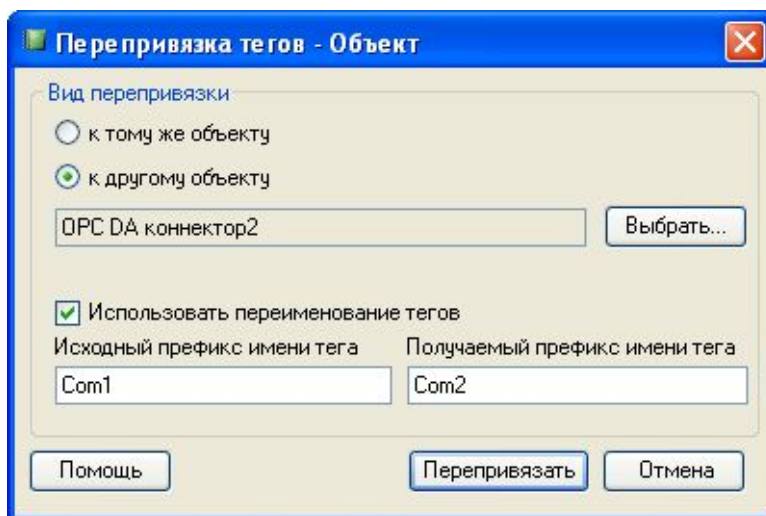
Как перепривязать теги к тегам другого объекта?

Если необходимо сменить объект привязки выбранного элемента проекта следует:

- 1 В контекстном меню объекта выбрать **Перепривязать теги**
- 2 В появившемся окне **Перепривязка тегов** для выбора нового объекта привязки нажать кнопку **Выбрать**
- 3 В появившемся окне **Выбор объекта перепривязки** выбрать объект привязки
- 4 После выбора объекта новой привязки в окне **Перепривязка тегов** ввести префиксы исходного и получаемого имен тегов.



Если флажок **Использовать переименование тегов** снят, то имена тегов привязки будут сохранены, иначе имена тегов будут заменены в соответствии с введенными значениями.

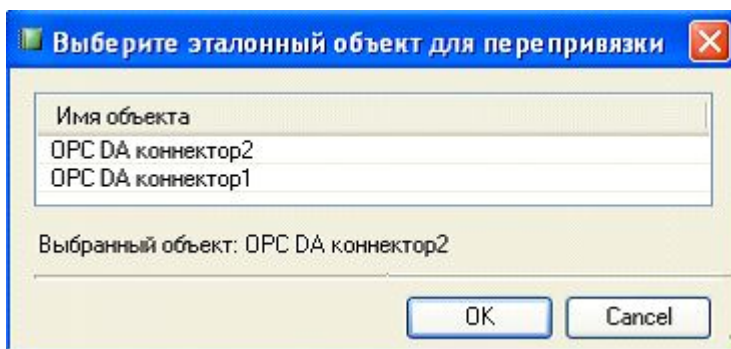


Как перепривязать теги, если они привязаны к нескольким объектам?

Если теги выбранного объекта привязаны не к одному, а к нескольким объектам, пользователю будет предложено выбрать объект текущей перепривязки, для сопоставления с объектом новой привязки.

ВНИМАНИЕ!

Перепривязываться к тегам нового объекта привязки будут только теги выбранного объекта.



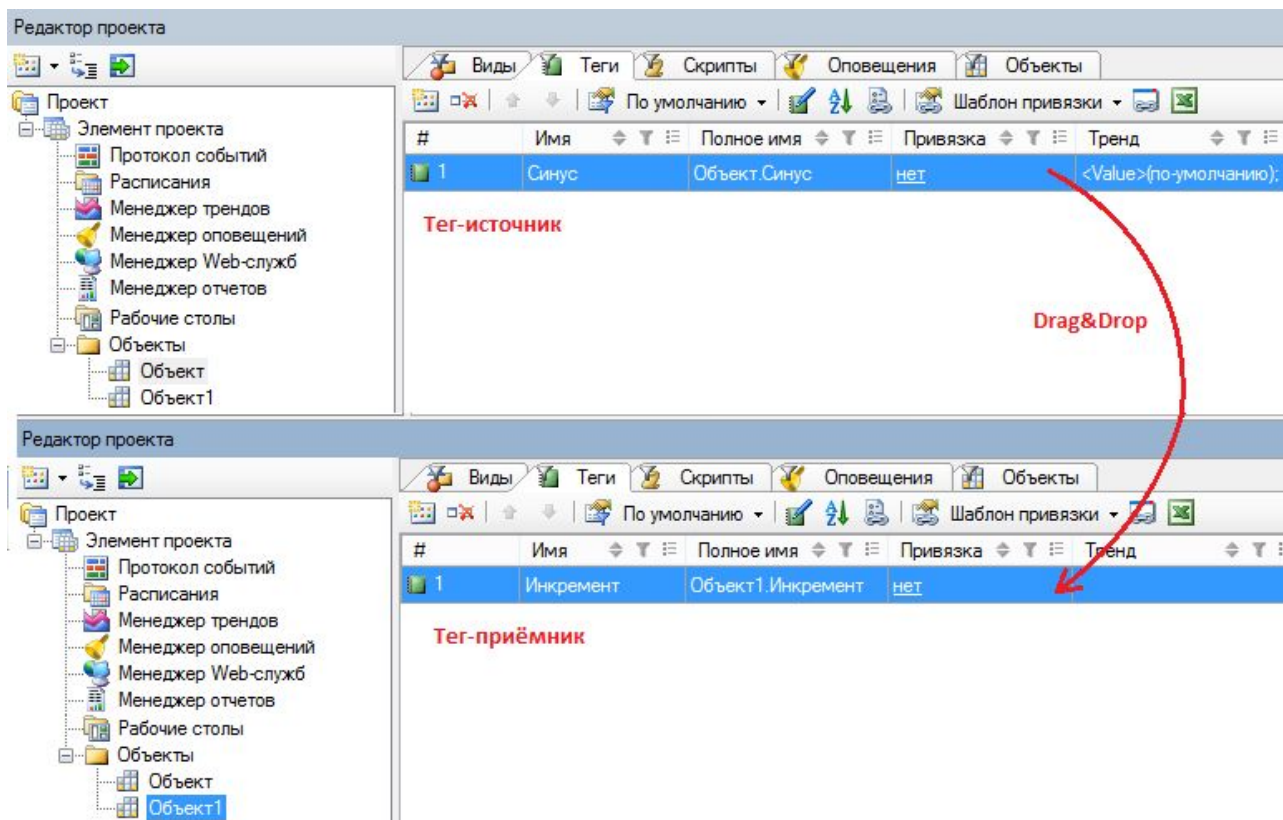
2.4.1.4 Групповая привязка тегов

Групповая привязка тегов – дополнительная команда для тегов, результатом выполнения которой является установка новых свойств привязки тегов. Групповая привязка служит для привязки атрибутов одной группы тегов к атрибуту другой группы тегов, а также позволяет изменить свойства привязки для уже привязанных тегов.

Как выполнить групповую привязку тегов

Для выполнения групповой привязки следует:

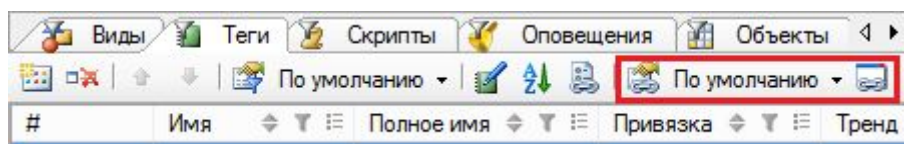
- 1 Открыть два экземпляра редактора проекта и в каждом из них выделить теги
- 2 Удерживая нажатой клавишу **Alt**, перетащить теги источника новой привязки на привязываемые теги. При перетаскивании привязка осуществляется в соответствии с порядком тегов: 1-й тег источника используется для 1-го привязываемого тега, 2-й тег источника – для 2-го и так далее.



Настройка шаблона групповой привязки

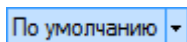
Шаблон привязки используется при установке привязки с помощью операции Drag&Drop.

Для настройки шаблона групповой привязки используются следующие элементы управления:





– Кнопка **Редактировать шаблоны привязки**.



– Выпадающий список доступных шаблонов привязок.



– Кнопка **Выбрать шаблон перед привязкой**.

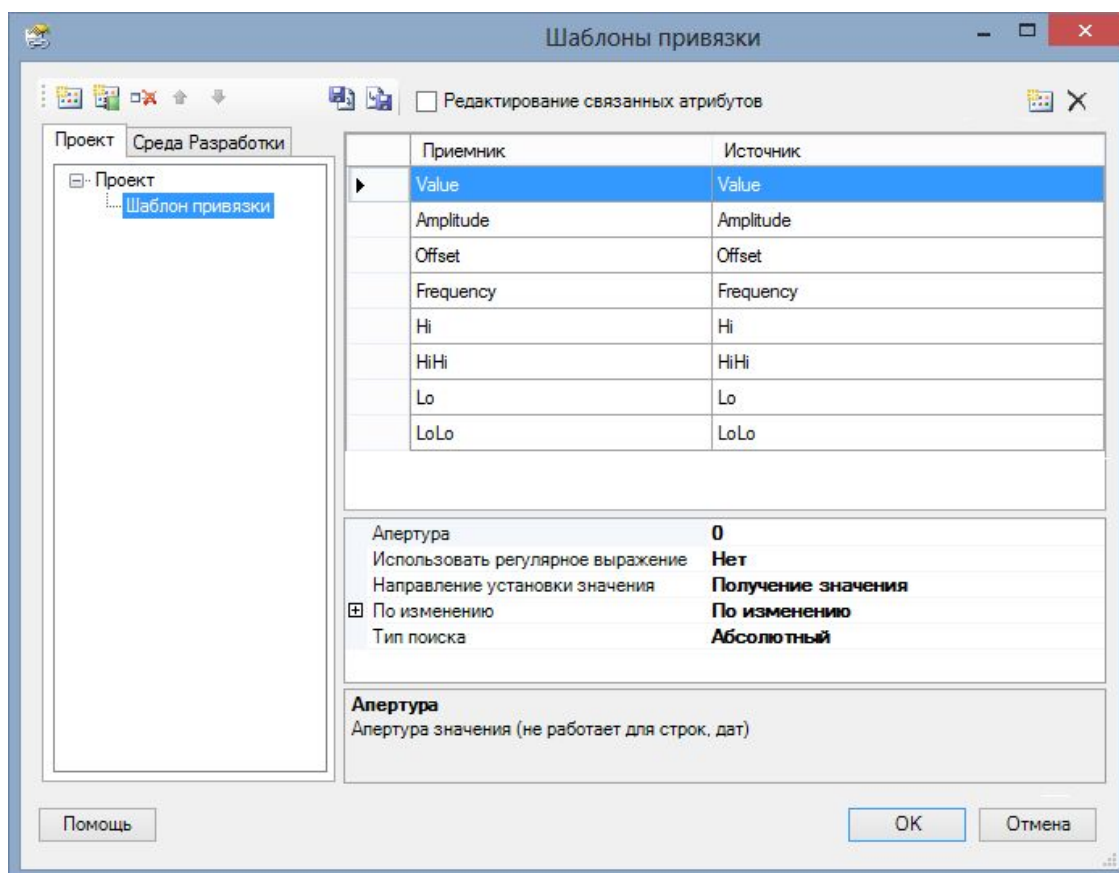
Если данная кнопка нажата в окне, в которое перетаскивается тег, то в конце операции Drag&Drop появляется окно выбора шаблона привязки.

Если данная кнопка отжата, то для привязки будет использоваться привязка по-умолчанию, и окно выбора шаблона привязки появляться не будет.

Создание/ редактирование шаблона привязки

Для создания/ редактирования шаблона следует:

- 1 Нажать на кнопку **Редактировать шаблоны привязки**
- 2 В появившемся окне **Шаблоны привязки** перейти на вкладку **Среда Разработки**, если необходимо, чтобы создаваемый шаблон был виден для всех проектов. Шаблоны, созданные на вкладке **Проект**, видны только для данного проекта.



- 3 Создать новый шаблон привязки, нажав на кнопку **Создать шаблон** или отредактировать уже существующий шаблон, выбрав его в дереве элементов. Для редактирования используйте кнопки:



– Добавить пару **Приемник/Источник**



– Удалить пару **Приемник/Источник**



- 4 Нажать на кнопку **OK**.

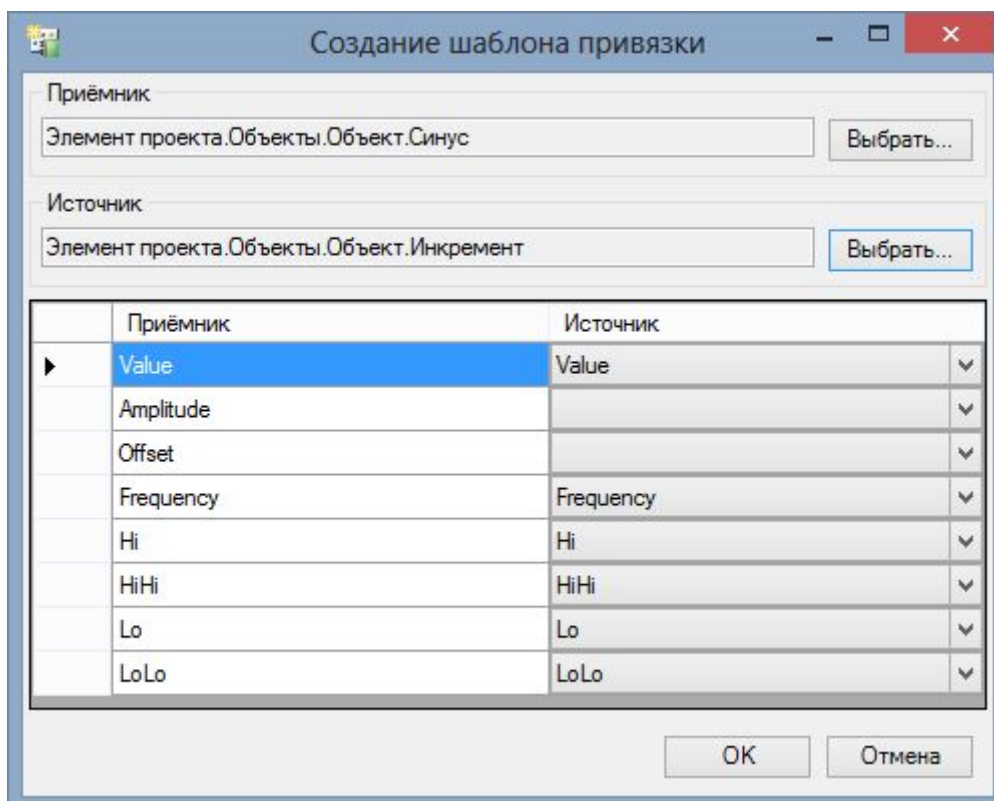
Свойства шаблона

- **Имя шаблона** – определяет уникальное имя шаблона. Для редактирования имени щёлкните дважды по нему в дереве шаблонов
- **Редактирование связанных атрибутов** – если данная опция выбрана, то при привязке будут связаны указанные в списке атрибуты, иначе будут связаны атрибуты, совпадающие по имени и типу
- **Апертура**
- **Использовать регулярное выражение**. Значение свойства:
 - Регулярное выражение для тега-приёмника
 - Регулярное выражение для тега-источника
- **Направление установки значения**. Имеет три варианта: Установка, Получение, Установка и получение
- **По изменению**. Тип обмена данными: По изменению/ По изменению не чаще чем.
- **Тип поиска**. Тип поиска связанного тега: Абсолютный/ Относительный.

Создание шаблона привязки на основе тегов

Для создания/ редактирования шаблона следует:

- 1 Нажать на кнопку **Редактировать шаблоны привязки** 
- 2 В появившемся окне **Шаблоны привязки** перейти на вкладку **Среда Разработки**, если необходимо, чтобы создаваемый шаблон был виден для всех проектов. Шаблоны, созданные на вкладке **Проект**, видны только для данного проекта
- 3 Создать новый шаблон привязки, нажав на кнопку **Создать шаблон на основе тегов** 
- 4 В открывшемся окне **Создание шаблона привязки** с помощью кнопок **Выбрать...** задать тег-приёмник и тег-источник.



Создание шаблона привязки

Приёмник
 Элемент проекта.Объекты.Объект.Синус Выбрать...

Источник
 Элемент проекта.Объекты.Объект.Инкремент Выбрать...

	Приёмник	Источник
▶	Value	Value
	Amplitude	
	Offset	
	Frequency	Frequency
	Hi	Hi
	HiHi	HiHi
	Lo	Lo
	LoLo	LoLo

OK Отмена


Пары атрибутов тегов подберутся автоматически, при условии, что название и тип атрибутов тега-приёмника и тега-источника совпадают. Связать атрибуты тегов можно вручную, выбирая в выпадающем списке атрибутов тега-источника атрибут, подходящий для атрибута тега-приёмника.

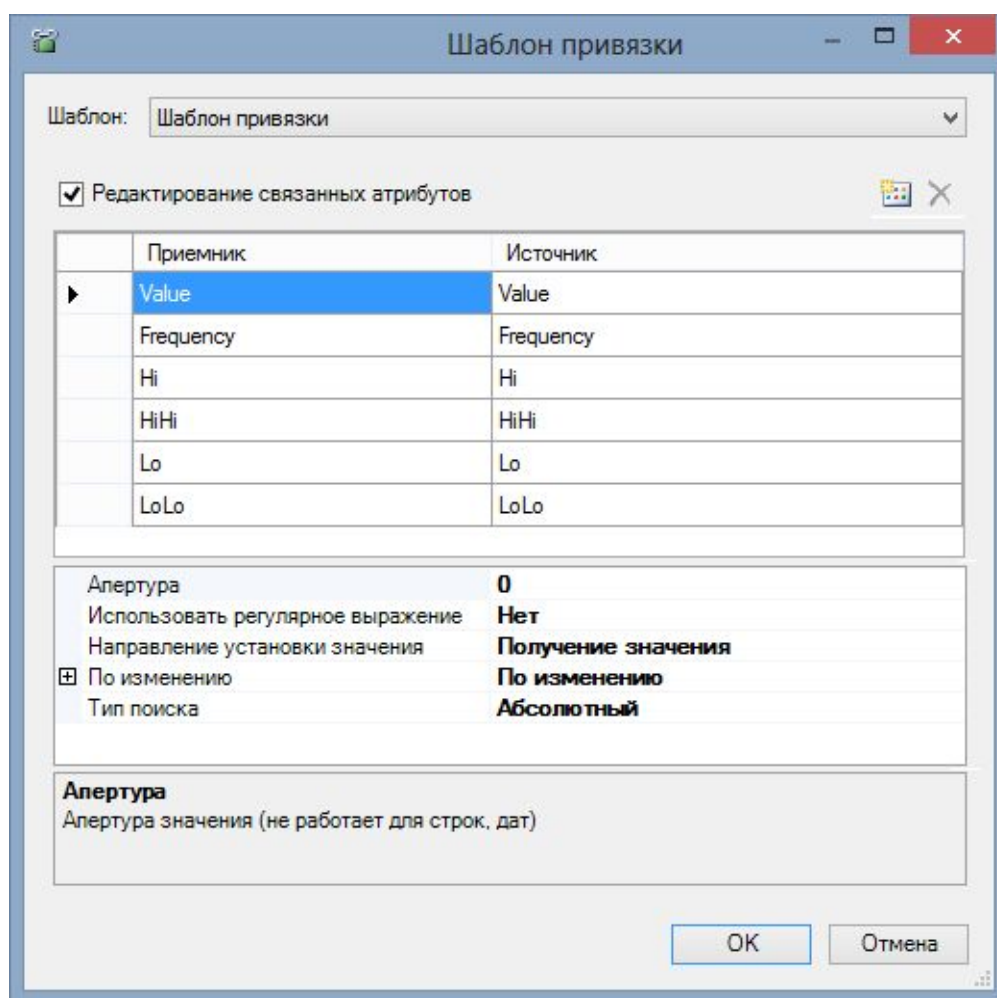
- 5 Нажать на кнопку **ОК**.

ВНИМАНИЕ!!!

Информация о выбранных тегах не сохраняется.

Выбор шаблона привязки при окончании операции Drag&Drop

Если нажата кнопка **Выбрать шаблон перед привязкой** , то после завершения операции Drag&Drop открывается окно **Шаблон привязки**.



Приемник	Источник
Value	Value
Frequency	Frequency
Hi	Hi
HiHi	HiHi
Lo	Lo
LoLo	LoLo

Апертура: 0

Использовать регулярное выражение: Нет

Направление установки значения: Получение значения

По изменению: По изменению

Тип поиска: Абсолютный

Апертура
Апертура значения (не работает для строк, дат)

ОК Отмена

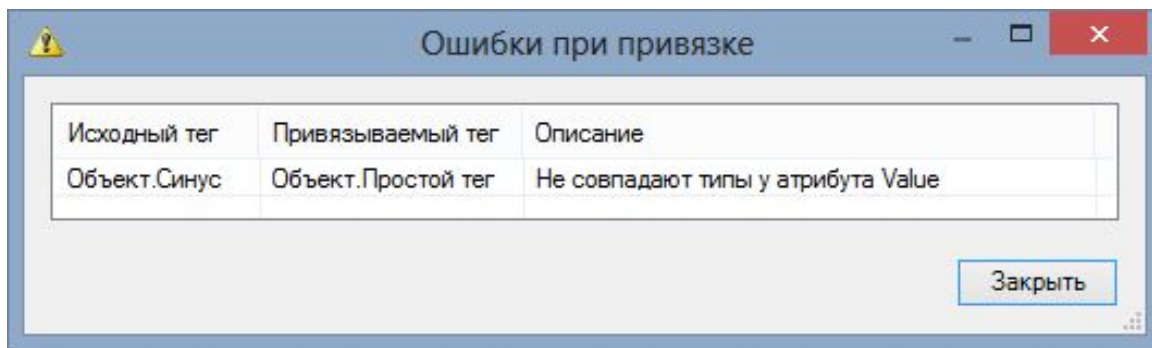
В этом окне из списка можно выбрать требуемый шаблон и, если необходимо, внести изменения в его свойства. Привязка тегов будет завершена в соответствии с выбранным шаблоном.

ВНИМАНИЕ!!!

Изменение шаблона в данном окне будут действовать только на данную операцию привязки.

Ошибки привязки

Ошибки привязки выводятся в отдельном окне.



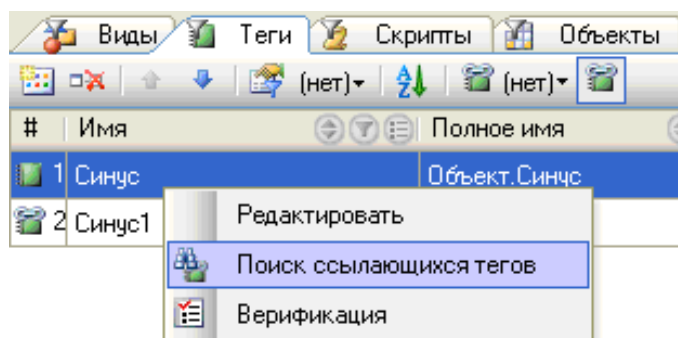
Данное сообщение может появиться, например, в следующих случаях:

- тег, привязку которого мы хотели изменить, не имеет привязки
- не совпадают типы атрибутов с одним и тем же именем в исходном и привязываемом теге.

2.4.1.5 Поиск тегов

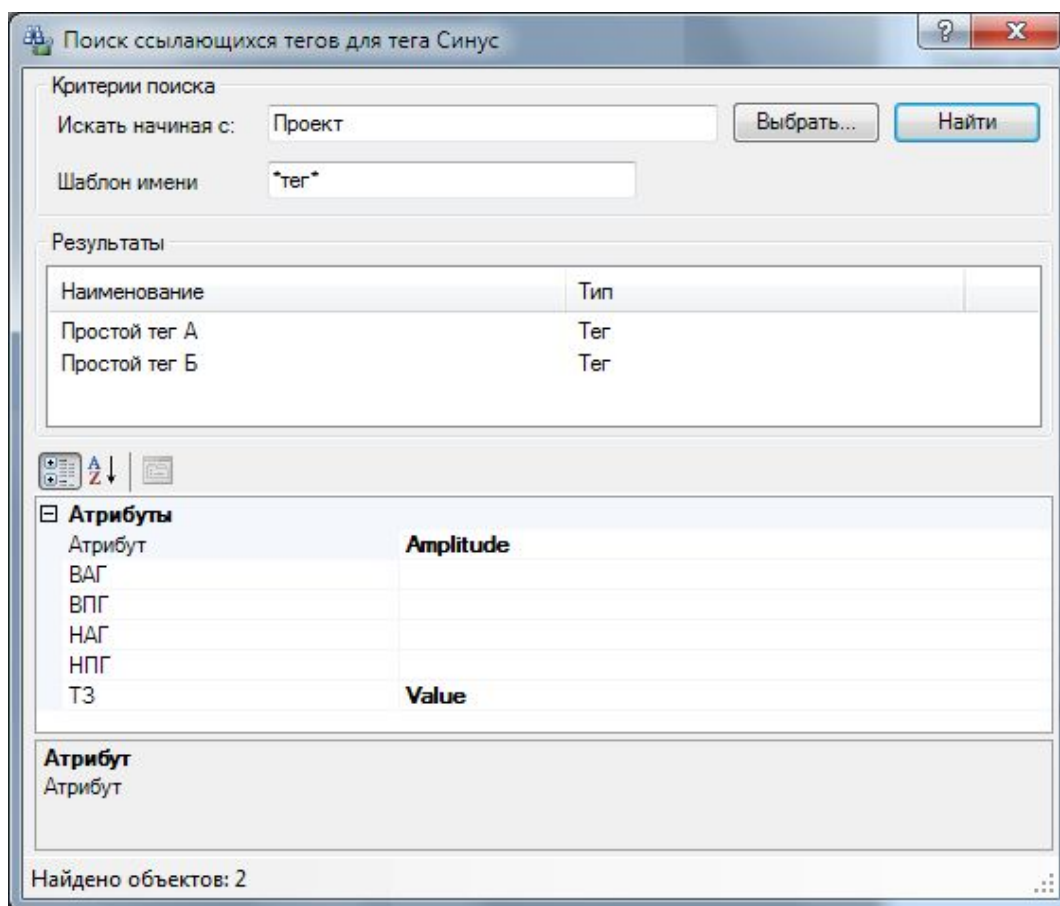
Поиск тегов, имеющих ссылки на заданный тег

Для того чтобы найти теги, которые ссылаются на заданный тег, выберите в редакторе проекта этот тег и в его контекстном меню вызовите команду **Поиск ссылающихся тегов**.



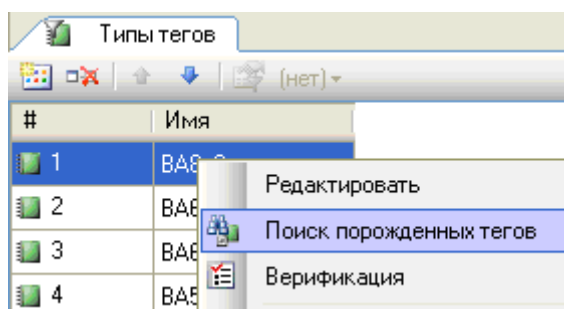
В появляющемся окне **Поиск ссылающихся тегов** можно задать параметры поиска, аналогичные параметрам поиска объектов.

Кроме возможностей, описанных в разделе «Поиск объектов», в окне поиска ссылающихся тегов добавлены возможности просматривать привязку тега по атрибутам и «перекидывать» одиночные атрибуты тегов.



Поиск тегов, порожденных от заданного тега

Для того чтобы найти теги, которые порождены от заданного типа тега, выберите этот тип тега в библиотеке и в его контекстном меню вызовите команду **Поиск порожденных тегов**.



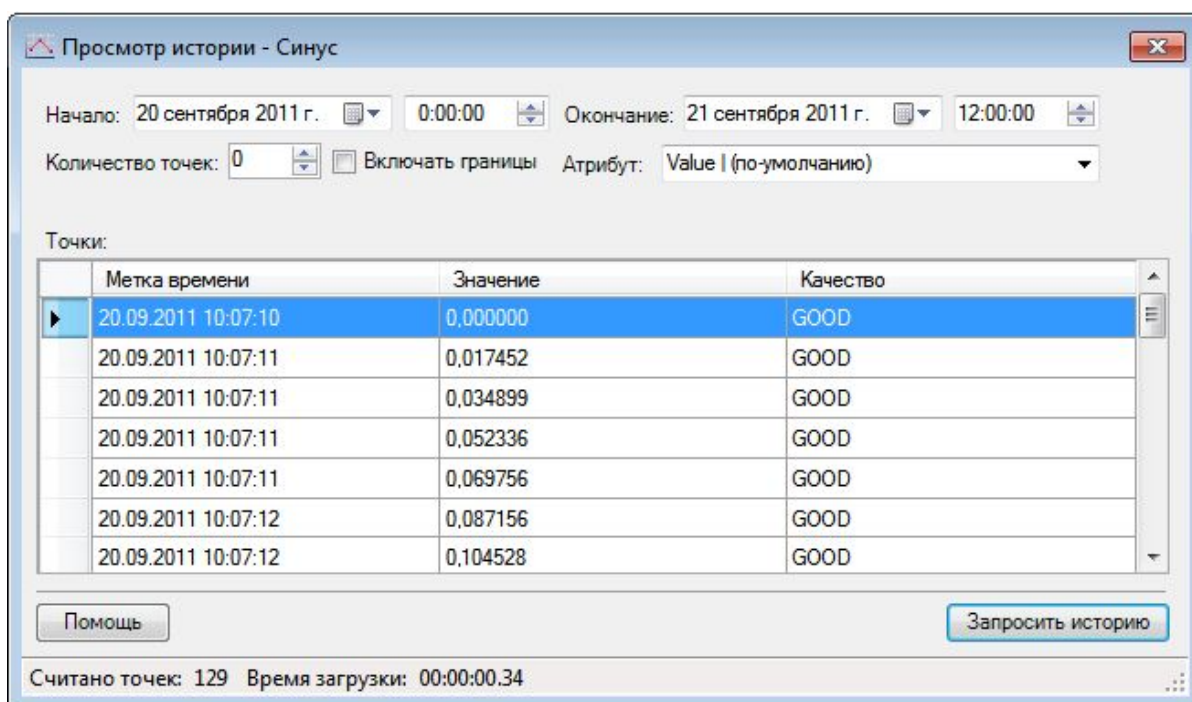
В диалоговом окне поиска можно задать параметры поиска. Описание параметров приведено в разделе «Поиск объектов».

2.4.1.6 Просмотр истории тега

Среда разработки *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 тег в редакторе проекта
- 2 В контекстном меню тега выбрать пункт **Просмотреть историю**
- 3 В появившемся окне **Просмотр истории** задать параметры запроса истории и нажать кнопку **Запросить историю**.




Пользователь должен задать следующие параметры:

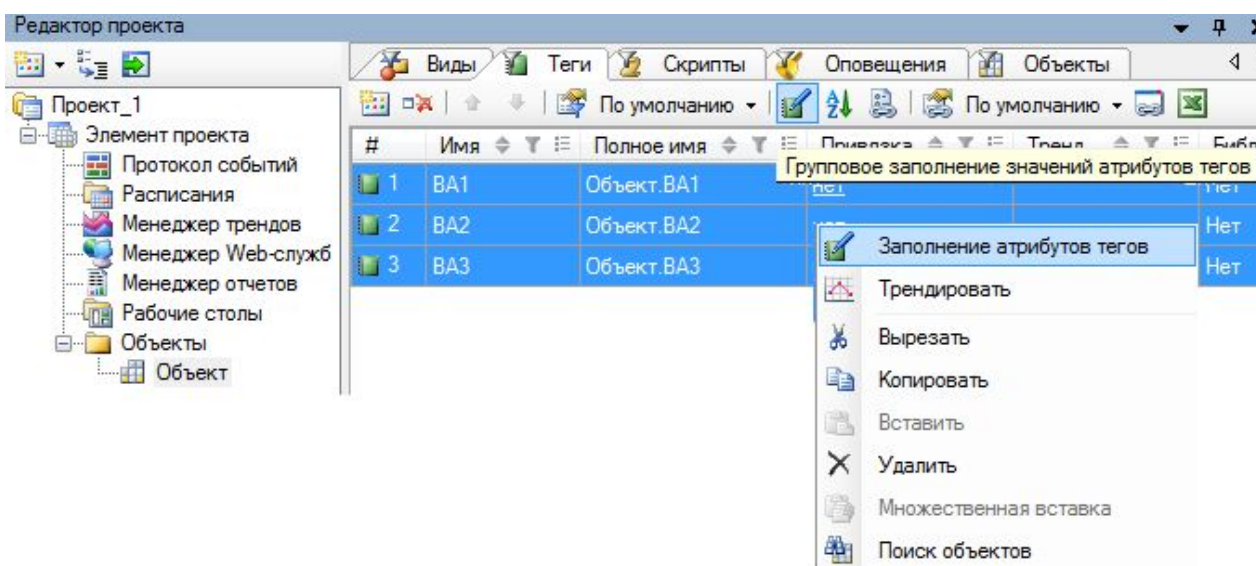
- **Начало/Окончание** – начальное и конечное время интервала запроса
- **Количество точек** – количество запрашиваемых на интервале точек. Если задан 0, то будут вычитываться все точки
- **Включать границы** – признак включения границ в интервал запроса
- **Атрибут** – атрибута тега.

2.4.1.7 Групповое заполнение значений атрибутов тегов

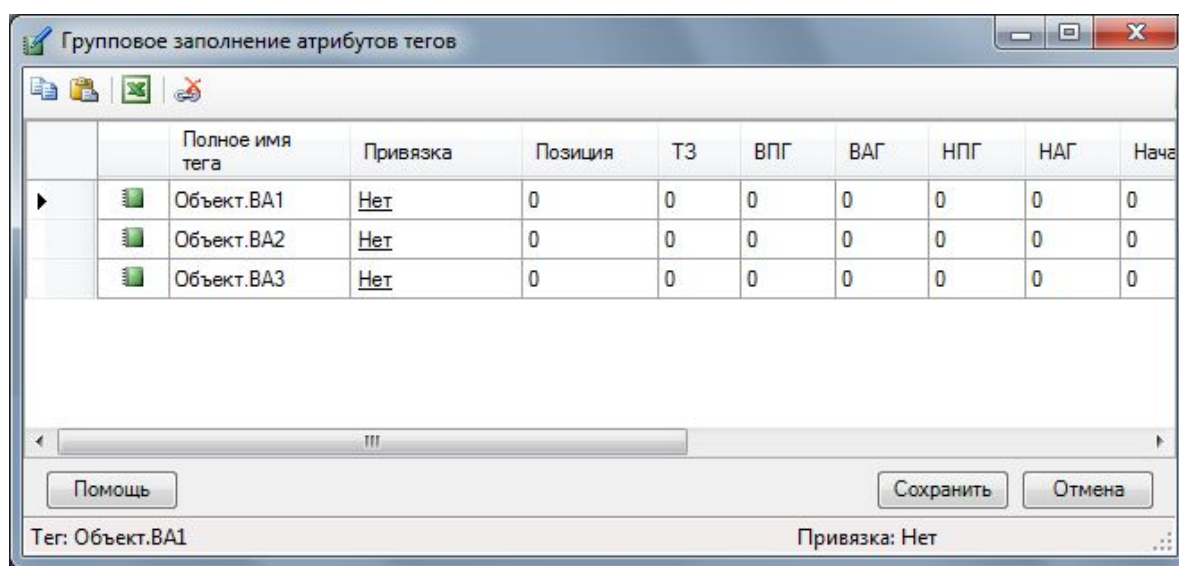
Начало редактирования

Чтобы начать редактирование значений атрибутов тегов следует:

- 1 В дереве проекта выделить один или несколько объектов
- 2 Выбрать вкладку **Теги** и выделить те теги, значения атрибутов которых необходимо задать
- 3 Нажать на кнопку , либо выбрать в контекстном меню пункт **Заполнение атрибутов тегов** (пункт появляется в меню только в случае, когда выбрано несколько тегов).





- 4 В появившемся окне **Групповое заполнение атрибутов тегов** задать необходимые значения.



Редактор значений атрибутов тегов

Редактор значений предназначен для редактирования значений тегов в окне **Групповое заполнение атрибутов тегов**.

Панель инструментов включает следующие кнопки управления:

- 
Копировать. Копирование выделенных ячеек редактора в буфер обмена. Для того чтобы скопировать содержимое ячеек в буфер обмена, необходимо с помощью мыши выделить один или несколько ячеек в редакторе и нажать кнопку **Копировать** либо сочетание клавиш **Control+C** / **Control+Insert**
- 
Вставить. Вставка содержимое буфера обмена в выделенные ячейки. Для того чтобы вставить значения в ячейки редактора, необходимо выделить мышью одно или несколько ячеек редактора, в которые будет произведена вставка, и нажать кнопку **Вставить** либо сочетание клавиш **Control+V** / **Shift+Insert**



– **Экспортировать.** Экспорт данных в **Csv**-файл (MS Excel). Экспортирование данных редактора осуществляется в текстовый файл в формате значений, разделенных запятой. Во время экспорта запрашивается имя файла и путь для записи данных.



– **Удалить привязку**

Список тегов и значений их атрибутов представлен в виде таблицы, состоящей из следующих колонок:

- **Графическое изображение тега.**
- **Полное имя тега.** Колонка доступна только для просмотра
- **Привязка.** Для того чтобы изменить привязку, необходимо щелкнуть мышью по ячейке с привязкой и в открывшемся окне редактора привязки настроить привязку тега
- **Колонки редактируемых атрибутов тега.** Список атрибутов определяется типом тега. Для редактирования значений атрибутов, необходимо выделить ячейку редактора и ввести новое значение. Вводить новые значения ячеек можно не только непосредственным вводом с клавиатуры, но также при помощи буфера обмена и механизма перетаскивания из других редакторов, например из Excel. При переходе от редактирования текущей ячейки к другой изменяется цвет ячейки:
 - Ячейки значения, которых были изменены, подсвечиваются зеленым цветом
 - Если в процессе редактирования была допущена ошибка и введенное значение не удастся преобразовать к типу соответствующего атрибута тега, то ячейки редактора с ошибкой подсвечиваются красным цветом.

		Полное имя тега	Привязка	Позиция	ТЗ	ВПГ	ВАГ
		Объект.ВА1	<u>Нет</u>	Позиция 1	XXX	0	0
		Объект.ВА2	<u>Нет</u>	0	0	0	0
		Объект.ВА3	<u>Нет</u>	0	0	0	0

Кнопки управления состоянием редактора

- **Помощь** – открывает это окно справки
- **Сохранить** – сохраняет изменения в проект и закрывает окно редактора. Если в процессе редактирования были допущены ошибки, то при сохранении необходимо сделать подтверждение закрытия окна
- **Отмена** – закрывает окно редактора без сохранения изменений. Если во время отмены в редакторе были сделаны изменения, то необходимо сделать подтверждение закрытия окна.

В статусной строке отображается наименование выделенного тега и его привязка.

2.4.1.8 База данных тегов

Значения атрибутов тегов хранятся в специально разработанной для этой цели базе данных - БД тегов.

В **Среде разработки DataRate** каждый элемент проекта содержит в себе экземпляр БД тегов. Назовем её – *база данных тегов времени разработки*.

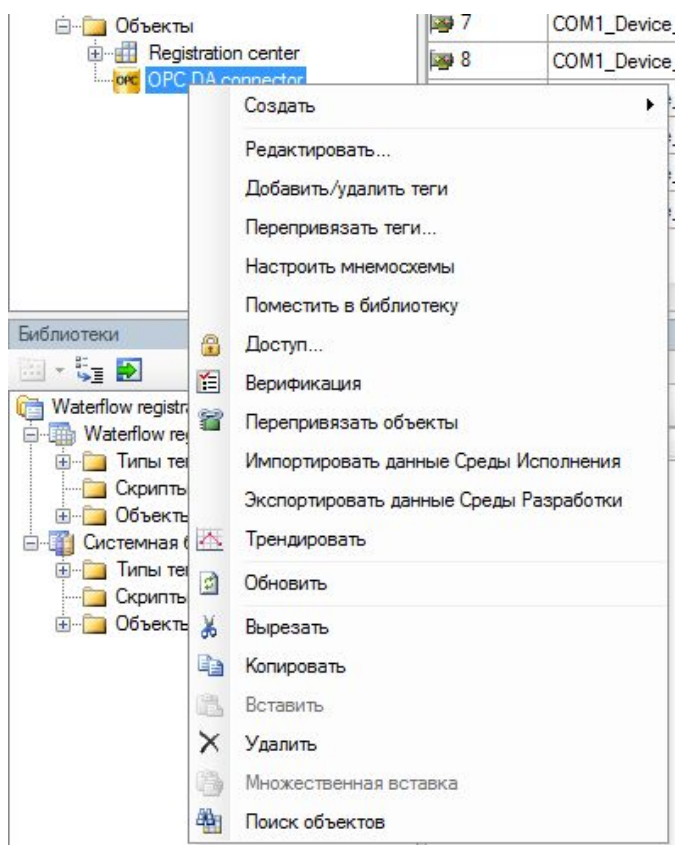
При запуске проекта на исполнения в **Среде исполнения** для каждого элемента проекта будет создана *база данных тегов времени исполнения*. БД времени исполнения создается и сохраняется в отдельном файле с именем **имя_элемента_проекта.rtdb**.

Такое решение для хранения атрибутов тегов обусловлено необходимостью разделять значения атрибутов тегов в периоды разработки проекта и исполнения, например, для хранения различных уставок.

Импорт данных среды исполнения

При необходимости значения атрибутов тегов времени исполнения можно перенести в БД Среды разработки. Для этого необходимо выбрать команду **Импортировать данные Среды Исполнения**, доступную из контекстного меню **Проекта**, **Элемента проекта** или **Объекта** в дереве проекта. В зависимости от того, где будет вызвано контекстное меню, импорт произойдет для всех элементов проекта, для одного выбранного элемента проекта или для одного выбранного объекта в дереве проекта.

В результате импорта значения атрибутов тегов БД времени разработки будут заменены значениями соответствующих атрибутов из БД времени исполнения.



Экспорт данных среды разработки

Также существует возможность переноса значений атрибутов тегов из БД времени разработки в БД времени исполнения. Для этого необходимо выбрать команду **Экспортировать данные Среды Разработки**, доступную из контекстного меню **Проекта**, **Элемента проекта** или **Объекта** в дереве проекта. В зависимости от того, где будет вызвано контекстное меню, экспорт произойдет для всех элементов проекта, для одного выбранного элемента проекта или для одного выбранного объекта в дереве проекта.

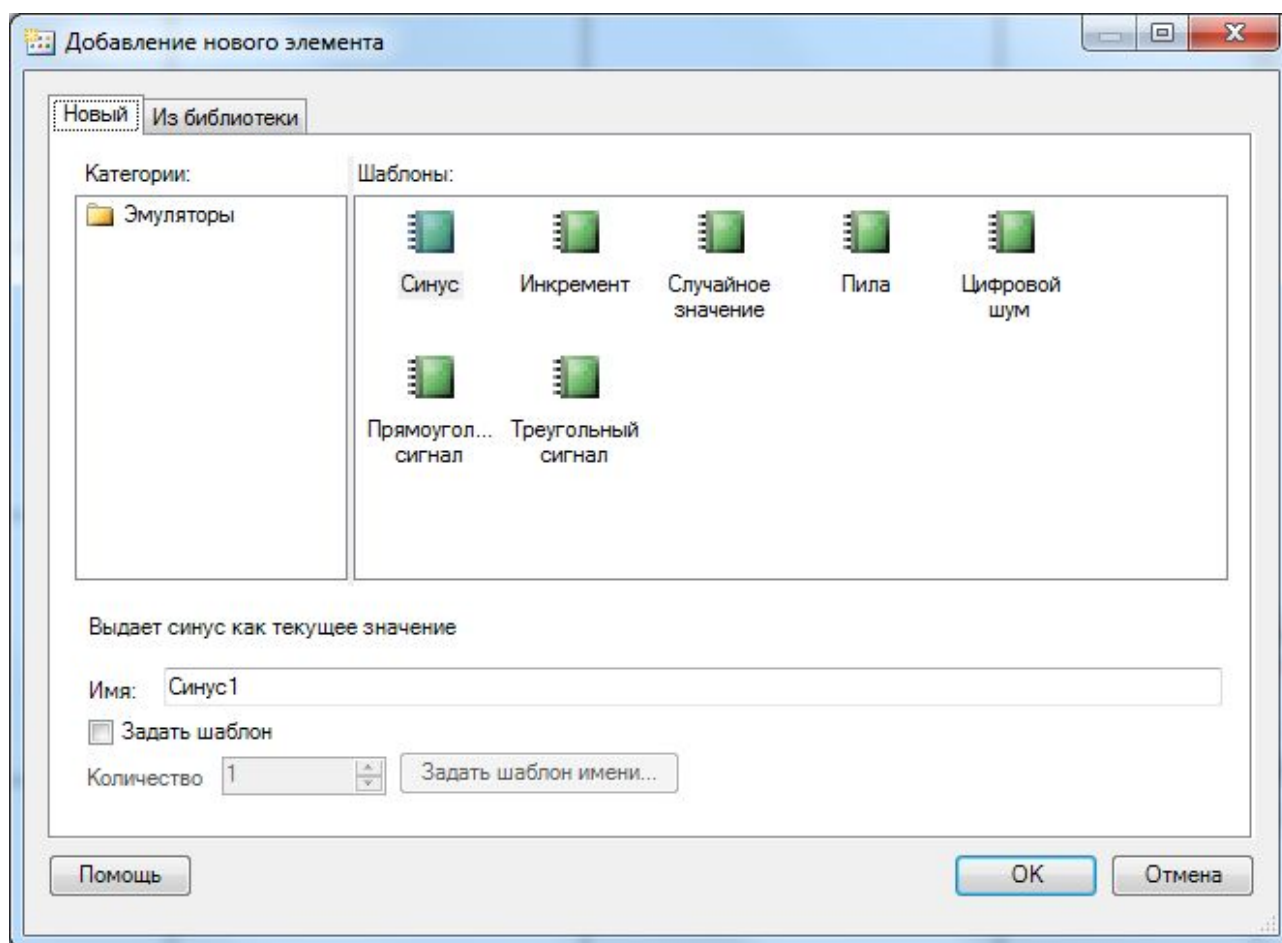
В результате экспорта, значения атрибутов тегов БД времени исполнения будут заменены значениями соответствующих атрибутов из БД времени разработки.

2.4.1.9 Теги эмуляторы

Тег Эмулятор предназначен для автоматической эмуляции значений выбранного атрибута, а также создания событий сигнализации **Превышение уровня**.

Создание

- 1 В контекстном меню объекта выбрать **Создать/Тег**
- 2 В появившемся окне **Добавление нового элемента** выбрать закладку **Новый**, категорию **Эмуляторы** и шаблон эмулятора
- 3 Ввести **Имя** тега и нажать на кнопку **ОК**



Для создания нескольких однотипных тегов воспользуйтесь флажком **Задать шаблон**. Введите количество тегов и задайте шаблон имени создаваемых тегов. Задать шаблон имени можно нажав на кнопку **Задать шаблон имени** (подробнее смотрите в разделе «Формирование шаблона имени»).

ВНИМАНИЕ!!!

При переименовании вида ссылки на этот тег становятся недействительными.

Для проверки необходимо запускать верификацию.

При копировании/вставке тега происходит копирование только модели тега, но не его данных.

Редактирование

Для изменения параметров **тега эмулятора** следует:

- 1 Выбрать в контекстном меню тега выбрать команду **Редактировать** или два раза щелкнуть левой кнопкой мыши по этому тегу
- 2 В открывшейся закладке изменить параметры тега

Пример 1. Тег эмулятор Синус

Тег эмулирует значения синуса от -1 до 1.

Объект. Синус							
#	Имя	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение	
1	Value	Текущее значение	Double		<input type="checkbox"/>	0,8	
2	Amplitude	Амплитуда	Double		<input type="checkbox"/>	1	
3	Offset	Смещение	Double		<input type="checkbox"/>	1	
4	Frequency	Частота	Int32		<input type="checkbox"/>	250	
5	Hi	Верхняя предупредительная граница	Double		<input type="checkbox"/>	0,8	
6	HiHi	Верхняя аварийная граница	Double		<input type="checkbox"/>	0,9	
7	Lo	Нижняя предупредительная граница	Double		<input type="checkbox"/>	-0,8	
8	LoLo	Нижняя аварийная граница	Double		<input type="checkbox"/>	-0,9	

Атрибуты тега

- **Value** – текущее значение тега
- **Amplitude** – амплитуда значения тега
- **Offset** – смещение относительно начала координат
- **Frequency** – частота изменения значения атрибута **Value** на **1 градус** (не меньше **100 мс**). На рисунке выше **Value** изменяется каждые **250 мс**
- **Hi** – верхняя предупредительная граница
- **HiHi** – верхняя аварийная граница
- **Lo** – нижняя предупредительная граница
- **LoLo** – нижняя аварийная граница.

Пример 2. Тег эмулятор Инкремент

Тег эмулирует значения с постоянным приращением.

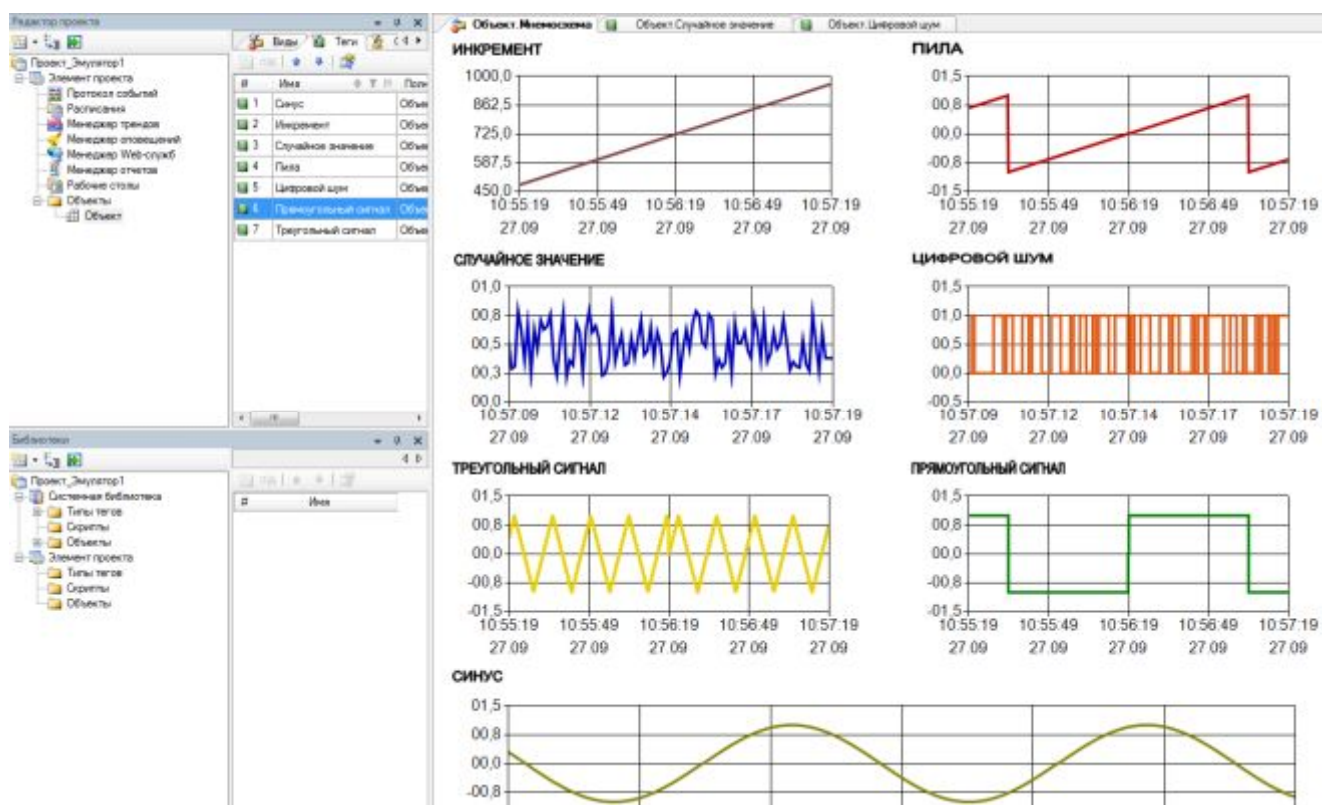
Объект1.Инкремент							
#	Имя	Пользовательское имя	Тип	Размер (для строк)	Только для чтения	Значение	
1	Value	Текущее значение	Double		<input type="checkbox"/>	0	
2	Delta	Разница	Double		<input type="checkbox"/>	1	
3	Frequency	Частота	Int32		<input checked="" type="checkbox"/>	250	
4	Hi	Верхняя предупредительная г...	Double		<input type="checkbox"/>	0,8	
5	HiHi	Верхняя аварийная граница	Double		<input type="checkbox"/>	0,9	
6	Lo	Нижняя предупредительная г...	Double		<input type="checkbox"/>	-0,8	
7	LoLo	Нижняя аварийная граница	Double		<input type="checkbox"/>	-0,9	

Атрибуты тега

- **Value** – текущее значение тега
- **Delta** – приращение, может быть как положительным так и отрицательным
- **Frequency** – частота изменения значения атрибута **Value** (не меньше **100** мс). Например, на рисунке, представленном выше, **Value** будет изменяться на **1** (значение атрибута **Delta**) каждые **250** мс
- **Hi** – верхняя предупредительная граница
- **HiHi** – верхняя аварийная граница
- **Lo** – нижняя предупредительная граница
- **LoLo** – нижняя аварийная граница.

Использование

При запуске среды исполнения через заданные промежутки времени во все объекты, "подписавшиеся" на изменение атрибута, выбранного Пользователем в поле **Имя**, будут приходить значения, определяемые тегом **Эмулятор**.



2.4.2 Виды

Вид – графическое изображение объекта. Создать изображение и анимировать его можно с помощью визуальных редакторов и скриптов.

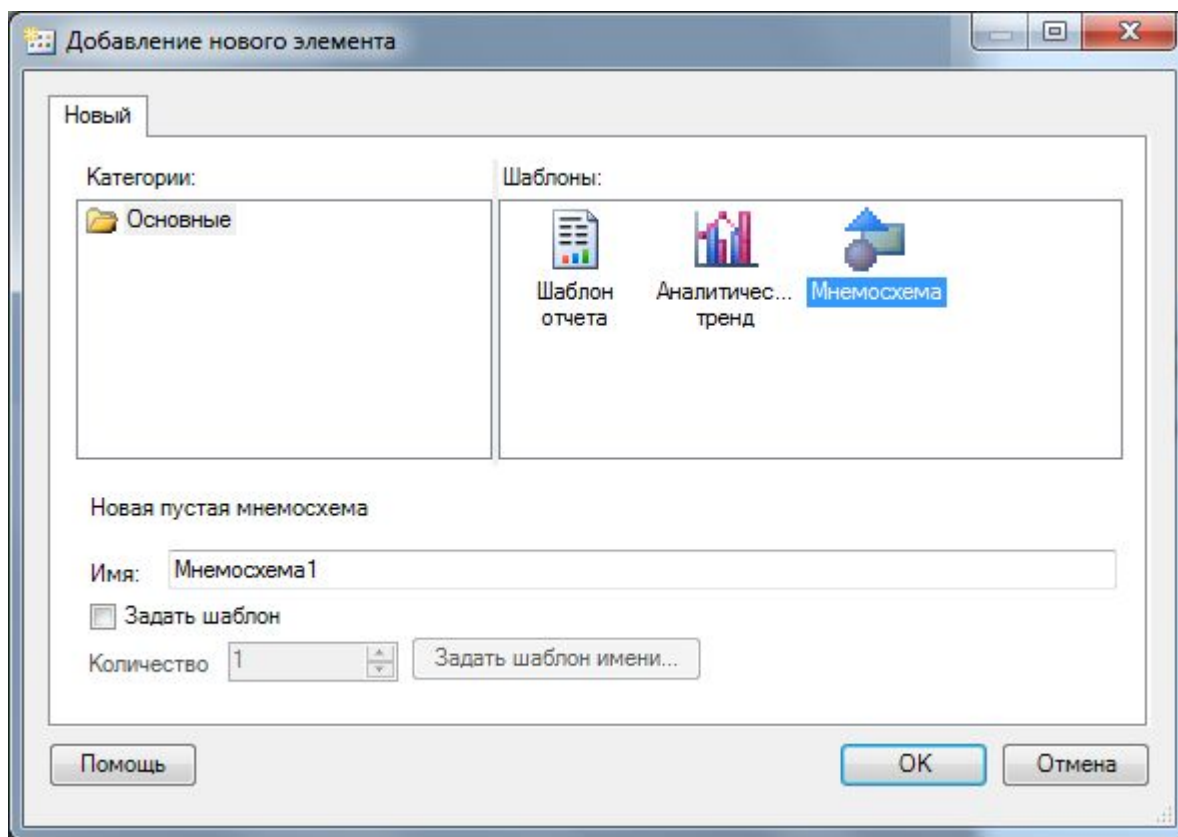
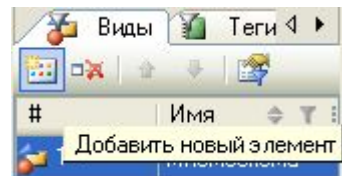
Мнемосхема – это анимированное изображение технологического объекта, состоящее из графических примитивов.

Мнемосхема является основной категорией вида объектов.

Графические примитивы – это основные объекты построения графических изображений с полным набором стандартных операций редактирования (таких как изменение размера; поворот; перемещение; настройка цвета, уровня прозрачности, степени сглаживания линий, процента заливки), возможностью анимации и динамической визуализации.

Создание

- 1 В контекстном меню объекта выполните команду **Создать/Вид** или в редакторе проекта на вкладке **Виды** для выбранного объекта нажмите кнопку **Добавить новый элемент**
- 2 В появившемся окне **Добавление нового элемента** выбрать категорию **Основные** и необходимый шаблон вида (например, Мнемосхема)
- 3 Ввести **Имя** вида и нажать **ОК**



Для создания нескольких однотипных видов воспользуйтесь флажком **Задать шаблон**. Введите количество видов и задайте шаблон имени создаваемых видов. Задать шаблон имени можно нажав на кнопку **Задать шаблон имени** (описание шаблона приведено в разделе «Формирование шаблона имени»).

Редактирование

Для редактирования графического изображения объекта выполните:

- Двойной щелчок на узле **Вид** в дереве проекта или
- Команду **Задачи/Редактировать** в контекстном меню.

ВНИМАНИЕ!!!

При переименовании вида ссылки на этот вид становятся недействительными. Для проверки необходимо запускать верификацию.

Индивидуальная настройка вида объекта


Если вы хотите индивидуально настроить вид объекта, который был порожден от библиотечного, и этот вид присутствует в библиотечном объекте, то свойство **Библиотечный** вида настраиваемого объекта нужно установить в **Нет**.

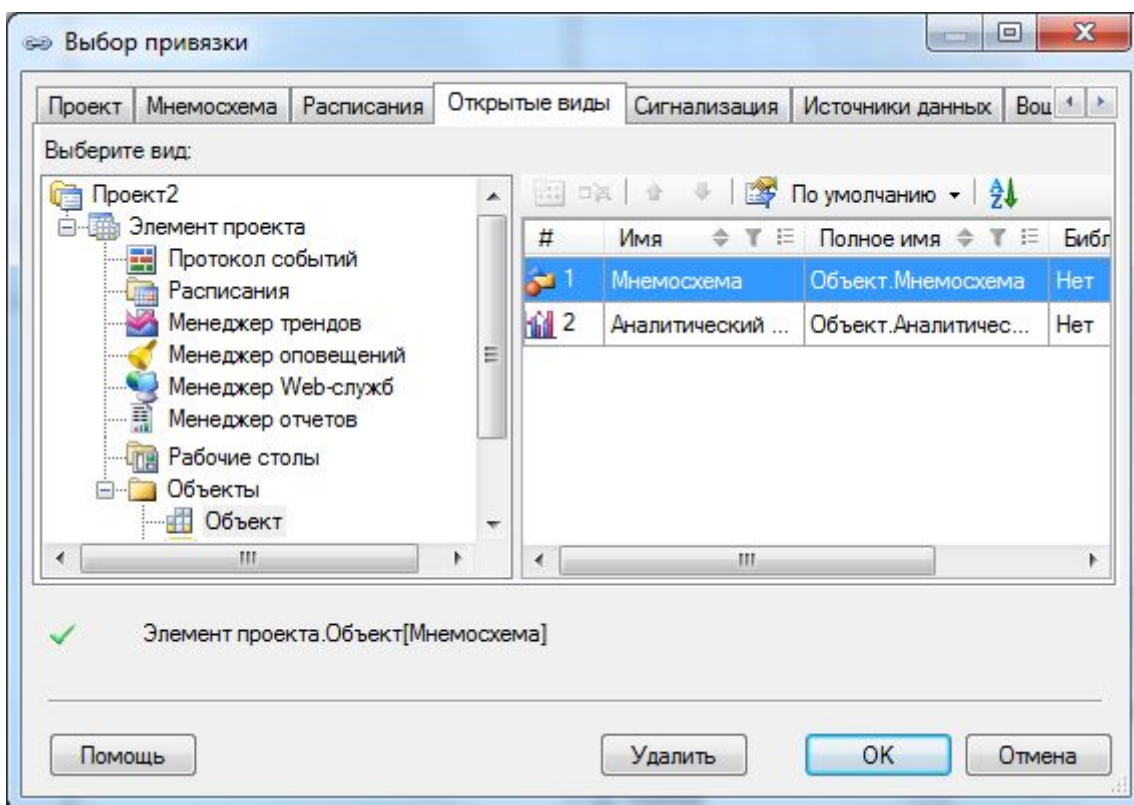
В случае установки свойства **Библиотечный** в **Да**, и при этом объект не имеет соответствующей библиотечной части, будет выдано сообщение: «**Элемент не может находиться в библиотеке**».

2.4.2.1 Привязка к открытию вида

Для привязки аргумента к открытию или закрытию вида предназначена вкладка **Открытые виды** окна **Выбор привязки**.

Для привязки аргумента к открытию и закрытию вида следует:

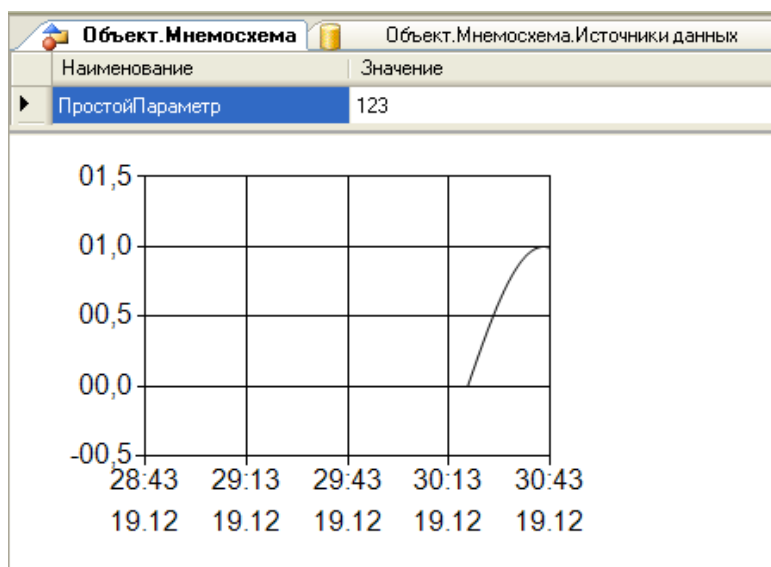
- 1 Вызвать окно **Выбор привязки**. Для этого в окне **Аргументы** (вызов по кнопке  панели инструментов) выделить ячейку в колонке **Привязки** и нажать на кнопку в ячейке
- 2 Выбрать вкладку **Открытые виды**
- 3 Выделить в дереве проекта объект (1), к видам которого нужно привязать аргумент
- 4 Выбрать необходимый вид (например, Мнемосхема) в списке дочерних элементов (2)



2.4.2.2 Параметры

Параметры предоставляет возможность настраивать свойства графических изображений объектов (видов) в процессе выполнения проекта.

Параметры используются совместно с **источниками данных**.



Каждый вид **DataRate** обладает коллекцией параметров, которые доступны для редактирования пользователем. Для любого вида можно задать до **100** различных параметров.

2.4.3 Скрипты

Скрипт – небольшая программа на языке C#.

С помощью скриптов можно:

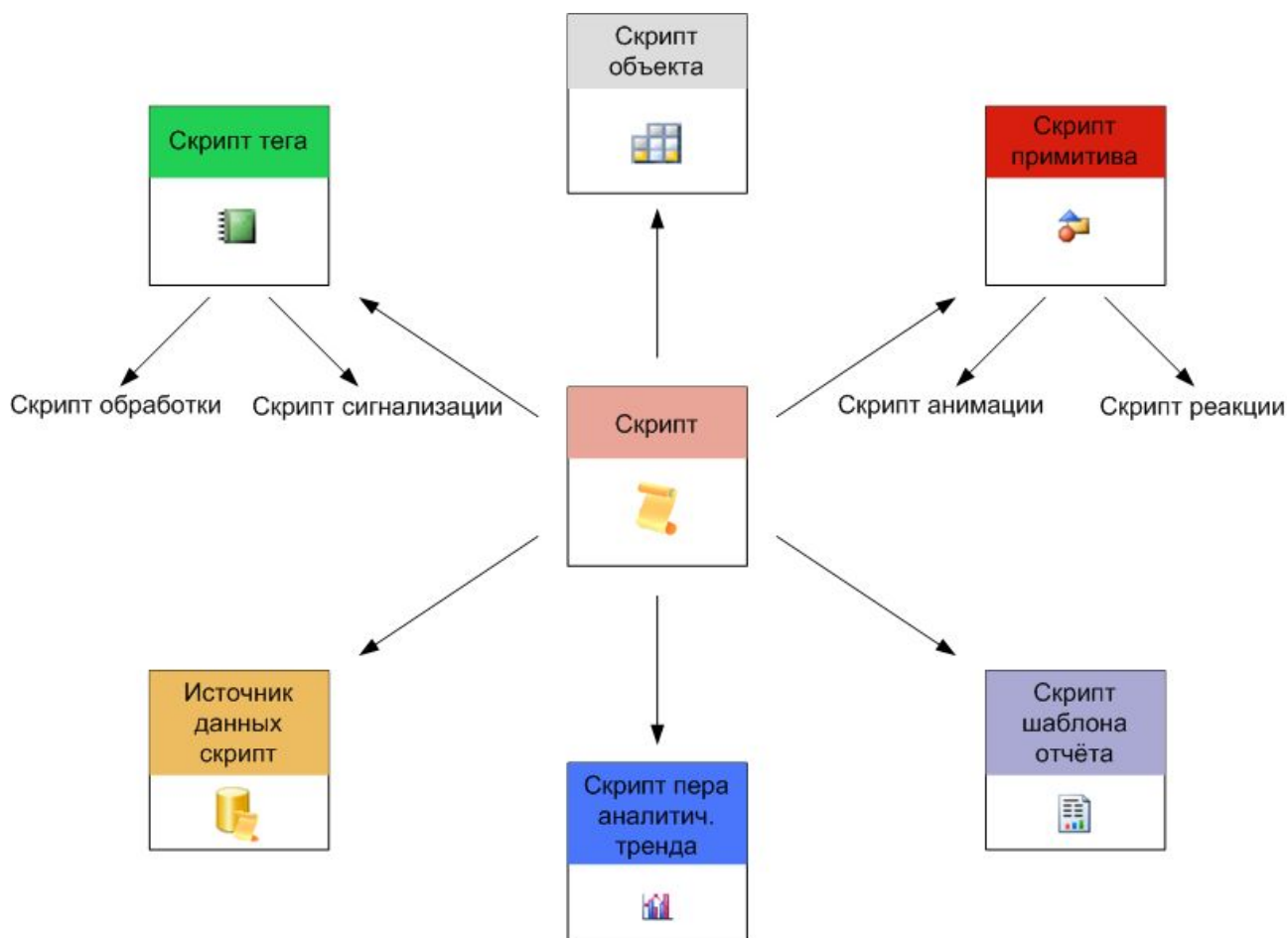
- Изменять свойства графического примитива
- Считывать/устанавливать значения аргументов
- Реализовать алгоритмы обработки данных, используя классы .NET Framework
- и многое другое.

Высокая скорость выполнения алгоритма, реализованного скриптом, достигается за счет предварительной компиляции скриптов в машинные коды.

Пример

Использование скрипта в управлении объектом:

```
// если пришла команда открыть и задвижка не открыта,
// и задвижка не находится в состоянии открытия, и не находится в
// состоянии закрытия
if( CommandOpen && !GateOpened && !GateOpening && !GateClosing )
{
    // устанавливаем состояние открытия задвижки
    GateOpening = true;
    // отправляем сообщение в роллинг о начале открытия задвижки
    AlarmServiceHelper.SendTrackingEvent(Client.OwnerObject, "ЗД_1_ОТК_С",
    "Вход", 3, "ЗД_1");
}
```



Для обращения к свойствам объектов в скрипте используется специальный объект **Client**.

Пример

Использования объекта **Client**

// присвоение значения свойству **Ширина** (*Width*)

```
Client.Width = 100;
```

//присвоение значения свойству **Текст**

```
Client.Text = "Норма";
```

//присвоение значения атрибуту свойства **Высота** (*Height*)

```
Атрибут1 = Client.Height + 100;
```

При наведении указателя мыши на слово "**Client**" во всплывающей подсказке показывается тип, соответствующий выбранному объекту.

Доступные свойства и методы объекта **Client** можно выбрать из раскрывающегося списка, который вызывается комбинацией **Ctrl+Пробел** или вводом символа «точка» за словом "**Client**".

Отображение свойств и методов в раскрывающемся списке обеспечивает механизм **Intellisense** (описание приведено в разделе «Редактор скриптов»).

Скрипт включает общий код и собственно код скрипта:

- **Код скрипта** – часть скрипта, содержащая тело функции, которая выполняется при анимации или в ответ на реакцию, заданную для объекта, с которым связан скрипт
- **Общий код** – часть скриптовой подсистемы любого объекта (примитив, тег и другие объекты, поддерживающие скрипты). Поля, методы, константы, свойства, события, делегаты и классы, расположенные в общем коде, доступны внутри скрипта (вкладки: Скрипт, Запуск, Останов):
 - **Запуск** – часть скрипта, выполняемая один раз, только при переходе в режим исполнения
 - **Останов** – часть скрипта, выполняемая один раз, только при выходе из режима исполнения
 - **Пространства имен** – часть скрипта, содержащая список используемых в скрипте пространств имён (Namespaces).

ВНИМАНИЕ!!!

Для скриптов в анимациях и реакциях следующие части являются общими: Общий код, Запуск, Останов и Пространства имен. Т.е. все анимации и реакции разделяют вышеуказанные части. Изменения этих частей в одном скрипте повлияют на другие скрипты в пределах примитива.

Например, при смене типа анимации со скрипта на простое преобразование и обратно затрется только текст на вкладке Скрипт, на остальных 4-х вкладках текст останется прежним. Сделано это для оптимизации загрузки и исполнения скриптов при большом их количестве.

Код скрипта и его выполнение

№	Вид скрипта	Аргументы	Скрипт	Общий код	Запуск	Останов
1	Скрипт объекта	Атрибуты тегов, Свойства оповещений, Открытые виды, Расписания, Сигнальные состояния и их свойства, Вошедший пользователь	Код выполняется при изменении аргументов, напротив которых выставлен флаг "Запустить скрипт"	Содержит определение функций, переменных и констант, которые используются в скрипте	Выполняется при запуске графического проекта	Выполняется при завершении работы графического проекта
2	Скрипт анимации примитива	Атрибуты тегов, Свойства оповещений, Открытые виды, Расписания, Сигнальные состояния и их свойства, Вошедший пользователь	Код выполняется при изменении любого из аргументов скрипта		Выполняется при открытии вида	Выполняется при закрытии вида
3	Скрипт реакции примитива	Свойства примитивов, Источники данных, Мнемосхемы, Параметры мнемосхем, Составной параметр	Код выполняется при наступлении события указанного в реакции			
4	Скрипт шаблона отчёта	Атрибуты тегов, Свойства оповещений, Открытые виды, Расписания, Сигнальные состояния и их свойства, Вошедший пользователь, Свойства примитивов, Источники данных, Мнемосхемы, Параметры мнемосхем, Составной параметр	Код выполняется при изменении аргументов, напротив которых выставлен флаг "Запустить скрипт"		Выполняется при запуске графического проекта	Выполняется при завершении работы графического проекта
5	Скрипт пера аналитического тренда	Атрибуты тегов, Свойства оповещений, Открытые виды, Расписания, Сигнальные состояния и их свойства, Вошедший пользователь, Свойства примитивов, Источники данных аналитического тренда, Параметры аналитического тренда, Составной параметр	Код выполняется при изменении массива данных на отображенных на тренде		Выполняется при открытии аналитического тренда	Выполняется при закрытии аналитического тренда
6	Источник данных скрипт	Атрибуты тегов, Свойства оповещений, Открытые виды, Расписания, Сигнальные состояния и их свойства, Вошедший пользователь, Свойства примитивов, Источники данных аналитического тренда, Параметры аналитического тренда, Составной параметр	Код выполняется при открытии вида и затем с периодом указанным в настройках (если период не равен 0)		Выполняется при открытии вида	Выполняется при закрытии вида
7	Скрипт обработки тега	Атрибуты тега	Код выполняется при изменении аргументов, напротив которых выставлен флаг "Запустить скрипт"	–	–	–
8	Скрипт сигнализации тега	Атрибуты тега, Сигнальные подсостояния				

Общий код скрипта

К общему коду скрипта относятся программные секции: **общий код, запуск, останов.**

Секция **Общий код** содержит определение функций, переменных и констант, которые используются в скрипте.

Пример

Определение переменной целого типа **tickers**, объекта **timer** класса **Timer** и обработчика **OnTimer** для его события **Tick**

```
System.Windows.Forms.Timer timer;
int tickers;
void OnTimer(object sender, EventArgs e)
{
    tickers += 1;
}
```

Секция **Запуск** содержит программный код, который выполняется при запуске проекта автоматизированной системы, создает объекты, определенные в общем коде, и выполняет их инициализацию.

Пример

Инициализация таймера. Определение таймера приведено выше, в примере общего кода.

```
timer = new System.Windows.Forms.Timer();
timer.Interval = 1000;
timer.Enabled = true;
timer.Tick += new EventHandler(OnTimer);
```

Секция **Останов** содержит программный код, который выполняется при завершении работы проекта, и освобождает ресурсы, выделенные в коде запуска.

Пример

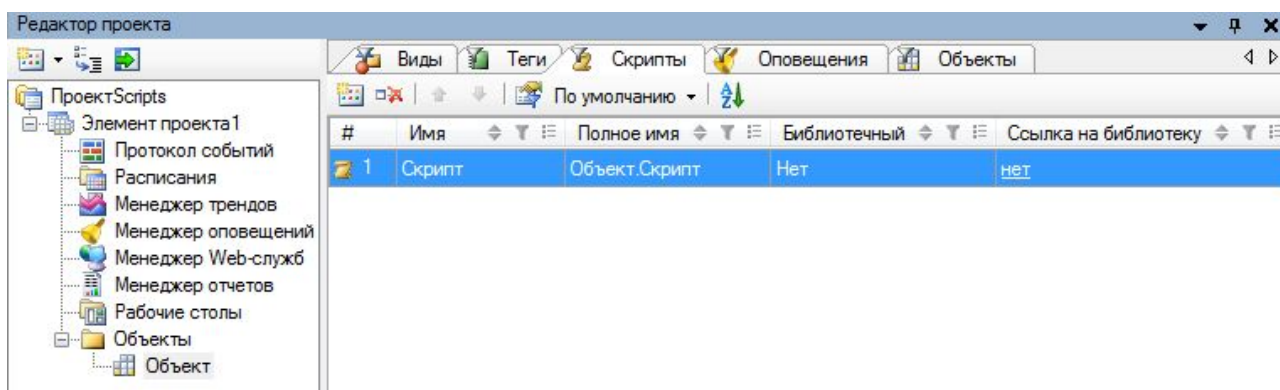
Остановка таймера

```
timer.Enabled = false;
timer.Tick -= new EventHandler(OnTimer);
```


Как посмотреть текст скрипта?

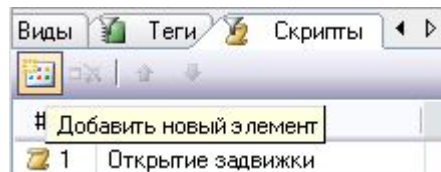
Тест скрипта можно посмотреть следующим образом:

- 1 В Редакторе объектов выделить в дереве проекта объект, скрипты которого Вас интересуют
- 2 В списке дочерних элементов узла дерева открыть вкладку **Скрипты**, в которой содержится список скриптов объекта.
- 3 Двойным щелчком на имени скрипта или с помощью **Браузера скриптов** открыть скрипт для просмотра и редактирования



Создание


- Скрипт объекта можно создать, нажав на кнопку **Добавить новый элемент**  в списке дочерних элементов узла дерева или
- Скрипт можно также создать на основе скрипта из библиотеки, используя операцию Drag&Drop.



Редактирование

Для редактирования скрипта используется редактор скриптов.

Аргументы

Каждый скрипт содержит коллекцию аргументов. Задание аргументов осуществляется в окне **Аргументы**. Для вызова окна нажмите на кнопку  панели инструментов.

Выражение

Выражение – это последовательность операций и операндов, которая определяет вычисляемое значение.

Выражения используются в тексте скрипта, в привязке **Простое преобразование**, в реакции **Присвоить значение**.

Индивидуальная настройка скрипта

Если вы хотите индивидуально настроить скрипт объекта, который был порожден от библиотечного, и этот скрипт присутствует в библиотечном объекте, то свойство **Библиотечный** скрипта настраиваемого объекта нужно установить в **Нет**.

В случае установки свойства **Библиотечный** в **Да**, и при этом объект не имеет соответствующей библиотечной части, будет выдано сообщение «**Элемент не может находиться в библиотеке**»

2.4.3.1 Браузер скриптов


Браузер скриптов предназначен для быстрого поиска и выбора скриптов в проекте.

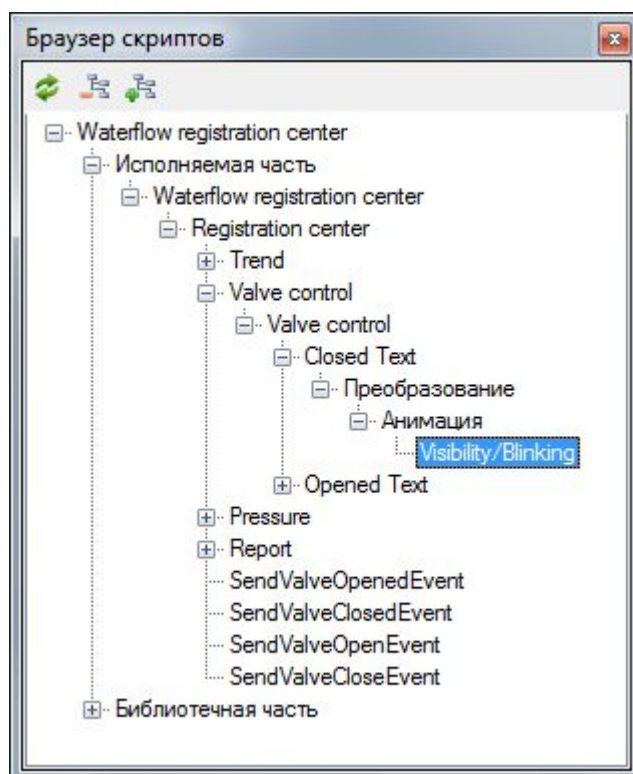
В окне **Браузер скриптов** отображается дерево скриптов.

Каждый узел в дереве – это объект или его дочерние элементы, которые имеют скрипты (например, объекты, виды).

При двойном щелчке по узлу дерева, открывается соответствующий скрипт.

Мнемосхема, которая имеет динамику, также отображается как узел дерева, раскрыв который можно пройти по всем графическим элементам, имеющим динамику.

Для обновления состояния браузера скриптов нажмите на кнопку **Обновить** – .



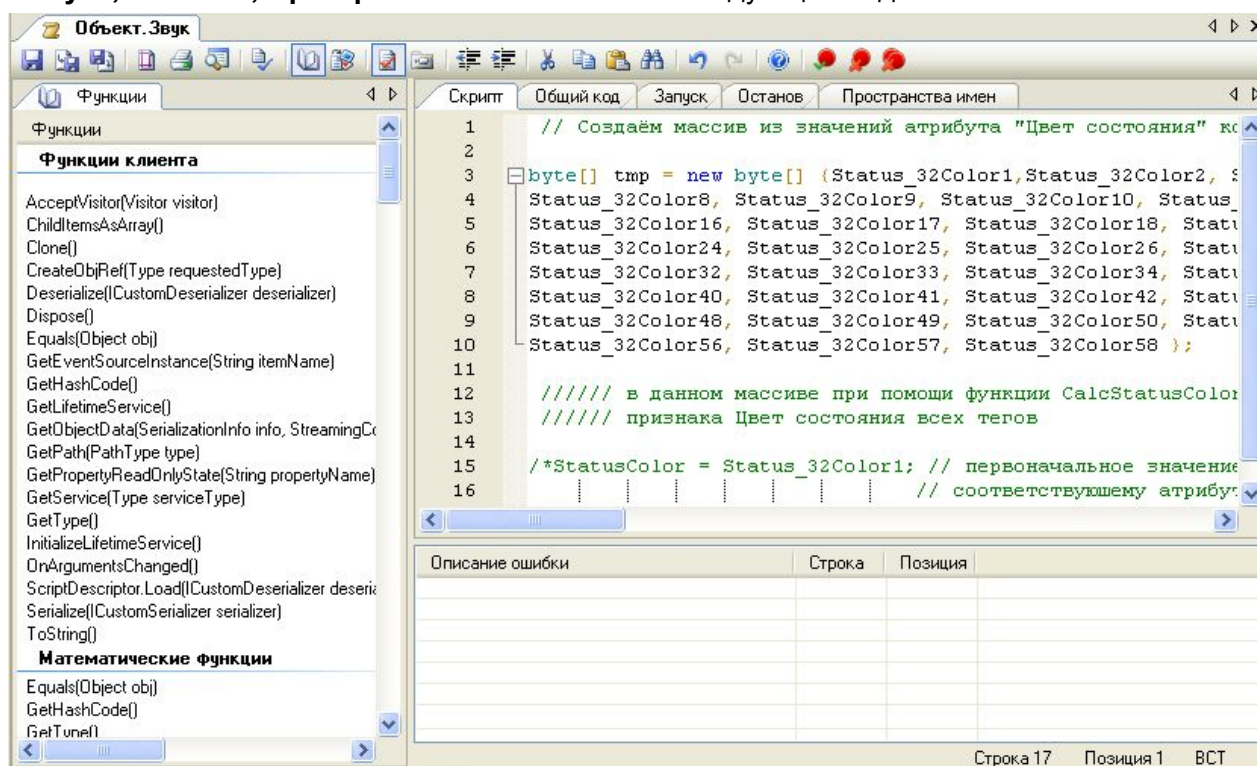
2.4.3.2 Редактор скриптов

Редактор скриптов предназначен для создания и редактирования текста скрипта.

Редактор обеспечивает:
















- Сохранение текста скрипта и загрузку скрипта из файла
- Эффективное редактирование текста (удалить/копировать/вставить, поиск/замена, проверка синтаксиса и другие операции)
- Печать текста скрипта (предварительный просмотр и настройка параметров страницы)
- и другие операции, необходимые для редактирования текста скрипта

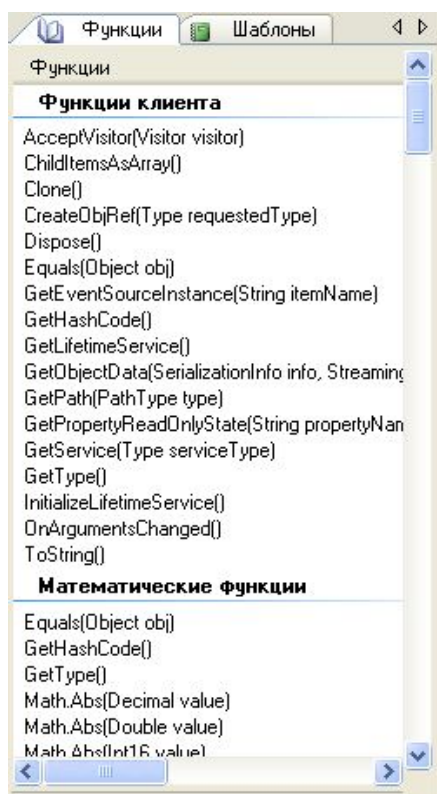
Окно редактора скриптов включает закладки: **Функции**, **Шаблоны**, **Скрипт**, **Общий код**, **Запуск**, **Останов**, **Пространство имен** и имеет следующий вид.



Панель инструментов состоит из следующих кнопок:

	сохранить скрипт – сохраняет изменения, сделанные в скрипте
	экспортировать скрипт – сохраняет текст скрипта в текстовый файл с расширением .script
	импортировать скрипт – загружает текст скрипта из выбранного файла, заменяя существующий текст
	параметры страницы – вызывает диалоговое окно настройки параметров страницы для вывода на печать
	параметры печати – вызывает диалоговое окно настройки параметров печати
	предварительный просмотр – вызывает диалоговое окно предварительного просмотра печати
	проверяет скрипт на ошибки – если ошибки есть, показывается окно ошибок с описанием, если нет – выводится сообщение Проверка прошла успешно
	ссылки на сборки – открывает окно редактирования ссылок на сборки

	закладка Функции – скрывает/показывает закладку Функции
	закладка Шаблоны – скрывает/показывает закладку Шаблоны
	скрыть/показать окно ошибок скрипта
	уменьшить отступ – уменьшает отступ для выделенного текста скрипта
	увеличить отступ – увеличивает отступ для выделенного текста скрипта
	вырезать – удаляет выделенный текст с помещением его в буфер обмена
	копировать – копирует текст в буфера обмена
	вставить – вставляет текст из буфера обмена
	найти – открывает диалоговое окно поиска и замены
	отменить – отменяет сделанные изменения
	вернуть – возвращает отмененные изменения
	помощь – открывает справку
	вставить точку останова в строку редактора скрипта, в которой установлен курсор
	удалить точку останова из строки редактора скриптов, в которой установлен курсор
	удалить все точки останова из текущего редактора скриптов

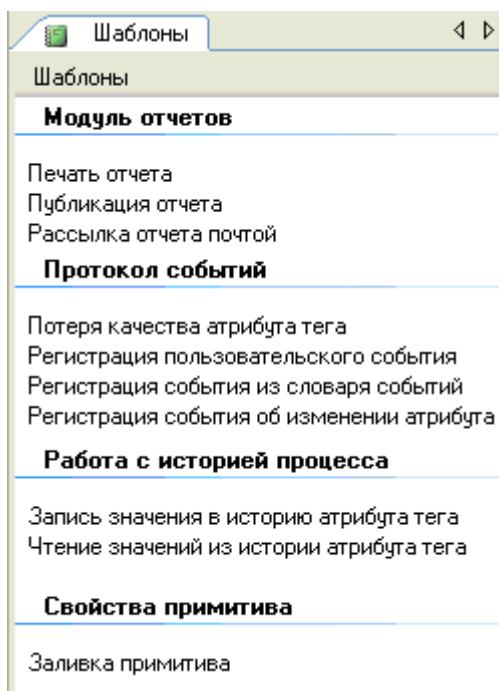


Закладка Функции предназначена для вставки вызова функции в текст скрипта.

Разделы закладки **Функции**:

- **Функции клиента** содержит функции предоставляемые **объектом-клиентом скрипта**.
- **Математические функции** содержит функции, определенные в классе **Math** из библиотеки NET Framework.
- **Функции работы с датой и временем** содержит функции, определенные в классе **DateTime** из библиотеки NET Framework.
- **«Функции работы со строками»** содержит функции, определенные в классе **String** из библиотеки NET Framework.

Для добавления нужной функции в текст скрипта установите курсор в поле ввода текста в то место, куда нужно добавить функцию, и дважды щелкните левой клавишей мыши на выбранной функции в закладке **«Функции»**.



Закладка Шаблоны предназначена для вставки заранее определенных пользовательских шаблонов в текст скрипта


Пользовательский шаблон - это текст, который пользователь заранее задает в **окне настройки пользовательских шаблонов**.

Шаблон характеризуется:

- **Группой**, к которой относится шаблон
- **Именем**, которое показывается в закладке
- **Описанием**, которое отображается на всплывающей подсказке при наведении указателя мыши на имя шаблона
- **Текстом**, который непосредственно вставляется в текст скрипта.

Для добавления нужного шаблона в текст скрипта установите курсор в поле ввода текста в то место, куда нужно добавить шаблон, и дважды щелкните левой клавишей мыши на выбранном шаблоне в закладке **Шаблоны**.

Окно ошибок скрипта предназначено для вывода сообщения об ошибке при проверке скрипта.

Окно открывается автоматически при нахождении ошибки во время проверки скрипта или кнопкой  на панели инструментов.

Закрывается окно при успешной проверке скрипта или щелчком по той же самой кнопке .

Окно ошибок содержит таблицу с тремя столбцами:

- **Описание ошибки** – содержит описание ошибки;
- **Строка** – содержит номер строки, в которой произошла ошибка;
- **Позиция** – показывает, в какой части скрипта произошла данная ошибка.

Описание ошибки	Строка	Позиция	
Имя 'I' отсутствует в текущем контексте	12	Скрипт	
Имя 'A' отсутствует в текущем контексте	12	Скрипт	
Имя 'B' отсутствует в текущем контексте	12	Скрипт	

При двойном щелчке левой клавиши мыши на строке с ошибкой открывается соответствующая вкладка в области ввода текста скрипта, и курсор позиционируется на строке, в которой произошла ошибка.

Область редактирования текста предназначена для ввода и редактирования текста скрипта и содержит вкладки, соответствующие секциям скрипта.


```

1  // Создаём массив из значений атрибута "Цвет состояния" контроллера
2
3  byte[] tmp = new byte[] { Status_32Color1, Status_32Color2, Status_32Color3,
4  Status_32Color8, Status_32Color9, Status_32Color10, Status_32Color11,
5  Status_32Color16, Status_32Color17, Status_32Color18, Status_32Color19,
6  Status_32Color24, Status_32Color25, Status_32Color26, Status_32Color27,
7  Status_32Color32, Status_32Color33, Status_32Color34, Status_32Color35,
8  Status_32Color40, Status_32Color41, Status_32Color42, Status_32Color43,
9  Status_32Color48, Status_32Color49, Status_32Color50, Status_32Color51,
10 Status_32Color56, Status_32Color57, Status_32Color58 };
11
12  // в данном массиве при помощи функции CalcStatusColor определены
13  // признаки Цвет состояния всех тегов
14
15  /*StatusColor = Status_32Color1; // первоначальное значение значения
16  // соответствующему атрибуту "Цвет состояния"
  
```

Аргументы скриптов

Каждый скрипт содержит свой набор аргументов. В любом скрипте существует аргумент с именем **Client**, тип которого соответствует типу объекта скрипта.

Список свойств и методов любого аргумента можно просмотреть с помощью механизма **Intellisense**.

Например, в скрипте "Обработка" для тега аргументами являются объекты строкового типа с именами **ProjectName**, **ObjectName**, **TagName**, **AttributeName** и объекты типа **AttributeValue** с именами **OldValue** и **NewValue**.

При этом в скрипте может быть вызван метод регистрации произошедших изменений:

```

AlarmServiceHelper.SendTrackingEvent (
    "Изменение значения атрибута {0}:{1}:{2}:{3} с {4} на {5}",
    ProjectName, ObjectName, TagName, AttributeName, OldValue.AsString,
    NewValue.AsString
);
  
```

Дополнительные возможности редактора скриптов

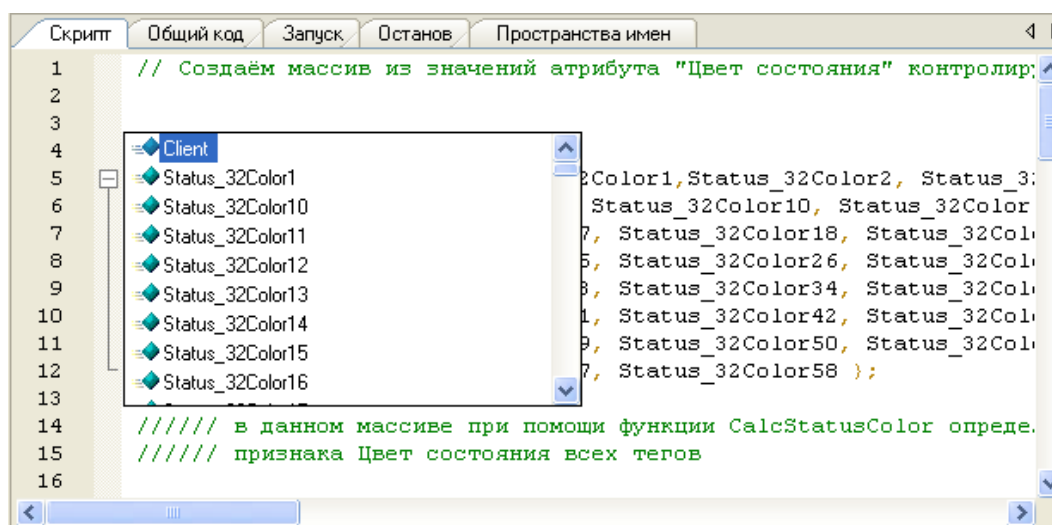
Редактор текста в области ввода скрипта поддерживает **подсветку** скрипта для **языка программирования C#**.

Для облегчения ввода скрипта реализован **механизм Intellisense** (всплывающий список с доступными в данный момент элементами в зависимости от контекста), **автоматически отступ** при переходе на новую строку, **«горячие» клавиши**.

Для определения типа объекта используемого в скрипте необходимо подвести указатель мыши на объект. Всплывающая подсказка покажет описание объекта (класс/множество/интерфейс/пространство имен и т.п.) и его тип (полное имя в сборке).

Механизм Intellisense

Для использования **механизма Intellisense** нужно нажать сочетание клавиш **Ctrl+Space**, и выбрать из списка нужный элемент



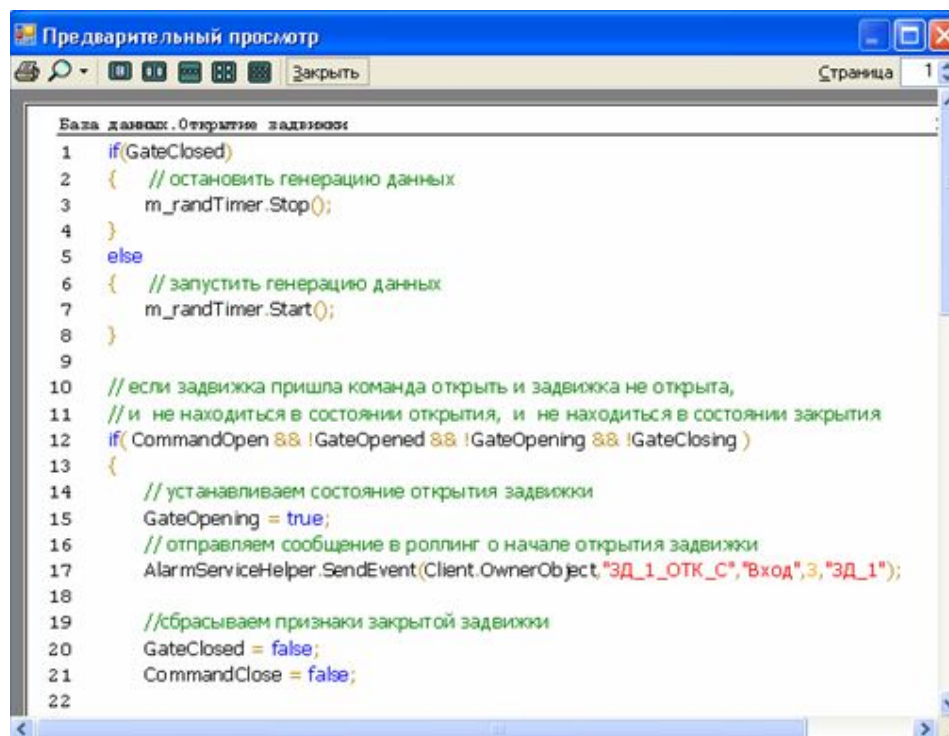
Для определения типа объекта, который показан в списке элементов, необходимо подвести указатель мыши на объект. Всплывающая подсказка покажет описание объекта и его тип.

«Горячие» клавиши

- Ctrl+S** – сохраняет изменения скрипта;
- Ctrl+F** – открывает диалоговое окно «поиск и замена» на закладке «Поиск»;
- Ctrl+H** – открывает диалоговое окно «поиск и замена» на закладке «Замена»;
- Ctrl+P** – открывает диалоговое окно «Печать»;
- F3** – если есть выделенный текст, ищет такой же текст ниже по скрипту, если находит – выделяет его, если не находит – выводит сообщение; если выделенного текста нет, открывает диалоговое окно «Поиск и замена».

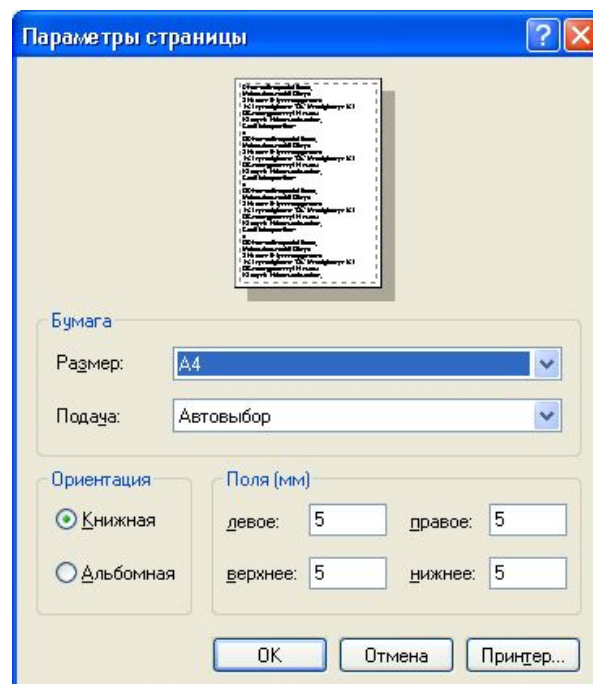
Окно Предварительный просмотр

позволяет предварительно, перед печатью, посмотреть, как будет выглядеть текст скрипта на странице.



Окно Параметры страницы предназначено для настройки параметров страницы при выводе скрипта на печать. Окно позволяет настроить размер бумаги, ориентацию и поля страницы.

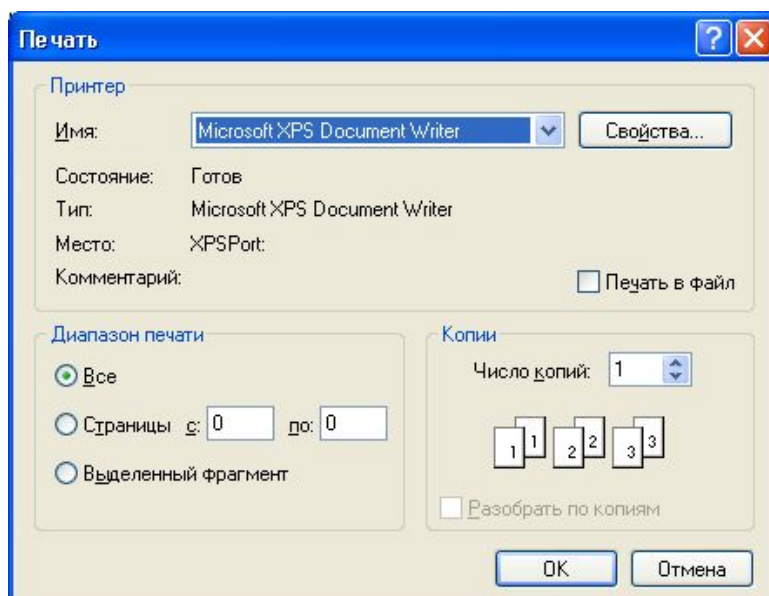
Кнопка **Принтер...** предназначена для вызова окна настроек принтера.



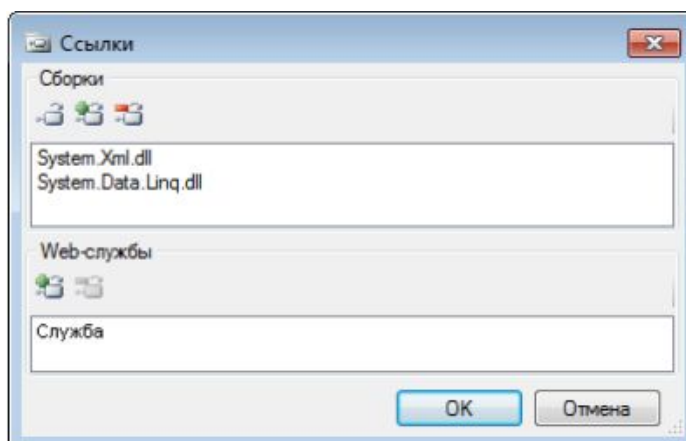
Окно Печать предназначено для настройки принтера и параметров печати скрипта.

Управляющие элементы:

- **Принтер** – выбор принтера из списка доступных в системе и настройка его свойств
- **Диапазон печати** – указание страниц для печати (все, по заданному номеру, выделенный фрагмент)
- **Копии** – задание количества копий, которое нужно напечатать.



Окно Ссылки предназначено для добавления и удаления ссылок на сборки.



Панель инструментов

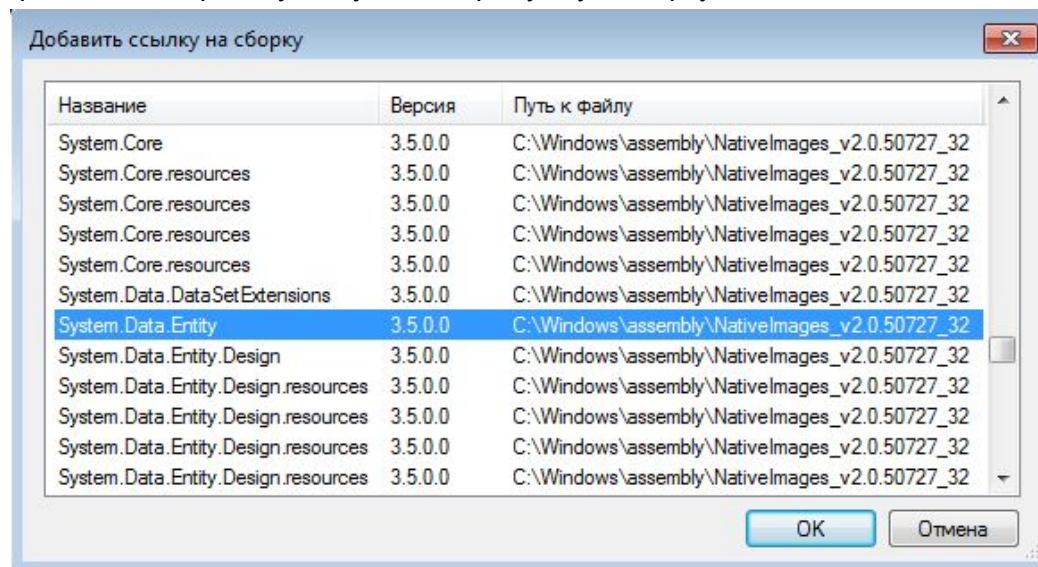
Сборки:



Выбрать сборку из глобального кэша сборок – позволяет выбрать сборку для общего пользования из глобального кэша сборок;



Выбрать сборку самостоятельно – открывает диалоговое окно выбора файла, в котором нужно указать требуемую сборку.

**ВНИМАНИЕ!!!**

Если Вы используете сборку не из папки Bin, в которой находятся все сборки, используемые в DataRate, то при переносе проекта на другую машину необходимо перенести также все используемые сборки, иначе скрипт при выполнении будет выдавать ошибку «Сборка не найдена».

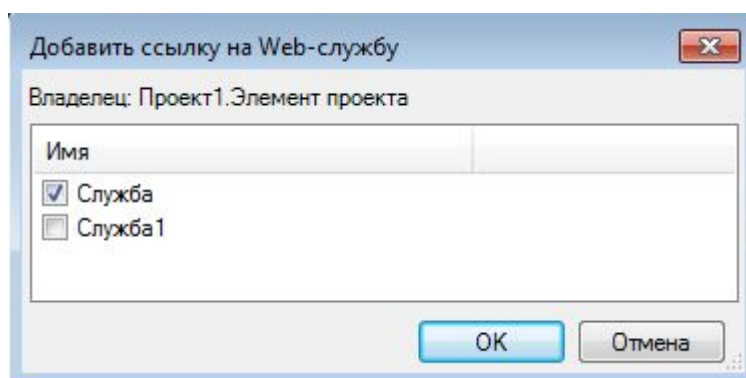


Удалить ссылку на сборку – удаляет ссылку на сборку. Если Вы удалите ссылку на сборку, которая используется в скрипте, то при выполнении скрипта генерируется ошибка **Сборка не найдена**.

Web-службы:



Выбрать Web-службу – позволяет выбрать Web-службы, которые будут использоваться в скрипте. При этом открывается окно добавления ссылок на Web-службы.

**ВНИМАНИЕ!!!**

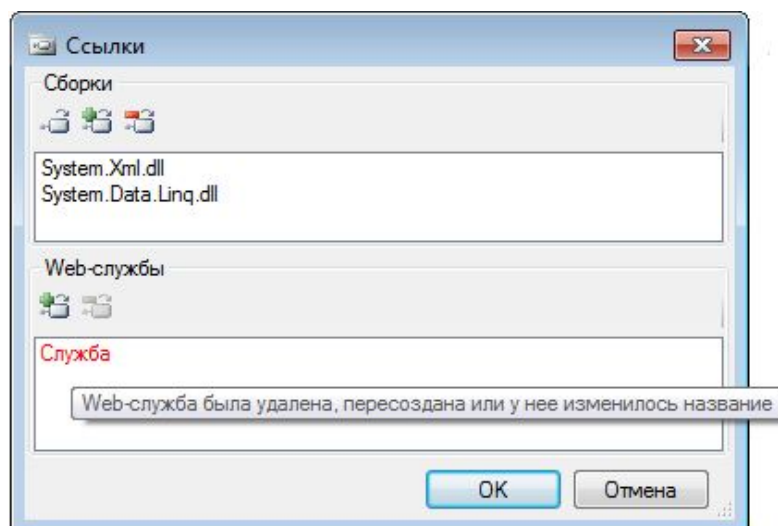
Кнопка доступна лишь тогда, когда элемент проекта содержит Web-службы и их перечень не совпадает со списком, представленным в группе "Web-службы".



Удалить ссылку на Web-службу – удаляет ссылку на Web-службу.

ВНИМАНИЕ!!!

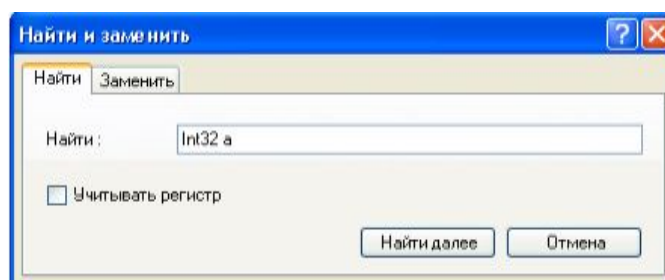
Если Вы удалите ссылку на Web-службу, которая используется в скрипте, при выполнении скрипт не запустится, а верификация сгенерирует ошибку "Web-служба <полное имя Web-службы> была удалена или пересоздана. Удалите или перепривяжите ссылку, скомпилируйте скрипт". Если по каким-либо причинам Web-служба была удалена, пересоздана или переименована, то ее название будет выделено красным цветом, а при наведении курсора выводится соответствующая подсказка.



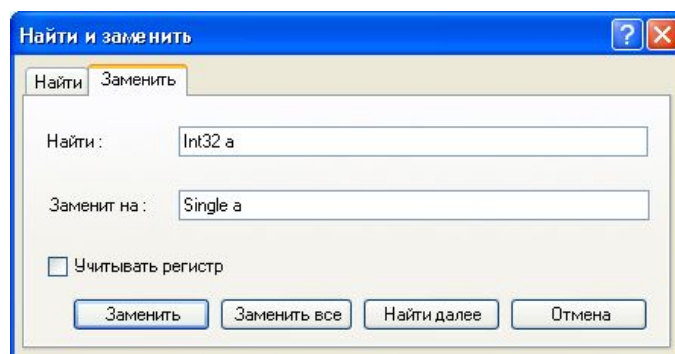
Чтобы выйти из окна без сохранения следует нажать на кнопку **Отмена**. Нажатие на кнопку **ОК** полностью обновляет список ссылок в скрипте.

Окно Найти и Заменить предназначено для поиска и замены набора символов в скрипте.

Вкладка **Найти** позволяет найти вхождения заданного набора символов в тексте скрипта с учетом регистра или без. При этом найденный набор символов будет подсвечиваться. Поиск начинается с начала текста скрипта. При достижении конца текста выдается сообщение.




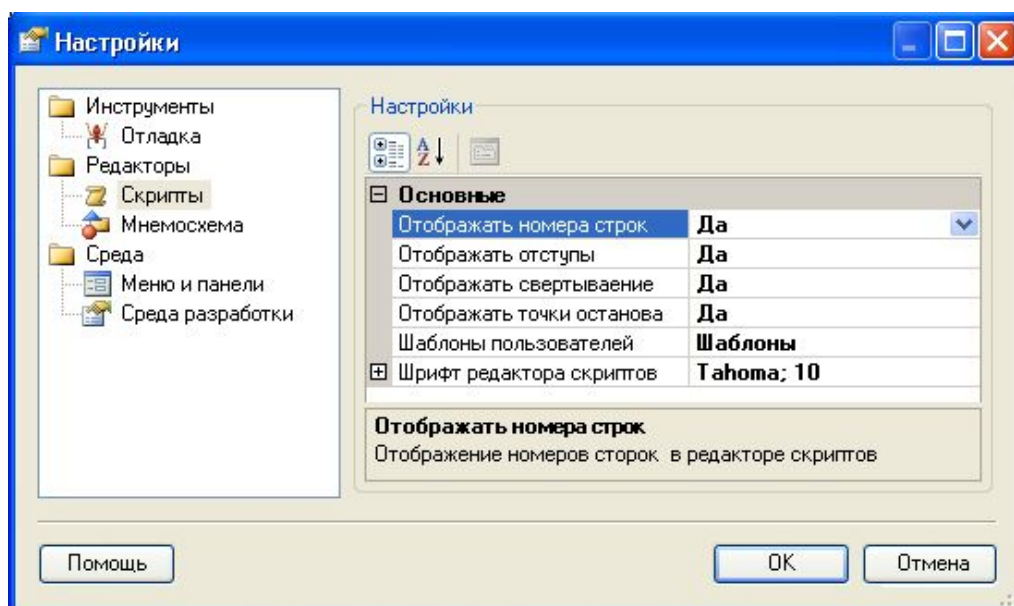
Вкладка **Заменить** позволяет заменить заданный набор символов другим набором с возможностью заменить все или выборочно.



Параметры настройки Редактора скриптов

Для настройки **Редактора скриптов** следует:

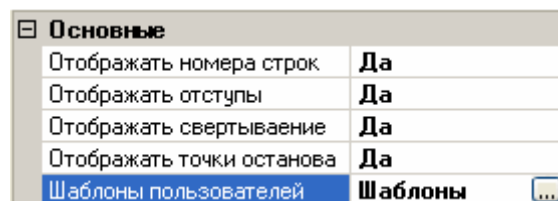
- 1 Вызвать окно **Настройки**. Для этого нажать на кнопку  на панели инструментов или выбрать **Сервис/Настройки** в главном меню. Также окно **Настройки** можно вызвать по комбинации клавиш **Ctrl+Alt+Shift+S**
- 2 В окне **Настройки** выбрать **Редакторы/Скрипты** и задать значения параметров настройки



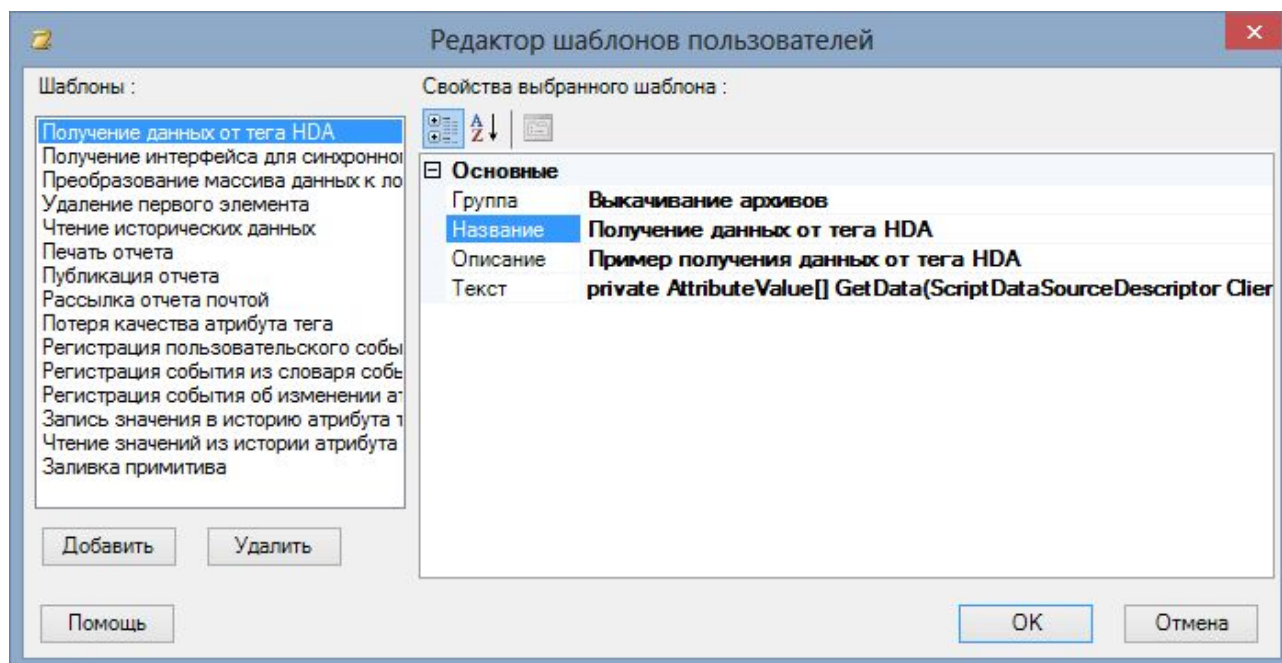
Настройки Шаблонов пользователей

Для настройки шаблонов в окне **Настройки** следует:

- 1 Выбрать свойство **Шаблоны пользователей** и нажать на появившуюся кнопку в строке свойств



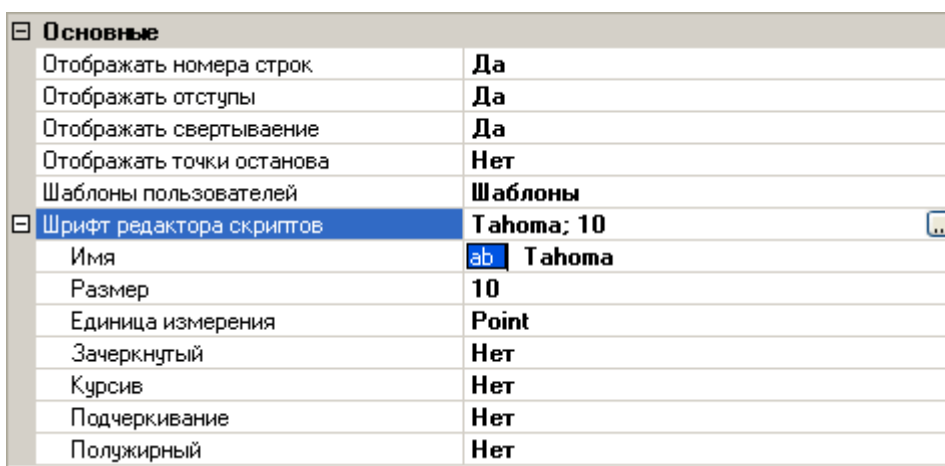
- 2 В открывшемся окне **Редактор шаблонов пользователей** задать свойства шаблона.



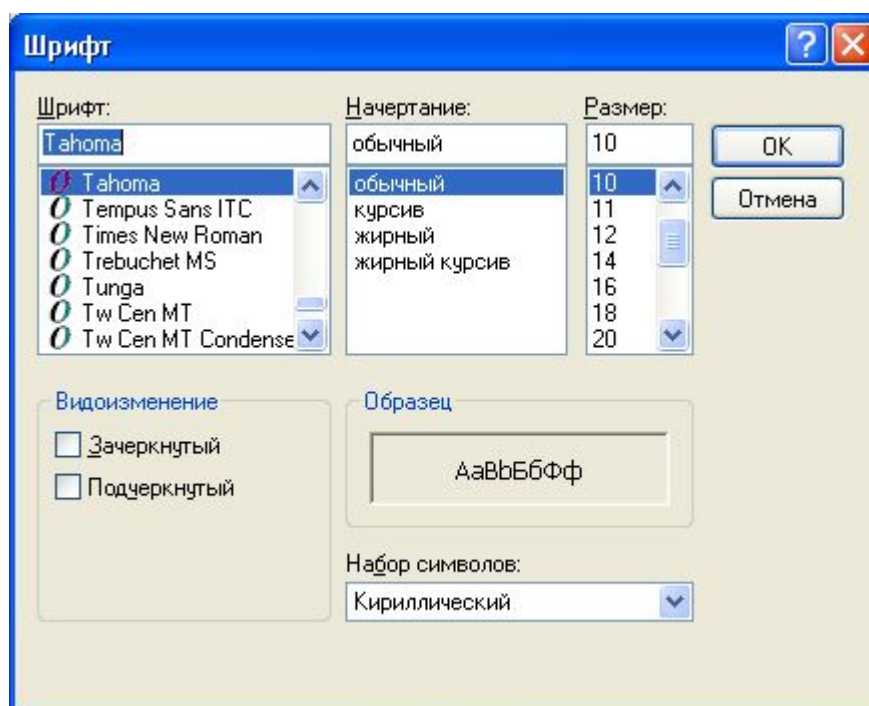
Настройки шрифта

Для настройки шрифта области редактирования текста следует:

1. Выбрать в окне **Настройки** свойство **Шрифт** редактора скриптов (при этом справа появится кнопка с многоточием)
2. Нажать на кнопку в строке свойств



3. В появившемся окне **Шрифт** настроить шрифт для редактора скриптов.



2.4.3.3. Поддержка языка КРУГОЛ

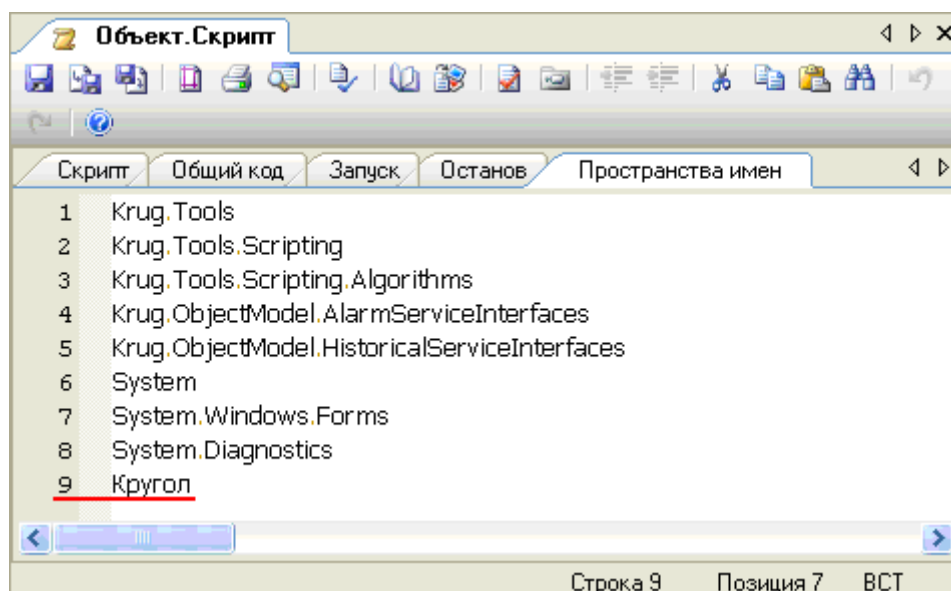
В проектах *DataRate* можно использовать функции библиотеки языка **КРУГОЛ**.

ВНИМАНИЕ!!!

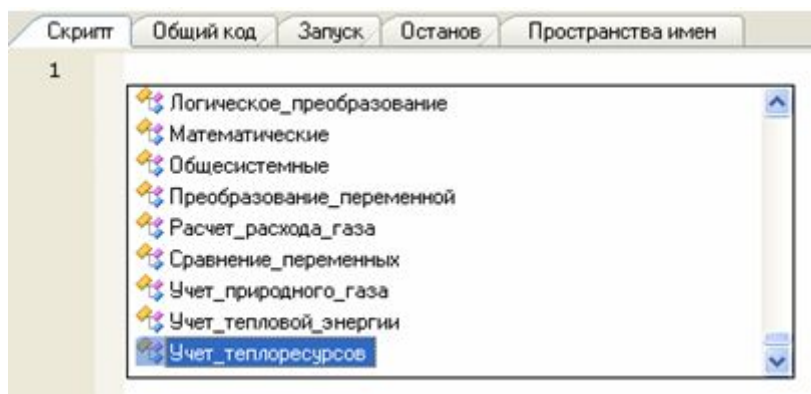
Для того, что бы пространство имён Кругол стало доступным, после установки *DataRate* необходимо установить Библиотеку функций языка КРУГОЛ (версия не ниже 2.1 SP1). Описание функций языка КРУГОЛ можно посмотреть в файле `funclib.hlp`, который устанавливается вместе с библиотекой.

Для использования функций **КРУГОЛ** в проекте *DataRate* следует:

1. Выбрать во вкладке **Пространства имён** пространство **Кругол**



- 2 После этого с помощью механизма **Intellisense** добавить требуемую функцию языка КРУГОЛ из выбранной библиотеки.



Пример

Вызов функции в интегрированной среде разработки КРУГОЛ:

```
a1 = Знак_в64(a2)
```

Вызов функции в **DataRate**:

```
double d;  
d = Математические.Знак_в64(a2);  
a1 = d;
```

ВНИМАНИЕ!!!

В **DataRate** невозможно использовать атрибут тега в качестве выходного параметра функции. Для этого необходимо использовать переменную. Первый выходной аргумент записывается слева от знака =, а остальные записываются в качестве аргументов функции с модификатором out. Например, вызов функции с 3-мя выходными аргументами будет иметь вид:

```
short a,b,c;  
a = Дата_и_время.СВремя(out b, out c);  
a1 = a; a2 = b; a3 = c;
```

2.4.3.4 Отладчик скриптов

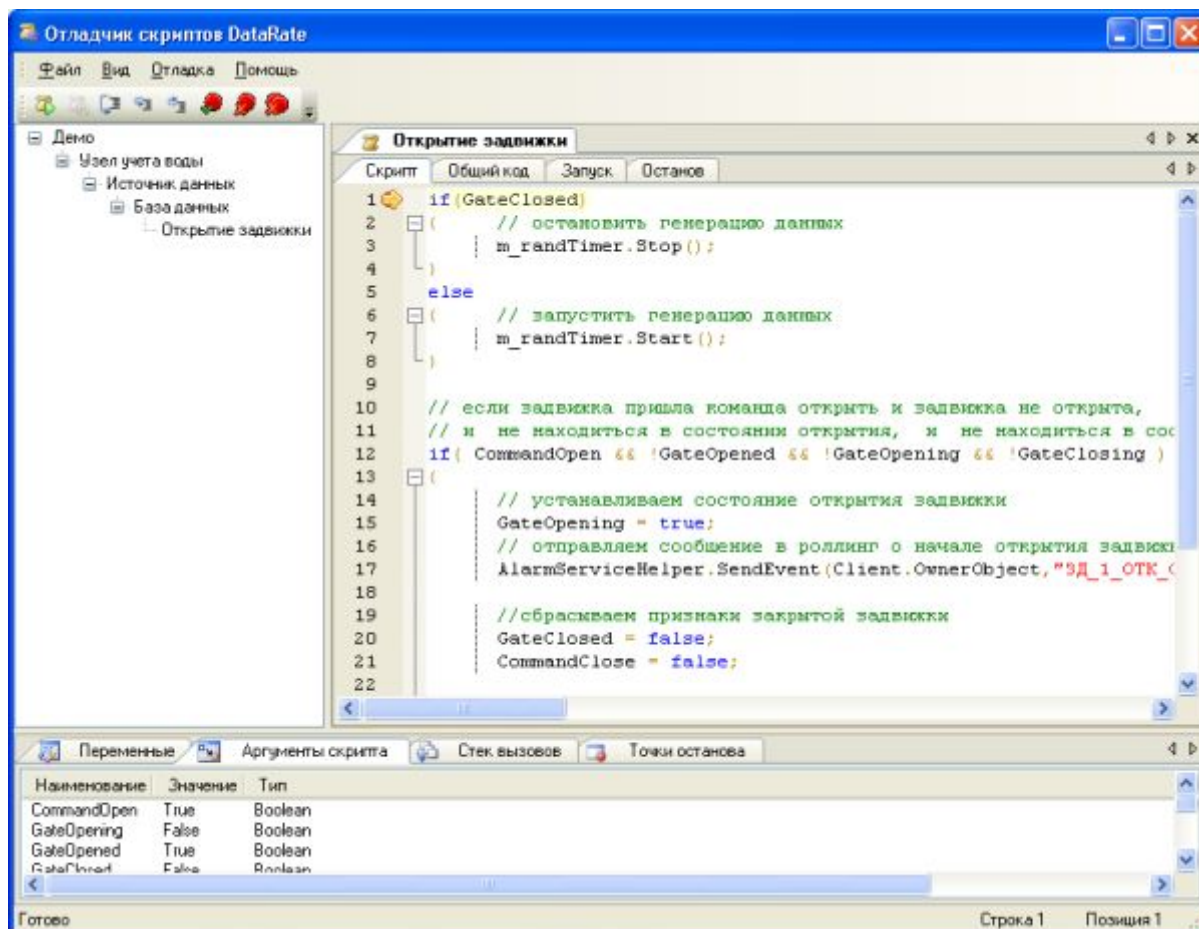
Отладчик скриптов – это модуль среды разработки, предназначенный для проверки правильности выполнения скрипта.

При имитации проекта **Отладчик скриптов** запускается автоматически, если в проекте есть хотя бы одна установленная точка останова. В процессе отладки **Отладчик скриптов** может быть открыт повторно по кнопке **Управление отладчиком скриптов** на панели **Отладка**.

Отладчик позволяет:

- Выполнять пошаговую трассировку
- Отслеживать значения переменных и аргументов в процессе выполнения скрипта
- Просматривать стек вызовов
- Устанавливать/отменять точки останова.

Главное окно **Отладчик скриптов** имеет следующий вид



Панель инструментов Отладчика скриптов состоит из следующих кнопок:



Запуск – продолжает работу проекта

Пауза – останавливает работу проекта

Шаг с обходом – переходит на следующую строку после функции вызова

Шаг с заходом – переходит на первую строку функции вызова

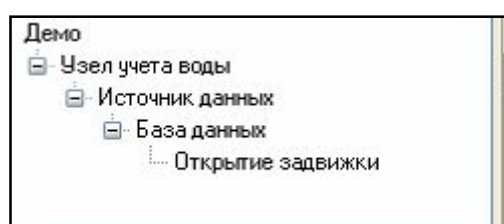
Шаг с выходом – вернуться к вызывающей функции

Поставить точку останова – устанавливает точку останова в выбранной строке

Удалить точку останова – удаляет точку останова из выбранной строки

Удалить все точки останова – удаляет все точки останова

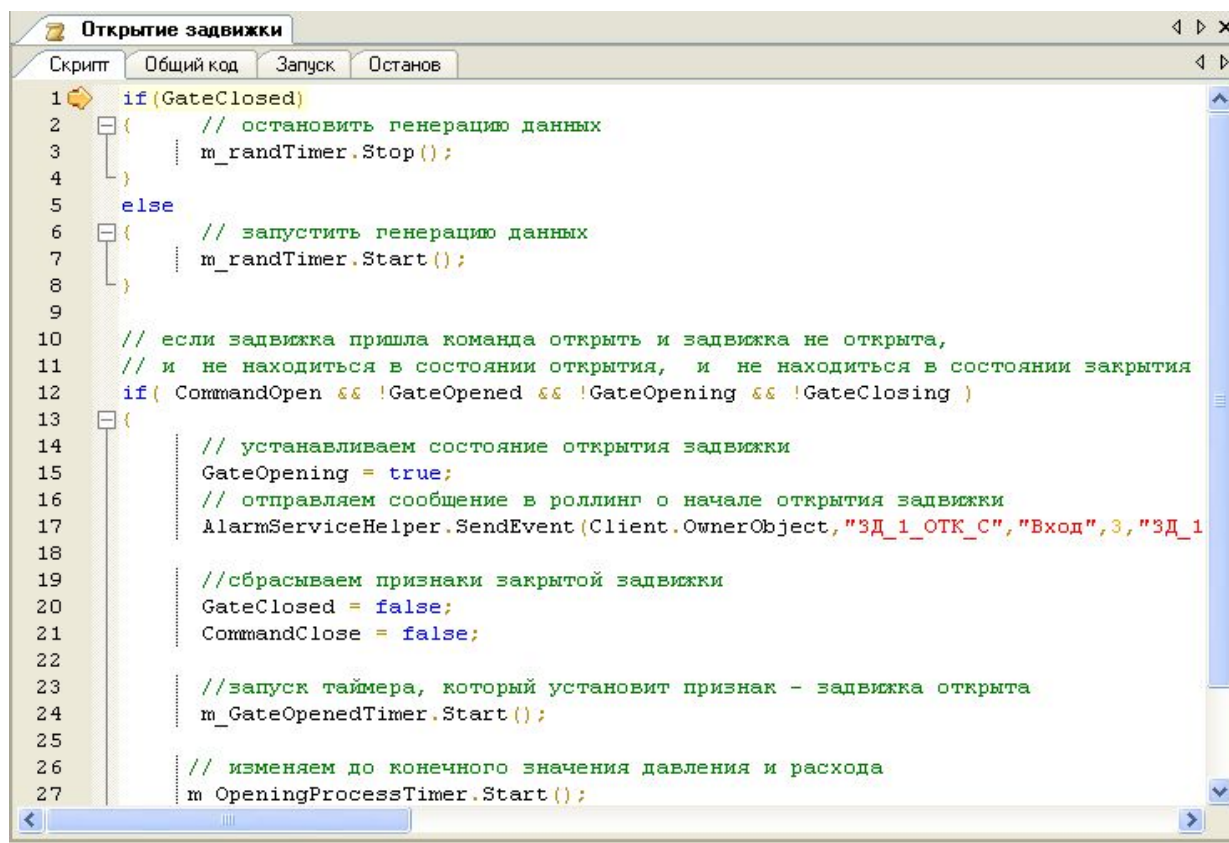
Дерево скриптов проекта показывает древовидную структуру скриптов проекта.



С помощью дерева можно найти нужный скрипт для отладки и открыть его в области просмотра текста скрипта.

Дерево скриптов позволяет фильтровать скрипты по типам родительских объектов и осуществлять поиск скриптов по полному имени.

Область просмотра текста скрипта состоит из четырех частей: **Скрипт**, **Общий код**, **Запуск**, **Останов**. В области просмотра пользователь может просматривать код скрипта и ставить в нужных местах точки останова простым щелчком мыши по строке.



Закладка Переменные предназначена для просмотра значений переменных скрипта в процессе отладки.

Наименование	Значение	Тип
openPhase	3	System.Int32

Каждой переменной скрипта соответствует одна строка таблицы в закладке **Переменные**. Строка таблицы включает:

- **Наименование** – название переменной в скрипте
- **Значение** – текущее значение переменной
- **Тип** – тип значения переменной.

Добавить переменную в окно просмотра переменных можно несколькими способами:

- Набрать имя переменной в области закладки **Переменные**
- Перетащить имя переменной из области редактирования скрипта в область закладки **Переменные**
- С помощью контекстного меню **Добавить просмотр** области редактирования данных.

Наименование	Значение	Тип
openPhase		System.Int32
<div> Добавить просмотр Удалить просмотр </div>		



- **Удалить все** – удаляет все точки останова из всех скриптов
- **Перейти к коду** – переходит к строке кода, на которой установлена соответствующая точка останова.

2.4.4 Аргументы

Аргумент – это ссылка на данные, значения которых используются в скриптах, анимациях и реакциях объекта.

В качестве данных может быть значение атрибута тега объекта или свойство графического элемента.

Задание аргументов осуществляется в окне **Аргументы** (для вызова нажать на кнопку  панели инструментов.).

Аргументы - Sensor Value Text				
	Имя в скрипте	Имя	Тип	Привязка
	Quality	Quality	String	Waterflow registration center.Registration center.Temperature[Value.Quality]
	Value	Current Value	Double	Waterflow registration center.Registration center.Temperature[Value.Value]

Свойства

Аргумент имеет свойства: **Имя в скрипте**, **Имя**, **Тип**, **Привязка**.

Имя в скрипте используйте для обращения к значению аргумента. При назначении **Имя в скрипте** следует руководствоваться правилами объявления переменных в языке **C#**:

- имя может начинаться с литерала (буквы) или знака подчеркивания _
- в имени можно использовать только буквы, цифры и _
- имя не должно содержать пробелов.



Имя содержит дружественное имя аргумента для отображения в диалогах.

Тип определяет тип значения аргумента (используйте для выбора выпадающий список).

Привязка определяет данные, на которые ссылается аргумент. Для установления/изменения используется окно **Выбор привязки**, которое вызывается щелчком по строке в колонке **Привязка**.

Аргумент скрипта имеет дополнительное свойство **Запускать скрипт**, которое позволяет настроить запуск скрипта при изменении аргумента.

Редактирование включает следующие операции:

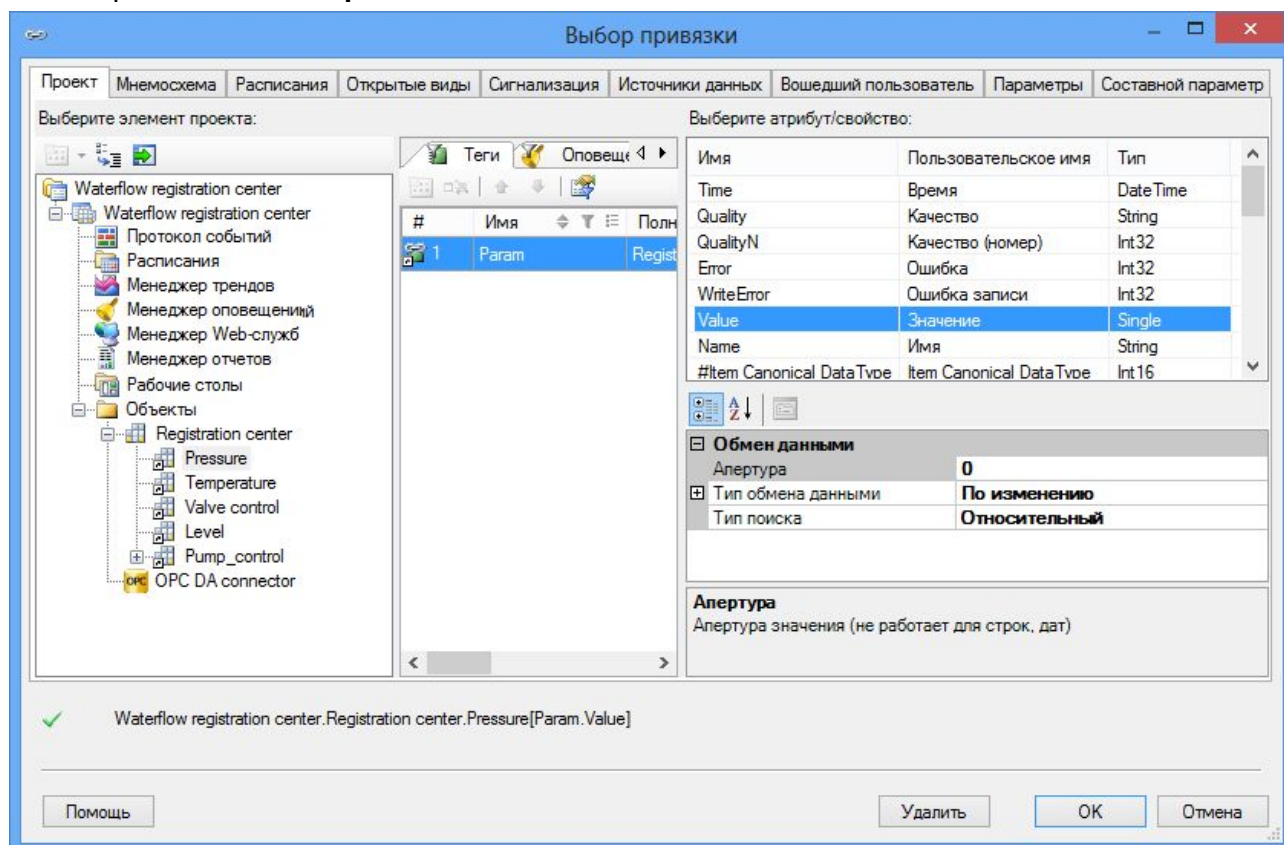
- **Добавление.** Для добавления нового аргумента нажмите на кнопку  панели инструментов окна **Аргументы**. В появившейся строке задайте необходимые параметры. Для добавления аргумента можно использовать операцию перетаскивания из дерева проекта
- **Удаление.** Для удаления аргумента нажмите на кнопку  панели инструментов окна **Аргументы**
- **Изменение типа.** Для изменения типа аргумента выберите нужный тип из выпадающего списка в колонке **Тип**.
- **Привязка.** Для установления/изменения привязки выделите редактируемую строку и щелкните по строке в колонке **Привязка**. В появившемся окне **Выбор привязки** задайте необходимые параметры.

2.4.4.1 Окно Выбор привязки

Окно **Выбор привязки** предназначено для определения («привязки») данных, на которые ссылается аргумент.

Вызов окна

Для вызова окна выделите в окне **Аргументы** редактируемую строку и щелкните по строке в колонке **Привязка**.



Элементы привязки

Аргумент можно привязать к одному из следующих элементов:

- **Атрибуту тега** (вкладка **Проект**)
- **Свойству графического примитива** (вкладка **Мнемосхема**)
- **Расписанию** (вкладка **Расписание**)
- **Открытию/закрытию вида** (вкладка **Открытые виды**)
- **Сигнализации** (вкладка **Сигнализация**)
- **Источнику данных** (вкладка **Источники данных**)
- **Вошедшему пользователю** (вкладка **Вошедший пользователь**)
- **Параметру вида** (вкладка **Параметры**)
- **Составному параметру вида** (вкладка **Составной параметр**)

Редактирование

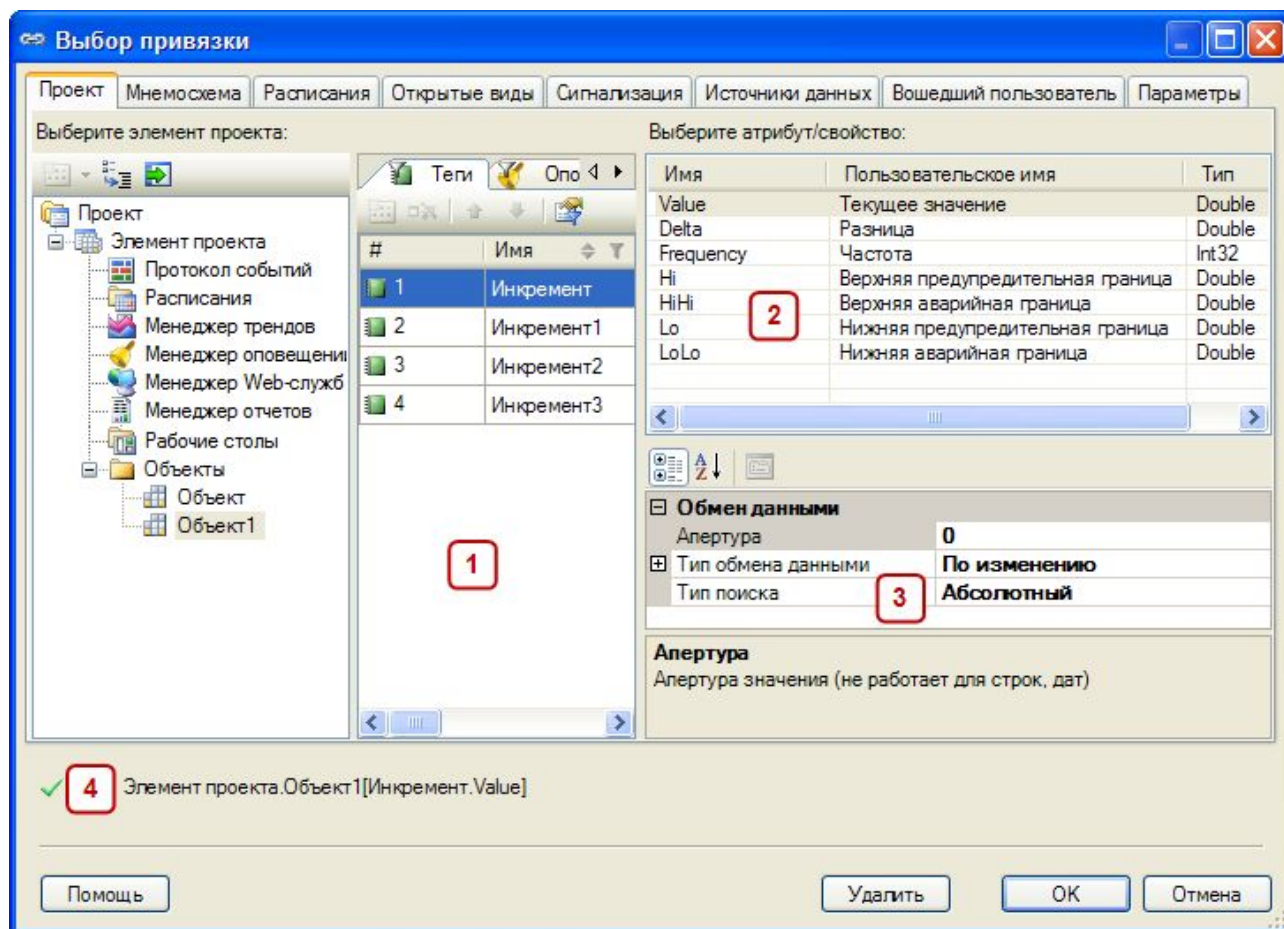
- **Удаление.** Для удаления ссылки нажмите на кнопку **Удалить**.
- **Подтверждение/Отмена.** Для подтверждения изменений нажмите кнопку **ОК**, для отмены – кнопку **Отмена**.

2.4.4.2 Привязка к атрибутам тега

Для привязки аргумента к атрибуту тега предназначена вкладка **Проект** окна **Выбор привязки**.

Для привязки аргумента к атрибуту тега следует:

- 1 Вызвать окно **Выбор привязки** и открыть вкладку **Проект**
- 2 Выбрать объект, к тегу которого осуществляется привязка
- 3 В закладке **Теги** выбранного объекта (1) выделить нужный **тег**
- 4 В списке атрибутов (2) выбрать нужный **атрибут**
- 5 Выбрать **Тип обмена данными** (3): **По изменению/По изменению не чаще чем**
- 6 Задать **Тип поиска** (3): **относительный** – связь с объектом осуществляется в зависимости от его местоположения в дереве; **абсолютный** – связь не зависит от местоположения объекта.



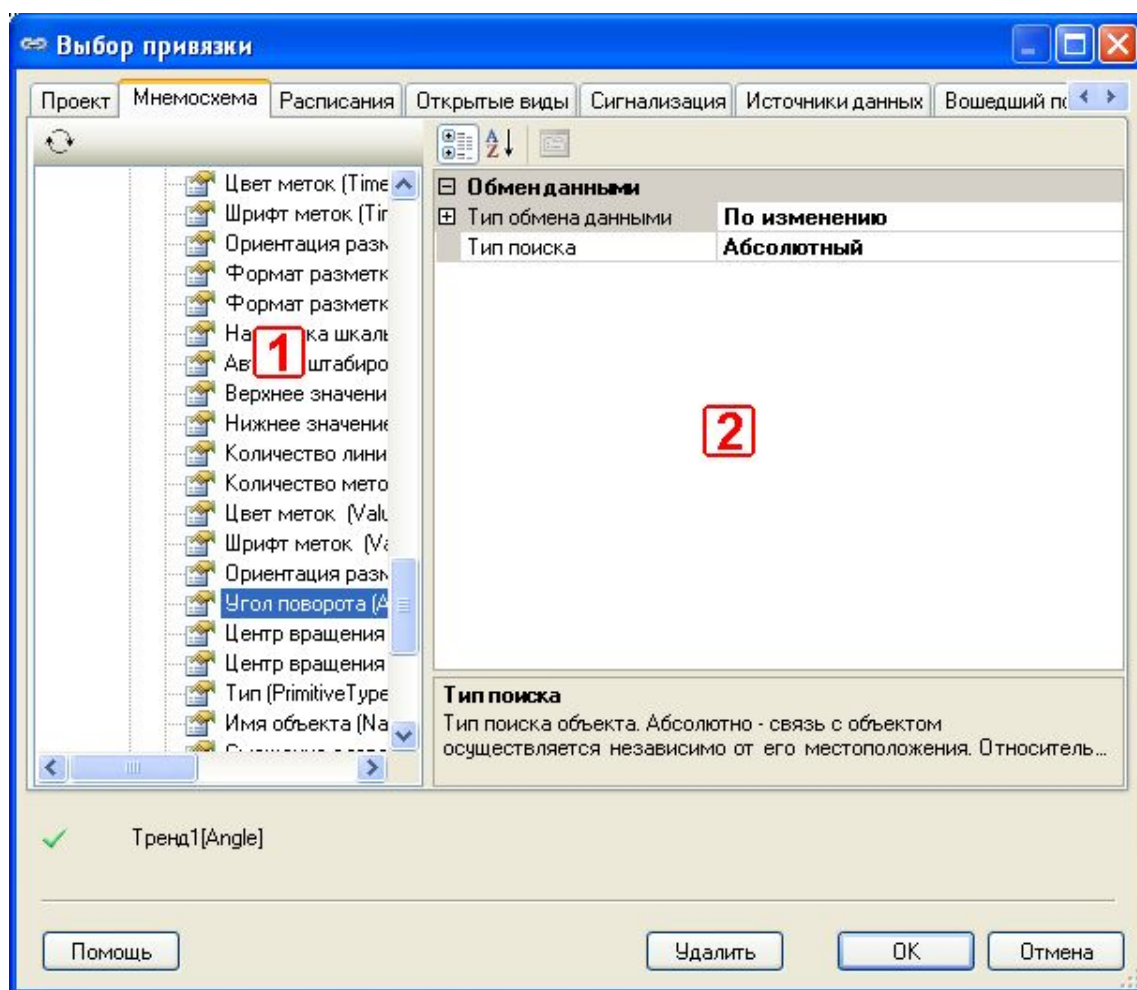
Если установленная ссылка корректна, на панели (4) отображается ✓, иначе — !.

2.4.4.3 Привязка к свойствам графического примитива

Для привязки аргумента к свойствам графического примитива предназначена вкладка **Мнемосхема** окна **Выбор привязки**.

Для привязки аргумента к свойству примитива следует:

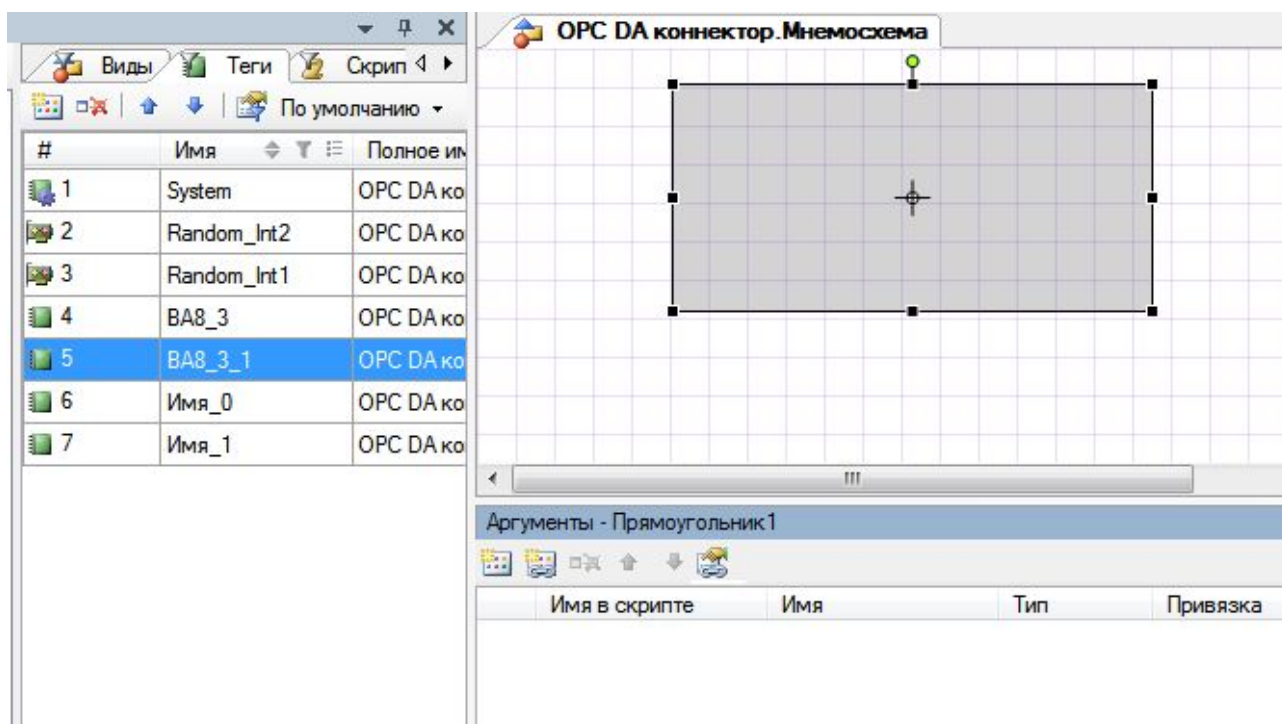
- 1 Вызвать окно **Выбор привязки** и открыть вкладку **Мнемосхема**
- 2 Выделить нужное свойство в дереве мнемосхемы (1)
- 3 Определить **Тип обмена данными** в (2): **По изменению/По изменению не чаще чем**
- 4 Определить **Тип поиска** (2): **Относительный/Абсолютный**.



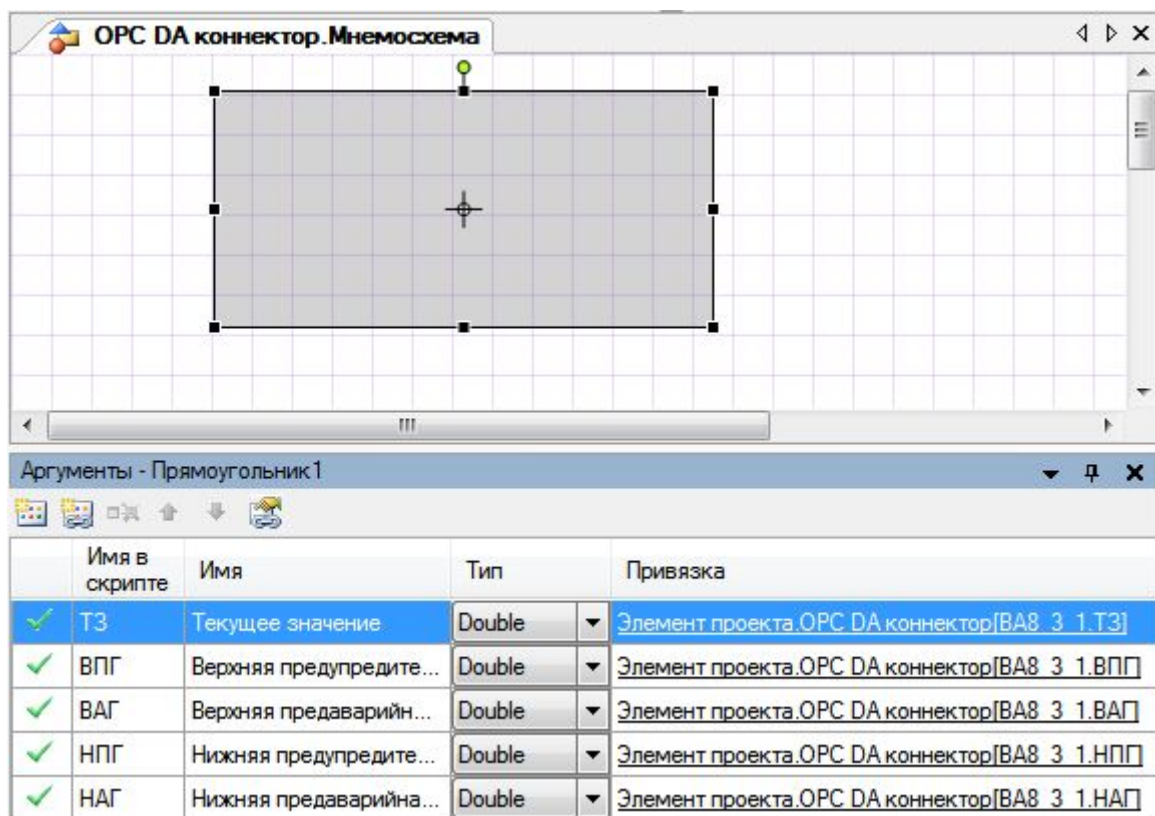
2.4.4.4 Привязка аргументов операцией Drag&Drop

Для добавления аргументов к графическому примитиву с помощью операции Drag&Drop («Перетащить и Оставить») следует:

- 1 Открыть мнемосхему и выделить графический примитив (окно **Аргументы** будет отображать текущий список привязок для примитива)
- 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 Щелкнуть левой клавишей мыши в выделенную область и, не отпуская клавишу мыши, переместить указатель мыши в окно **Аргументы**. После чего отпустить клавишу мыши.




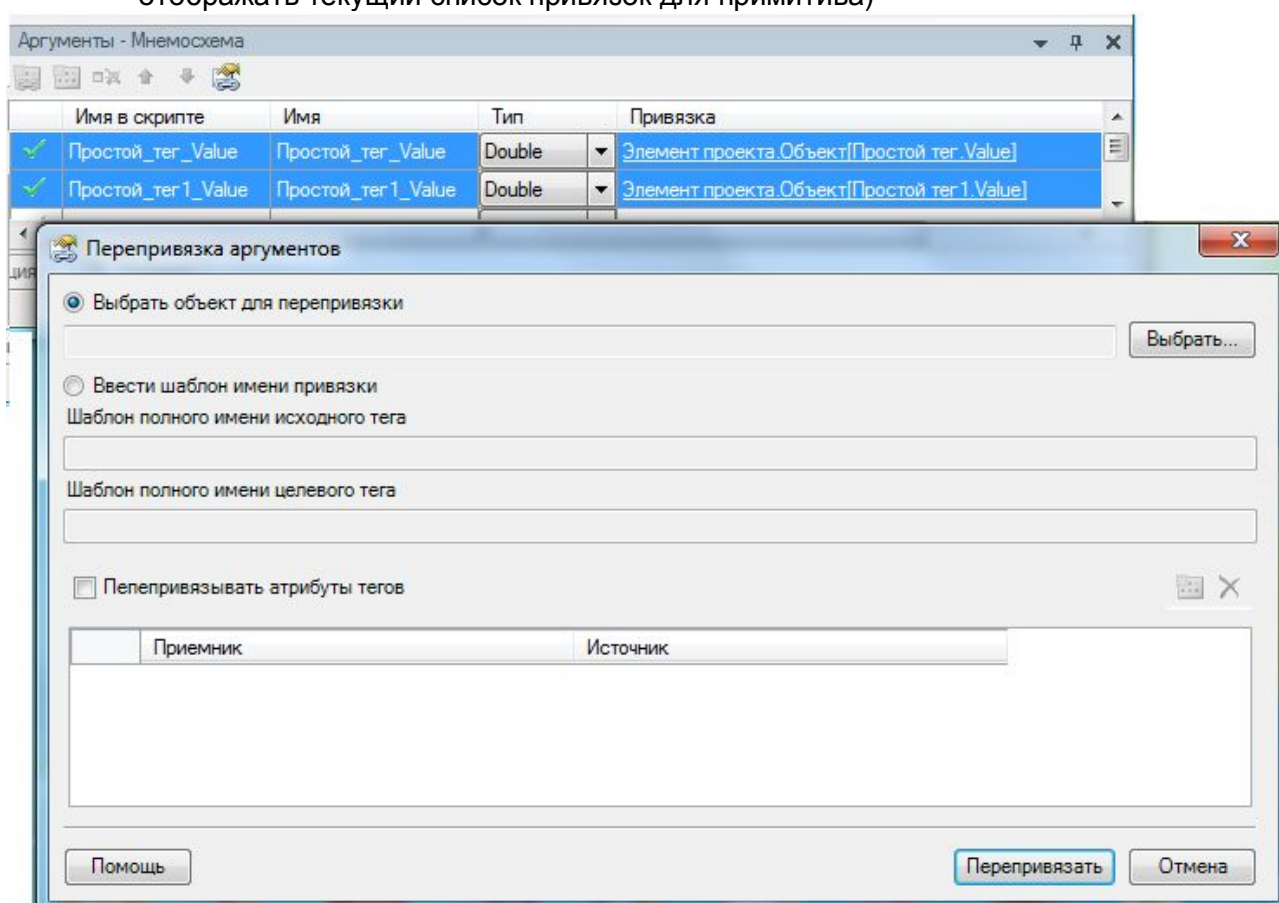
В результате произведенных действий откроется окно редактора связей. При нажатии на кнопку **ОК** помеченные атрибуты и свойства сигнальных состояний тегов будут добавлены в качестве аргументов для графического примитива.



2.4.4.5 Перепривязка аргументов

Для групповой перепривязки аргументов следует:

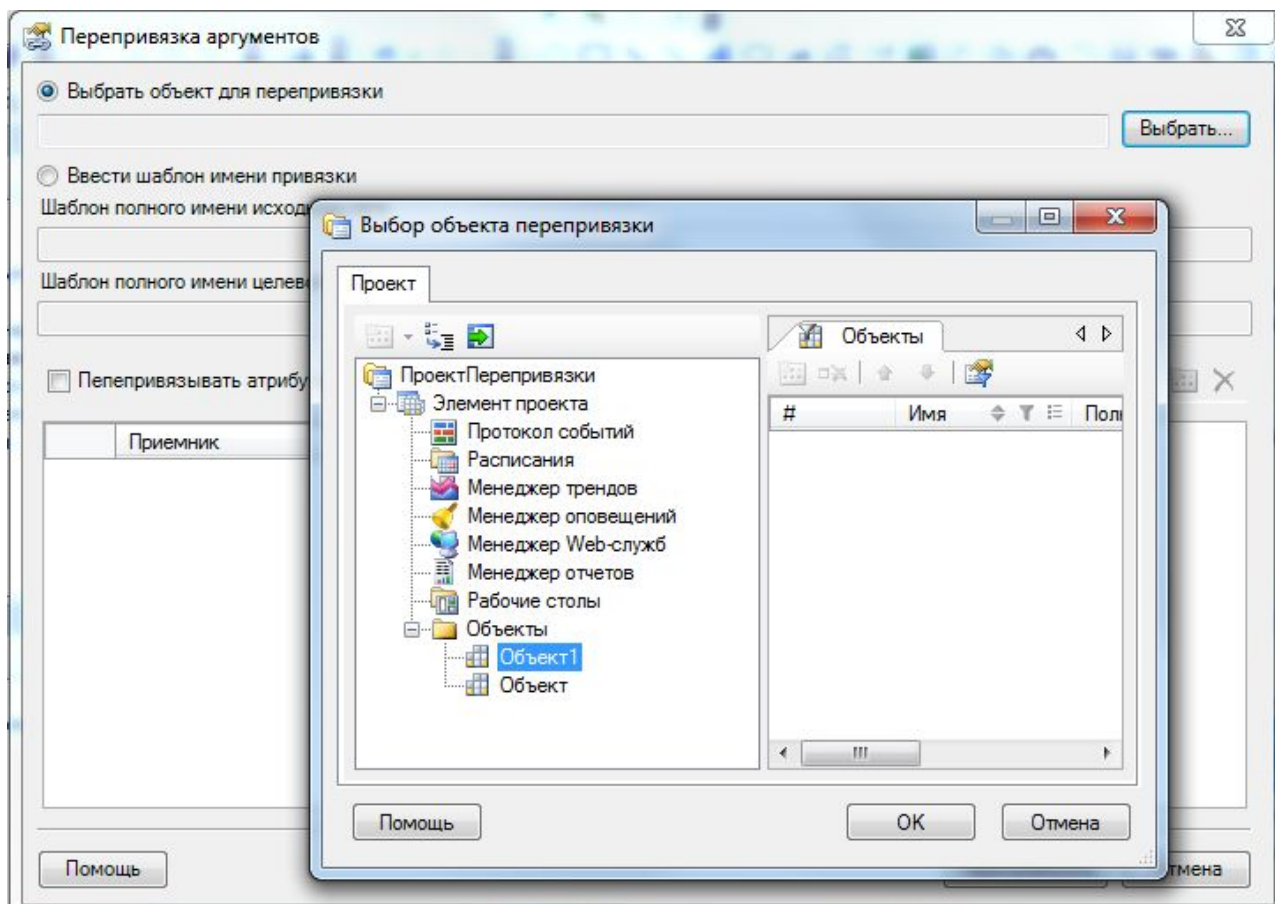
- 1 Выделить в окне **Аргументы** строки с описанием аргументов, привязку которых необходимо изменить
- 2 Нажать на кнопку **Групповая перепривязка** 
- 3 В открывшемся окне **Перепривязка аргументов** задать необходимые параметры
- 4 Открыть мнемосхему и выделить графический примитив (окно **Аргументы** будет отображать текущий список привязок для примитива)



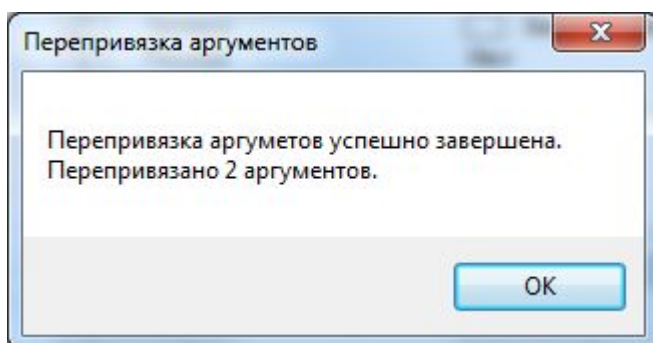
Перепривязка к объектам

Для групповой перепривязки аргументов к объектам следует:

- 1 Выделить в окне **Аргументы** строки с описанием аргументов, привязку которых необходимо изменить
- 2 Нажать на кнопку **Групповая перепривязка**
- 3 В открывшемся окне **Перепривязка аргументов** установить переключатель в положение **Выбрать объект для перепривязки**
- 4 Нажать на кнопку **Выбрать**
- 5 В открывшемся окне **Выбор объекта перепривязки** выбрать объект и нажать на кнопку **ОК**



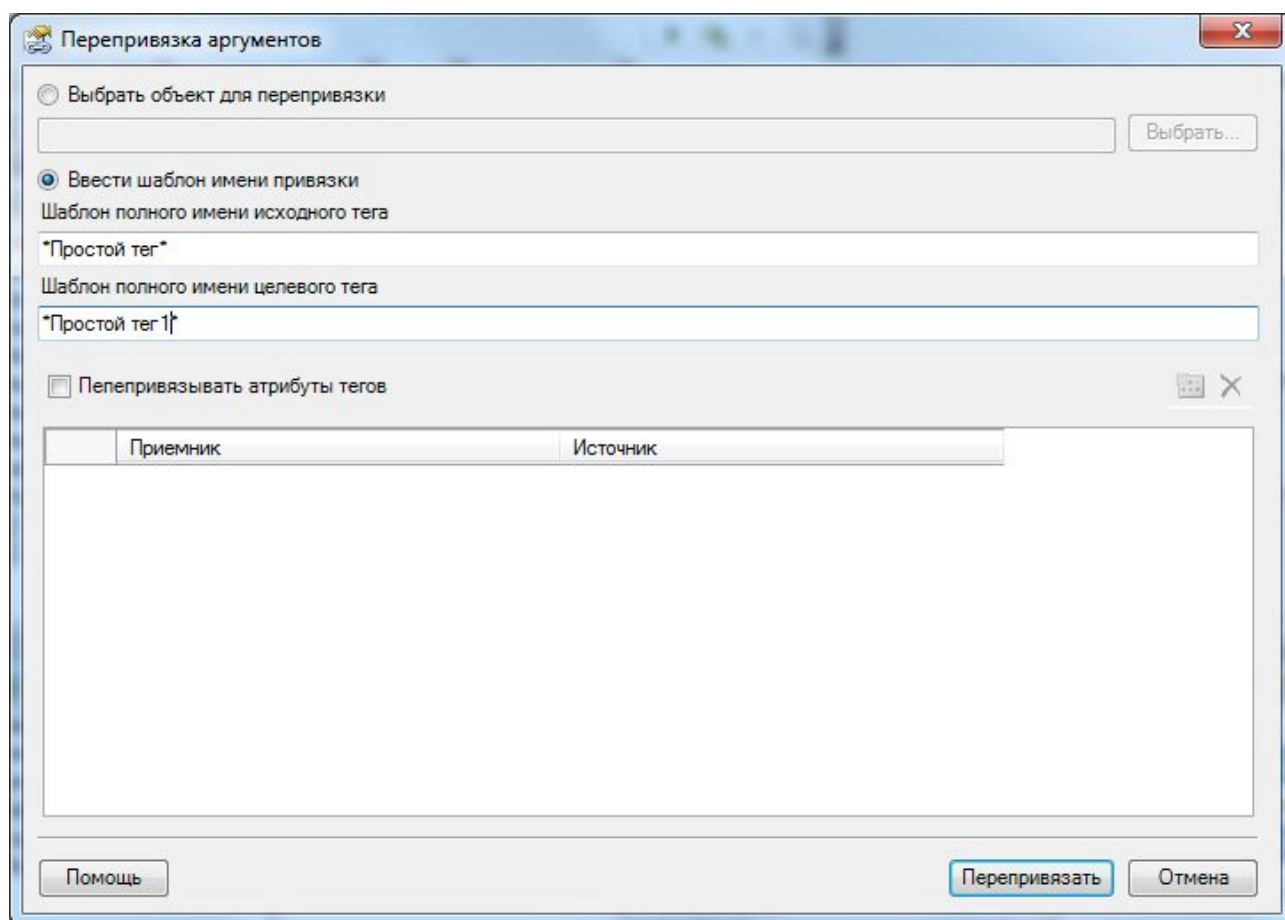
После завершения операции перепривязки появляется сообщение с указанием количества перепривязанных аргументов



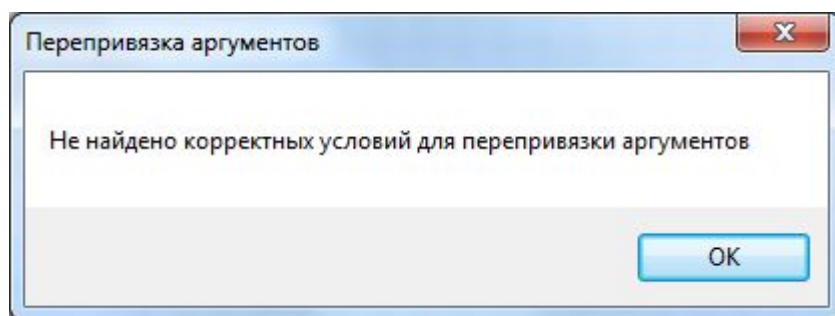
Перепривязка по шаблону наименования аргументов

Для групповой перепривязки аргументов по шаблону наименования следует:

- 1 Выделить в окне **Аргументы** строки с описанием аргументов, привязку которых необходимо изменить
- 2 Нажать на кнопку **Групповая перепривязка**
- 3 В открывшемся окне **Перепривязка аргументов** установить переключатель в положение **Ввести шаблон имени привязки**
- 4 Задать шаблоны наименования исходного и целевого тега, нажать на кнопку **Перепривязать**



После завершения операции перепривязки появляется сообщение с указанием количества перепривязанных аргументов или с сообщением об ошибке




2.4.5 Оповещение

Оповещение - это действие, выполняемое для оповещения о достижении определенного состояния объекта с учетом заданного уровня серьезности. Доступны оповещения следующих типов: **Звук**, **Почта** и **SMS**.

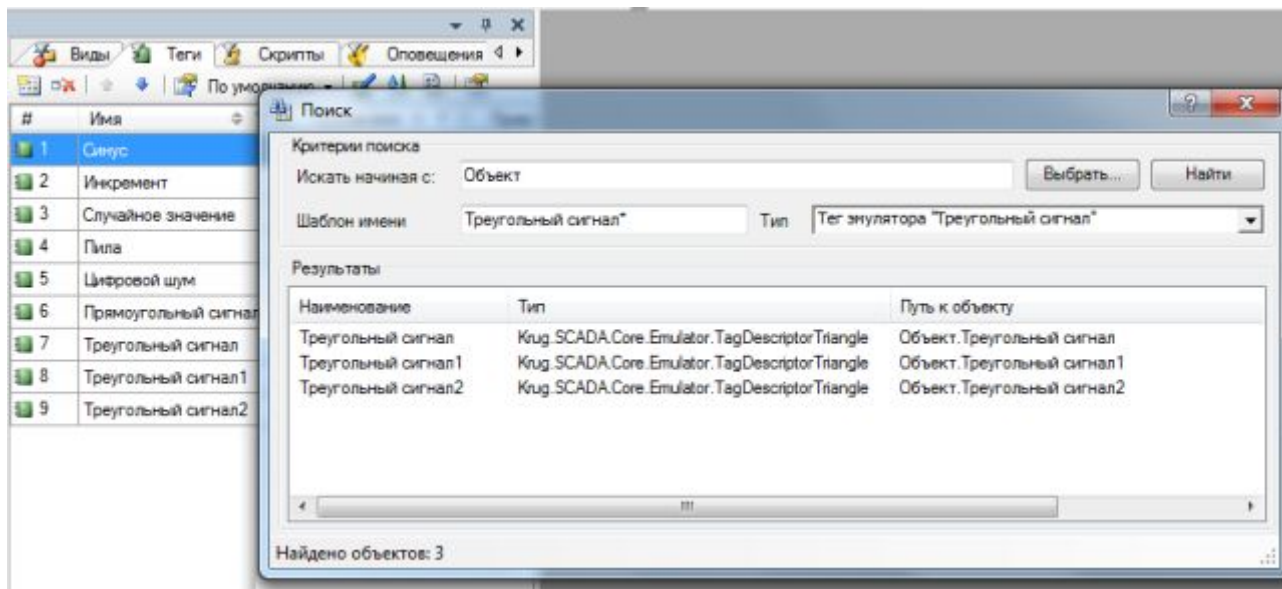
2.4.6 Поиск объектов

Для поиска **объектов/видов/тегов/скриптов** следует:

1. Выбрать команду **Поиск объектов** контекстного меню или щелкнуть мышью по иконке  на панели инструментов

- 2 В появившемся окне **Поиск** задать параметры поиска и щелкнуть мышью по кнопке **Найти**.

Если найдены объекты, удовлетворяющие условиям поиска, то они отображаются в списке **Результаты**. Переход к найденному объекту осуществляется двойным щелчком мыши по элементу списка.



Критерии поиска:

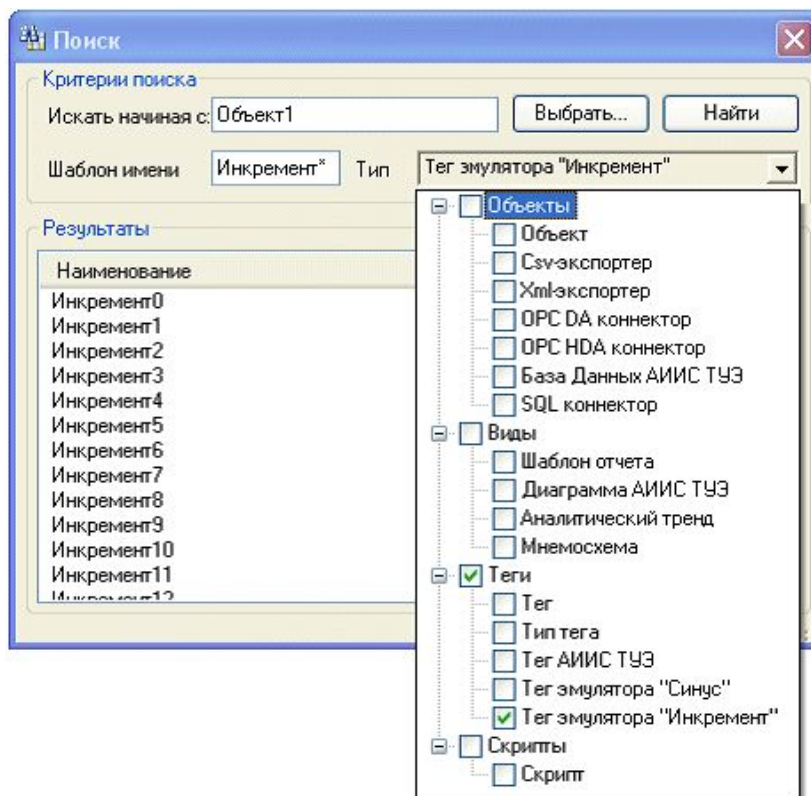
- По имени (по шаблону)
- По типу
- По расположению в дереве проекта (начиная с выбранного объекта и вниз по иерархии)

Поиск по имени

Для нахождения элемента объектной модели по имени необходимо в поле **Шаблон имени** ввести маску поиска. Пользователь может использовать при задании маски символы ***** – для выделения всех символов и символ **?** – для выделения одного символа.

ВНИМАНИЕ!!!

Если шаблон имени не задается, то в списке **Результаты** будут отображаться все элементы, независимо от их имен.

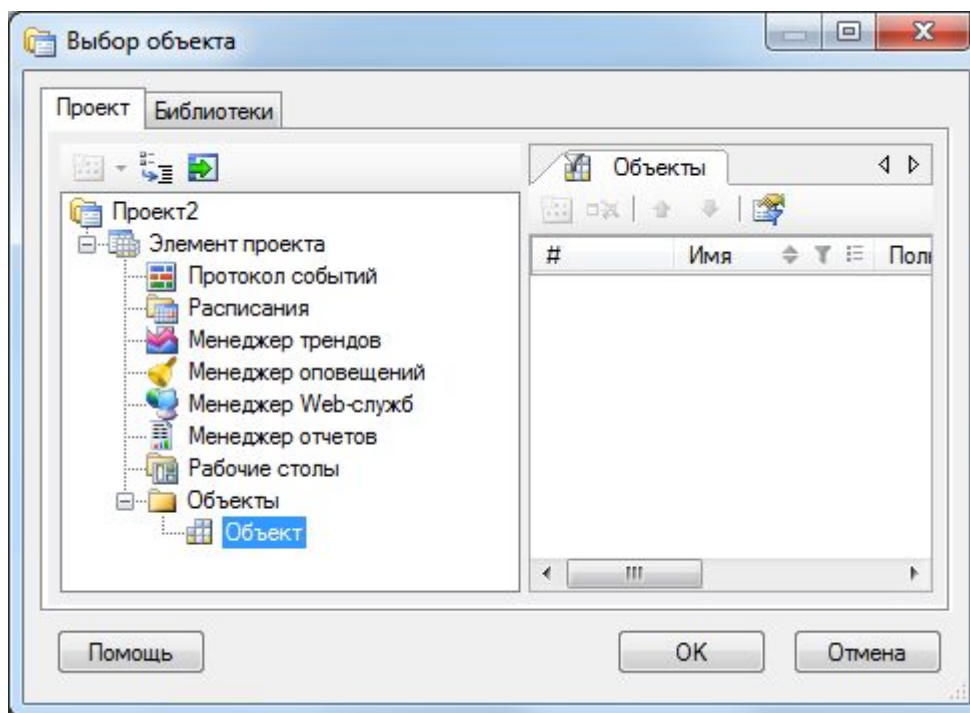


Поиск по типу

Для нахождения элемента объектной модели по типу следует открыть выпадающий список **Тип** и отметить галочками требуемые типы для поиска.

Поиск по расположению

Для того чтобы указать объект, с которого начинается поиск, следует щелкнуть мышью по кнопке **Выбрать** и в открывшемся окне **Выбор объекта** указать элемент проекта или сам проект.



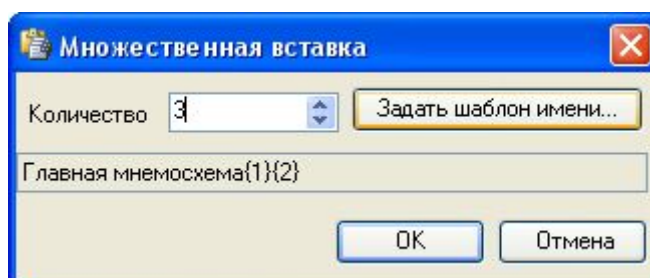
Поиск объектов начинается с выбранного объекта и осуществляется по заданному типу и шаблону имени.

2.4.7 Множественная вставка

На панели инструментов **Редактирование** и в контекстном меню объектов/видов/тегов/скриптов доступна команда **Множественная вставка**.

При вызове этой команды открывается окно **Множественная вставка**.

Множественная вставка работает аналогично простой вставке объектов. Отличие заключается в том, что вставляется не один экземпляр, скопированного в буфер обмена элемента(ов), а столько, сколько укажет пользователь.



Количество вставляемых объектов задается в поле **Количество**. Имя вставляемого элемента определяется шаблоном.

Для задания шаблона щелкните мышью по кнопке **Задать шаблон имени** и выполните действия, описанные в разделе «Формирование шаблона имени».

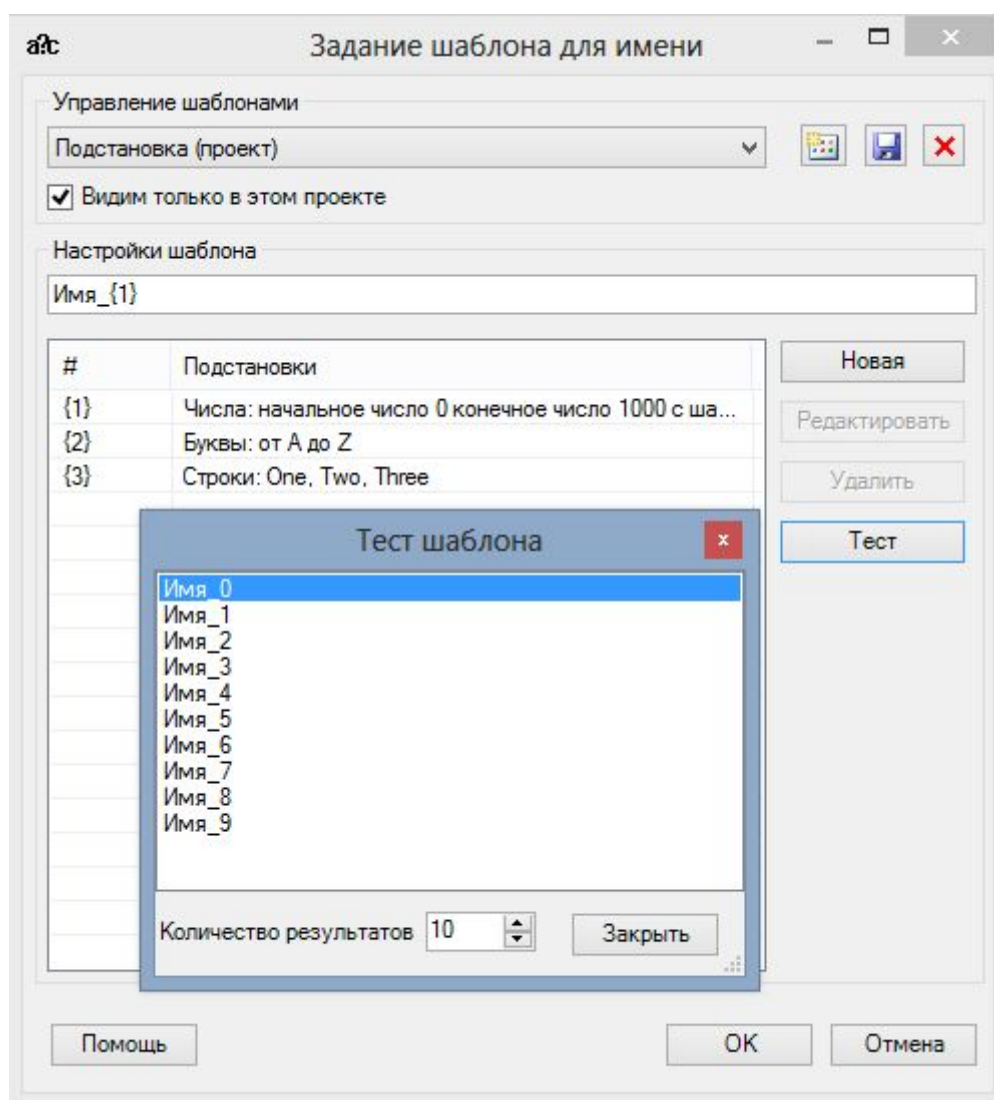
2.4.8 Формирование шаблона имени

Окно **Задание шаблона для имени** предназначено для создания списка регулярных имен на основе шаблонной строки.

Регулярное имя включает неизменяемую часть и подстановку.

Подстановка задается в виде **{N}**, где N – номер подстановки.

В качестве подстановок шаблона для имени могут выступать числа, буквы и строки.



Элементы **Управление шаблонами**: – создать шаблон; – сохранить шаблон; – удалить шаблон; ☒ Видим только в этом проекте – если галочка отсутствует, то шаблон может использоваться не только в текущем проекте, но и в других проектах.

Создание, редактирование и просмотр подстановки

При нажатии на кнопку **Новая** выпадает список типов подстановок. При выборе типа подстановки открывается окно с настройками:

- **для чисел**: начальное число, конечное число, шаг и количество знаков в числе
- **для строк**: список строк
- **для букв**: начальная и конечная буква (в прямом или обратном алфавитном порядке).

Кнопка **Редактировать** предназначена для редактирования выделенной подстановки.

Кнопка **Удалить** предназначена для удаления выделенной подстановки.

С помощью кнопки **Тест** можно посмотреть результирующий список имен в появляющемся окне **Тест шаблона**. В этом окне в поле **Количество результатов** можно указать количество имен, генерируемых по выбранной подстановке.

Для сохранения шаблонной строки и подстановок задайте имя, тип сохранения и затем щелкните мышью по кнопке **ОК**.

Для того чтобы файл с шаблонами имени сохранился в папке с проектом, установите флаг **Видим только в этом проекте**, иначе по умолчанию файл с шаблонами имени сохранится по одному из путей в зависимости от ОС:

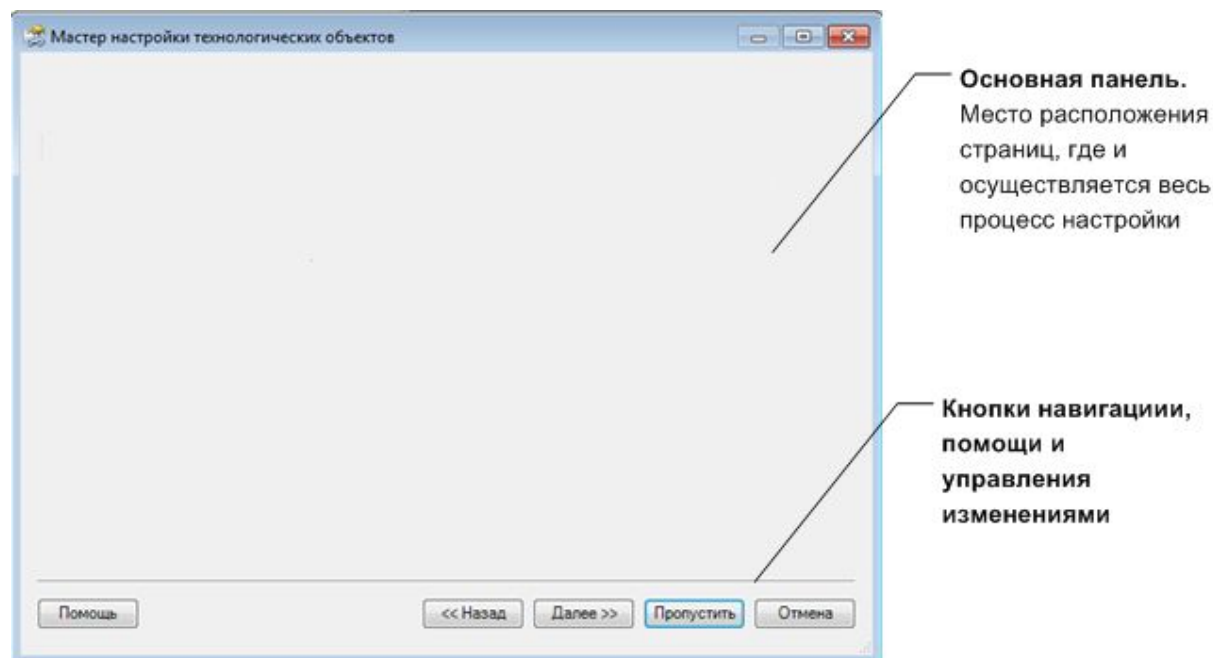
<Системный диск>:\Users\<User>\AppData\Local\Energokrug\DataRate 4.x\Settings\
<Системный диск>:\Documents and Settings\<Users>\Local Settings\Application
Data\Energokrug\DataRate 4.x\Settings\).

Выбрать сохраненные настройки можно из выпадающего списка (если настроек несколько) при нажатии на кнопку **Выбрать**

2.4.9 Мастер настройки технологических объектов

Для ускорения процесса разработки и внедрения типовых решений в *DataRate* предусмотрен мастер настройки технологических объектов.

Окно мастера условно разделено на две панели: верхняя, основная панель, является местом расположения страниц, где и осуществляется весь процесс настройки; нижняя – отведена под кнопки навигации, помощи и управления изменениями



Мастер содержит 4 страницы:

- **Страница выбора объекта**
- **Страница настройки сцен**
- **Страница привязки тегов**
- **Информационная страница**

В зависимости от выбранных опций и способа вызова, страницы отображаются в различных сочетаниях и с разным набором кнопок.

К кнопкам навигации относятся:

- **Назад** – перейти к предыдущей странице мастера. Кнопка видима лишь в случае, когда переход с предыдущей страницы был осуществлен по кнопке **Пропустить**. Кнопка доступна всегда
- **Далее** – перейти к следующей странице мастера, зафиксировав изменения. Видимость кнопки определяется наличием следующей страницы. Кнопка доступна тогда, когда на текущей странице есть настройки, требующие применения
- **Пропустить** – перейти к следующей странице мастера, отказавшись от изменений на текущей странице. Кнопка видима только на тех страницах, которые являются необязательными. Это страница настройки сцен и страница привязки тегов. Кнопка доступна всегда.

Кнопки управления:

- **ОК** – если есть изменения, то зафиксировать их и закрыть окно мастера. Кнопка видима и доступна только на последней странице мастера.
- **Отмена** – отказаться от изменений и закрыть окно мастера. Кнопка видима и доступна всегда, что позволяет завершить работу на любом этапе настройки.

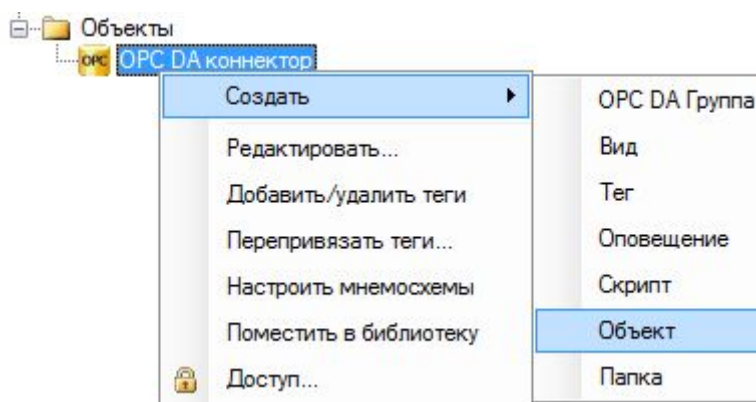
Кнопка **Помощь** дает возможность пользователю раскрыть раздел файла справки, соответствующий отображаемой странице и получить информацию по работе с ней. Видимость и доступность кнопки определяется наличием справочных сведений.

Мастер может быть активирован одним из следующих способов:

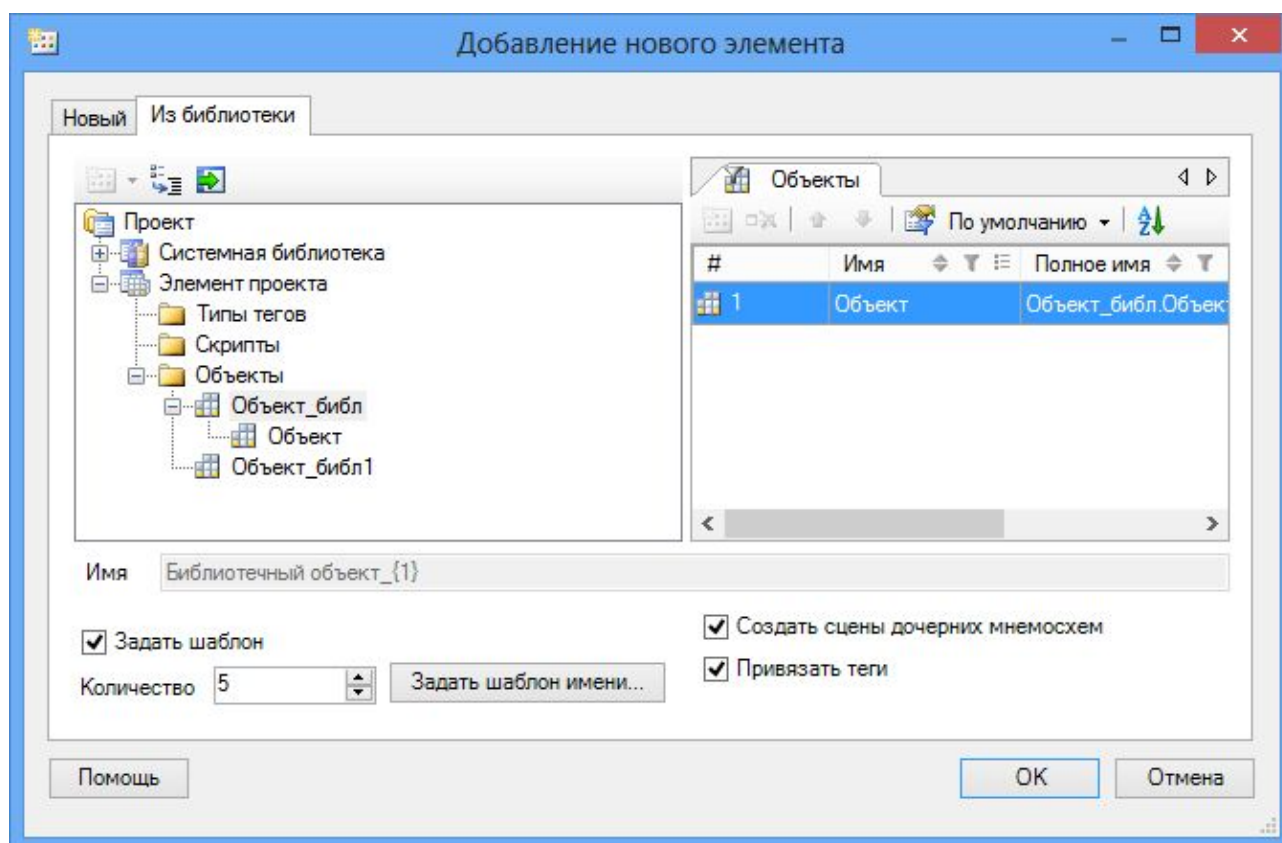
- Из мастера добавления новых элементов
- При выборе пункта **Настроить мнемосхемы** или **Перепривязать объекты** в контекстного меню объекта, который может обладать тегами
- В результате операции **Drag&Drop** (перетащить и оставить) при перетаскивании объекта из библиотеки на мнемосхему объекта, расположенного в редакторе проекта

Для активизации мастера при добавлении нового объекта следует:

- 1 В контекстном меню объекта (в дереве редактора проекта или редактора библиотек) выбрать пункт **Создать/Объект**



- 2 В появившемся окне **Добавление нового элемента** перейти на закладку **Из библиотеки**, выбрать библиотечный объект, настроить шаблон имени (в случае множественного создания), выставить опции.

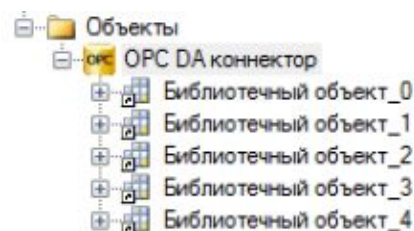


Пользователю предлагается следующие опции:

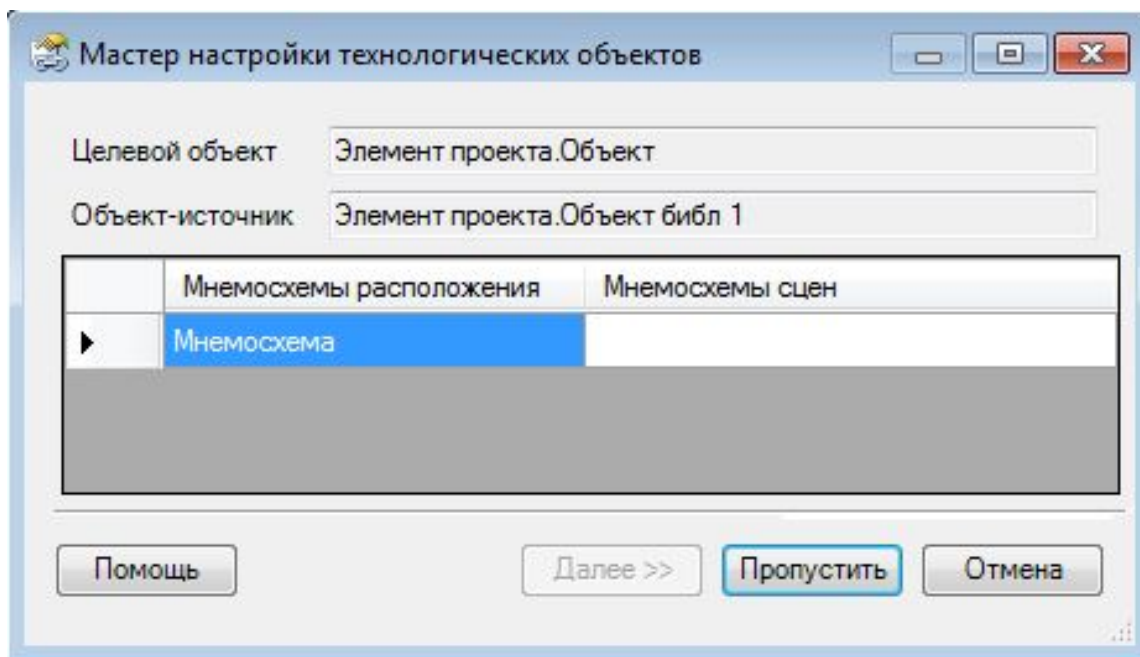
- **Создать сцены дочерних мнемосхем.** Эта опция позволяет использовать мнемосхемы новых объектов в качестве сцен на мнемосхемах родительского объекта, т.е. объекта в который они вставляются
- **Привязать теги.** Эта опция позволяет осуществить привязку тегов новых объектов к тегам источников данных. Такими источниками могут выступать коннекторы, а также любые объекты, содержащие теги.

3 Нажать на кнопку **ОК**

В результате произведенных действий в объект **Объект** дерева проекта будет добавлено **5** дочерних объектов (по значению, указанному в поле **Количество**) с базовым именем **Библиотечный_объект**.



Если отмечена хотя бы одна опция, то после завершения процесса создания библиотечных объектов в исполняемой части проекта на экране появится окно Мастера настройки технологических объектов.



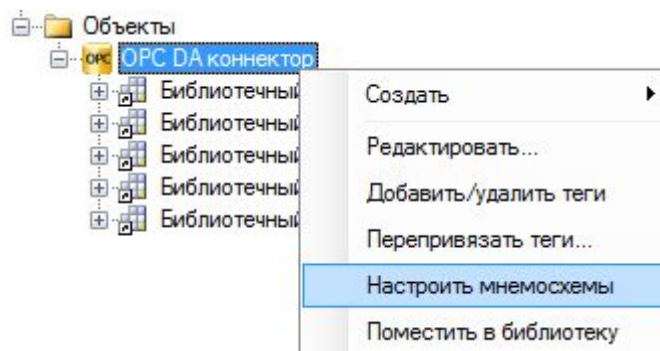
2.4.9.1 Страница выбора объекта

На данной странице пользователь должен определиться с объектом-источником, мнемосхемы которого будут выступать в качестве сцен для мнемосхем выделенного (целевого) объекта.

Страница может быть активирована:

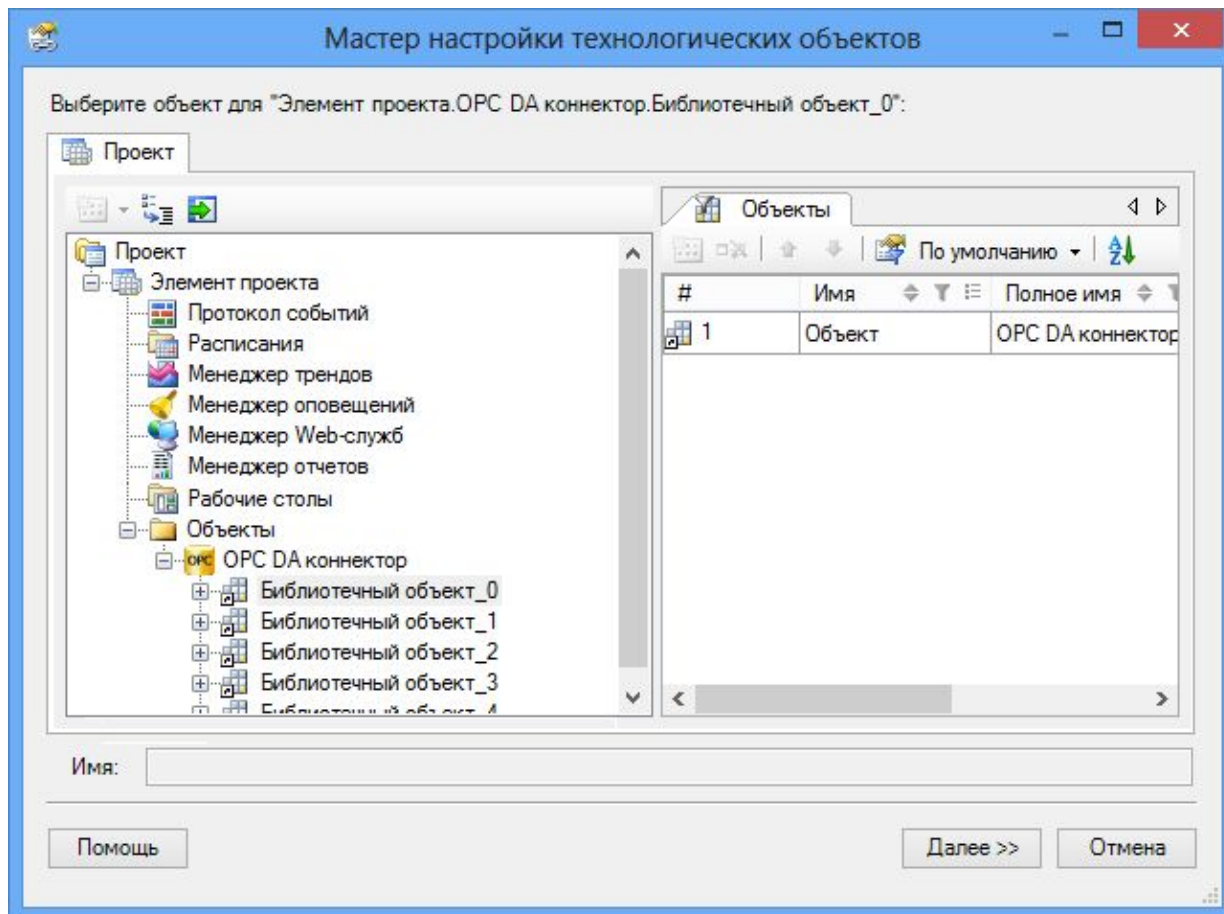
- При выборе пункта **Настроить мнемосхемы** в контекстном меню объекта
- При перетаскивании объекта (операция **Drag&Drop**) из библиотеки на мнемосхему объекта, расположенного в редакторе проекта.

Активирование из контекстного меню объекта



Если выделенный объект расположен в исполняемой части, то выбор разрешается только из числа объектов редактора проекта.

Страница выбора будет содержать закладку **Проект**.



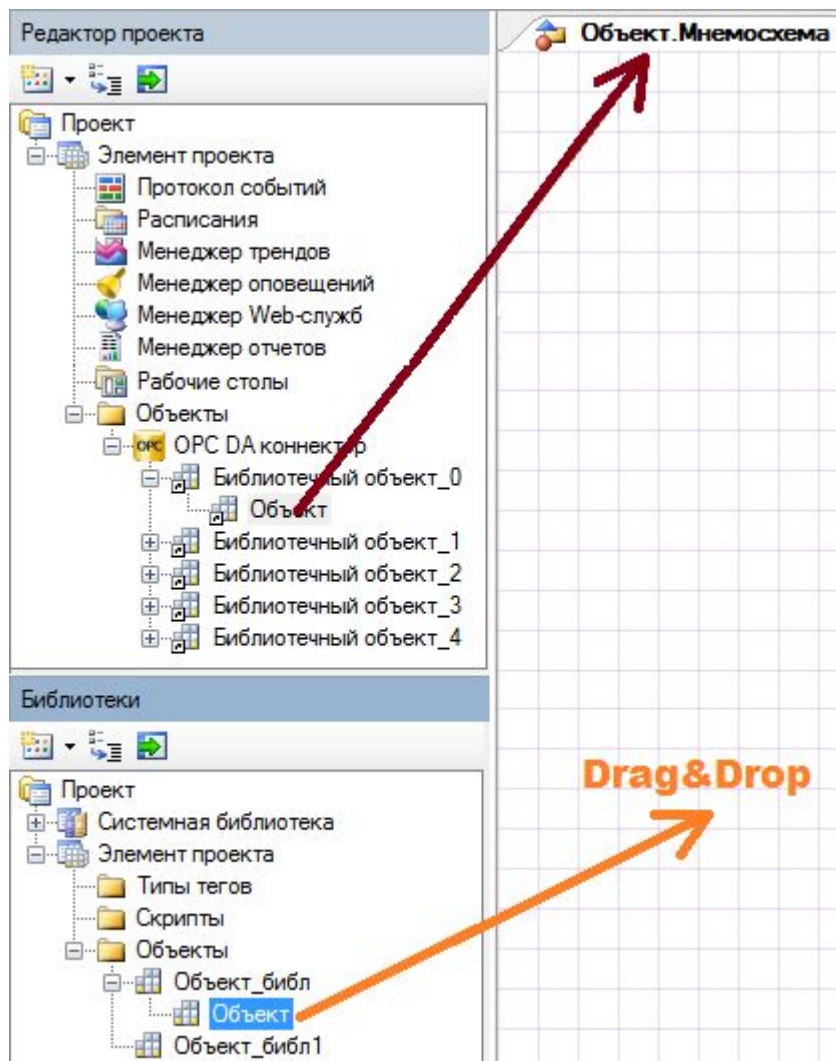
Если контекстное меню относится к объекту, выделенному в библиотеке, то выбор разрешается только среди библиотечных объектов.

На странице выбора будет присутствовать закладка **Библиотеки**.

По умолчанию, при открытии страницы, объектом-источником будет целевой объект.

Активирование при операции Drag&Drop

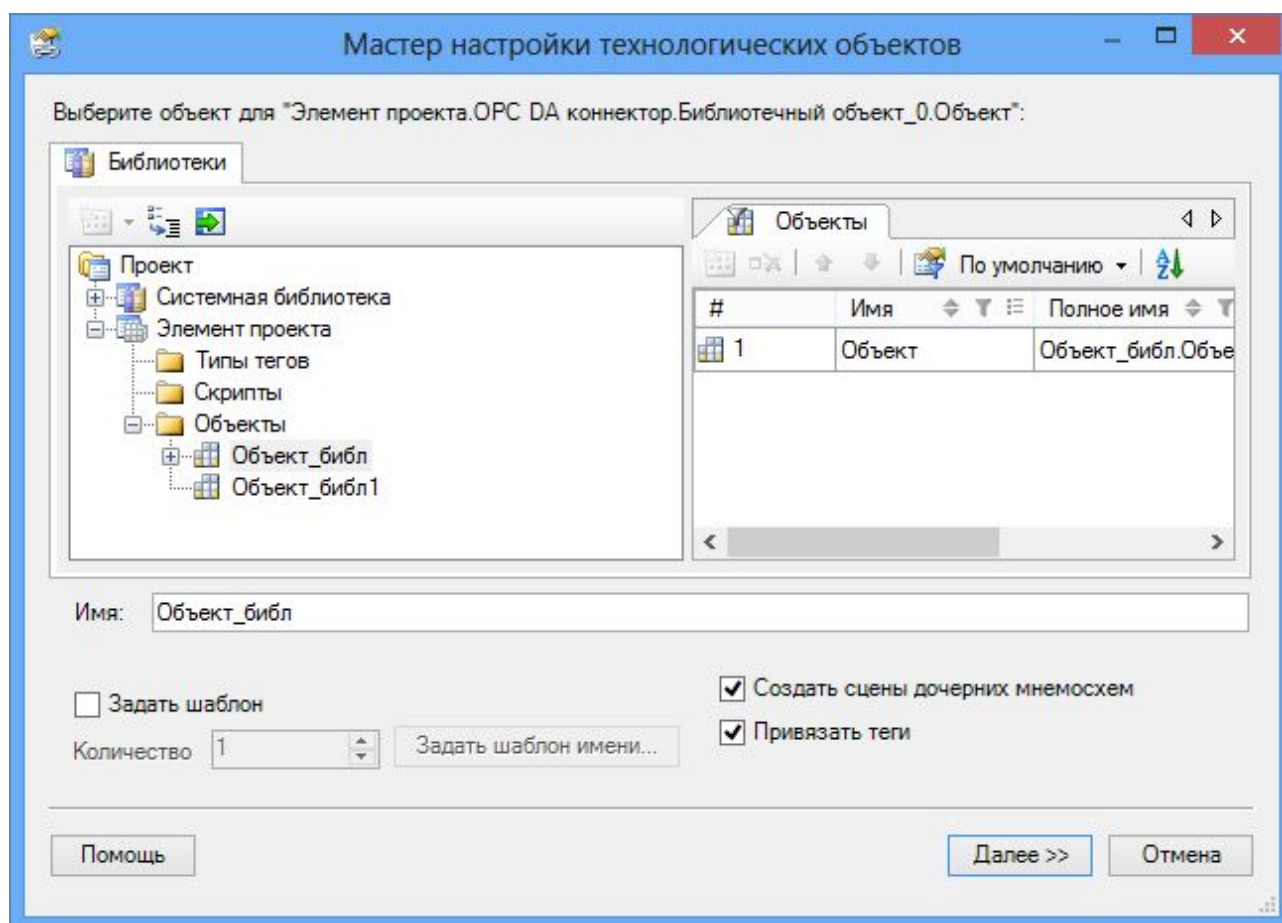
Выполняется операция перетаскивания библиотечного объекта на мнемосхему объекта исполняемой части.



В результате перетаскивания объекта из библиотеки на мнемосхему открывается окно **Мастер настройки технологических объектов** с закладкой **Библиотеки**.

В этом случае целевым объектом будет являться объект, которому принадлежит открытая мнемосхема.

По умолчанию, при открытии страницы, объектом-источником будет перетаскиваемый объект.

**ВНИМАНИЕ!!!**

Дополнительно данный вариант активации мастера предполагает создание в исполняемой части проекта одного (название указывается в текстовом поле "Имя") или нескольких объектов в соответствии с указанным шаблоном (флажок "Задать шаблон" должен быть установлен).

2.4.9.2 Страница настройки сцен

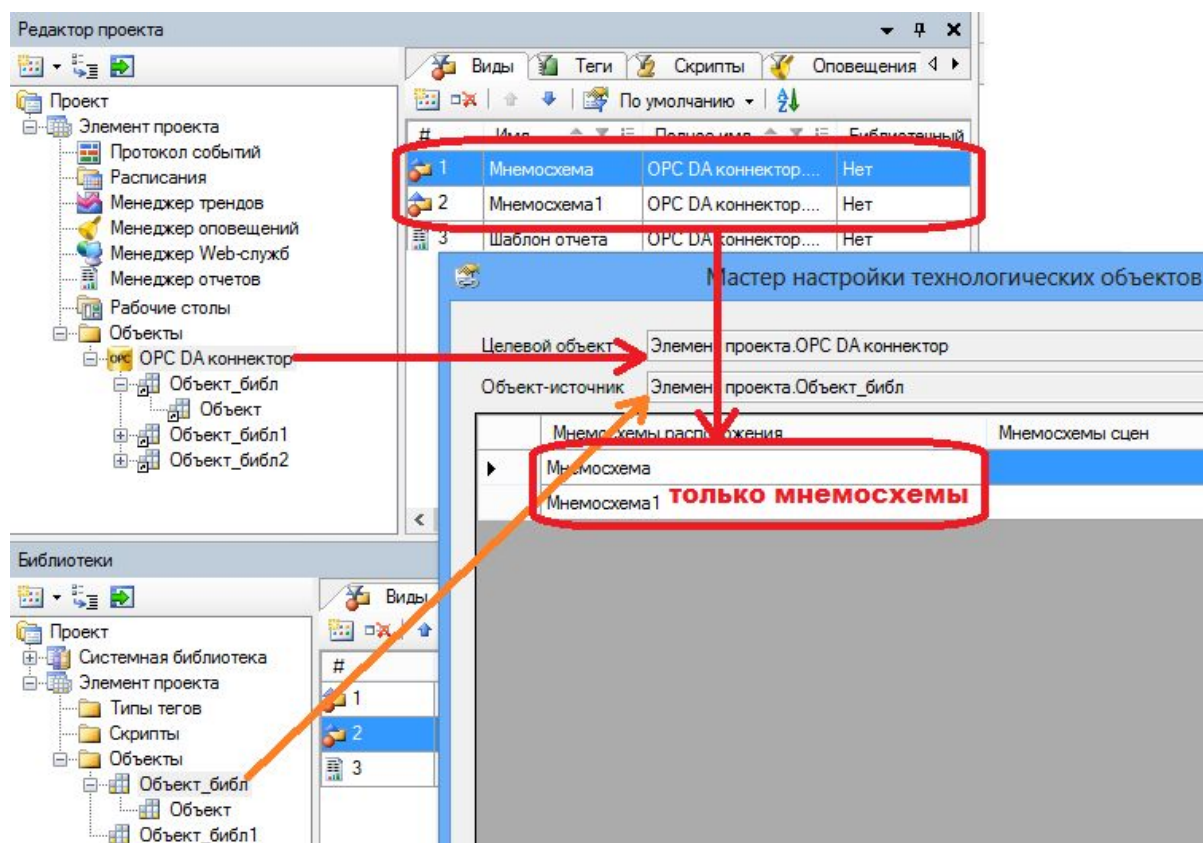
Страница настройки сцен является необязательной и может быть пропущена пользователем.

На данной странице пользователь должен определиться с набором сцен для каждого вида целевого объекта.

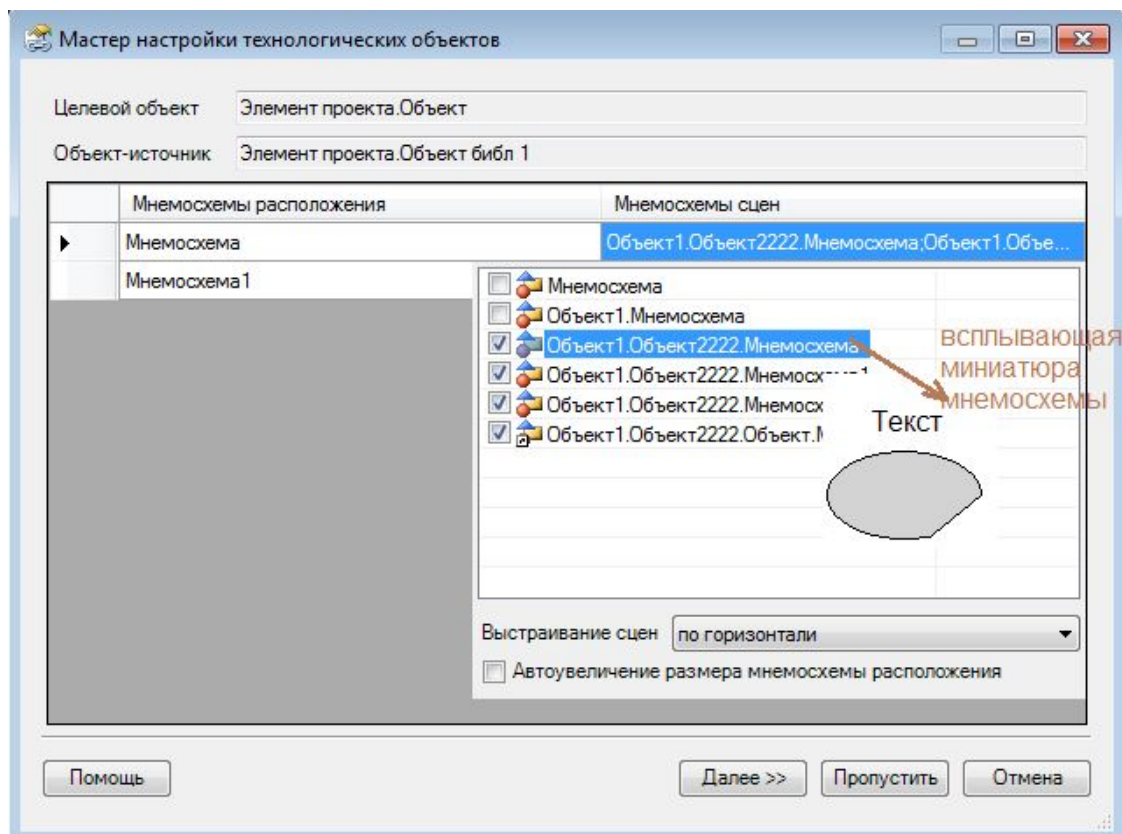
Виды целевого объекта располагаются в колонке **Мнемосхемы расположения**, виды объекта-источника и его дочерних объектов заносятся в столбец **Мнемосхемы сцен**.

ВНИМАНИЕ!!!



При настройке сцен мастером используется только вид «Мнемосхема». Библиотечные мнемосхемы, в силу невозможности их редактирования, не выводятся в список мнемосхем расположения.



Правая часть таблицы предназначена для настройки изображения сцен на мнемосхеме целевого объекта («хост-мнемосхеме»).



Для выбора мнемосхем в качестве сцен следует:

- 1 Щелкнуть левой клавишей мыши по ячейке столбца **Мнемосхемы сцен**
- 2 В появившейся панели настроек, в которой перечислены все мнемосхемы, отметить («поставить галочку») необходимые мнемосхемы.
Пиктограмма, расположенная рядом с названием, ассоциируется с типом мнемосхемы:
 -  – обычная мнемосхема, не имеющая ссылки на библиотеку
 -  – библиотечная мнемосхема.

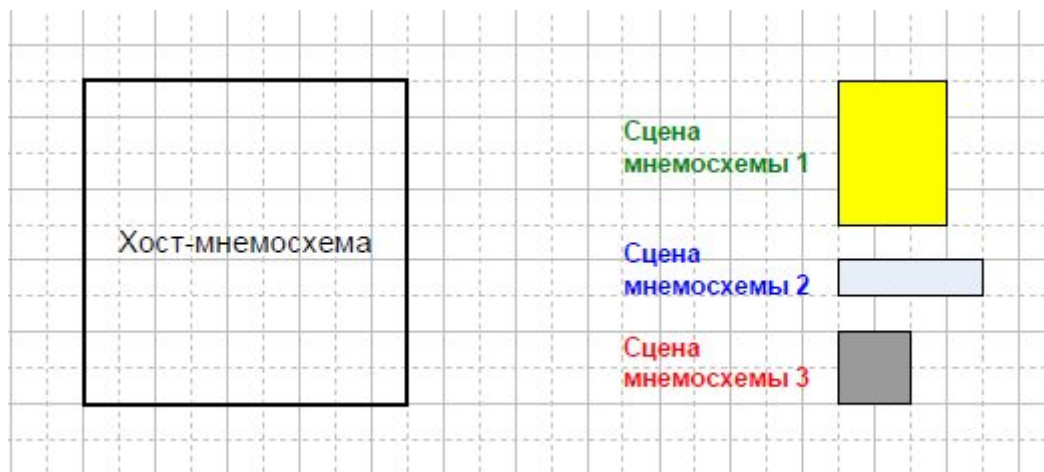
При наведении курсора мыши на название мнемосхемы всплывает ее миниатюрное изображение. Тот же самый эффект можно наблюдать и при подведении курсора на ячейку в столбце **Мнемосхемы расположения**.

В нижней части панели пользователю предоставляются дополнительные настройки:

- **Выстраивание сцен.** Настройка определяет каким образом будет осуществляться расположение элементов на мнемосхеме (по горизонтали или по вертикали) при автоувеличении ее размера.
- **Автоувеличение размера мнемосхемы расположения.** При создании достаточно большого количества сцен на хост-мнемосхеме зачастую невозможно предугадать размеры области, требуемые для корректного расположения примитивов. Поэтому, в *DataRate* предусмотрена возможность автоматического увеличения ширины и высоты мнемосхемы. Чтобы воспользоваться этим сервисом, надо установить данный флажок.

Рассмотрим основные принципы работы механизма выстраивания сцен на конкретных примерах.

Введем следующие графические обозначения:



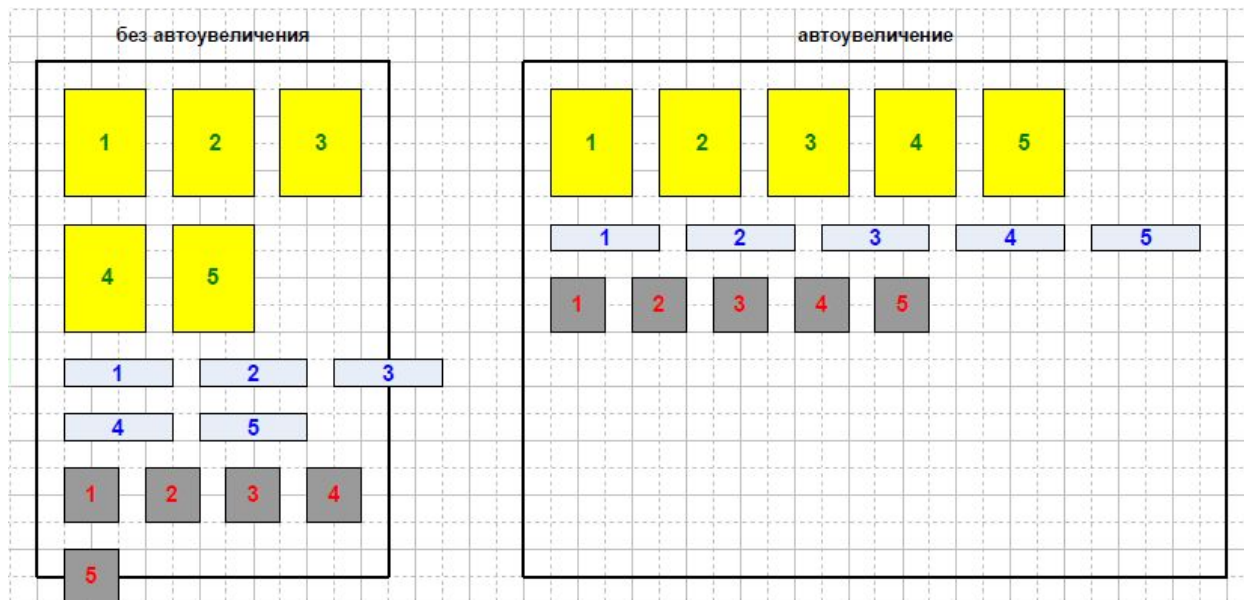
Согласно рисунку, хост-мнемосхема – это одна из мнемосхем в столбце **Мнемосхемы расположения**, а сцены мнемосхем 1,2,3 – отмеченные мнемосхемы в колонке **Мнемосхемы сцен**.

Предположим, что в исполняемую часть проекта добавляется *N* библиотечных объектов. Тогда сцены мнемосхемы 1 для этих объектов будут отображаться зелеными цифрами 1,2,3,...,N, сцены мнемосхемы 2 – синими цифрами 1,2,3,...,N, а сцены мнемосхемы 3 – цифрами 1,2,3,...,N красного цвета

Выстраивание сцен по горизонтали

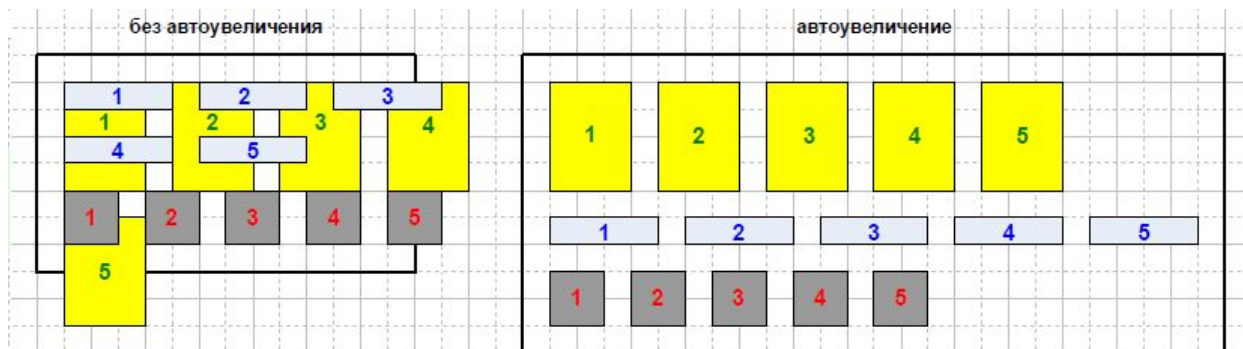
Пример 1

Для 5 объектов, когда на хост-мнемосхеме достаточно места, чтобы сцены не накладывались друг на друга без автоматического увеличения ее размеров.



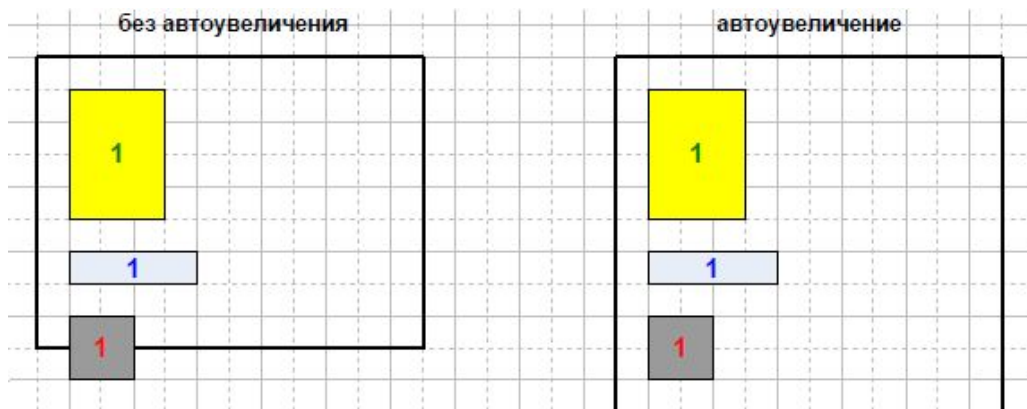
Пример 2

Для 5 объектов, когда размеры хост-мнемосхемы настолько малы, что без автоматического увеличения ее ширины и высоты происходит наложение сцен.



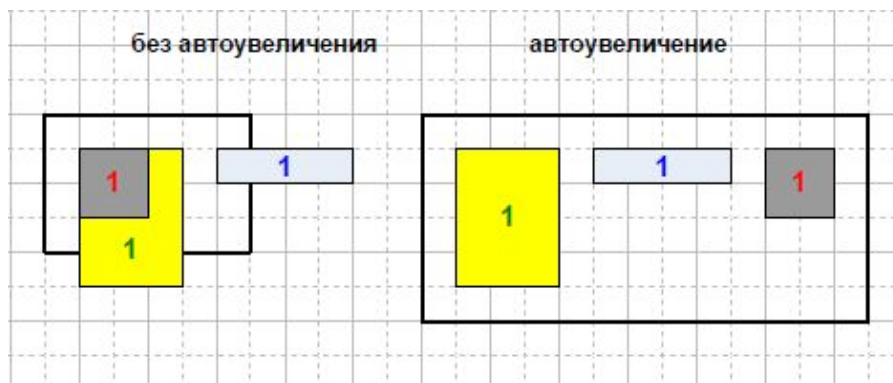
Пример 3

Для 1 объекта, когда высота и ширина хост-мнемосхемы достаточны, чтобы сцены не накладывались друг на друга без автоматического увеличения ее размеров.

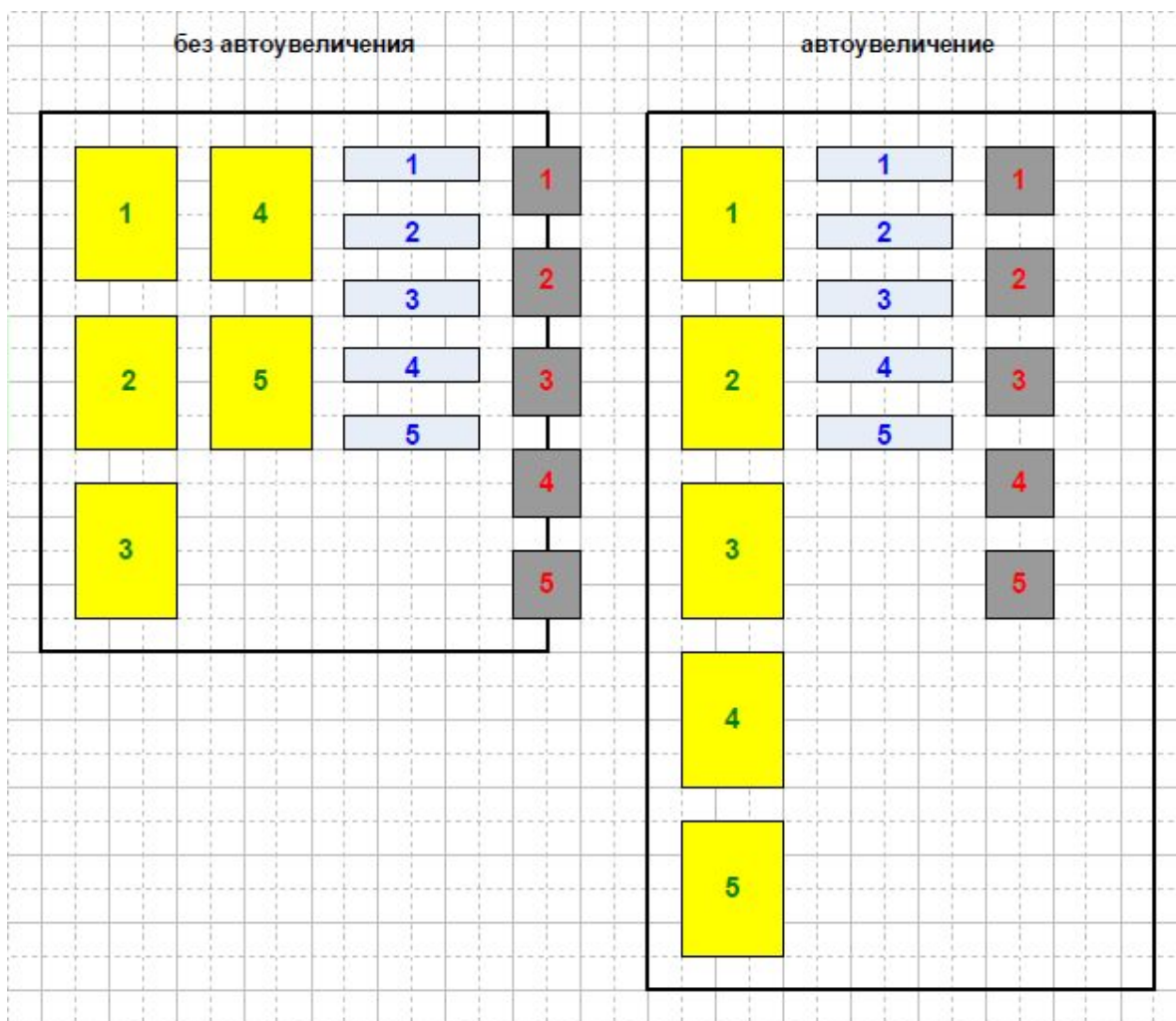


Пример 4

Для 1 объекта, когда высота хост-мнемосхемы мала и без автоматического увеличения ее размеров происходит наложение сцен.

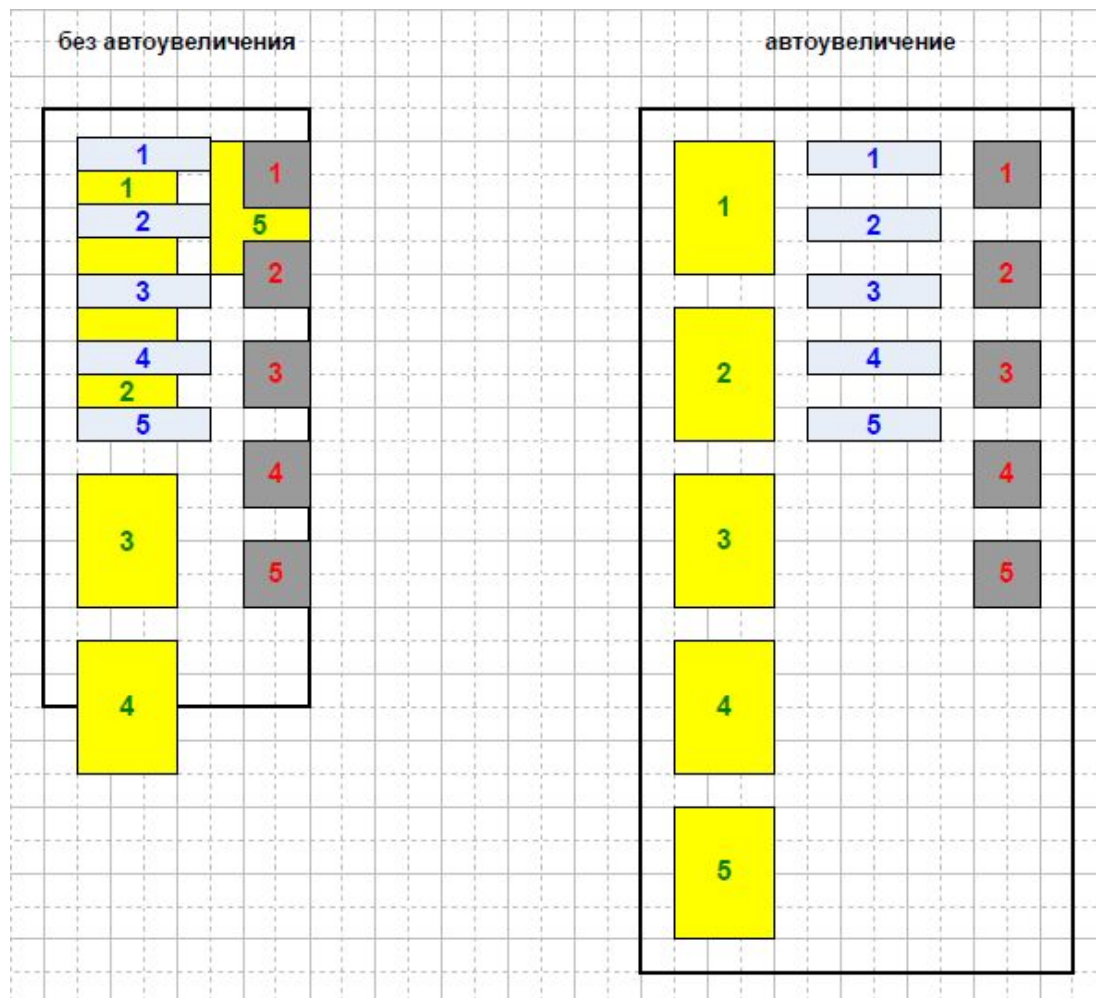
**Выстраивание сцен по вертикали****Пример 5**

Для 5 объектов, когда на хост-мнемосхеме достаточно места, чтобы сцены не накладывались друг на друга без автоматического увеличения ее размеров.

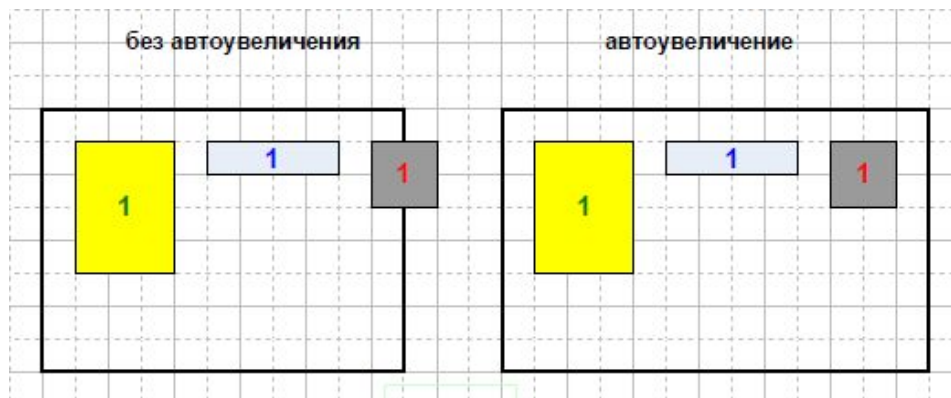


Пример 6

Для 5 объектов, когда размеры хост-мнемосхемы настолько малы, что без автоматического увеличения ее ширины и высоты происходит наложение сцен.

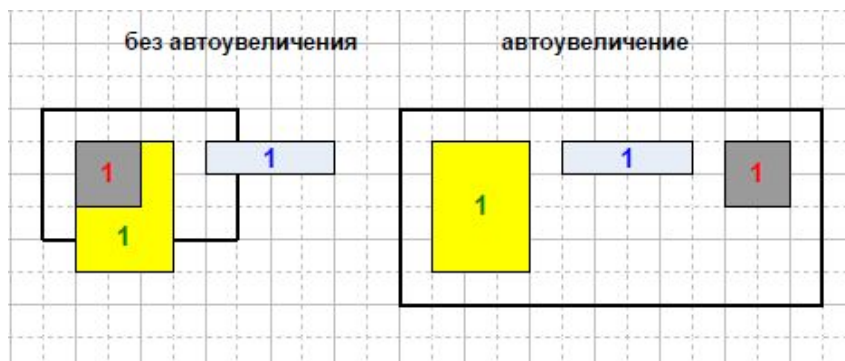
**Пример 7**

Для 1 объекта, когда высота и ширина хост-мнемосхемы достаточны, чтобы сцены не накладывались друг на друга без автоматического увеличения ее размеров.



Пример 8

Для 1 объекта, когда ширина хост-мнемосхемы мала и без автоматического увеличения ее размеров происходит наложение сцен.



2.4.9.3 Страница привязки тегов

Страница привязки тегов является необязательной и может быть пропущена.

При открытии, страница привязки содержит список объектов для привязки.

Мастер настройки технологических объектов

Вид привязки
☒ Один объект ☐ Несколько объектов

Обзор...

Объект	Привязывать	Маска для привязки
Библиотечный объект_0	<input checked="" type="checkbox"/>	(нет)
Библиотечный объект_1	<input checked="" type="checkbox"/>	(нет)
Библиотечный объект_2	<input checked="" type="checkbox"/>	(нет)
Библиотечный объект_3	<input checked="" type="checkbox"/>	(нет)
Библиотечный объект_4	<input checked="" type="checkbox"/>	(нет)

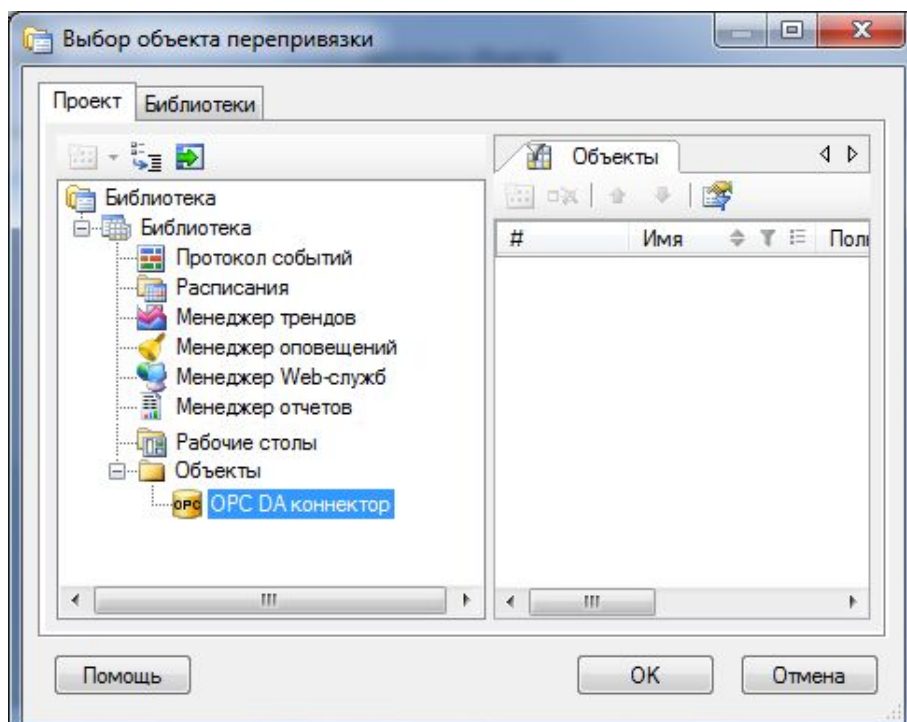
Помощь << Назад Далее >> Пропустить Отмена

В начале, выбираем вид привязки: один или несколько объектов. По-умолчанию выбран – **Один объект**. Если необходимо осуществить привязку к нескольким объектам, выбираем – **Несколько объектов**.

Выбор объектов

Для выбора объектов, с которыми необходимо установить связь следует:

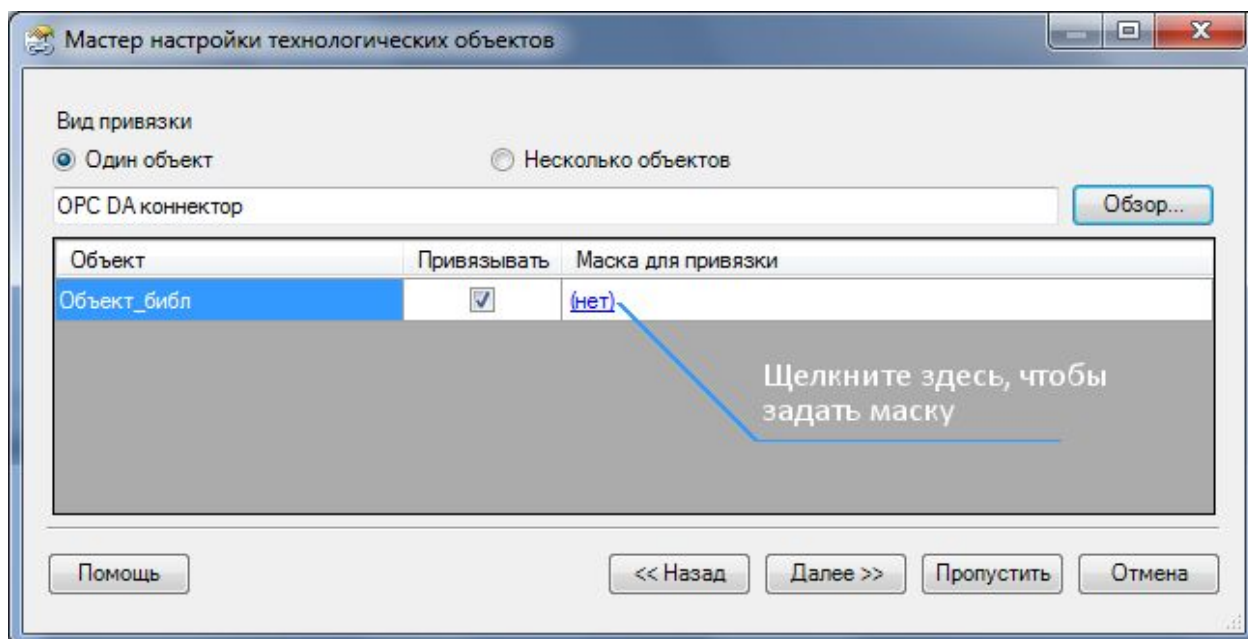
- 1 Нажать на кнопку **Обзор...**
- 2 В появившемся окне **Выбор объекта перепривязки** выбрать необходимые объекты (объект).



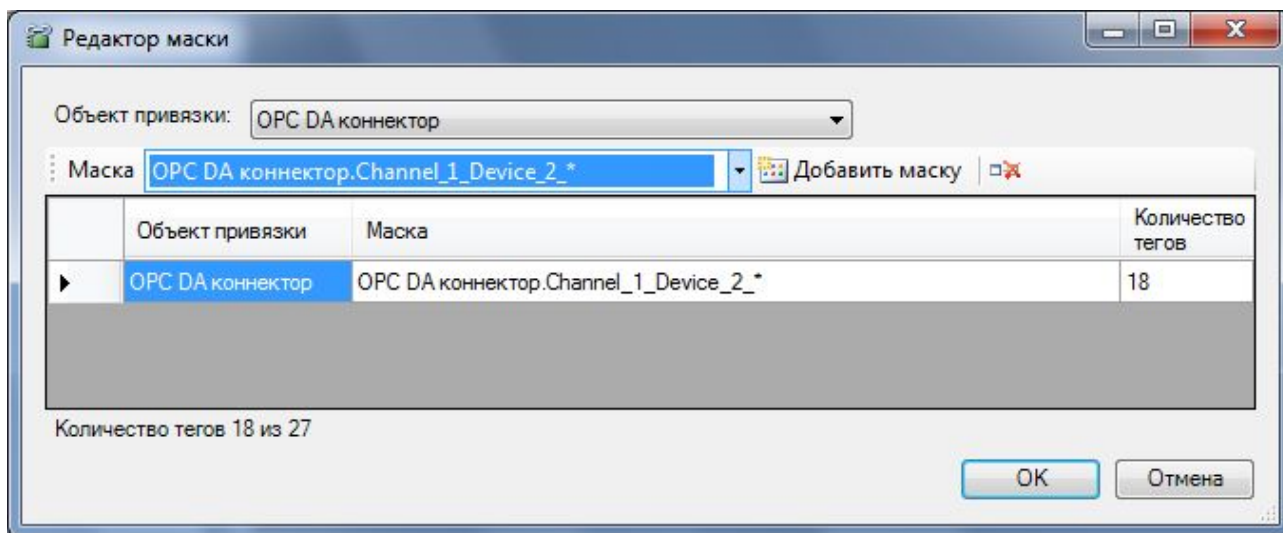
Добавление маски для привязки

После выбора объектов для каждого перепривязываемого объекта необходимо задать маску. Для задания маски следует:


- 1 В окне **Мастер настройки технологических объектов** щелкнуть левой клавишей мыши по ссылке, отображаемой в колонке **Маска для привязки**



- 2 В появившемся окне **Редактор маски** выбрать необходимый объект из выпадающего списка **Объект привязки** (если ранее был выбран вид привязки **Несколько объектов**)
- 3 Выбрать из выпадающего списка **Маска** маску для привязки
- 4 Нажать на кнопку **Добавить маску**



Каждому объекту можно задать несколько масок для привязки, для чего нужно повторить действия 2 – 4.

Удалить ненужные маски можно нажатием клавиши **Del** либо щелчком по кнопке  на панели редактора маски.

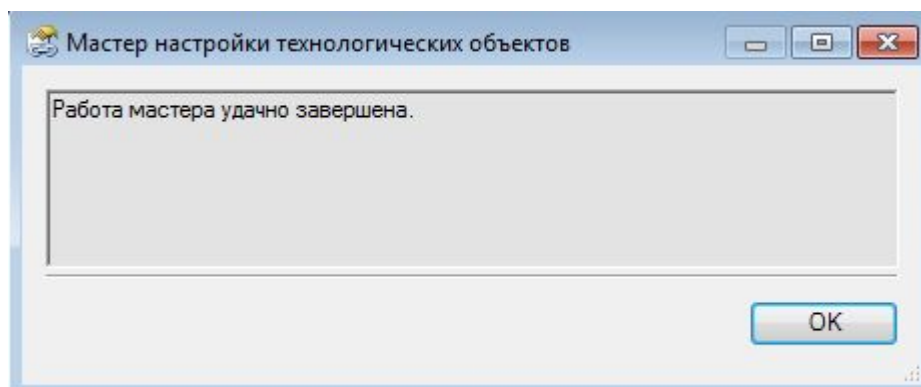
ВНИМАНИЕ!!!

Задать привязку тегов можно позднее из контекстного меню объекта «Перепривязать объекты».

2.4.9.4 Информационная страница

Информационная страница является последней страницей мастера, если его работа не была прервана по кнопке **Отмена**.

Как правило, на ней отображается сообщение с результатами выполнения мастером своих действий.

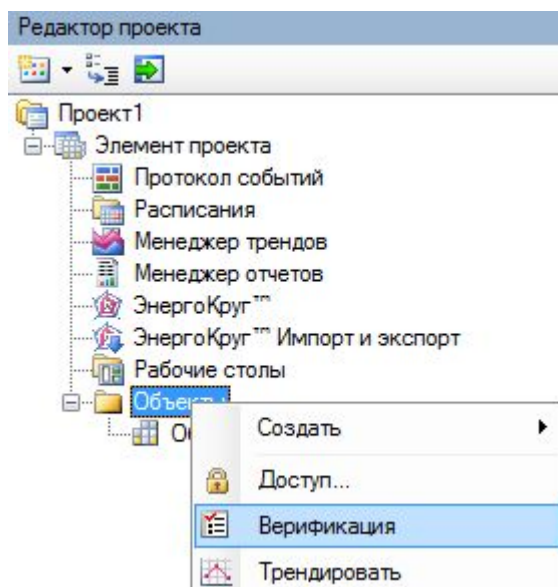


2.4.10 Верификация объектов

Верификация дает возможность пользователю произвести комплексную проверку работоспособности элементов объектной модели и помогает выявить ошибки на этапе разработки. Верификация доступна как для основной, так и для библиотечной части проекта. Каждый объект обладает своим набором правил, определяющих правильность настройки его свойств. Соответствие или несоответствие этим правилам проверяется при верификации.

Чтобы запустить процесс верификации необходимо:

- 1 Выделить в редакторе проекта объект
- 2 В контекстном меню объекта выбрать пункт **Верификация**

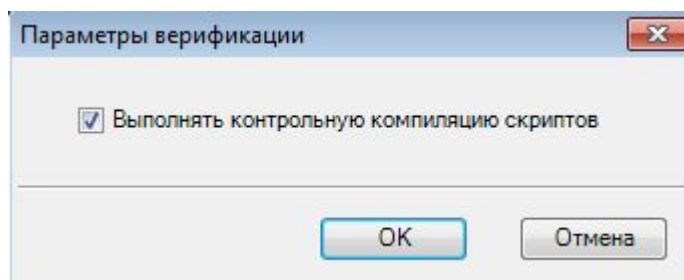


ВНИМАНИЕ!!!

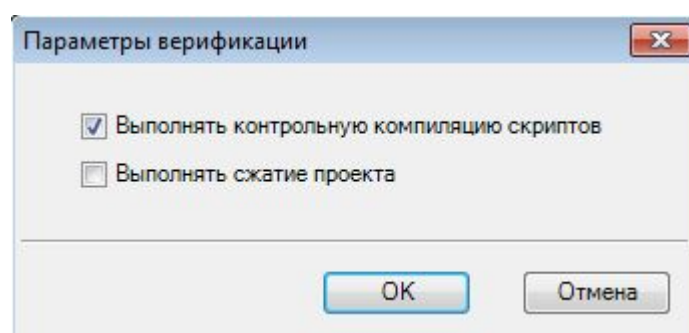
Объекты, не поддерживающие функцию верификации, не будут иметь соответствующий пункт **Верификация**.

Если верифицируемый объект является составным, то проверяться будут и все дочерние вложенные объекты.

- 3 В открывшемся окне **Параметры верификации** выставить флажки для опциональных параметров:
 - **Выполнять контрольную компиляцию скриптов.** По умолчанию данная опция находится во включенном состоянии



- **Выполнять сжатие проекта.** По умолчанию данная опция выключена. Под сжатием проекта понимается удаление из файла элемента проекта (файл с расширением **kritem**) недействительных или невалидных объектов, которые когда-то были удалены из дерева элемента проекта, но их данные остались в файле. Необходимость в такой ситуации возникает, если не проводилась операция принудительной упаковки. Если верифицируемым элементом является проект, то сжиматься будут все элементы проекта, входящие в него.



3 ДИНАМИЧЕСКАЯ ВИЗУАЛИЗАЦИЯ ДАННЫХ

Графическая подсистема **DataRate** позволяет разработчикам легко и быстро создавать объектно-ориентированные графические представления управляемого процесса, а операторам в интерактивном режиме контролировать и управлять этим процессом.

3.1 Графическая подсистема

Графическая подсистема предназначена для построения интерактивного графического интерфейса пользователя автоматизированной системы.

Графическая подсистема представляет собой совокупность графических объектов - **рабочих столов**, **мнемосхем**, **графических примитивов**, а также **изображений** из библиотеки **Image Library**.

3.1.1 Рабочие столы

Рабочий стол – объект проекта, который предназначен для размещения мнемосхем в рабочем пространстве интерфейса пользователя автоматизированной системы.

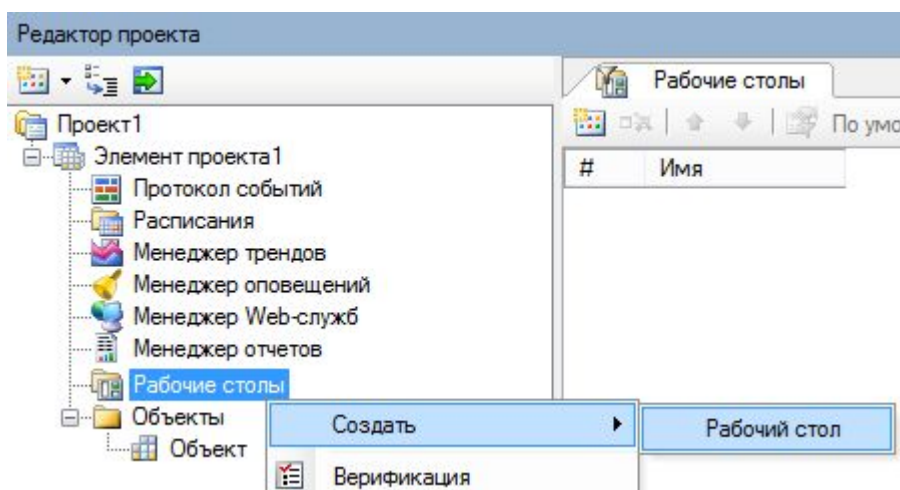
Мастер рабочих столов предоставляет Пользователю возможность настроить окно рабочего стола, разместить в нем мнемосхемы, задать их свойства и расположение.

ВНИМАНИЕ!!!

Для исполнения проекта системы автоматизации необходимо создать хотя бы один рабочий стол!

Добавление в проект

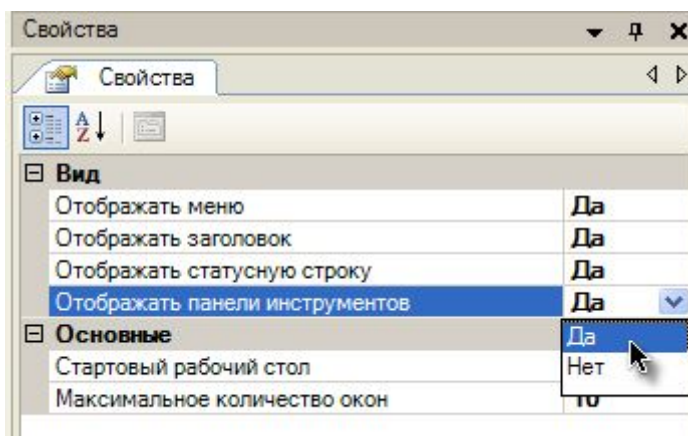
Для добавления рабочего стола в проект выполните команду **Создать\Рабочий стол** в контекстном меню элемента **Рабочие столы**.



Свойства рабочих столов

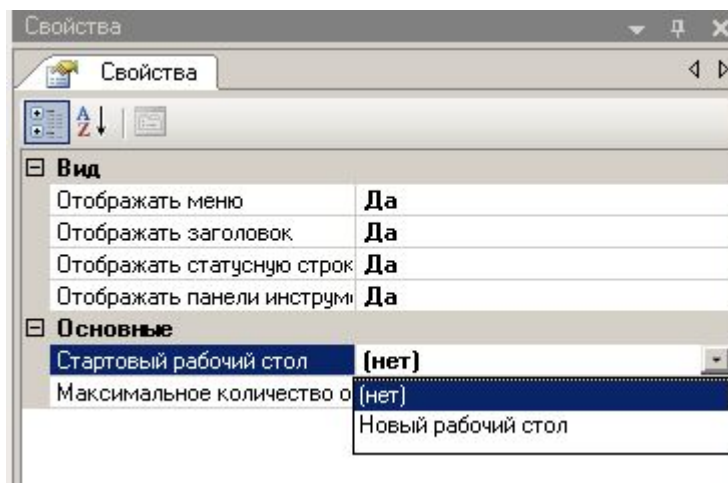
Вид

Эта группа свойств определяет, какие элементы рабочего стола будут отображаться в среде исполнения. Например, если установить свойство **Отображать меню** в значение **Нет**, то в среде исполнения в окне, в котором будет находиться рабочий стол, меню показано не будет.



Основные

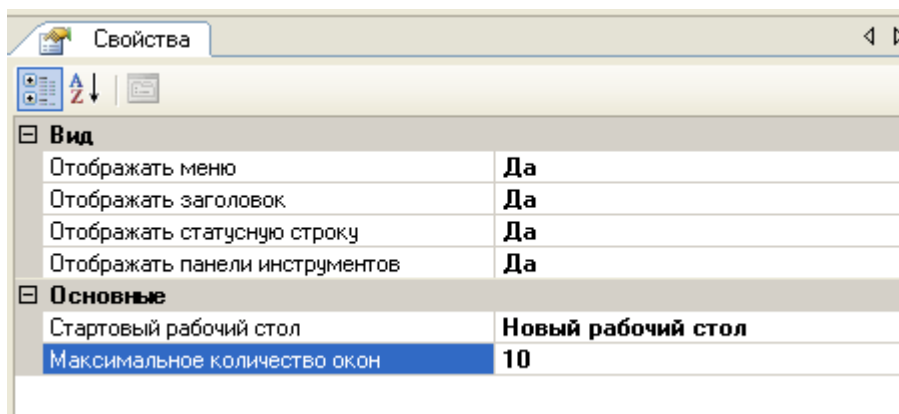
- Стартовый рабочий стол.** Для правильной работы системы необходимо у объекта **Рабочие столы** выбрать в свойстве **Стартовый рабочий стол** один из доступных рабочих столов, который будет загружаться при старте приложения по умолчанию.
 В списке отображаются только те рабочие столы, для которых не назначен пользователь/роль/группа ролей. Пользователю необходимо пройти авторизацию (Авторизация пользователей проекта).
 Если стартовый рабочий стол не назначен, то при старте приложения не будет загружаться ни один рабочий стол.
 По умолчанию свойство **Стартовый рабочий стол** имеет значение – **нет**, т.е. у данного элемента проекта нет рабочего стола.



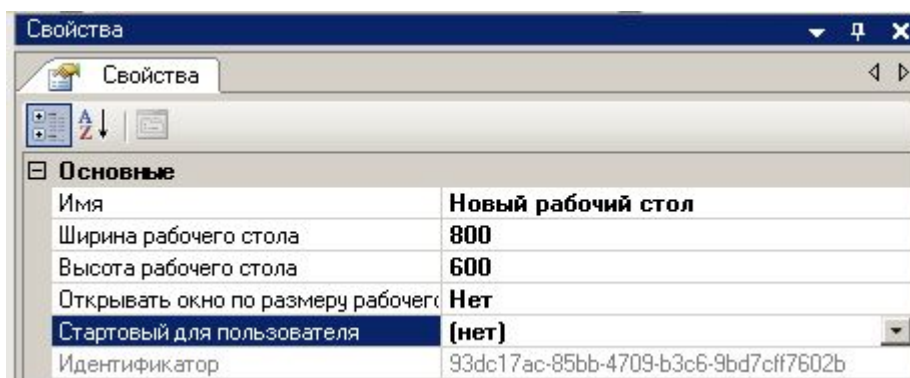
ВНИМАНИЕ!!!

Если Вы не наблюдаете, рабочий стол в списке для назначения его стартовым столом, выполните верификацию рабочих столов. Возможно, у этого стола есть ошибки. Например, для рабочего стола был назначен пользователь, а затем он (пользователь) был удален.

- Максимальное количество окон с мнемосхемами.** Для задания максимального количества окон, которые будут открываться в режиме исполнения проеекта, используйте свойство **Максимальное количество окон** объекта **Рабочие столы**.



Основные свойства рабочего стола



Каждый рабочий стол имеет следующие свойства:

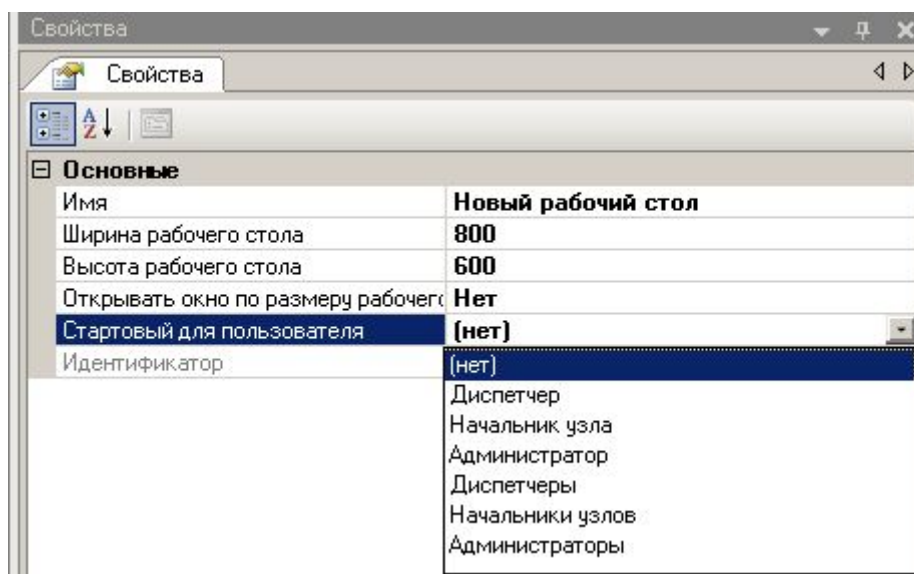
- **Имя** – имя рабочего стола, которое также отображается в среде исполнения
- **Ширина рабочего стола** – ширина рабочего стола в среде исполнения
- **Высота рабочего стола** – высота рабочего стола в среде исполнения
- **Открывать окно по размеру рабочего стола в среде исполнения** – открывать окно по размеру рабочего стола в среде исполнения
- **Стартовый для пользователя** – устанавливается **пользователь/роль/группа ролей**, для которого при входе в среду исполнения открывается данный рабочий стол.

Если пользователь не выбран для текущего рабочего стола, то он попадает в разряд стартового рабочего стола, который загружается при открытии проекта в среде исполнения.

Можно назначить рабочий стол стартовым, как для пользователя, так и для роли или группы ролей. Если рабочий стол назначается стартовым для пользователя, то при входе пользователя открывается этот рабочий стол. Если назначается стартовым для роли, то при входе любого пользователя, относящегося к данной роли, будет открываться этот рабочий стол. Если назначается стартовым для группы ролей, то при входе любого пользователя, относящегося к определенной роли, входящей в заданную группу ролей, будет открываться этот рабочий стол.

ВНИМАНИЕ!

Если Вы удаляете пользователя, для которого есть стартовый рабочий стол, не забудьте изменить для рабочего стола параметр "Стартовый для пользователя" на "Нет", либо выполните верификацию рабочих столов и исправьте ошибку.



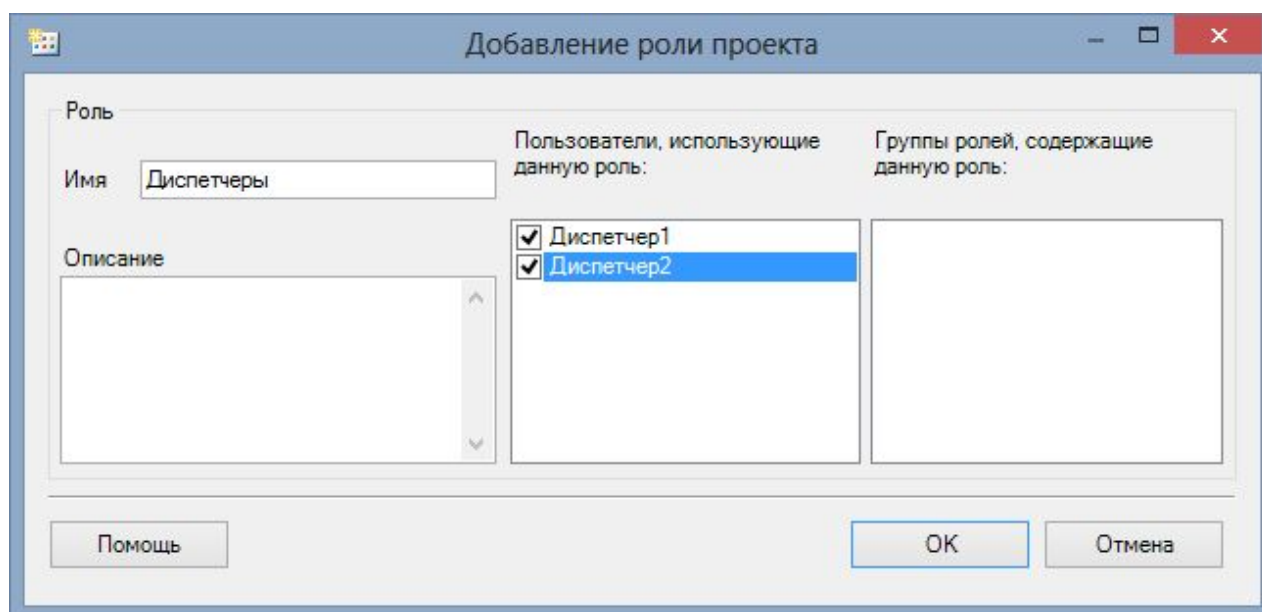
- **Идентификатор.** Идентификатор рабочего стола может использоваться в скриптах.

Пример

Рассмотрим ситуацию, когда для разных пользователей при их входе в автоматизированную систему должен загружаться один и тот же рабочий стол. В этом случае необходимо создать рабочий стол для пользователей на основе их общей роли.

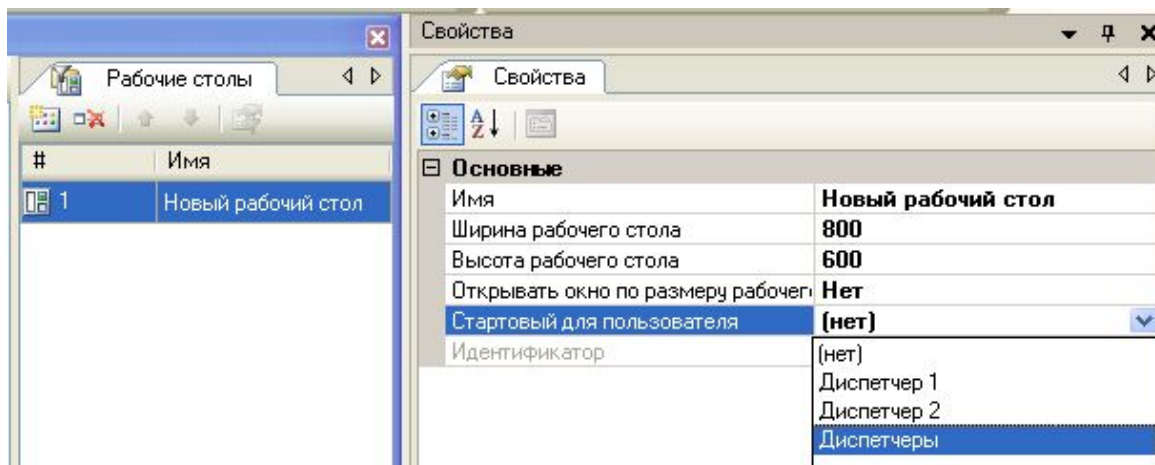
В примере рассмотрим создание рабочего стола для двух пользователей **Диспетчер 1** и **Диспетчер 2**, относящихся к одной роли **Диспетчеры**:

- 1 Сначала создадим двух пользователей (**Диспетчер 1** и **Диспетчер 2**) и объединим их общей ролью (**Диспетчеры**) с помощью настройки пользователей и ролей проекта.



- 2 Затем следует создать мнемосхему, рабочий стол, и задать их непосредственное связывание.

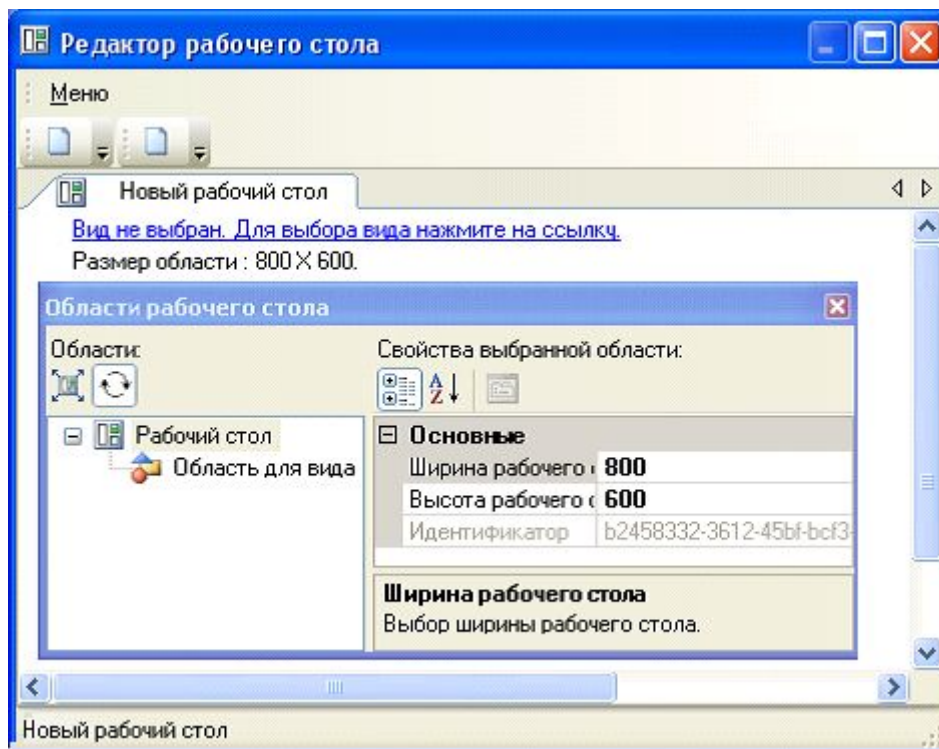
- 3 Далее в свойствах рабочего стола задать параметр **Стартовый для пользователя**, выбрав из выпадающего списка созданную ранее роль **Диспетчеры**.



Редактор рабочего стола

Для редактирования рабочего стола следует:

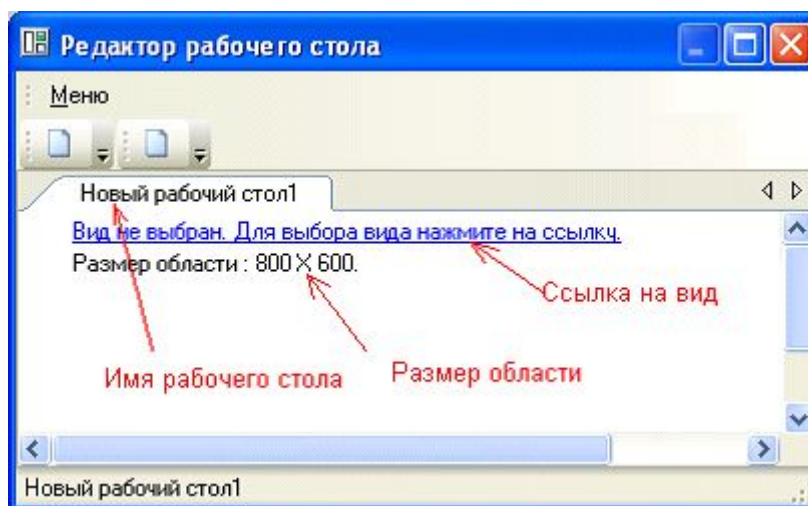
- 1 Выбрать из контекстного меню рабочего стола команду **Редактировать** или щелкнуть два раза мышкой по имени рабочего стола
- 2 В открывшемся окне **Редактор рабочего стола** с открытым окном **Области рабочего стола** задать необходимые значения свойств



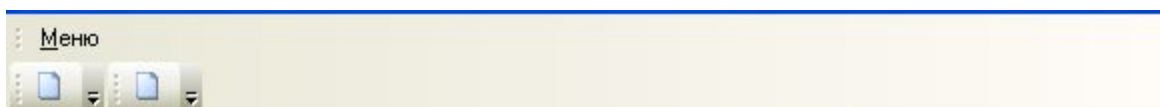
Окно **Области рабочего стола** является немодальным, то есть можно, не закрывая его, работать в окне **Редактор рабочего стола**. Все сделанные изменения будут отражены и в окне **Области рабочего стола**.

Компоненты редактора рабочего стола

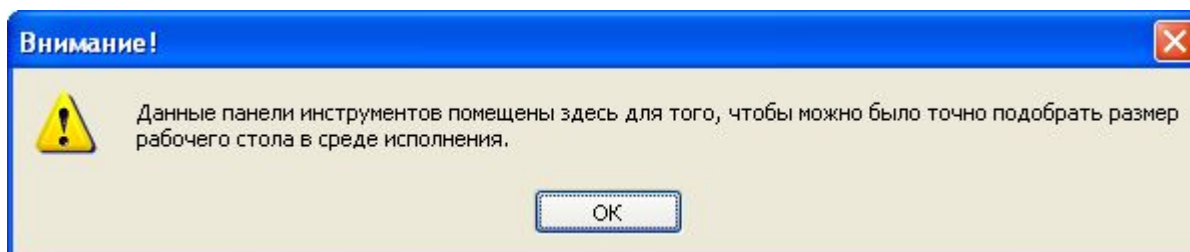
Окно редактора включает элементы, имитирующие меню и панель инструментов графического интерфейса пользователя автоматизированной системы, а также рабочее пространство настройки отображения вида объекта.



Элементы **Меню** и **панель инструментов** редактора имитирует элементы окна графического интерфейса пользователя следующим образом.

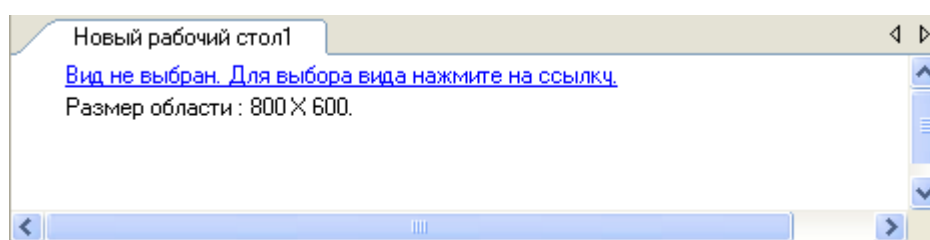


При щелчке мышью по этим элементам будет выдано соответствующее сообщение.




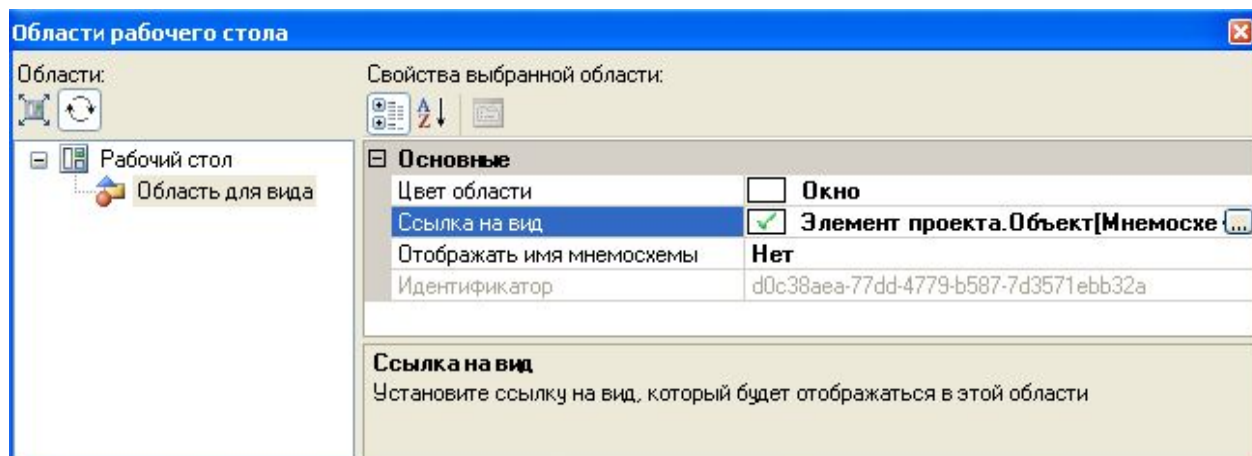
Рабочее пространство редактора отображают свойства рабочего стола:

- **Название рабочего стола.** В данном примере – **Новый рабочий стол1**
- **Размер области** – размер рабочего стола в пикселях
- **Ссылка на вид** в проекте. Ссылка отображается синим цветом и указывает название вида **DataRate**, отображаемого на рабочем столе. Если ссылка не задана, отображается соответствующий текст.

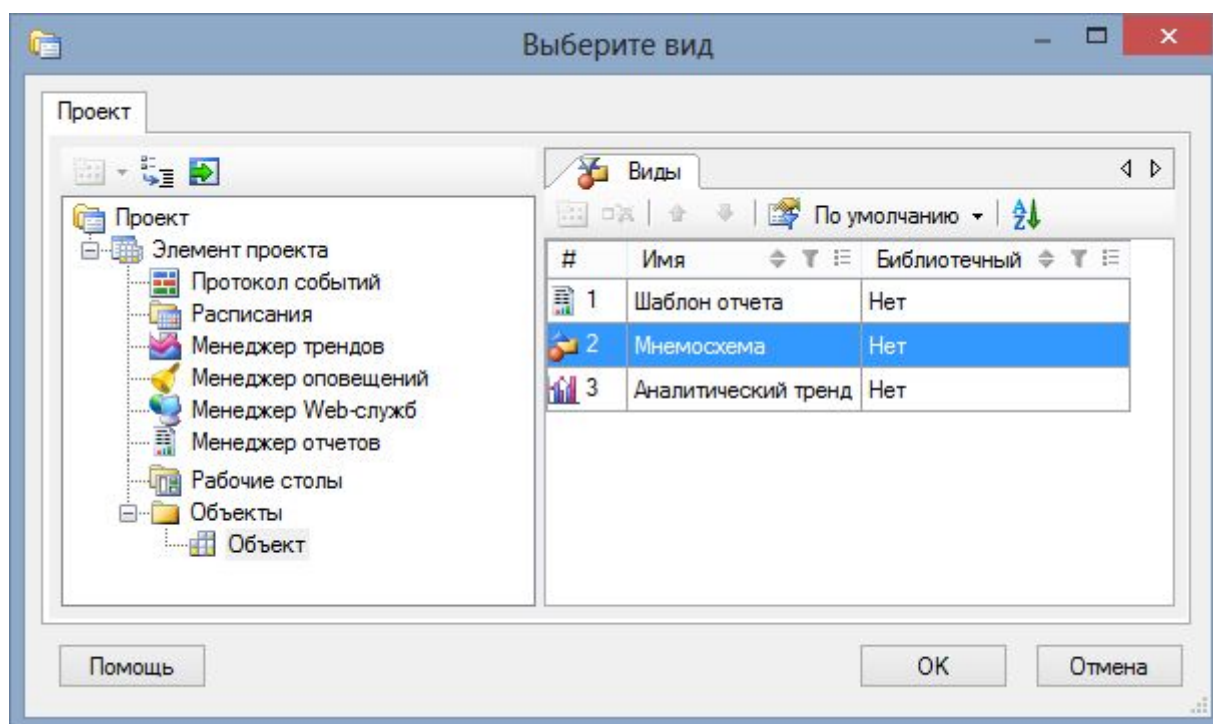


Для задания вида, который будет отображаться в области рабочего стола, следует:

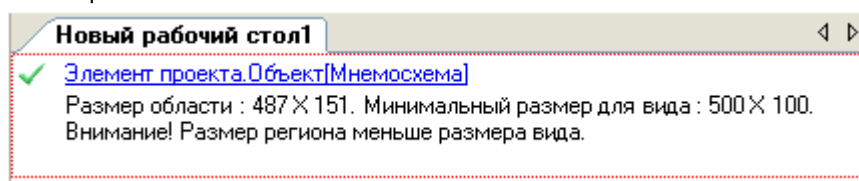
- 1 Активизировать свойство **Ссылка на вид** в окне **Области рабочего стола** и щелкнуть мышью по кнопке 



- 2 В появившемся окне **Выбор элемента проекта** выбрать необходимый вид объекта.

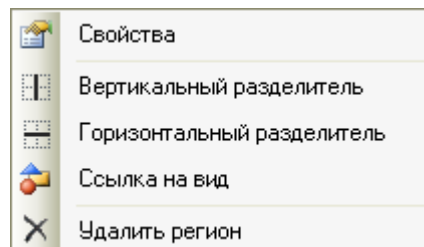


Красный прямоугольник показывает размеры выбранного вида. Если размер области будет меньше размера вида, то об этом выводится сообщение.

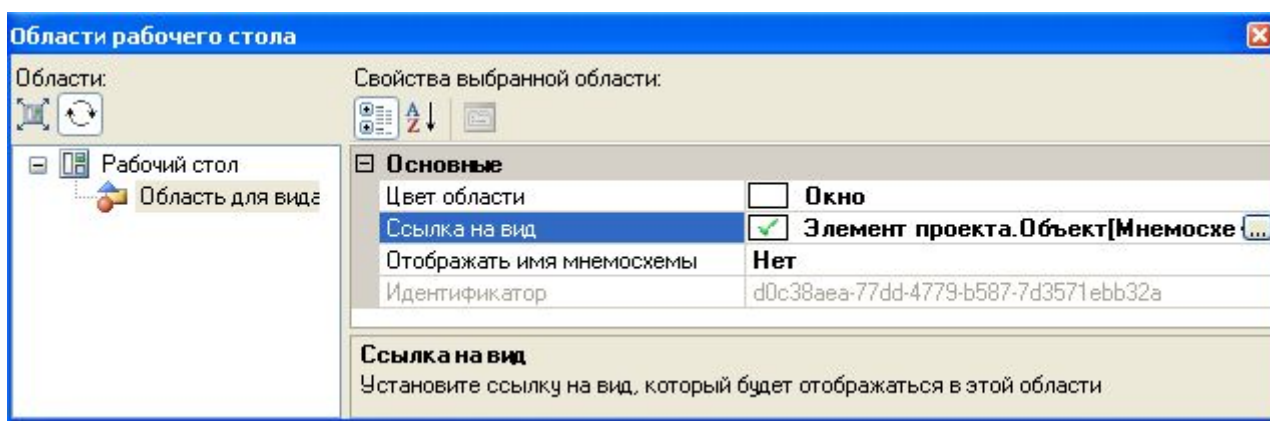


Функции редактирования

Для вызова функций редактора рабочего стола щелчком правой кнопки мыши внутри области вида откройте **контекстное меню** и активизируйте необходимую функцию:



- **Свойства** вызывает диалоговое окно **Области рабочего стола**.



Дерево элементов рабочего стола отображает элементы рабочего стола и позволяет для каждой **Область для вида** настроить следующие свойства:


- **Цвет области** – выбор цвета области. Он будет применен при открытии рабочего стола в режиме исполнения
- **Ссылка на вид** – открывает окно выбора вида
- **Отображать имя мнемосхемы** – отображает имя мнемосхемы в заголовке рабочего стола в режиме исполнения. Может устанавливаться только для одной области.

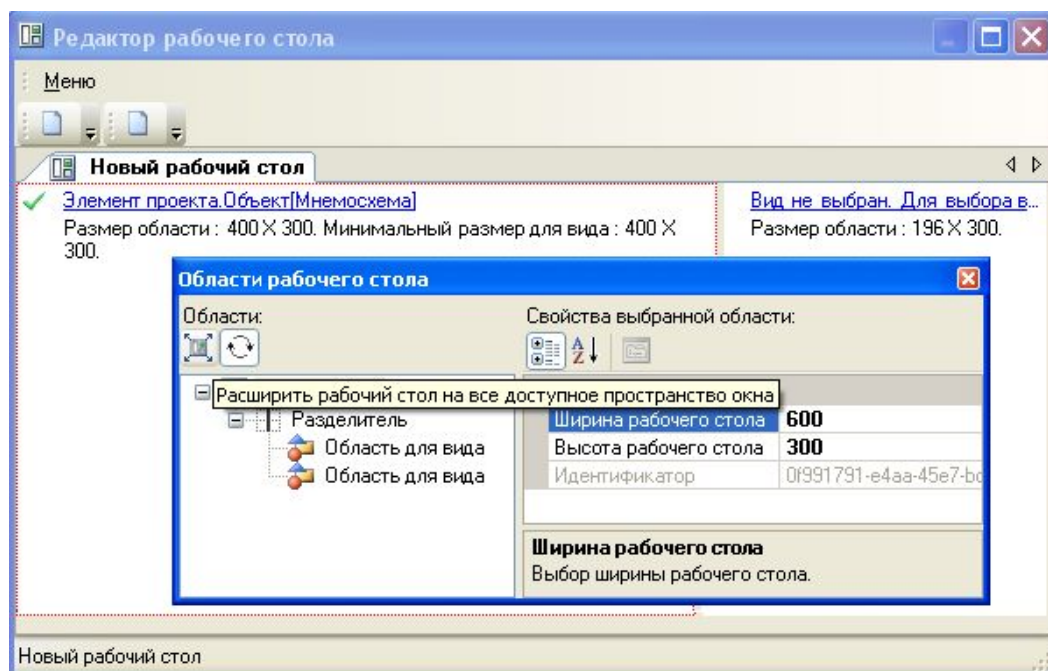
Окно **Области рабочего стола** содержит следующие кнопки:




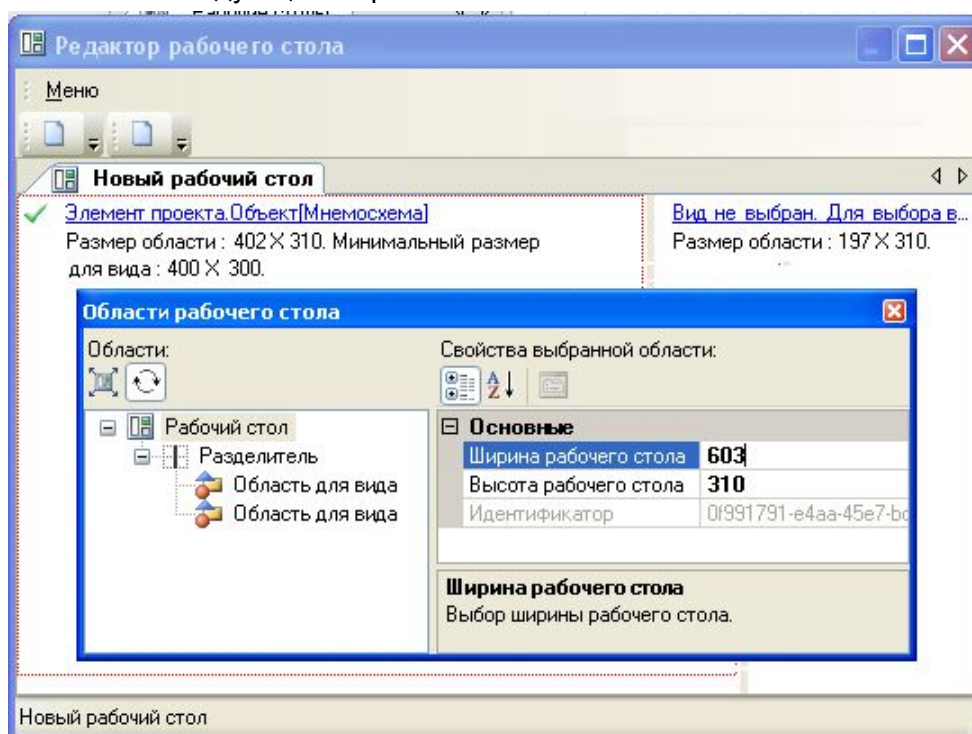
– подсветка региона при выделении. При выделении региона появится мигающий прямоугольник точно соответствующий размеру области рабочего стола.



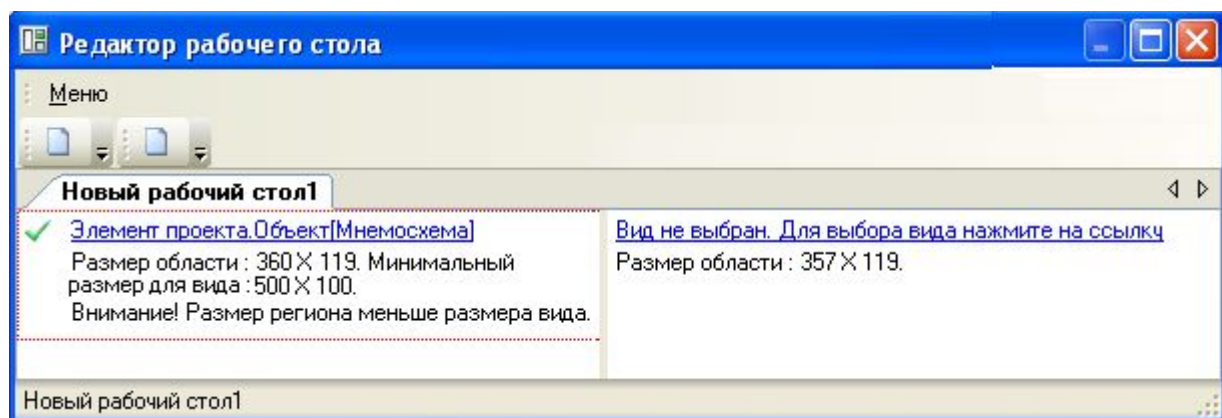
– подгоняет размер рабочего стола по размеру рабочего пространства в окне Редактор рабочего стола. Если рабочая область Редактора рабочего стола меньше размера рабочего стола, то при нажатии на кнопку  размер рабочего стола будет уменьшен. Все области рабочего стола при подгонке изменяют свой размер пропорционально. Например, до подгонки размера рабочий стол имел следующие размеры



После подгонки размера (нажатие на кнопку ) размеры рабочего стола изменятся следующим образом



- **Вертикальный разделитель** добавляет новую область справа. Изменять размер области можно с помощью мыши

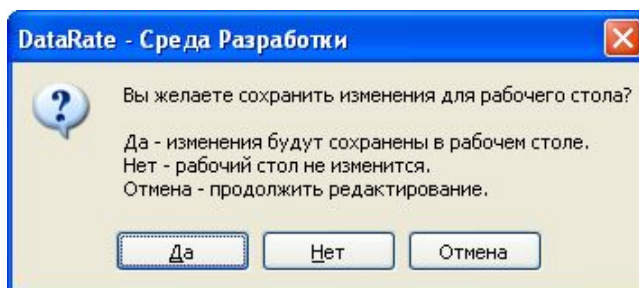


- **Горизонтальный разделитель** делит область пополам, соответственно появляется верхняя и нижняя область. Новая область добавляется снизу. Изменять размер области можно с помощью мыши
- **Ссылка на вид** – открывает окно выбора вида.
- **Удалить регион** – удаляет текущий регион (область).

Первоначально рабочий стол содержит только одну область. Для того чтобы добавить еще одну область, нужно добавить вертикальный или горизонтальный разделитель.

Сохранение изменений сделанных в редакторе

Во время закрытия редактора рабочего стола будет выдано сообщение с предложением сохранить все сделанные изменения.

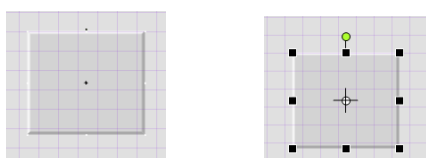


3.1.2 Мнемосхема

Мнемосхема – это анимированное изображение технологического объекта, состоящее из графических примитивов.

Мнемосхема является основной категорией вида объектов.

Свойство **Размер модификаторов** позволяет задать размер модификаторов примитива используемых для различных действий связанных с трансформацией объекта.



3.1.2.1 Система координат и привязка к сетке

Координаты – пара чисел (x, y), определяющих положение каждой точки на мнемосхеме.



Координатная ось – это воображаемая линия, определяющая направление изменения координаты. На мнемосхеме имеется две стандартные оси - X и Y.

Начало координат – точка пересечения координатных осей, имеющая координаты (0;0). Она находится в левом верхнем углу мнемосхемы.

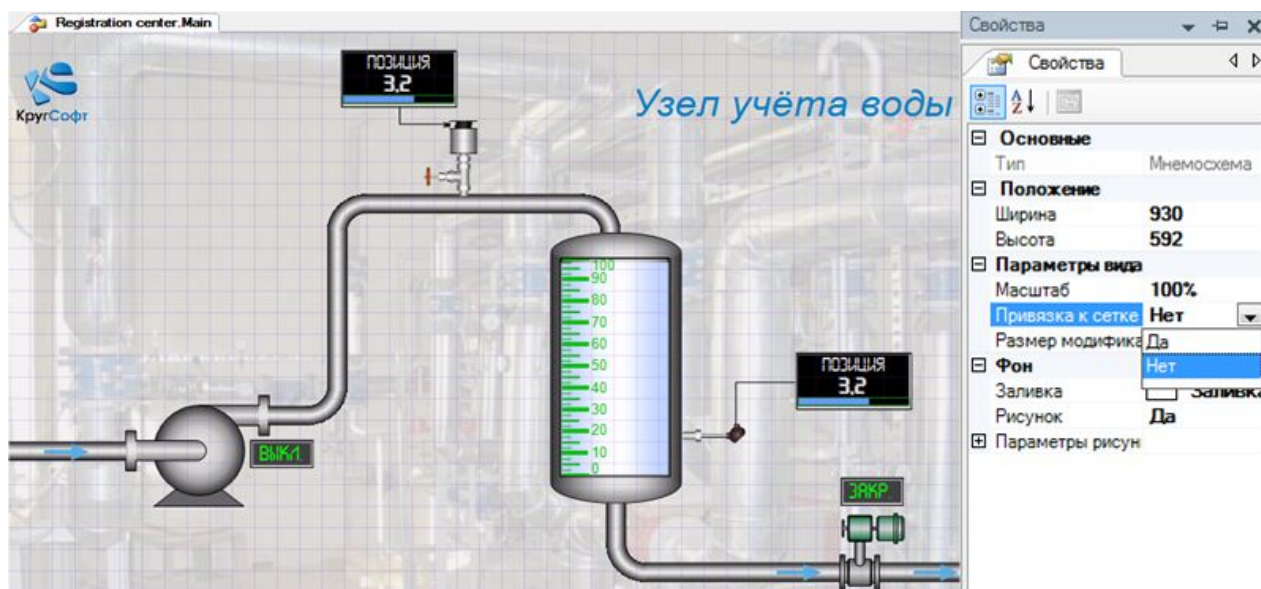
Привязка к сетке

Свойство мнемосхемы **Привязка к сетке** имеет два значения:

- **Да** – графический примитив перемещается строго по узлам сетки
 - **Нет** – графический примитив перемещается произвольно.
- Значение **Нет** устанавливается по умолчанию.

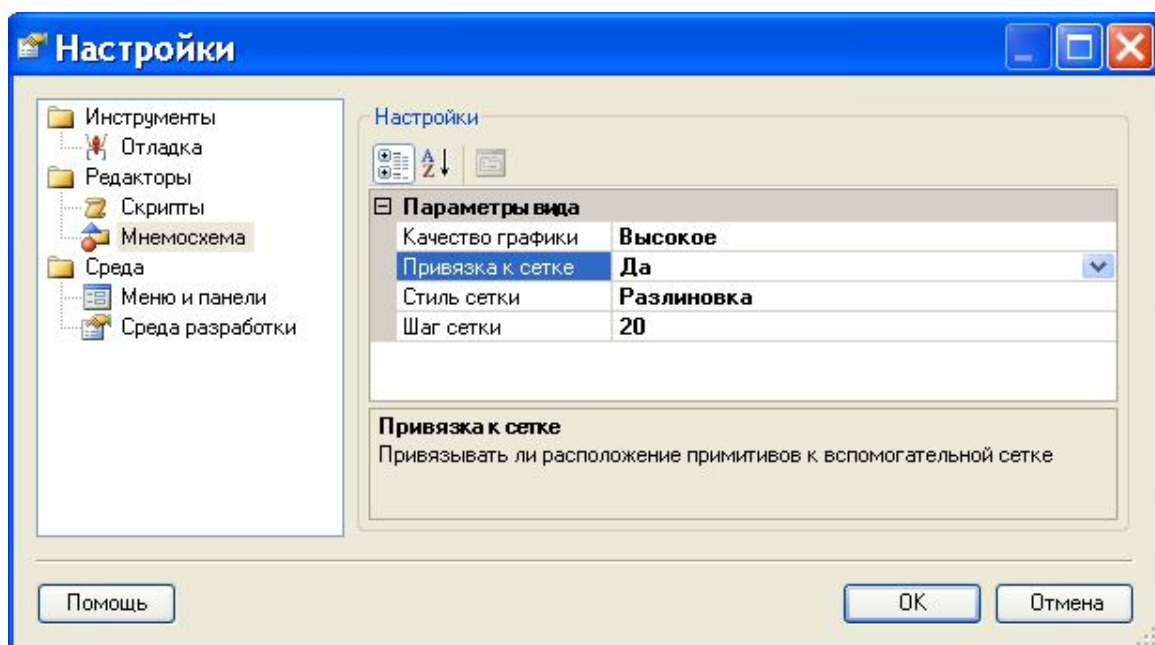
Если пользователю необходимо во время редактирования мнемосхемы временно изменить эту настройку, то это возможно сделать через окно **Свойства**.

Изменения свойств самой мнемосхемы не сохраняются.



Если необходимо, чтобы всегда при открытии мнемосхем свойство **Привязывать к сетке** было выставлено в **Да**, следует изменить эту настройку через окно **Настройки** (Редактор мнемосхем) среды разработки DataRate.

Настройка **Привязка к сетке** является общей для среды разработки и поэтому распространяется на все открываемые мнемосхемы




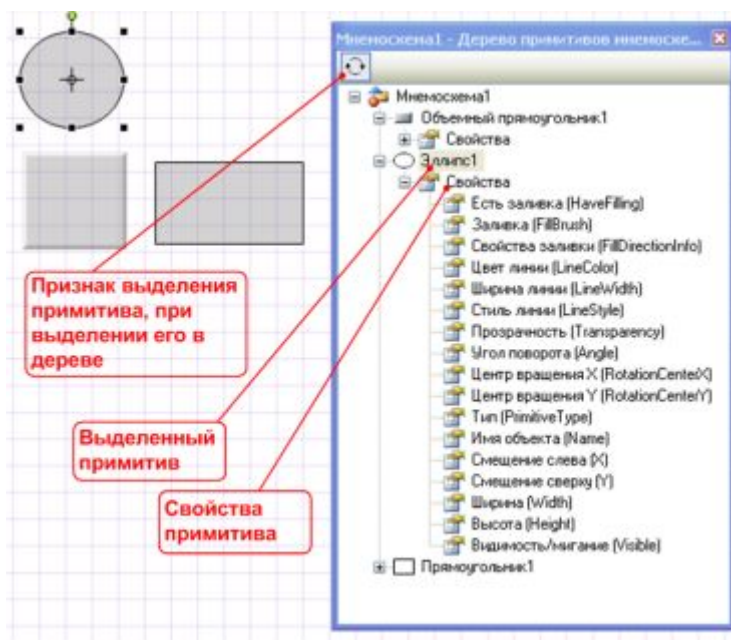
3.1.2.2 Дерево примитивов мнемосхемы

Дерево примитивов мнемосхемы — это отображение списка графических примитивов мнемосхемы и их свойств в отдельном окне в иерархическом виде.

В дереве каждый узел графического примитива содержит узел **Свойства** примитива.

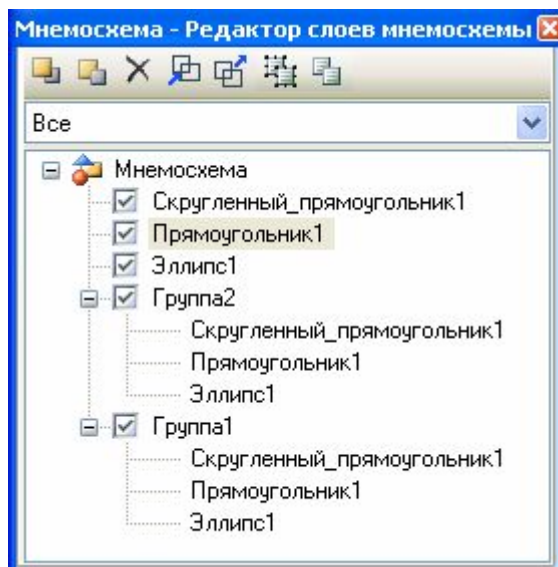
Узлы свойств используются в окне аргументов для добавления ссылок на свойства примитива.

Кнопка  — переключатель **режима выделения**. При нажатой кнопке включается режим, при котором выделенный в дереве примитив становится выделенным на мнемосхеме.



3.1.2.3 Редактор слоев мнемосхемы

Редактор слоев имеет панель инструментов, выпадающий список для выбора фильтра и список в виде дерева, отображающий выделенные примитивы.



Панель инструментов содержит следующие команды:

- Переместить выделенный в списке примитив вперед на одну позицию.
- Переместить выделенный в списке примитив назад на одну позицию.
- Удалить примитив с мнемосхемы.
- Вынести примитив из группы. Примитив будет располагаться перед группой.
- Добавить примитив в группу. Примитив помещается на самый верх группы.
- Выделить все (установить все галочки).
- Убрать выделение (снять все галочки).

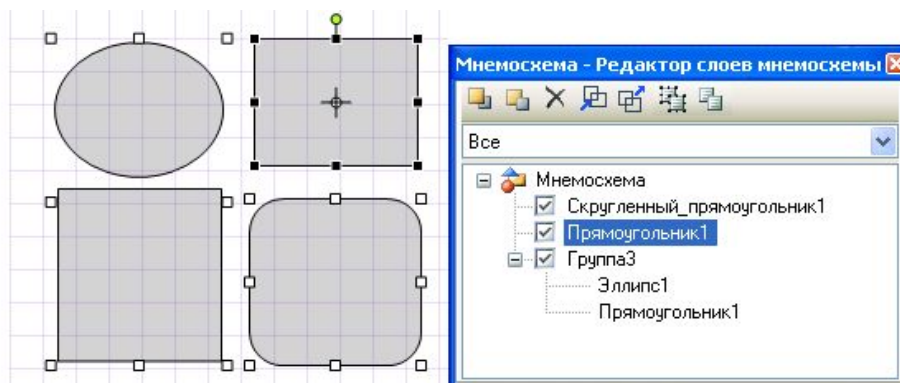
Управление выделением примитивов

Если в редакторе слоев снять галочку, то снимется выделение примитива на мнемосхеме.

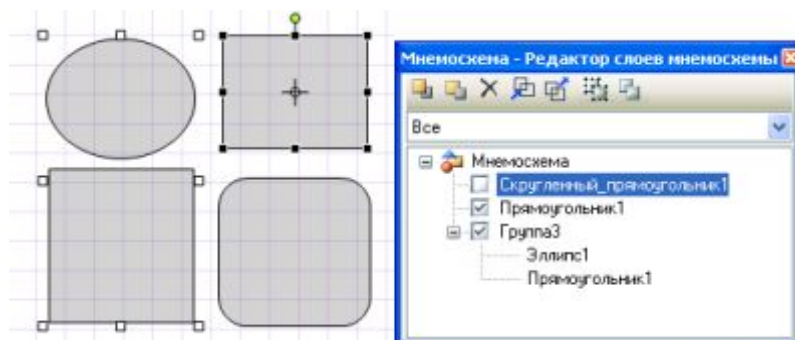
Выделять можно только примитивы и группы, лежащие непосредственно на мнемосхеме. Примитивы и группы, находящиеся внутри группы, выделять нельзя.

Пример:

Пусть на мнемосхеме выделена группа и два примитива.

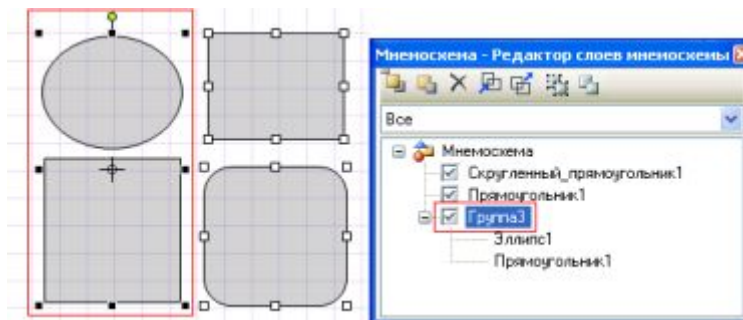


Снимем галочку напротив **Скругленный_прямоугольник1** в редакторе слоев, одновременно снимется выделение с примитива в мнемосхеме.



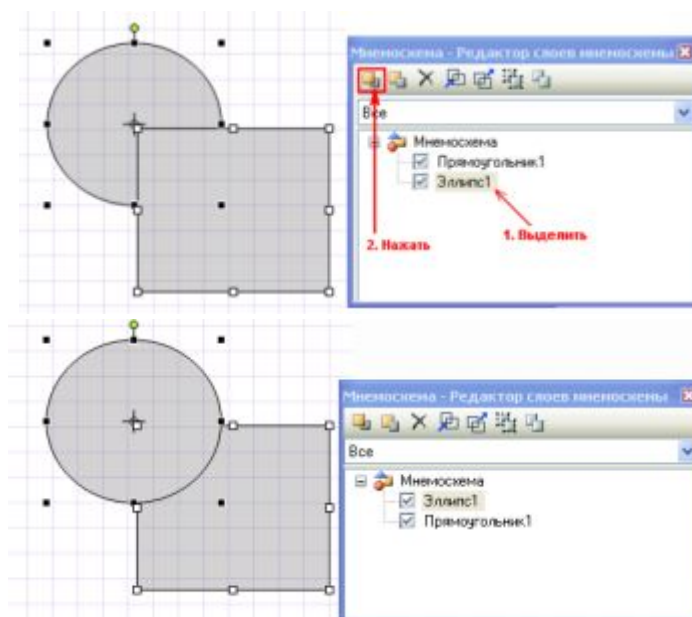
Управление активностью примитива

Выделенный в редакторе слоев примитив будет активным – выделен черными прямоугольниками – и его свойства будут отображаться в окне свойств.



Перемещение примитива вперед

Для перемещения примитива вперед нужно выделить его в редакторе слоев и нажать на кнопку **Вперед**. На мнемосхеме примитив переместится вперед, а в редакторе слоев - вверх.



Команда доступна только при выделении какого-либо примитива или группы. Действие команды может быть отменено с помощью команды **Отмена**.

Перемещение примитива назад

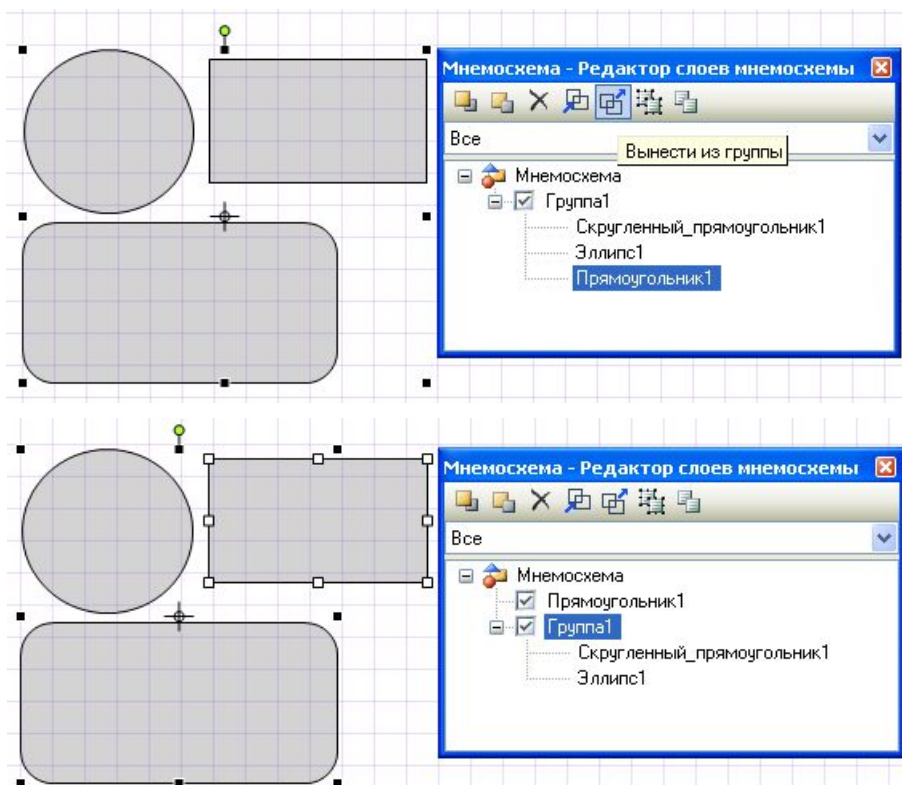
Для перемещения примитива назад нужно выделить его в редакторе слоев и нажать на кнопку **Назад**. На мнемосхеме примитив переместится назад, а в редакторе слоев - вниз. Команда доступна только при выделении какого-либо примитива или группы. Действие команды может быть отменено с помощью команды **Отмена**.

Удаление примитива

Для удаления примитива нужно выделить его в редакторе слоев и нажать на кнопку **Удалить**. Примитив будет удален с мнемосхемы и из редактора слоев. Команда доступна, если выделен примитив или группа. Можно также удалять примитивы и группы, лежащие в группе. Действие команды может быть отменено с помощью команды **Отмена**.

Вынос примитива за пределы группы


Для выноса примитива из группы нужно выделить примитив внутри группы и нажать на кнопку **Вынести из группы**. Примитив будет вынесен из группы и расположится перед группой.

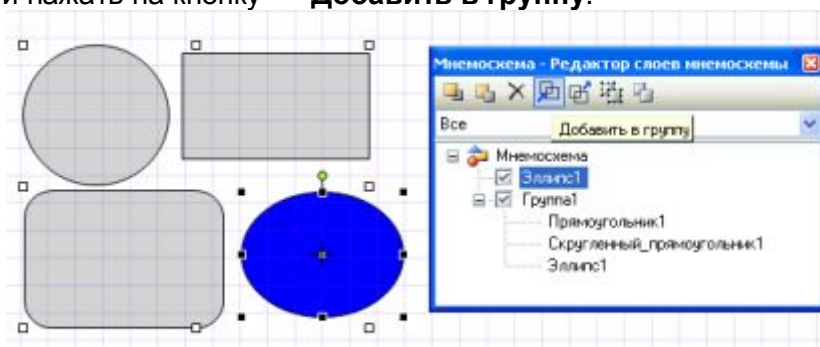


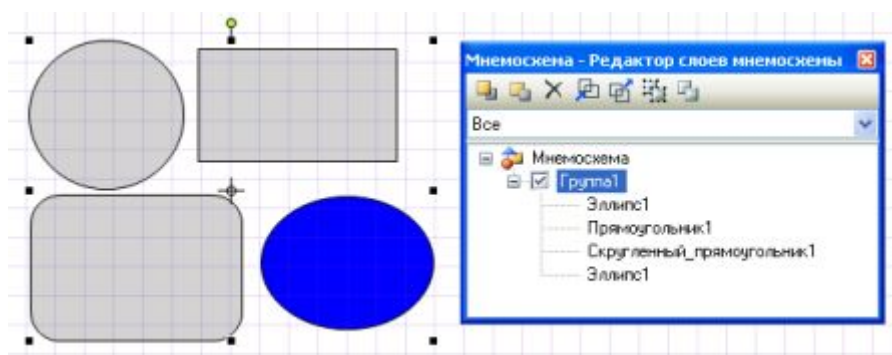
Для выноса примитива из группы можно также воспользоваться мышкой, и просто перетащить примитив из группы на мнемосхему.

Если примитив вложен во внутреннюю группу, то его можно перетащить как за пределы группы, в которую он входит, так и за пределы внешней группы.

Добавление примитива в существующую группу

Для добавления примитивов в группу нужно отметить галочками примитивы и группы, которые нужно добавить в группу, выделить группу, в которую будут добавлены примитивы, и нажать на кнопку  **Добавить в группу**.

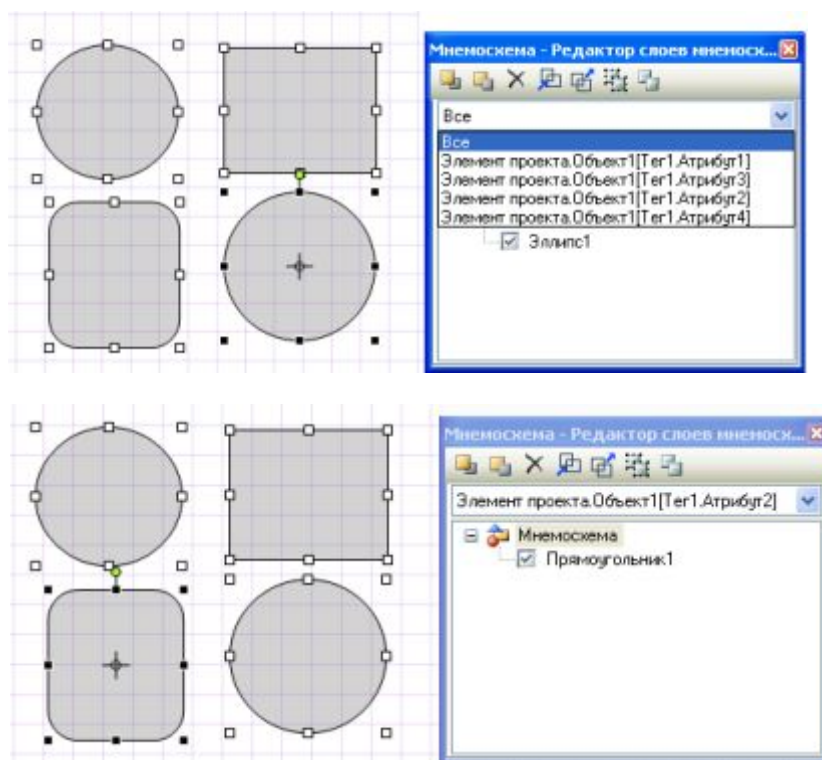




Для добавления примитива в группу можно также воспользоваться мышью. Для этого нужно захватить примитив мышью и дотащить до нужной группы и отпустить.

Использование фильтра

Редактор слоев позволяет из всего списка выделенных примитивов отобразить только те, которые содержат заданную ссылку на атрибут тега или свойство примитива. Выпадающий список содержит все ссылки в выделенных примитивах.




В примере ссылку на **Тег1.Атрибут2** имеет только **Прямоугольник1**.

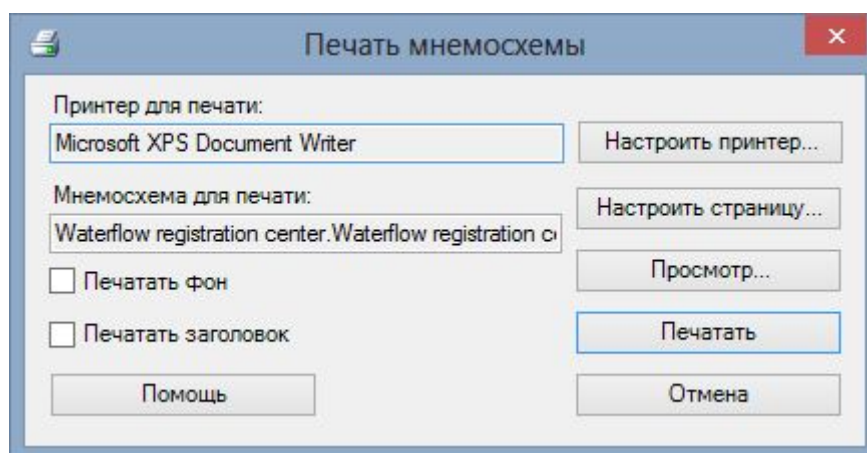
3.1.2.4 Печать мнемосхемы

Печать мнемосхемы позволяет передавать на печатающее устройство, текущее состояние мнемосхемы.

Мнемосхема может быть напечатана, как в режиме исполнения (имитации), так и в режиме редактирования мнемосхемы в среде разработки.

Для печати мнемосхемы следует:

- 1 Нажать на панели инструментов на кнопку **Печатать выделенный элемент** 
- 2 В появившемся окне **Печатать мнемосхемы** настроить параметры печати
 - **Принтер для печати** – отображает стандартный выбранный в операционной системе принтер
 - **Мнемосхема для печати** – отображает имя выводимой на печать мнемосхемы.



Для настройки используйте кнопки:

- **Настроить принтер** – вызовет стандартный диалог настроек принтера для печати.
- **Настроить страницу** – вызовет стандартный диалог настроек параметров страницы.
- **Просмотр** – вызовет окно предварительного просмотра печатаемой мнемосхемы. Нажатие на кнопку **Печатать** в окне предварительного просмотра отправит на печать мнемосхему.
- **Печатать** – отправит выбранную мнемосхему на выбранный для печати принтер.
- **Помощь** – вызовет этот раздел справочной системы.
- **Отмена** – вызовет диалог настроек печати мнемосхемы.






















При установке флажка **Печатать фон** при печати мнемосхемы будет выведен фоновый рисунок или заливка фона мнемосхемы. Иначе мнемосхема будет напечатана без фонового рисунка или заливки фона.

При установке флажка **Печатать заголовок** при печати в левом верхнем углу листа будет выведен заголовок мнемосхемы в формате: **Дата/Время и имя мнемосхемы**.

3.1.3 Графические примитивы

Графические примитивы – это основные объекты построения графических изображений с полным набором стандартных операций редактирования (таких как изменение размера; поворот; перемещение; настройка цвета, уровня прозрачности, степени сглаживания линий, процента заливки), возможностью анимации и динамической визуализации.

Графическая подсистема включает следующие примитивы:

-  – эллипс
-  – прямоугольник
-  – скругленный прямоугольник
-  – объемный прямоугольник
-  – прямая
-  – вспомогательная линия
-  – ломаная
-  – многоугольник
-  – надпись
-  – рисунок
-  – сегмент
-  – сектор
-  – большая дуга
-  – дуга
-  – текст
-  – таблица
-  – сцена
-  – тренд
-  – табличный тренд
-  – протокол событий
-  – элемент управления.

3.1.3.1 Свойства графических примитивов

Каждый графический примитив имеет две группы свойств – **Основные** и **Положение**.

Основные

- **Имя объекта** – определяет уникальное имя примитива
- **Прозрачность** – определяет процент прозрачности примитива
- **Видимость/мигание** – определяет видимость примитива
- **Цвет линии** – определяет цвет линии контура примитива
- **Стиль линии** – определяет стиль линии образующей примитив. Возможно, установить пять стилей: Сплошная, Пунктир, Тире-точка, Тире-точка-точка, Точки
- **Ширина линии** – определяет ширину линии образующей примитив
- **Заливка** – определяет стиль заливки примитива. Свойство **заливка** имеют не все примитивы
- **Свойства заливки** – данное свойство поддерживают примитивы, у которых есть свойство **Заливка**.

Положение

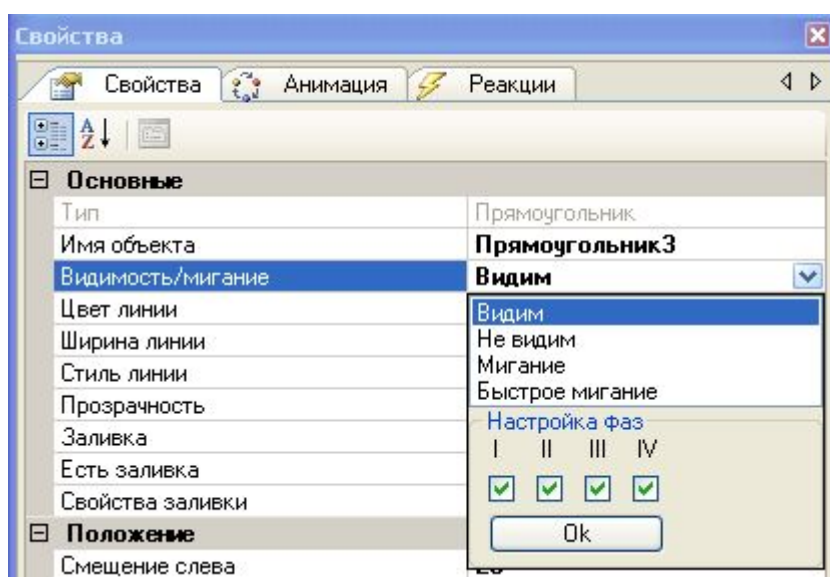
- **Смещение сверху** – определяет координату верхней левой точки примитива по оси X
- **Смещение слева** – определяет координату верхней левой точки примитива по оси Y
- **Ширина** – определяет ширину примитива
- **Высота** – определяет высоту примитива
- **Угол поворота** – определяет угол поворота примитива
- **Центр вращения X/Центр вращения Y** – определяет координату центра вращения примитива по оси X/ по оси Y.

Кроме основных свойств некоторые графические примитивы имеют свои, дополнительные группы свойств.

Управление свойствами графического примитива происходит через окно свойств.

Видимость/мигание.

Данное свойство устанавливает способ отображения графического примитива на мнемосхеме и может использоваться для анимации примитива. Предопределенные значения: **Видим, Не видим, Мигание, Быстрое мигание.** Настройка свойства происходит в окне **Свойства** графического примитива.



В панели **Настройка фаз** можно произвольно указать фазы видимости.

Цвет

Выбор цвета происходит в окне **Свойства** графического примитива. Выбрать нужный цвет поможет:

- **Палитра цветов**
- **Пипетка**
- Кнопка **Подобрать цвет...**

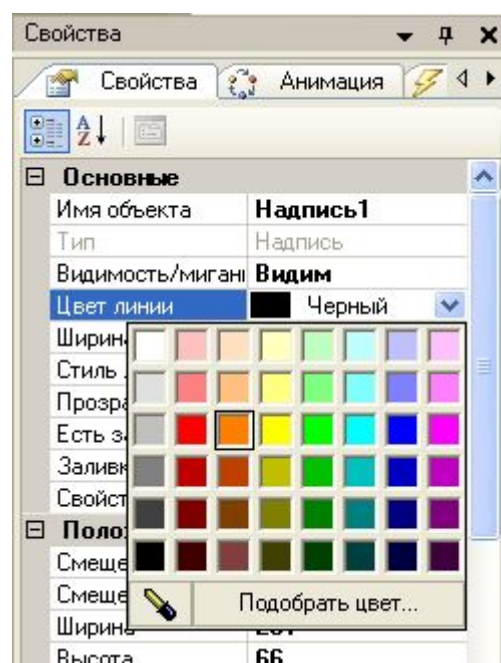
Пример


Рассмотрим выбор цвета линии, используемой для обводки символов надписи.

В окне **Свойства** объекта щелкните на кнопку в строке **Цвет линии**.

В открывшейся палитре цветов выберите необходимый цвет.

Если нужного цвета в палитре нет, но в «пределах приложения» есть необходимый цвет, можно воспользоваться пипеткой.



Щелкните по кнопке с пипеткой . Появится пипетка и рядом с ней окно, залитое цветом, который в данный момент находится под курсором в виде пипетки. После нажатия левой кнопки мыши цвет будет считаться выбранным.

В окне свойств появится выбранный цвет, и вместо пипетки будет отображаться обычный курсор. Соответственно изменится цвет линии.

Если нужный цвет так и не был найден предыдущими способами, тогда необходимо нажать на кнопку **Подобрать цвет...**

В появившемся окне **Выбор цвета** можно по спектру подобрать необходимый цвет, а также настроить палитру цветов.

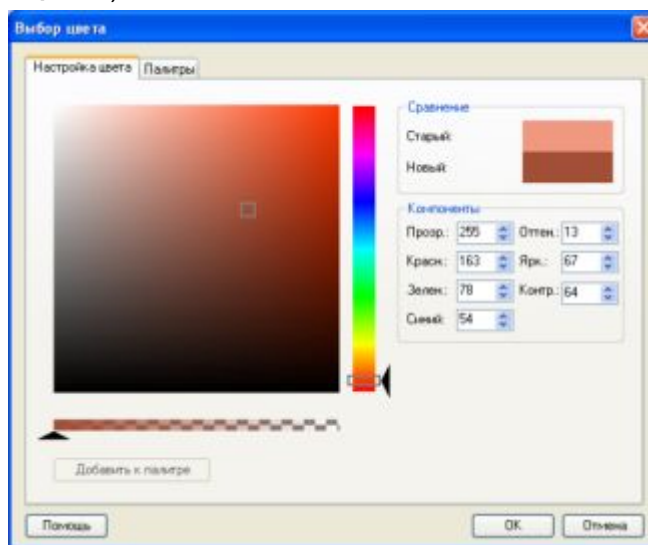
Выбор из спектра (закладка **Настройка цвета**):

- С помощью ползунка справа выберите область спектра
- Выбрав необходимую область в спектре, для выбора цвета щелкните мышкой внутри прямоугольной области
- Ползунком снизу задайте уровень прозрачности.

Панель **Сравнение** показывает текущий цвет (Старый) и цвет, полученный в результате настройки (Новый).

Панель **Компоненты** позволяет задать цвет, установив точные числовые значения составляющих цвета.

Кнопка **Добавить к палитре** добавляет выбранный цвет в текущую палитру. Кнопка становится доступной, если выбрана палитра не по умолчанию.



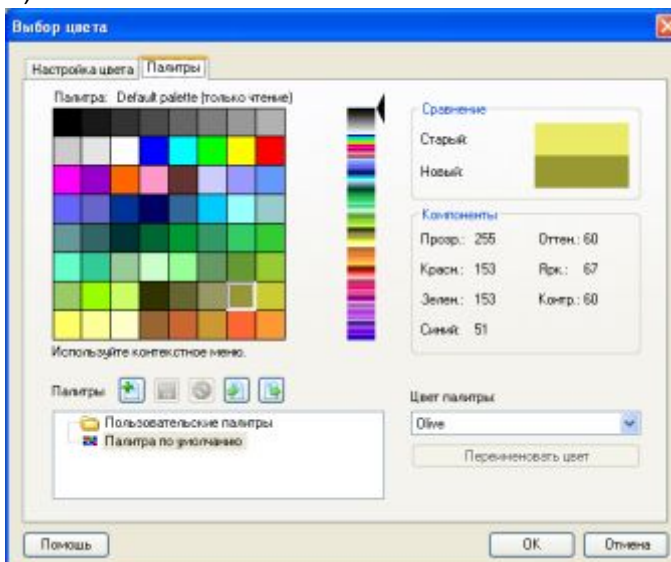
Настройка палитры (закладка **Палитры**)

Ползунок справа позволяет прокрутить палитру и отображает в уменьшенном виде цвета составляющие палитру.

Панель **Сравнение** показывает текущий цвет и цвет, выбранный в палитре.

Панель **Компоненты** отображает составляющие выбранного в палитре цвета.

В выпадающем списке **Цвет палитры** отображается выбранный в палитре цвет. Если выбрать другой цвет в списке, то в панели спектра автоматически будет выбран цвет, имеющий соответствующее имя.




ВНИМАНИЕ!!!

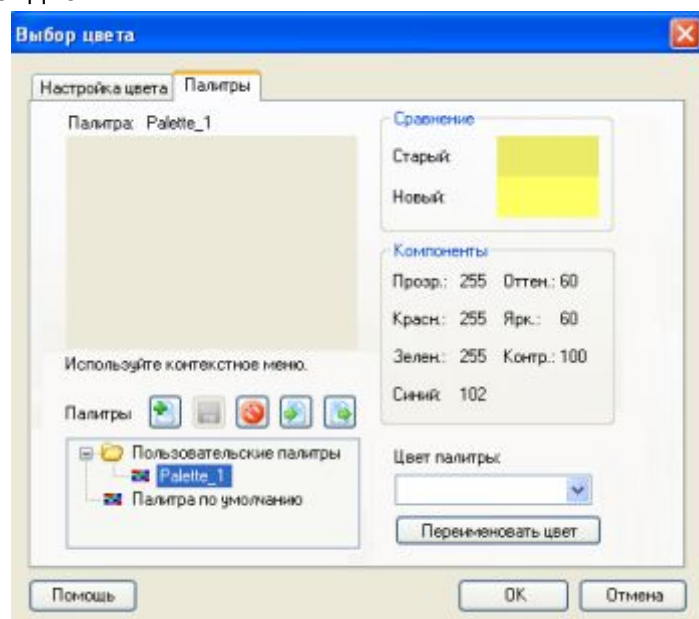
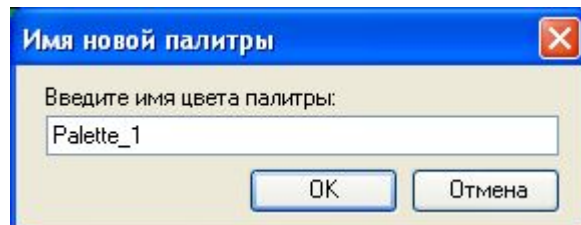
Менять заданные по умолчанию названия цветов нельзя.

Создание пользовательской палитры

Для создания пользовательской палитры следует:

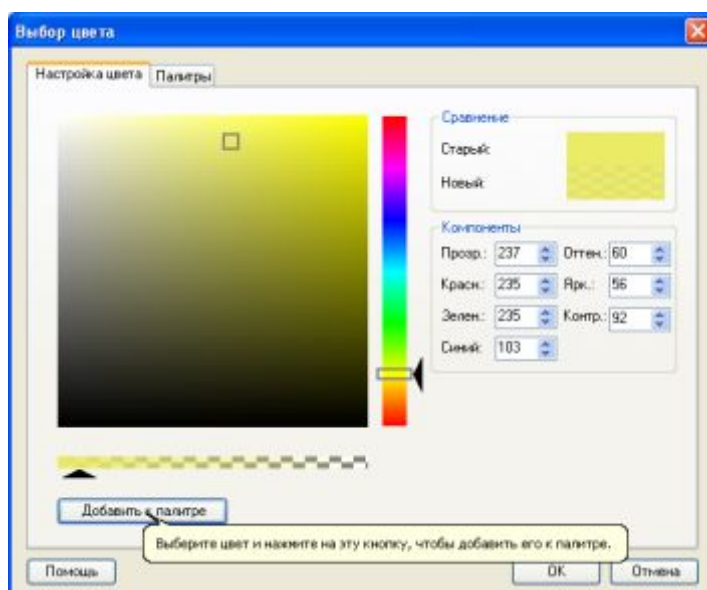
- 1 Щелкнуть по кнопке 
- 2 В появившемся окне **Имя новой палитры** задать имя создаваемой палитры
- 3 Для сохранения палитры на диск нажать на кнопку **ОК**.
Палитры хранятся в системном каталоге:
<диск, на который установлена ОС>:\Documents and Settings\<имя пользователя>\Local Settings\Application Data\KrugSoft\DataRate <версия>\Palettes

Палитра создается пустой и будет добавлена в список пользовательских палитр.

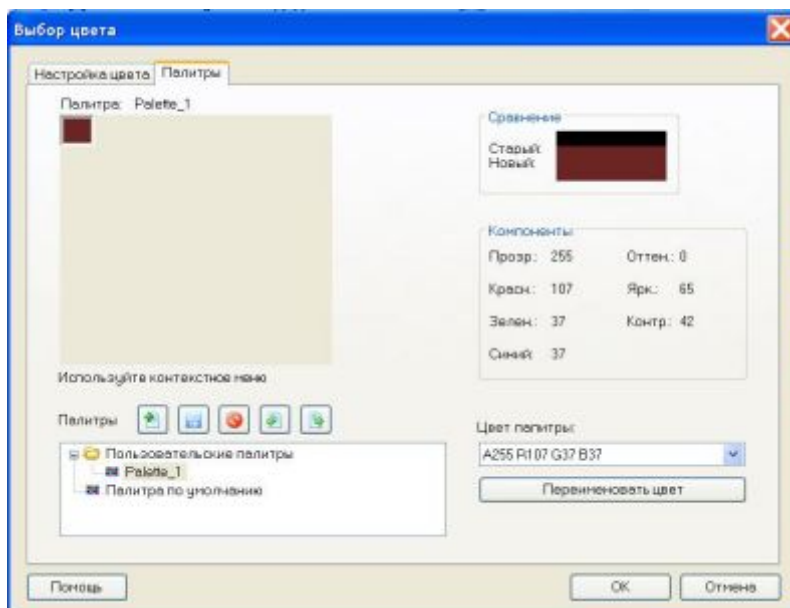
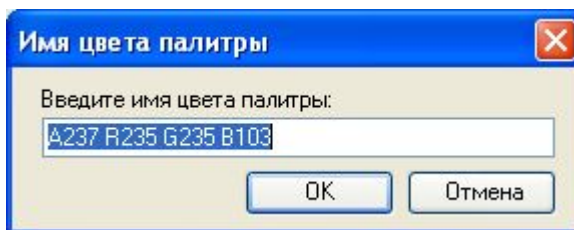
**Добавить цвет в пользовательскую палитру.**

Цвет можно добавить по кнопке **Добавить к палитре** во вкладке **Настройка цвета** или использовать контекстное меню палитры.

В первом случае перейдите на вкладку **Настройка цвета**, выберите необходимый цвет и нажмите на кнопку **Добавить к палитре**.



После нажатия на кнопку автоматически откроется вкладка **Палитры** и появится окно **Имя цвета палитры**, в котором отобразится имя цвета, состоящее из значений составляющих цвета. После ввода имени цвета и нажатия на кнопку **ОК** цвет добавится в палитру



ВНИМАНИЕ!!!

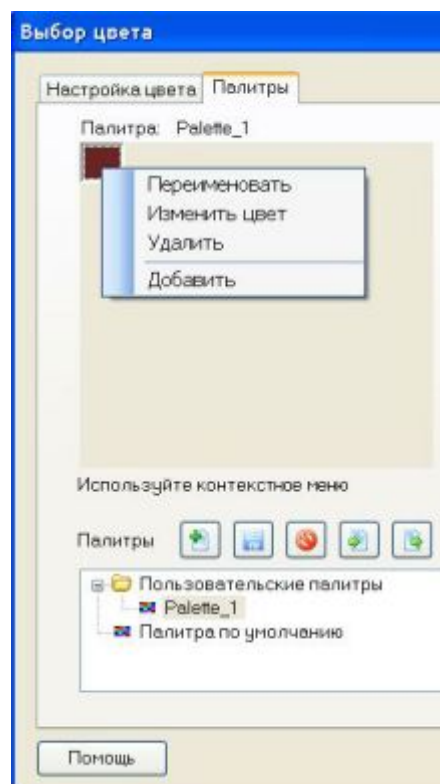
Если такой цвет уже есть в палитре, то будет выдано сообщение «Палитра уже содержит элемент с таким цветом» и цвет добавлен не будет.

Во втором случае для добавления цвета в палитру используется контекстное меню.

Если вызвать контекстное меню, щелкнув на пустом месте палитры, то появится только один пункт **Добавить**. При выборе этого пункта меню откроется вкладка **Настройка цвета** и станет активной кнопка **Добавить к палитре**.




Если вызвать контекстное меню, щелкнув по цвету палитры, то появится следующее контекстное меню:

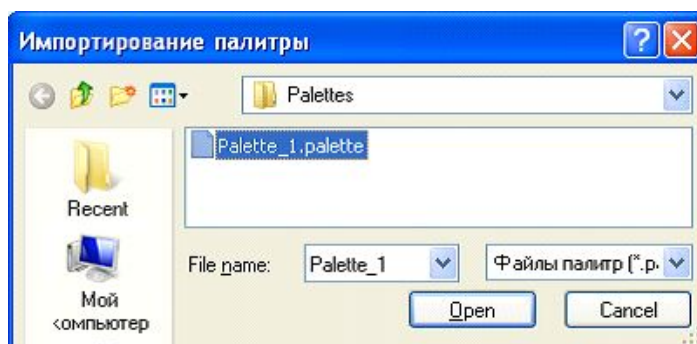
- **Переименовать** – будет выдан диалог ввода нового имени. Переименовать цвет можно, выбрав в выпадающем списке цвет и нажав на кнопку **Переименовать цвет**
- **Изменить цвет** – будет открыта вкладка **Настройка цвета**, в которой дополнительно доступна кнопка **Изменить цвет палитры**
- **Удалить** – удалит выбранный цвет из палитры
- **Добавить** – откроется вкладка **Настройка цвета** и станет активной кнопка **Добавить к палитре**.



Управление палитрами

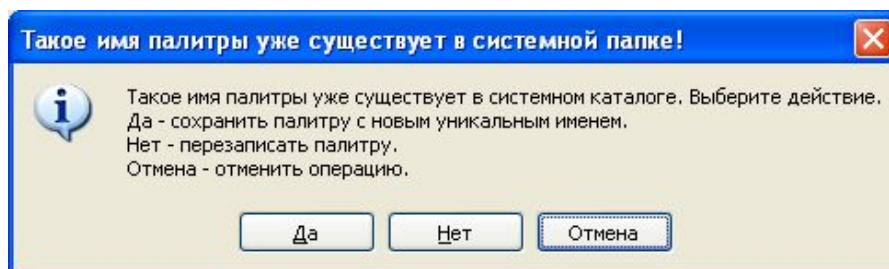
Управление палитрами осуществляется с помощью следующих кнопок:

-  – **Сохранение изменений** в пользовательской палитре
-  – **Удаление** пользовательской палитры из списка палитр. Файл палитры также будет удален
-  – **Импорт** пользовательской палитры.

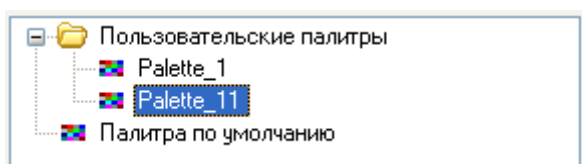



Палитра по умолчанию загружается всегда. Это не редактируемая палитра с набором основных цветов.

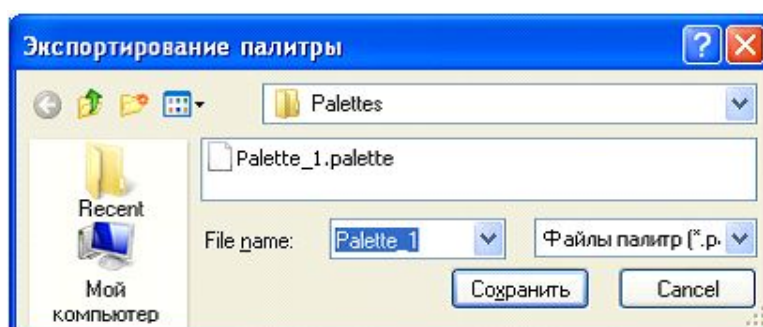
Палитры, созданные пользователем, загружаются из файлов с расширением **.palette**. При импорте файл с палитрой копируется в системную папку, в которой хранятся палитры. Если возникает конфликт, то отображается диалоговое окно для разрешения конфликта.



При выборе **Да** палитре будет присвоено новое уникальное имя



- **Экспорт** пользовательской палитры осуществляется по кнопке . Палитра будет сохранена в файл указанный пользователем.

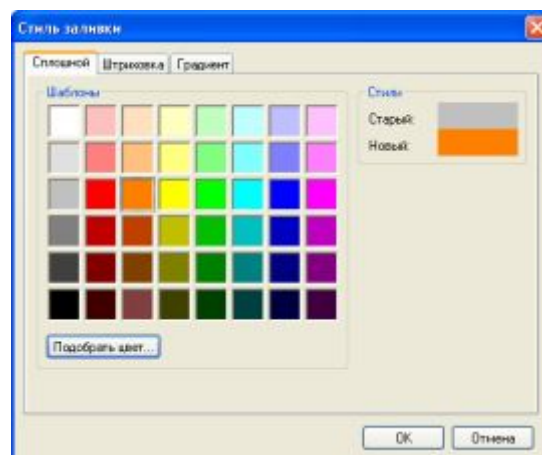


Заливка

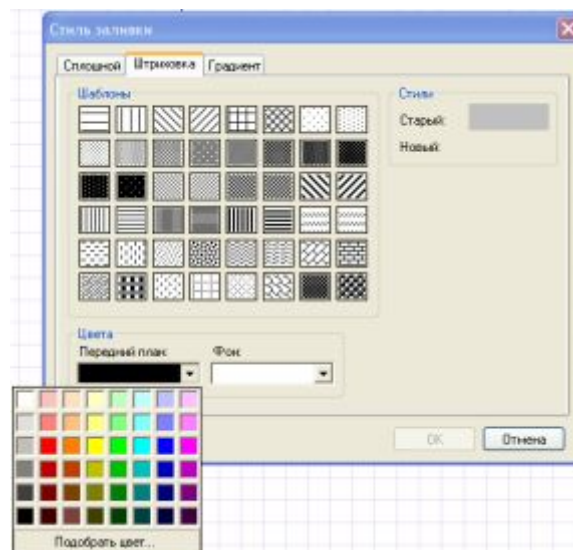
Для настройки свойства **Заливка** следует:

- 1 Щелкнуть по кнопке в строке **Заливка** окна Свойства
- 2 В появившемся окне **Стиль заливки** установить требуемые значения параметров заливки

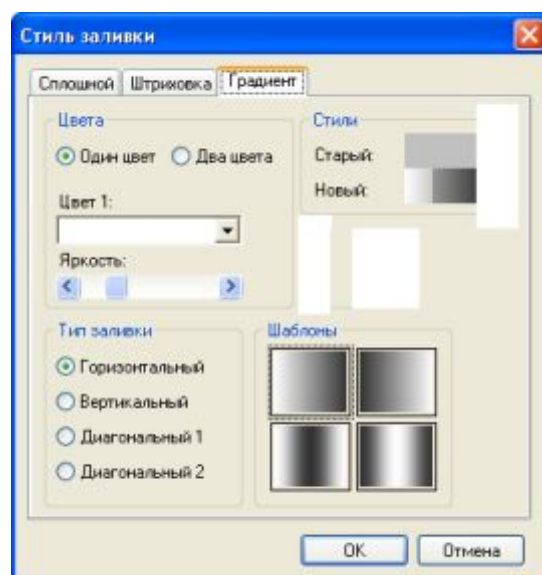
Закладка **Сплошной** позволяет выбрать заливку цветом. На панели **Шаблоны** можно выбрать predetermined цвет или, нажав на кнопку **Подобрать цвет**, произвести точную настройку цвета.



Закладка **Штриховка** позволяет выбрать **стиль** (панель **Шаблоны**) и **цвет** штриховки (панель **Цвета**).



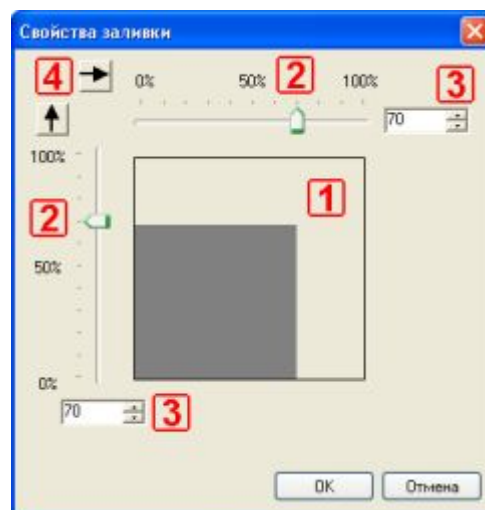
Закладка **Градиент** позволяет установить градиентную заливку.



Свойства заливки

Для настройки **Свойства заливки** следует:

- 1 Щелкнуть по кнопке в строке **Свойства заливки** окна **Свойства**
- 2 В появившемся окне **Свойства заливки** установить требуемые значения:
 - **Процент заливки** (по горизонтали и вертикали) устанавливают/отображают ползунки (2) и поля ввода (3)
 - **Направление заливки** (влево/вправо, вверх/вниз) устанавливают кнопки (4). Поле (1) отображает результат настройки.



3.1.3.2 Использование свойства графических примитивов в скриптах

В скриптах при обращении к свойствам графического примитива используются имена свойств на английском языке, эквивалентные русским именам.

Имя свойства	Эквивалентное наименование свойства по-английски	Тип
Имя объекта	Name	string
Прозрачность	Transparency*	float
Видимость/мигание	Visible	int
Цвет линии	LineColor	Color
Стиль линии	LineStyle	DashStyle
Ширина линии	LineWidth	float
Заливка	FillBrush	BrushX
Свойства заливки	FillDirectionInfo	FillDirectionInfo
Смещение сверху	Y	float
Смещение слева	X	float
Ширина	Width	float
Высота	Height	float
Угол поворота	Angle	float
Центр вращения X	RotationCenterX	float
Центр вращения Y	RotationCenterY	float

* *Transparency* принимает значения от 0 до 100.

Для обращения к свойствам в скриптах используется системный объект **Client**. Например, обращение к свойству **Высота** записывается так:

`Client.Height = 100.`

Client.Visible

Используется в скрипте для задания свойства графического примитива **Видимость/мигание** и может принимать следующие значения:

- невидим — 0
- видим в фазе 1 — 8
- видим в фазе 2 — 4
- видим в фазе 3 — 2
- видим в фазе 4 — 1
- быстрое мигание — 10
- мигание — 8
- всегда видим — 15.

Пример

```
Client.Visible = 15; // примитив всегда видим
Client.Visible = 8;  // примитив мигает
```

Client.LineColor

Используется для задания в скрипте цвета графического примитива, имеющего тип **Color**, с помощью predefined цвета или значений **RGB**.

Пример

```
// установка красного цвета, используя predefined цвет Red
Client.LineColor = Color.Red;
// установка красного цвета, используя значения 8-bit RGB (red, green, blue)
Client.LineColor = Color.FromArgb(255, 0, 0);
```

Client.FillBrush

Используется в скрипте для задания свойства Заливка графического примитива

Пример

```
// Назначение сплошной заливки красным цветом
Client.FillBrush = BrushX.CreateSolid(Color.Red);
// Назначение заливки «решеткой»
// Параметры используемого метода
// CreateHatch(HatchStyle hatchStyle, Color foreColor, Color backColor):
// hatchStyle — значение структуры System.Drawing.Drawing2D.HatchStyle
// foreColor — цвет решетки
// backColor — цвет фона
BrushX.CreateHatch(System.Drawing.Drawing2D.HatchStyle.DashedVertical,
Color.Black, Color.White);
```

Client.FillDirectionInfo

Используется в скрипте для задания **Свойства заливки** графического примитива.

Пример

```
// Создаем объект info типа FillDirectionInfo
FillDirectionInfo info = new FillDirectionInfo(50,50);

// Назначаем направление заливки по X слева направо
// Значения перечисления FillOrientationX:
// FillOrientationX.LeftToRight — заливка слева направо
// FillOrientationX.RightToLeft — заливка справа налево
info.FillOrientationX = FillOrientationX.LeftToRight;
```

```
//Назначаем направление заливки по Y снизу вверх
//Значения перечисления FillOrientationY:
//FillOrientationY.UpToDown - заливка сверху вниз
//FillOrientationY.DownToUp - заливка снизу вверх
info.FillOrientationY = FillOrientationY.DownToUp;
```


















```
//Присваиваем назначенные свойства заливки графическому примитиву
Client.FillDirectionInfo = info;
```





3.1.3.3 Действия над примитивами

Над примитивами можно выполнять следующие действия:

- Изменение размера и положения
- Группирование
- Копирование реакций и анимаций

На панели инструментов среды разработки доступны следующие действия:

Иконка	Описание
	Сгруппировать выделенные примитивы. В результате операции примитивы будут объединены в группу. Описано в разделе Группирование.
	Разгруппировать. Сгруппированные примитивы будут разгруппированы. Описано в разделе Группирование.
	Выделенные примитивы будут перенесены на передний план. Примитивы окажутся на самом переднем плане.
	Выделенные примитивы будут помещены на самый нижний уровень.
	Выделенные примитивы будут перенесены на один уровень вверх.
	Выделенные примитивы будут перенесены на один уровень вниз.
	Левые границы выделенных примитивов будут выровнены по левой границе активного выделенного элемента.
	Правые границы выделенных примитивов будут выровнены по правой границе активного выделенного элемента.
	Верхние границы выделенных примитивов будут выровнены по верхней границе активного выделенного элемента.
	Нижние границы выделенных примитивов будут выровнены по нижней границе активного выделенного элемента.
	Выровнять центры выделенных примитивов по центру по горизонтали активного выделенного примитива.
	Выровнять центры выделенных примитивов по центру по вертикали активного выделенного примитива.
	Выравнивает ширину всех примитивов по ширине активного выделенного примитива.
	Выравнивает высоту всех примитивов по высоте активного выделенного примитива.
	Выравнивает расстояние по горизонтали между выделенными примитивами.
	Выравнивает расстояние по вертикали между выделенными примитивами.
	Поворачивает выделенные примитивы на угол 90 градусов в лево.

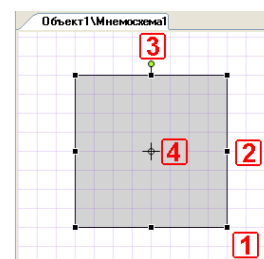
-  Поворачивает выделенные примитивы на угол 90 градусов в право.
-  Отражение выделенных примитивов слева направо.
-  Отражение выделенных примитивов сверху вниз.
-  Копирование реакций и анимаций.



Изменение размера и положения


Изменять размер и положение можно или при помощи мыши, или указывая точные значения в окне **Свойства**.

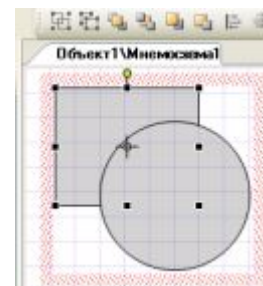
Каждый графический примитив имеет манипуляторы для управления размером и положением при помощи мыши:

- Угловые манипуляторы (1) позволяют одновременно изменять длину и ширину
- Боковые манипуляторы (2) позволяют изменять ширину или высоту
- Манипулятор поворота (3) позволяет менять угол поворота.
- Манипулятор вращения (4) позволяет менять центр вращения.




При наведении на манипуляторы курсор принимает вид .
При наведении мыши на манипулятор угла поворота курсор принимает вид .

Если курсор принимает вид , то примитив можно перемещать, или копировать (при нажатой кнопке **Ctrl**).



Группирование

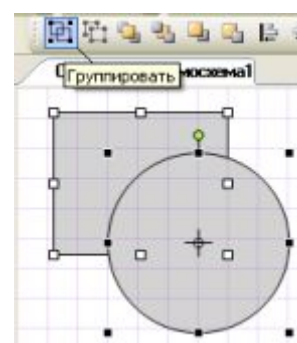
Для объединения графических примитивов в группу следует:

- Нажать на кнопку **Группировать** .
- Выделить нужные графические примитивы

Внутри **группы** возможно выделять отдельный **графический примитив**. Для этого щелкните по нему мышью.


ВНИМАНИЕ!!!

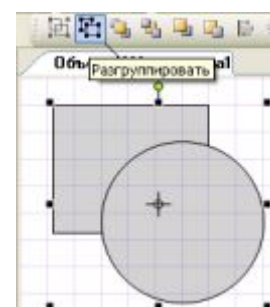
Координаты графических примитивов в группе отсчитываются относительно верхнего левого угла группы.



Разгруппирование

Для разгруппирования графических элементов следует:

- 1 Выделить группу
- 2 Нажать на кнопку **Разгруппировать** .



3.1.3.4 Эллипс

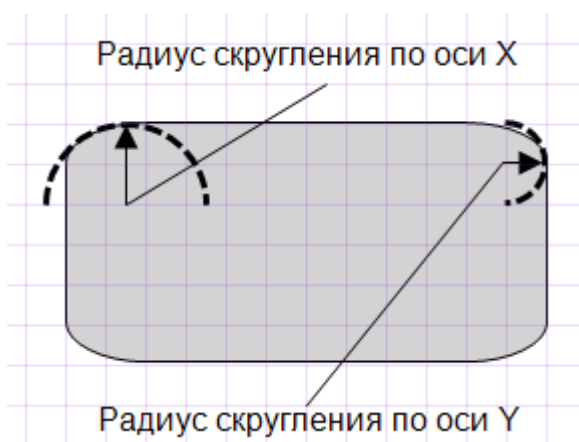
Эллипс – графический примитив, предназначенный для отображения эллипса.

3.1.3.5 Прямоугольник

Прямоугольник – графический примитив, предназначенный для отображения прямоугольника.

3.1.3.6 Скругленный прямоугольник

Скругленный прямоугольник – графический примитив, предназначенный для отображения прямоугольника со скругленными углами.

**Дополнительные свойства**

- **Радиус скругления по оси X** – определяет радиус скругления углов прямоугольника по оси X.
- **Радиус скругления по оси Y** – определяет радиус скругления углов прямоугольника по оси Y.

3.1.3.7 Прямая

Прямая – графический примитив, предназначенный для отображения прямой линии.

Дополнительные свойства

- **Стиль начала линии / Стиль конца линии** – определяет вид линии в начальной и конечной точке. Свойство может принимать одно из четырех значений: **Нет**, **Стрелка**, **Залитая стрелка**, **Круг**, **Залитый круг**.



- **Размер начала линии / Размер конца линии** – может принимать три значения: **Нормальный**, **Большой**, **Огромный**.

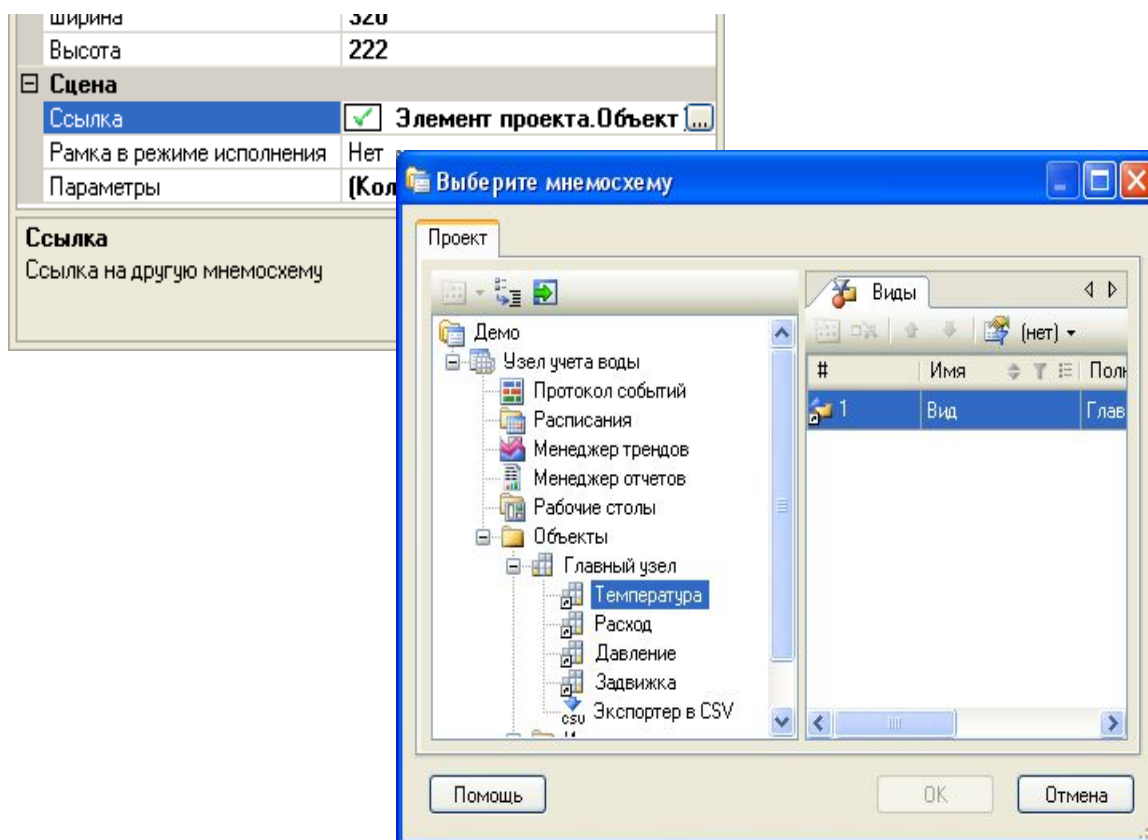
3.1.3.8 Сцена

Сцена – графический примитив, предназначенный для отображения на мнемосхеме (в рамках размера сцены) другой мнемосхемы.

Сцену можно создать, как указанием ссылки на мнемосхему, так и путем перетаскивания мнемосхемы из окна редактирования проекта на другую мнемосхему.

Дополнительные свойства

- **Ссылка** – устанавливает мнемосхему, которая будет показана на сцене. Для настройки свойства **Ссылка** следует:
 - 1 Выделить графический примитив **Сцена**
 - 2 Нажать на кнопку в строке **Ссылка** в окне **Свойства**
 - 3 В открывшемся окне **Выберите мнемосхему** выбрать мнемосхему, которая будет показываться на сцене



Сцену можно создать, как указанием ссылки на мнемосхему, так и путем перетаскивания мнемосхемы из окна редактирования проекта на другую мнемосхему.

ВНИМАНИЕ!!!

Сцена не копирует свойства мнемосхемы! Сцена клонирует примитивы лежащие на мнемосхеме

ВНИМАНИЕ!!!

При изменении мнемосхемы (количества источников данных, параметров или примитивов), которую показывает сцена, необходимо принудительно обновить сцену (команда "Обновить" контекстного меню сцены)! В противном случае сделанные в мнемосхеме изменения в сцене отображаться не будут!

- **Рамка в режиме исполнения** – устанавливает, будет ли показываться рамка по границе сцены в режиме исполнения проекта
- **Параметры** – используется для редактирования параметров мнемосхемы. После задания ссылки на мнемосхему, имеющую параметры, автоматически открывается окно **Задайте параметры** для редактирования параметров мнемосхемы.

Параметр	Тип	Привязка
Device type id	Константа	10
Device name	По умолчанию	
Device id	Аргумент	Сцена1-Device id value
Version	Константа	5
Description	По умолчанию	

Описание
Описание прибора

Помощь OK Отмена

В этом окне перечислен список параметров вида и дана возможность задать тип привязки параметра и значение привязки. Тип параметра и привязки может иметь следующие значения:

- **По умолчанию** – для параметра будет использовано значение по умолчанию;
- **Константа** – строковая или числовая константа;
- **Аргумент** – имя аргумента из списка аргументов мнемосхемы;
- **Поле источника данных** – имя колонки источника данных.

3.1.3.9 Рисунок

Рисунок – графический примитив, предназначенный для размещения на мнемосхеме растровых изображений формата **bitmap**, **metafile**, **SVG**, **SVGZ**, **icon**, **JPEG**, **GIF** или **PNG**.

Дополнительные свойства


- **Файл** - указывает файл с изображением (*.bmp, *.wmf, *.emf, *.icon, *.jpg, *.gif, *.png, *.svg, *.svgz)
- **Способ задания пути** – принимает два значения: **Относительный путь** (путь до файла с изображением определяется относительно файла проекта), **Абсолютный путь** (путь до файла с изображением определяется, как указано в свойстве **Файл**)
- **Режим сохранения** – принимает два значения: **Ссылка на файл** (изображение будет храниться в отдельном файле), **С мнемосхемой** (изображение будет храниться в мнемосхеме в файле проекта).
- **Прозрачность** – указывает процент прозрачности изображения.
- **Прозрачный цвет** – указывает цвет, который будет для изображения прозрачным.
- **Использовать растяжение** – указывает, будет ли изображение растянуто по границам рисунка или показано в оригинальном виде.

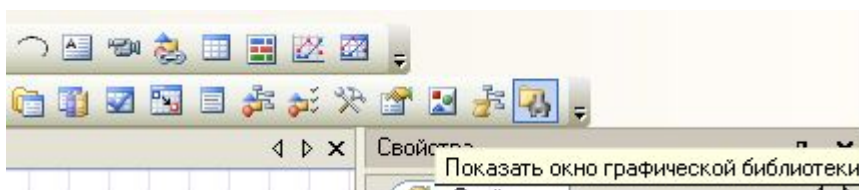
Использование графической библиотеки

Библиотека изображений **DataRate** содержит рисунки «технологических» объектов. Это, например, изображения резервуаров и колонн, ёмкостей, исполнительных механизмов, насосов и печей.

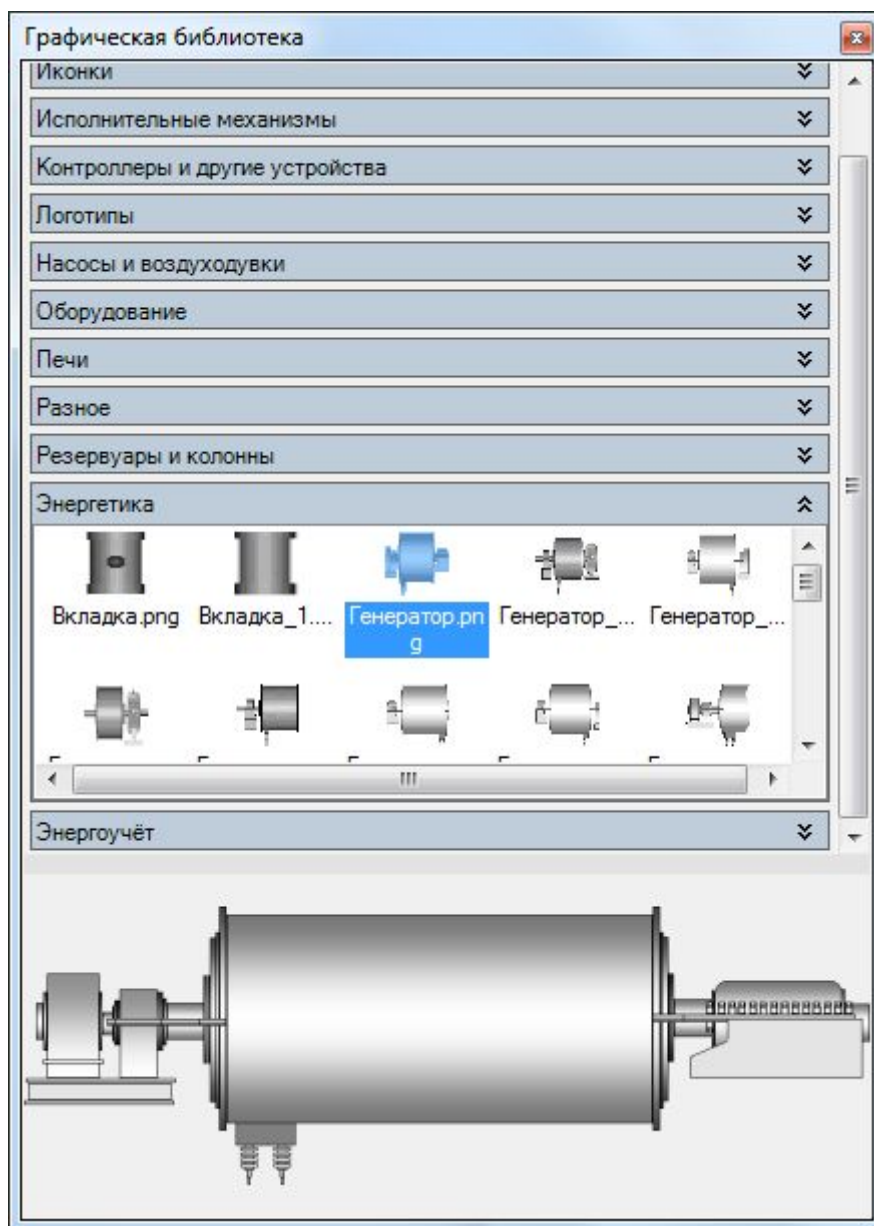
При использовании этих рисунков на мнемосхеме автоматически формируется графический примитив **Рисунок** со встроенным изображением, выбранном из библиотеки.

Для размещения библиотечных изображений на мнемосхеме проекта следует:

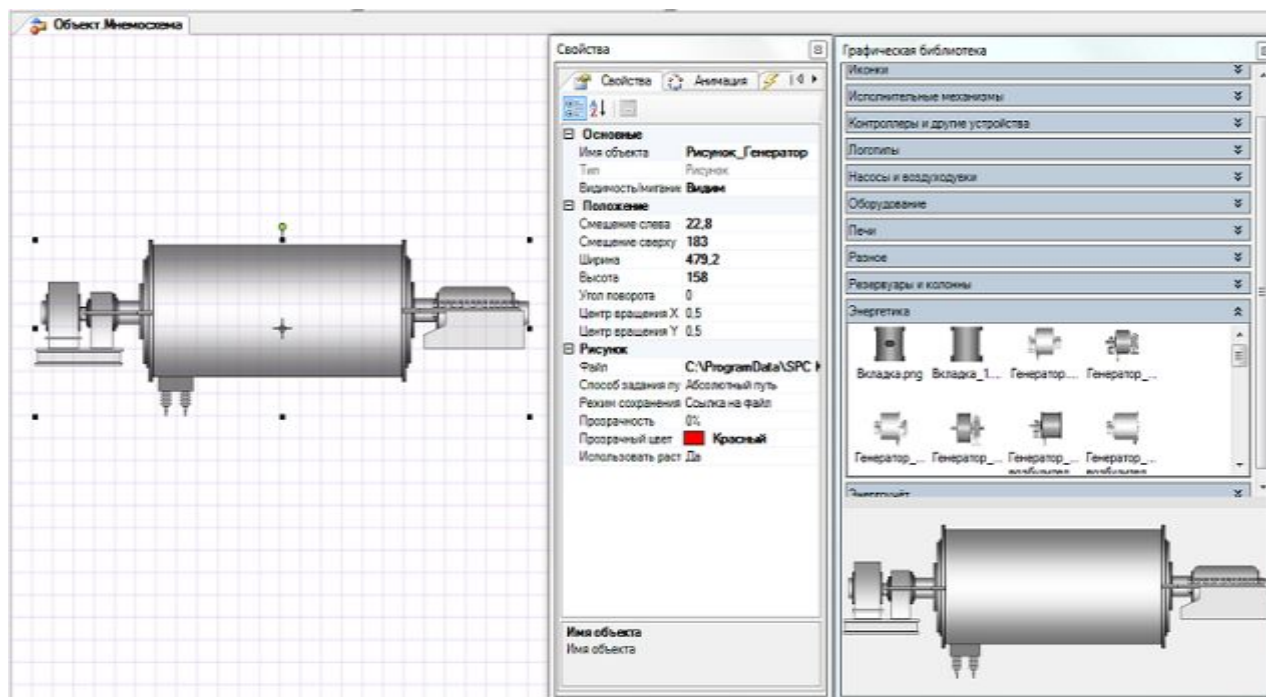
- 1 Открыть окно графической библиотеки (иконка  на панели инструментов)



- 2 В появившемся окне открыть закладку библиотеки по требуемой тематике и выбрать необходимый рисунок



- 3 Щелкнуть мышью по изображению в области закладки или в области просмотра и, не отпуская кнопку мыши, переместить изображение на мнемосхему
- 4 В окне свойств задать необходимые значения свойств вставленного рисунка (аналогично графическому примитиву **Рисунок**)



3.1.3.10 Текст

Текст – графический примитив, предназначенный для отображения **многострочного текста**.

Дополнительные свойства

- **Текст** – устанавливает текст, который будет отображаться
- **Шрифт** – устанавливает шрифт, которым будет отображаться текст
- **Цвет текста** – цвет выводимого текста.

3.1.3.11 Надпись

Надпись – графический примитив, предназначенный для отображения символов **на одной строке**.

Для выделения или перетаскивания примитива **Надпись** курсор мыши должен находиться над любым из символов в тексте. Щелчок мыши по области, не принадлежащей символам или манипуляторам, приведет к снятию выделения.

Дополнительные свойства

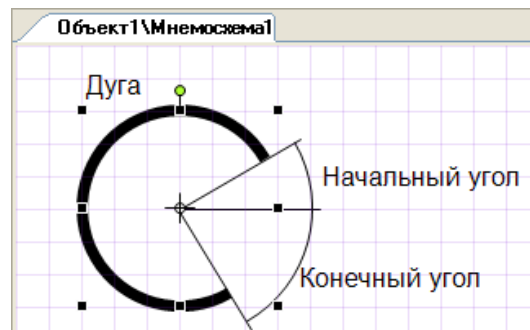
- **Шрифт** – устанавливает шрифт, которым будет отображаться текст
- **Текст** – устанавливает текст, который будет отображаться.

3.1.3.12 Дуга

Дуга – графический примитив, предназначенный для отображения дуги.

Дополнительные свойства

- **Начальный угол** – определяет начало дуги. Угол между горизонтальной линией проходящей через центр дуги и линией проведенной от центра дуги к началу дуги
- **Конечный угол** – определяет конец дуги. Угол между горизонтальной линией проходящей через центр дуги и линией проведенной от центра дуги к концу дуги.

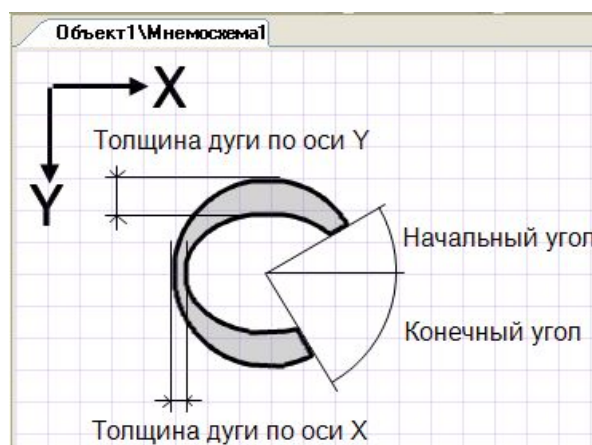


3.1.3.13 Большая дуга

Большая дуга – графический примитив, предназначенный для отображения дуги с расширенными свойствами.

Дополнительные свойства

- **Толщина дуги по оси X**
- **Толщина дуги по оси Y**
- **Начальный угол** – определяет начало дуги. Угол между горизонтальной линией проходящей через центр дуги и линией проведенной от центра дуги к началу дуги
- **Конечный угол** – определяет конец дуги. Угол между горизонтальной линией проходящей через центр дуги и линией проведенной от центра дуги к концу дуги.



3.1.3.14 Сектор

Сектор – графический примитив, предназначенный для отображения сектора.

Свойства положения

- **Начальный угол** – определяет начальную сторону сектора. Угол между горизонтальной линией проходящей через центр сектора и начальной стороной сектора
- **Конечный угол** – определяет конечную сторону сектора. Угол между горизонтальной линией проходящей через центр сектора и конечной стороной сектора.



3.1.3.15 Сегмент

Сегмент – графический примитив, предназначенный для отображения сегмента.

Свойства положения

- **Начальный угол** – определяет начало хорды сегмента. Угол между горизонтальной линией проходящей через центр сегмента и линией проведенной от центра сегмента к началу хорды
- **Конечный угол** – определяет конец сегмента. Угол между горизонтальной линией проходящей через центр сегмента и линией проведенной от центра сегмента к концу хорды.

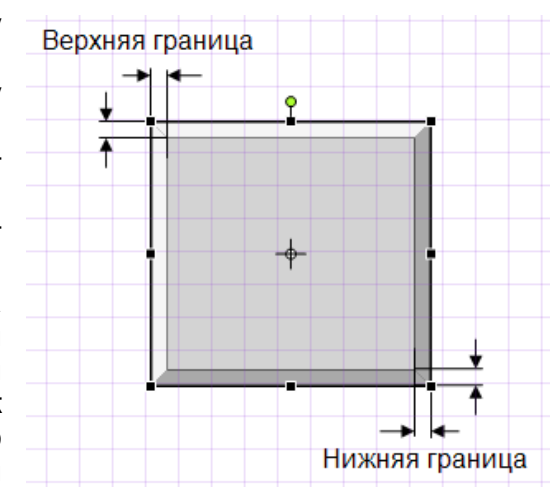


3.1.3.16 Объемный прямоугольник

Объемный прямоугольник – графический примитив, предназначенный для отображения прямоугольника с расширенными свойствами границ.

Дополнительные свойства

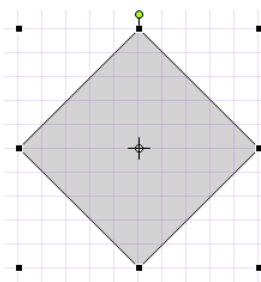
- **Верхняя граница** – определяет толщину верхней границы
- **Нижняя граница** – определяет толщину нижней границы
- **Заливка верхней границы** – определяет способ заливки верхней границы
- **Заливка нижней границы** – определяет способ заливки нижней границы
- **Инверсия границ** – если выбрана инверсия границ то свойства верхней границы будут применены к нижней границе, а свойства нижней границы – к верхней. Используя это свойство, можно получить изображение нажатой/отжатой кнопки.



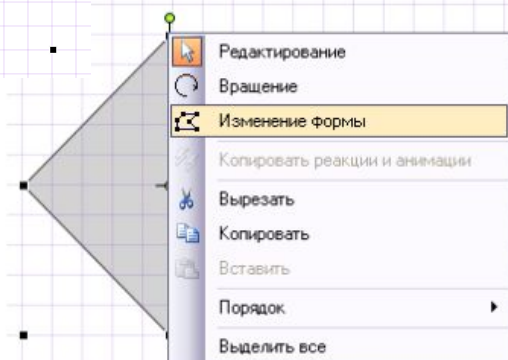
3.1.3.17 Вспомогательная линия

Вспомогательная линия – используется для настройки реакции Ползунок.

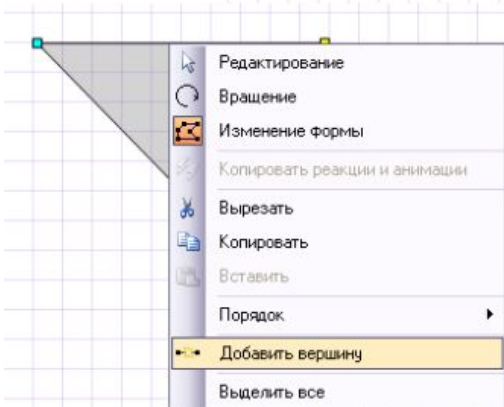
3.1.3.18 Многоугольник



Многоугольник – графический примитив, предназначенный для отображения замкнутой фигуры с количеством углов большим или равным трем.

**Изменение формы**

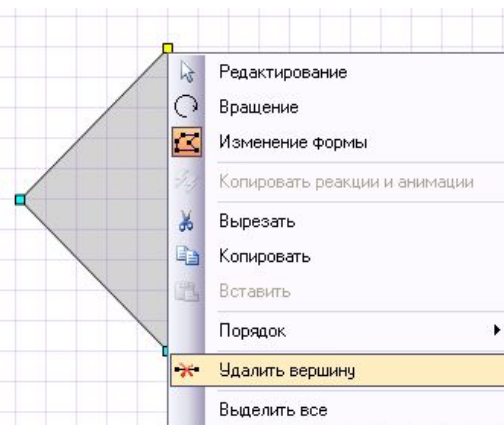
Для редактирования формы многоугольника в контекстном меню многоугольника выберите режим **Изменение формы**.



Изменение положения вершин. Изменять положения вершин можно при помощи манипуляторов.

Добавление вершин. Для добавления вершины следует:

- 1 Щелкнуть правой кнопкой мыши на стороне многоугольника, где предполагается создать вершину
- 2 В появившемся контекстном меню выполнить команду **Добавить вершину**.

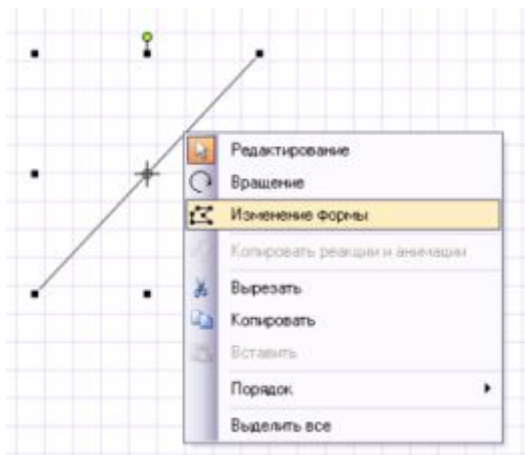


Удаление вершин. Для удаления вершины в контекстном меню манипулятора вершины выберите команду **Удалить вершину**.

Для выхода из режима **Изменение формы** в контекстном меню многоугольника выберите команду **Редактирование**.

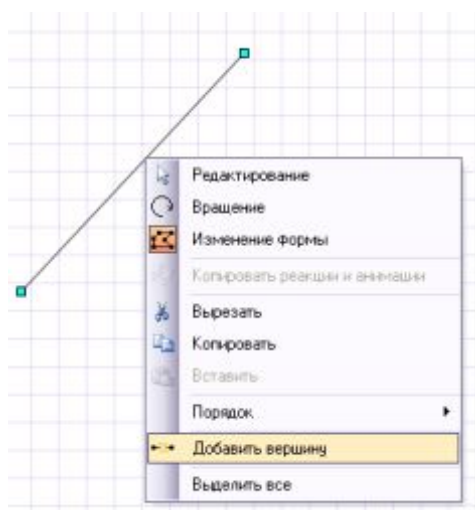
3.1.3.19 Ломаная

Ломаная – графический примитив, предназначенный для отображения ломаной линии.

**Изменение формы**

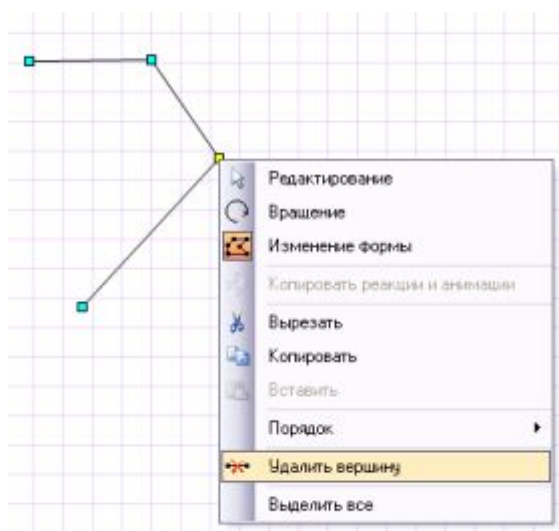
Для редактирования формы и изменения количества вершин в контекстном меню ломаной выберите режим **Изменение формы**.

Изменение положения вершин. Изменять положения вершин можно при помощи манипуляторов .



Добавление вершин. Для добавления вершины следует:

- 1 Щелкнуть правой кнопкой мыши на стороне ломаной, где предполагается создать вершину
- 2 В появившемся контекстном меню выполнить команду **Добавить вершину**.

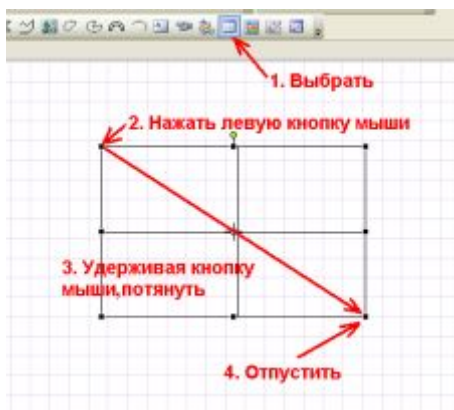


Удаление вершин. Для удаления вершины в контекстном меню манипулятора вершины выберите команду **Удалить вершину**.


Для выхода из режима **Изменение формы** в контекстном меню **ломаной** выберите команду **Редактирование**.

3.1.3.20 Таблица

Таблица – графический примитив, предназначенный для отображения таблицы, в ячейках которой могут располагаться статические данные.

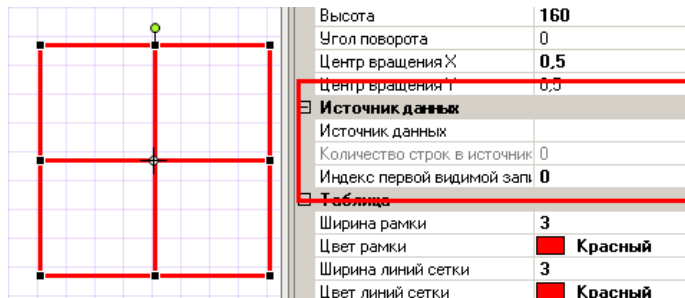


Для размещения таблицы на мнемосхеме следует:

- 1 Нажать на панели инструментов кнопку  (1)
- 2 Щелкнуть левой клавишей мыши на мнемосхеме (2) и, не отпуская ее (3), установить размеры таблицы (4)

Свойства источника данных

- **Источник данных** – это имя аргумента, связанного с источником данных вида. У примитива Таблица может быть только один источник данных.
- **Количество строк в источнике** – это количество строк в источнике данных в режиме исполнения (доступно только для чтения).
- **Индекс первой видимой записи** – индекс записи в источнике данных, начиная с которого отображаются данные в таблице. Это свойство позволяет организовать "листание" данных с использованием механизма постраничного отображения информации, когда количество записей в источнике данных превышает заданное количество строк в таблице.

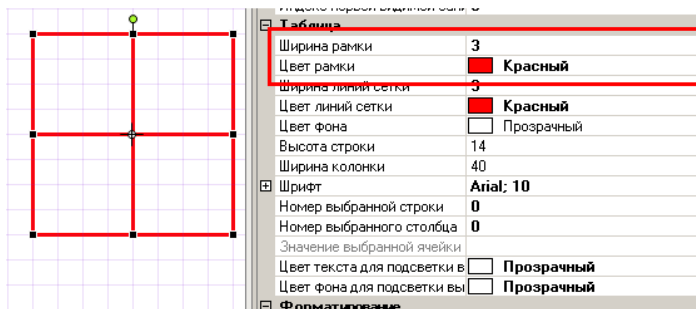


Для отображения данных из источника данных вида необходимо произвести следующую настройку:

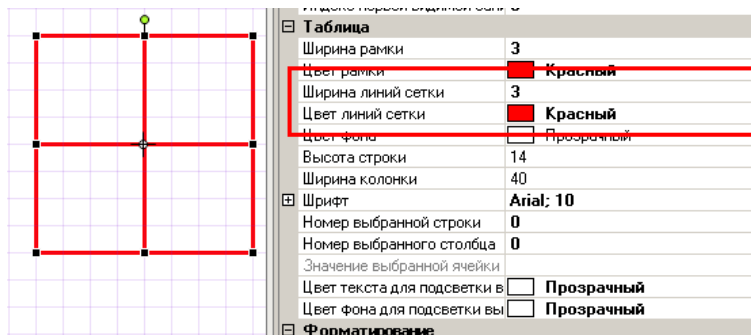
- 1 Создать аргумент для примитива
- 2 В окне **Выбор привязки** на вкладке **Источники данных** задать в качестве привязки аргумента источник данных
- 3 В свойстве **Источник данных таблицы** выбрать аргумент, привязанный к источнику данных.

Свойства таблицы

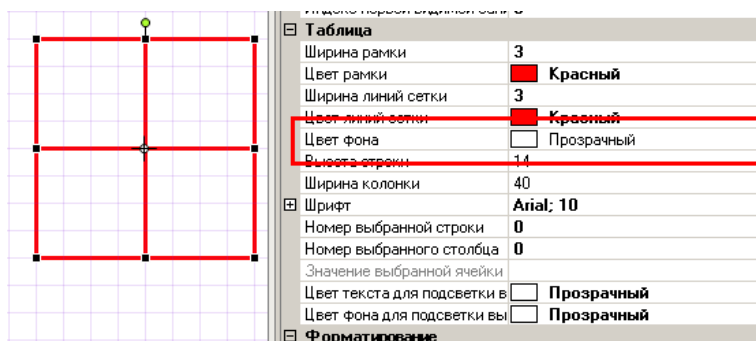
- Внешняя рамка таблицы:
 - **Ширина рамки** – ширина рамки вокруг таблицы
 - **Цвет рамки** – цвет рамки вокруг таблицы



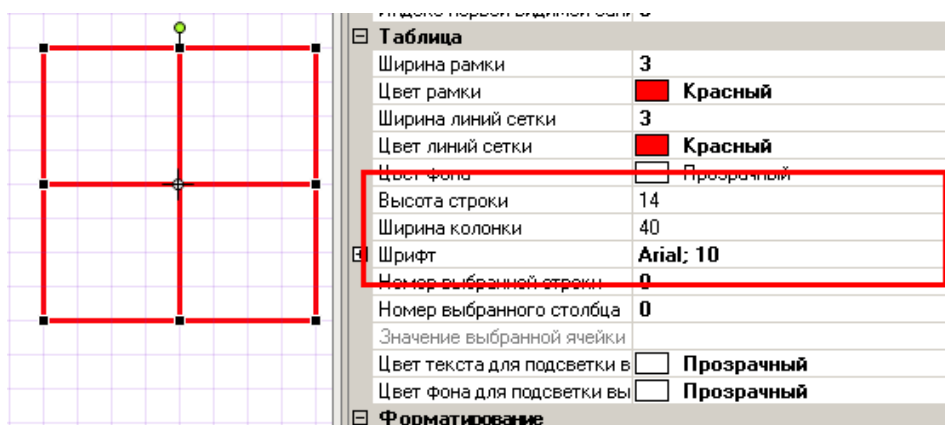
- Сетка внутри таблицы:
 - **рина линий сетки** – ширина линий сетки внутри таблицы
 - **Цвет линий сетки** – цвет линий сетки внутри таблицы.



- **Цвет фона.** Используется в качестве цвета фона для всех элементов таблицы.

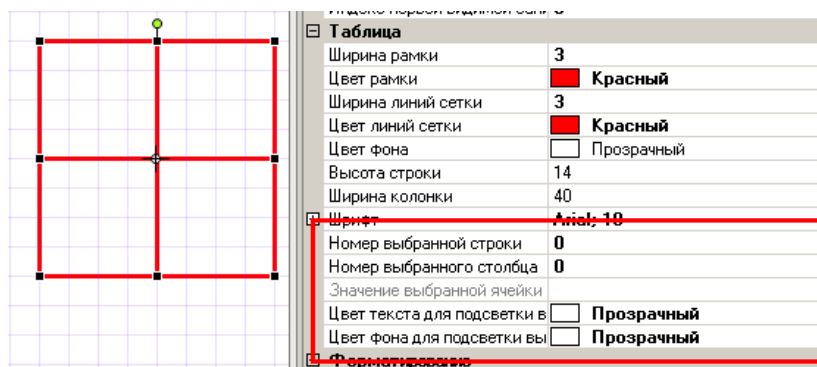


- **Высота строки** – высота строки таблицы по умолчанию. Используется для задания высоты строки вновь добавляемых строк, заголовков и подписей
- **Ширина колонки** – ширина колонки таблицы по умолчанию. Используется для задания ширины колонки вновь добавляемых колонок и заголовков
- **Шрифт** – шрифт таблицы. Используется в качестве шрифта по умолчанию для всех элементов таблицы.



- **Номер выбранной строки** – это номер выбранной в таблице строки. Первая строка имеет номер 0. По умолчанию выбрана первая строка.
- **Номер выбранного столбца** – это номер выбранного в таблице столбца. Первый столбец имеет номер 0.

- **Значение выбранной ячейки** – это значение, находящееся в выбранной ячейке таблицы.
- **Цвет текста и Цвет фона для подсветки выбранной строки** позволяют задать цвет и фон для подсветки строки, выбранной в таблице. Выбор строки осуществляется щелчком клавиши мыши в строке.



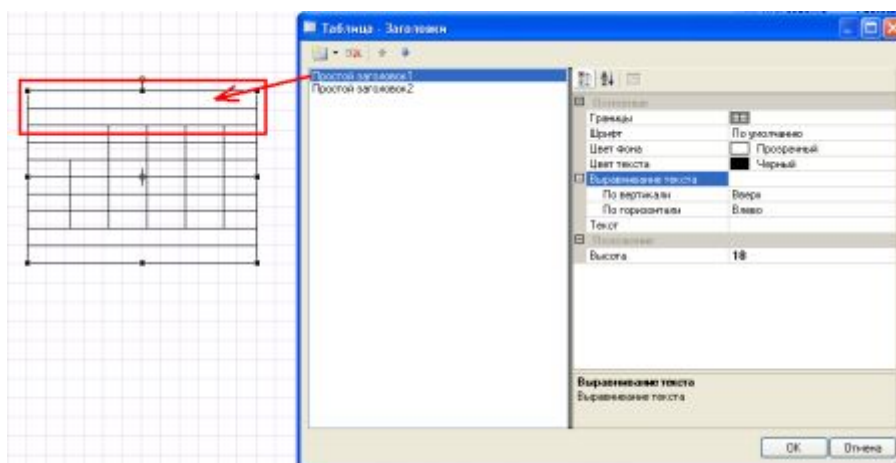
Свойства форматирования

Таблица состоит из следующих элементов:

- **Заголовки таблицы**
- **Колонки**
- **Заголовки колонок**
- **Строки**
- **Заголовки строк**
- **Подписи.**

Свойства этих элементов используются для форматирования таблицы.

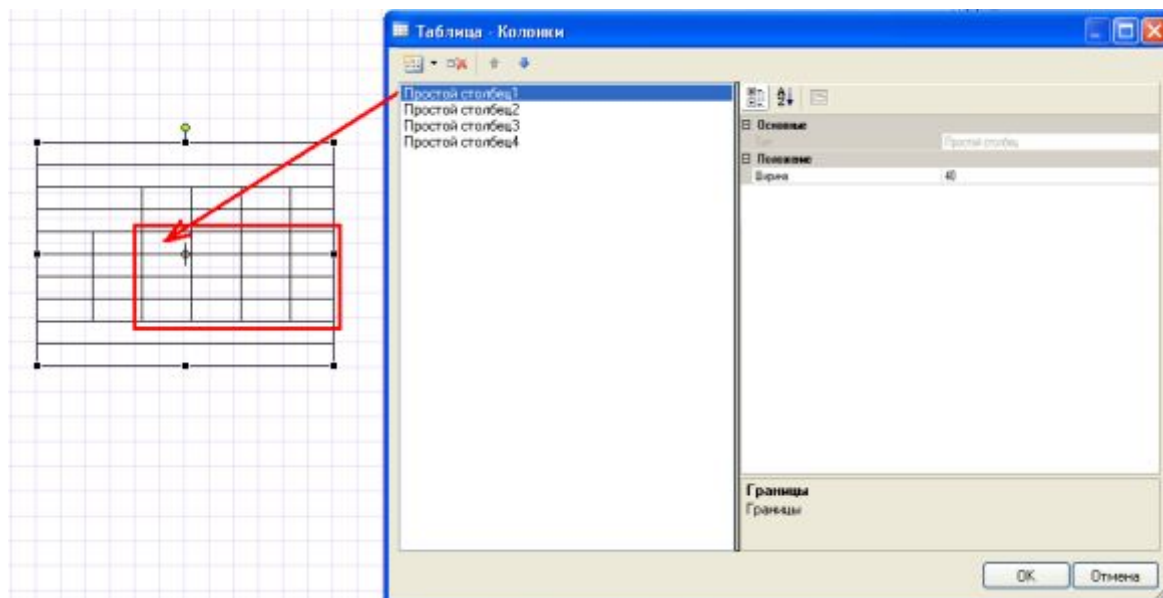
Заголовки – список заголовков таблицы.



Свойства простого заголовка:

- **Тип** – тип элемента таблицы
- **Цвет фона** – цвет фона заголовка. Если не задан, то используется цвет фона таблицы
- **Шрифт** – шрифт текста заголовка. Если не задан, то используется шрифт таблицы
- **Цвет текста** – цвет текста заголовка
- **Текст** – текст заголовка
- **Выравнивание текста** – выравнивание текста заголовка

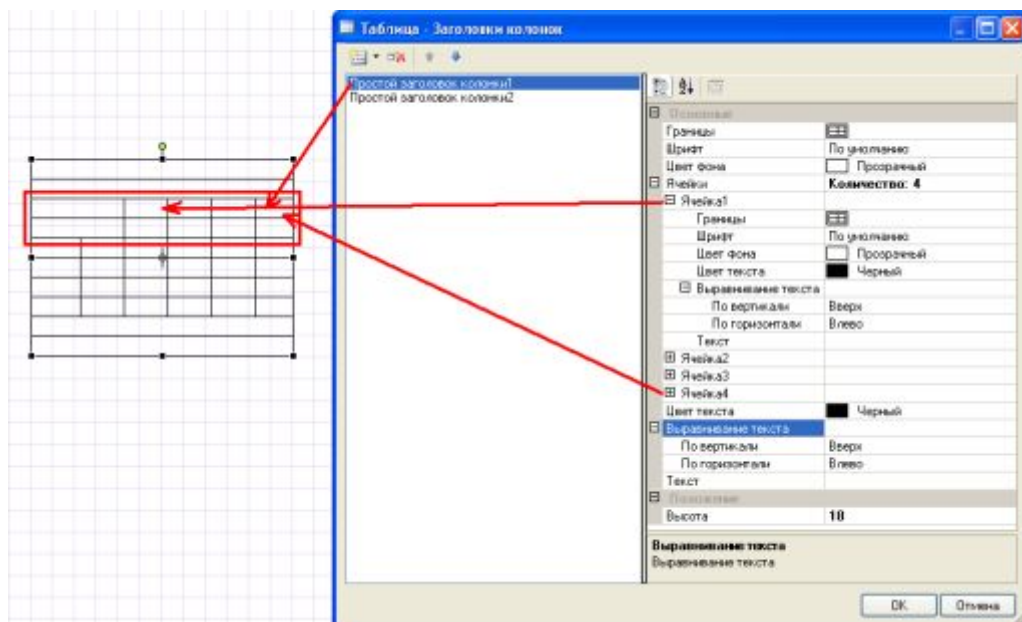
- **Границы** – отображение границ вокруг заголовка
 - **Высота** - высота заголовка.
- Колонки** – список колонок (столбцов) таблицы.



Свойства простого столбца:

- **Тип** – тип элемента таблицы
- **Ширина** – ширина столбца.

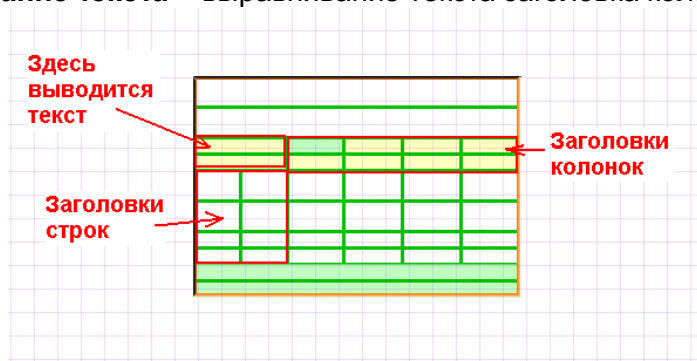
Заголовки колонок – список заголовков колонок таблицы.



Свойства простого заголовка колонки:

- **Тип** – тип элемента таблицы
- **Цвет фона** – цвет фона заголовка. Если не задан, то используется цвет фона таблицы
- **Шрифт** – шрифт текста заголовка

- **Цвет текста** – цвет текста заголовка
- **Текст** – текст заголовка. Этот текст выводится над заголовками строк. Если в таблице нет заголовков строк, то этот текст выводится, не будет
- **Выравнивание текста** – выравнивание текста заголовка колонки

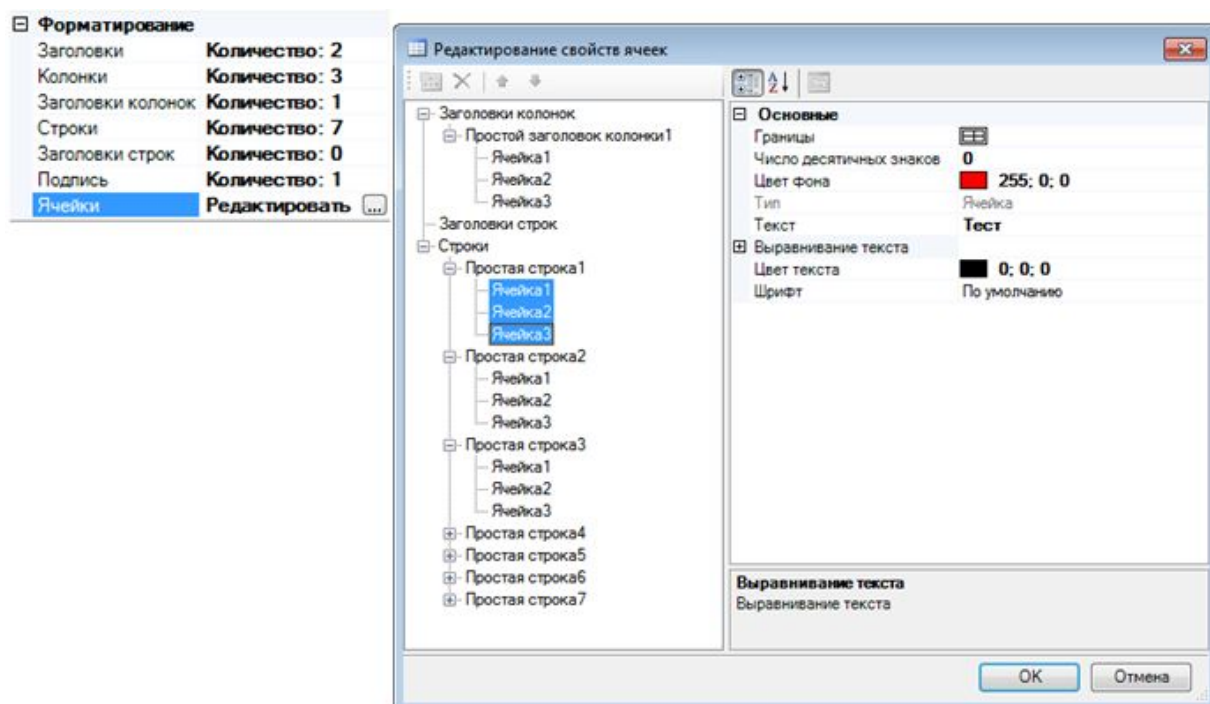


- **Границы** – отображение границ вокруг заголовка колонки
- **Ячейки** – список ячеек заголовка. Количество ячеек всегда равно количеству столбцов. Это свойство предоставляет доступ к свойствам ячеек

Свойства ячейки:

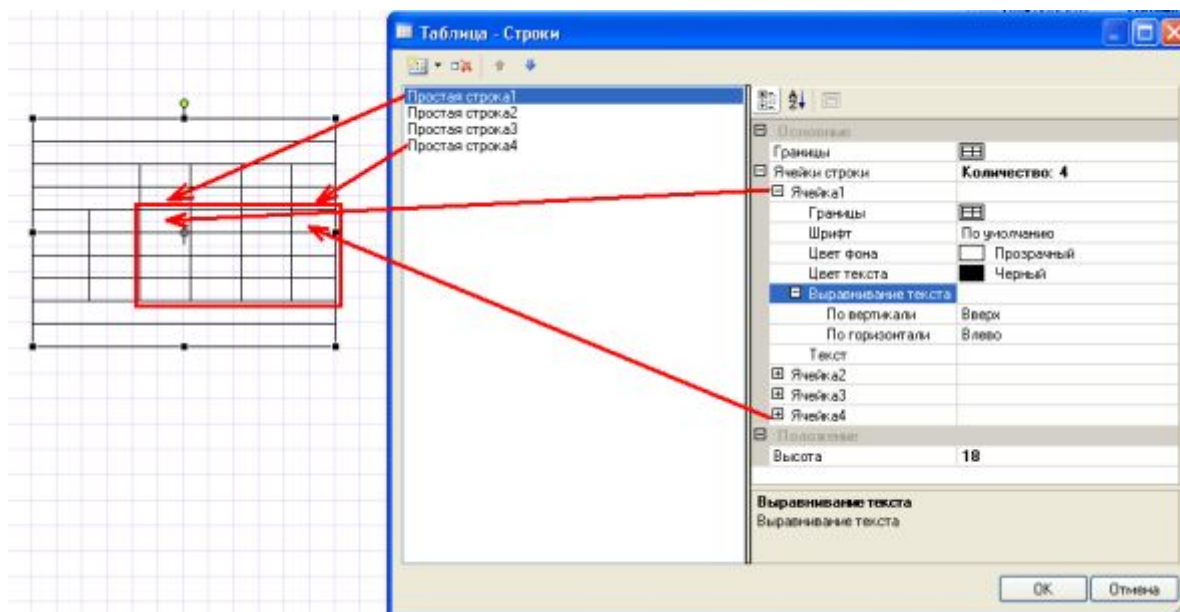
- **Тип** – тип элемента таблицы
- **Цвет фона** – цвет фона ячейки
- **Шрифт** – шрифт текста ячейки
- **Цвет текста** – цвет текста ячейки
- **Текст** – текст ячейки
- **Выравнивание текста** – выравнивание текста ячейки
- **Границы** – отображение границ вокруг текста ячейки.

Для редактирования свойства нескольких ячеек используется **Редактор свойств ячеек**. Для вызова редактора следует выделить свойство **Ячейки** и нажать на кнопку в строке свойств.



- **Высота** – высота заголовка колонки.

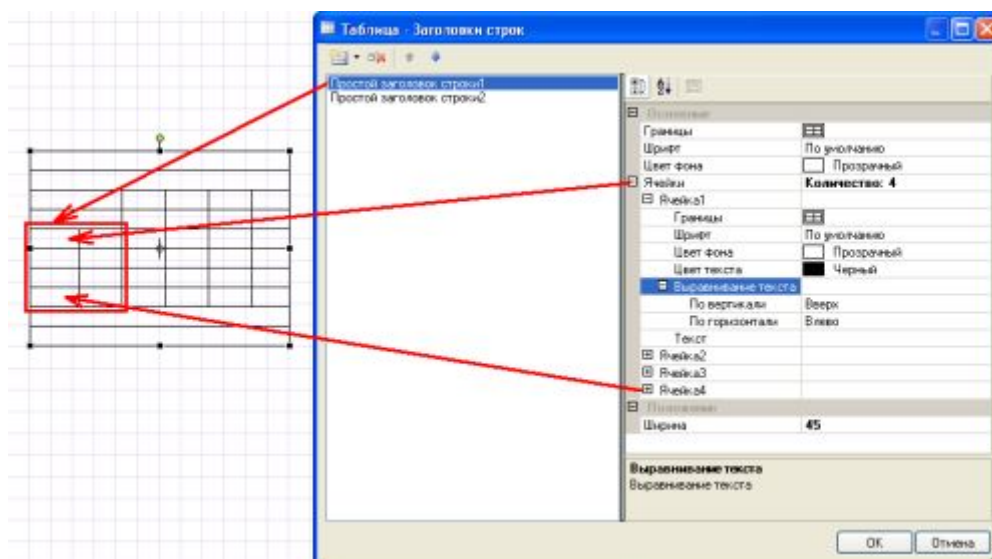
Строки – список строк таблицы.



Свойства простой строки:

- **Тип** – тип элемента таблицы
- **Границы** – отображение границ вокруг строки
- **Ячейки** – список ячеек строки. Количество ячеек всегда равно количеству столбцов
- **Высота** – высота строки.

Заголовки строк – список заголовков строк.

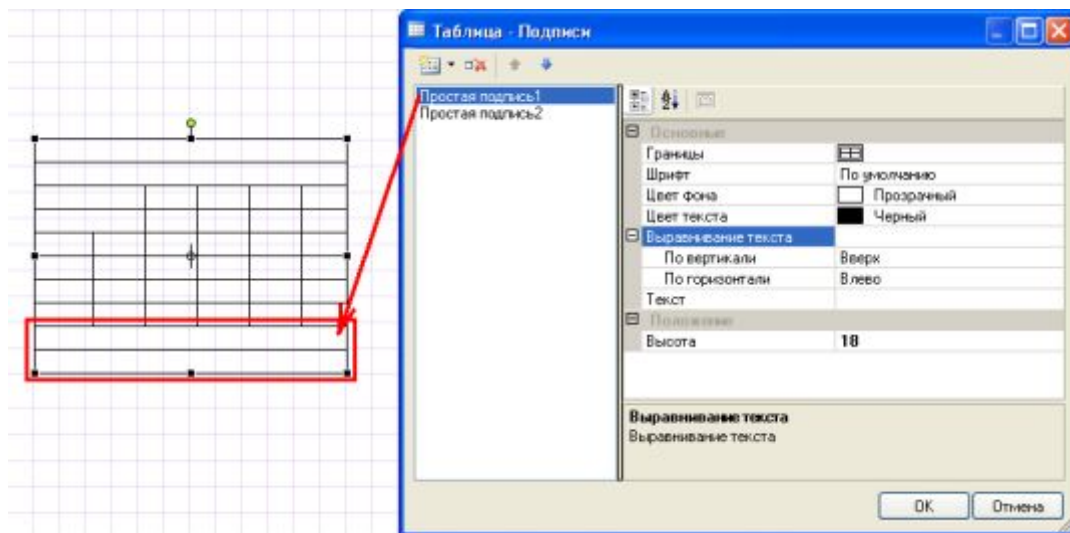


Свойства простого заголовка строки:

- **Границы** – отображение границ вокруг текста заголовка строки
- **Шрифт** – шрифт текста заголовка строки
- **Цвет фона** – цвет фона заголовка строки

- **Ячейки** – список ячеек строки. Количество ячеек всегда равно количеству строк
- **Ширина** – ширина заголовка строки.

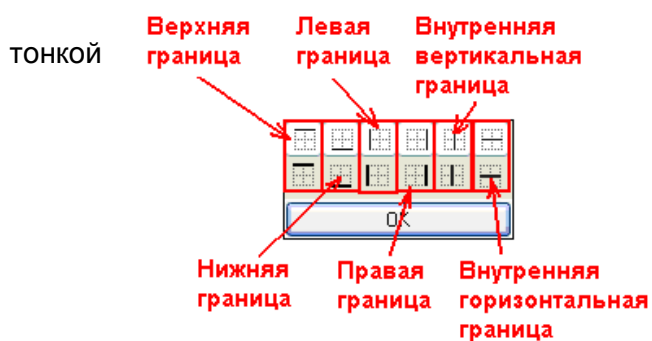
Подписи – список подписей таблицы.



Свойства простой подписи:

- **Тип** – тип элемента таблицы
- **Цвет фона** – цвет фона подписи
- **Шрифт** – шрифт текста подписи
- **Цвет текста** – цвет текста подписи
- **Текст** – текст подписи
- **Выравнивание текста** – выравнивание текста подписи
- **Границы** – отображение границ вокруг подписи
- **Высота** – высота подписи.

Редактирование границ осуществляется с помощью элемента управления, приведенного на рисунке ниже, и позволяет настраивать отображение границ как вокруг элемента таблицы, так и внутри него.



Граница может отображаться линией, толстой линией или не отображаться.

3.1.3.21 Элемент управления

Элемент управления – графический примитив, предназначенный для отображения элементов управления (**control-ов**) на мнемосхеме. Примитив используется для расширения функциональности мнемосхем.

Элементы управления могут быть разработаны и поставлены, как ООО"ЭнергоКруг", так и сторонними производителями.

Доступны два типа элементов управления, зарегистрированных на машине пользователя:


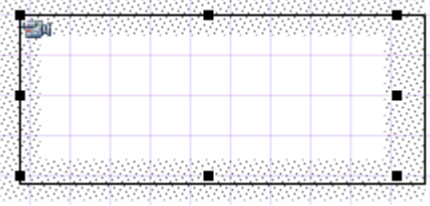
- **ActiveX (COM компоненты)**
- **Элементы управления, поставляемые с библиотекой классов .NET Framework.**

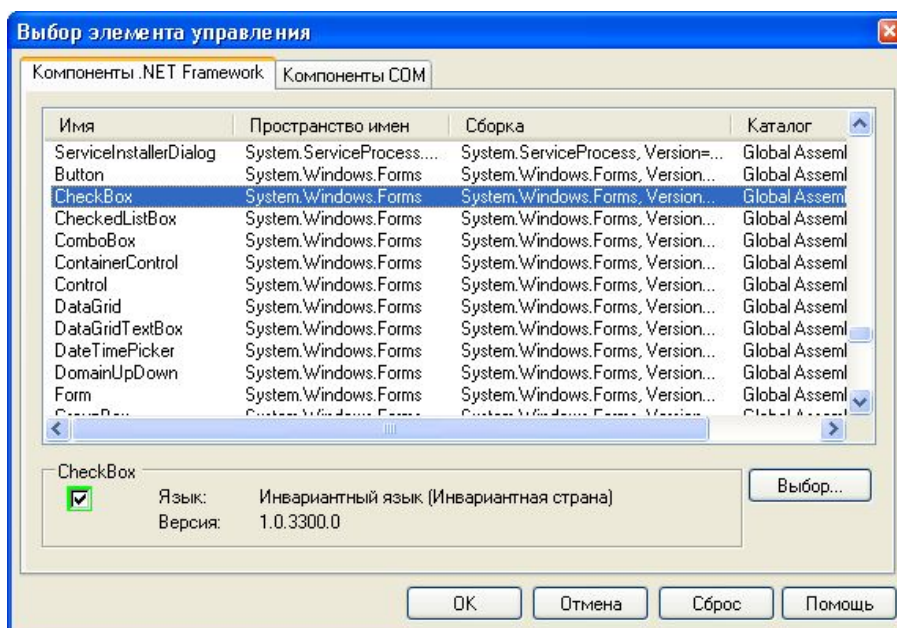
Свойства примитива Элемент управления

- **Элемент управления** – позволяет выбрать конкретный элемент управления из элементов, зарегистрированных на машине пользователя. При задании примитиву свойства **Элемент управления** список свойств расширяется свойствами выбранного элемента управления.
- **Тип** – определяет имя типа примитива
- **Имя объекта** – определяет уникальное имя примитива на мнемосхеме

Как разместить элемент управления на мнемосхеме?

Для размещения элемента управления на мнемосхеме следует:

- 1 Нажать на панели инструментов кнопку: 
 - 2 Щелкнуть левой клавишей мыши на мнемосхеме и, не отпуская ее, установить размеры элемента управления:
- 
- 3 Выбрать конкретный элемент управления: для этого щелкнуть мышью в поле свойства **Элемент управления** и в появившемся окне **Выбор элемента управления** указать необходимый элемент управления.



При выборе элемента управления из списка (или при выборе его из конкретного файла, в результате которого расширяется список) элемент управления размещается на мнемосхеме в заданной области.

Свойства выбранного элемента управления, отображаются на вкладке Анимация



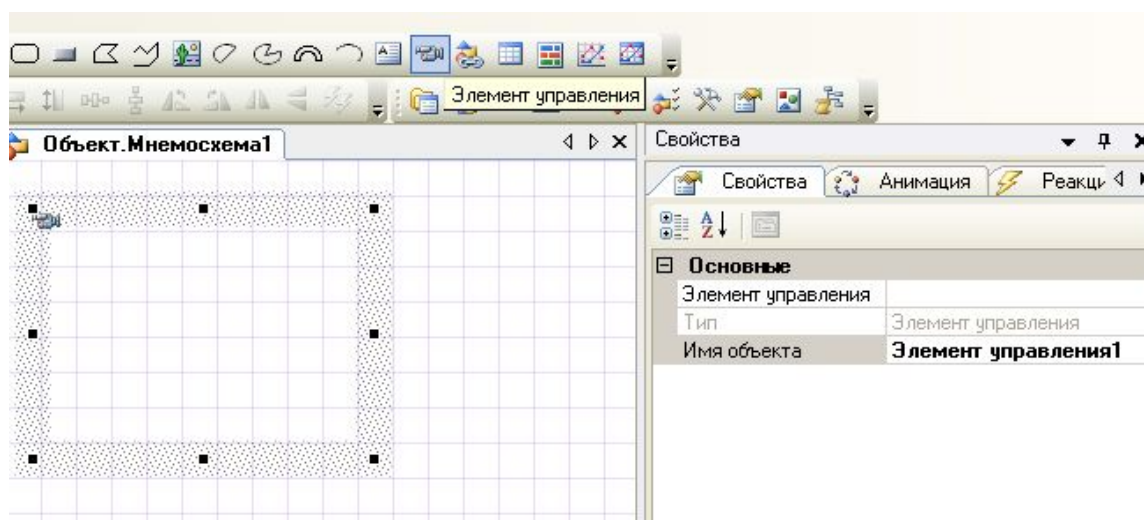
Элементом можно управлять при помощи скриптов, обращаясь к его свойствам и вызывая его методы.

Пример

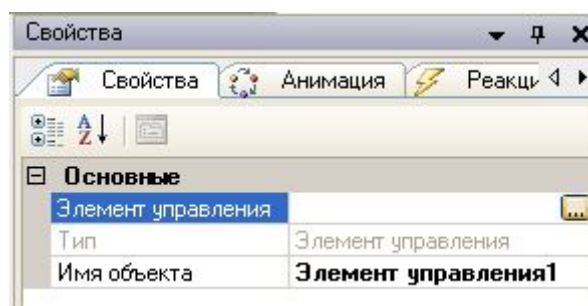
В качестве примера рассмотрим отображение на мнемосхеме результатов **http-запроса**.

Для отображения на мнемосхеме результатов http-запроса следует:

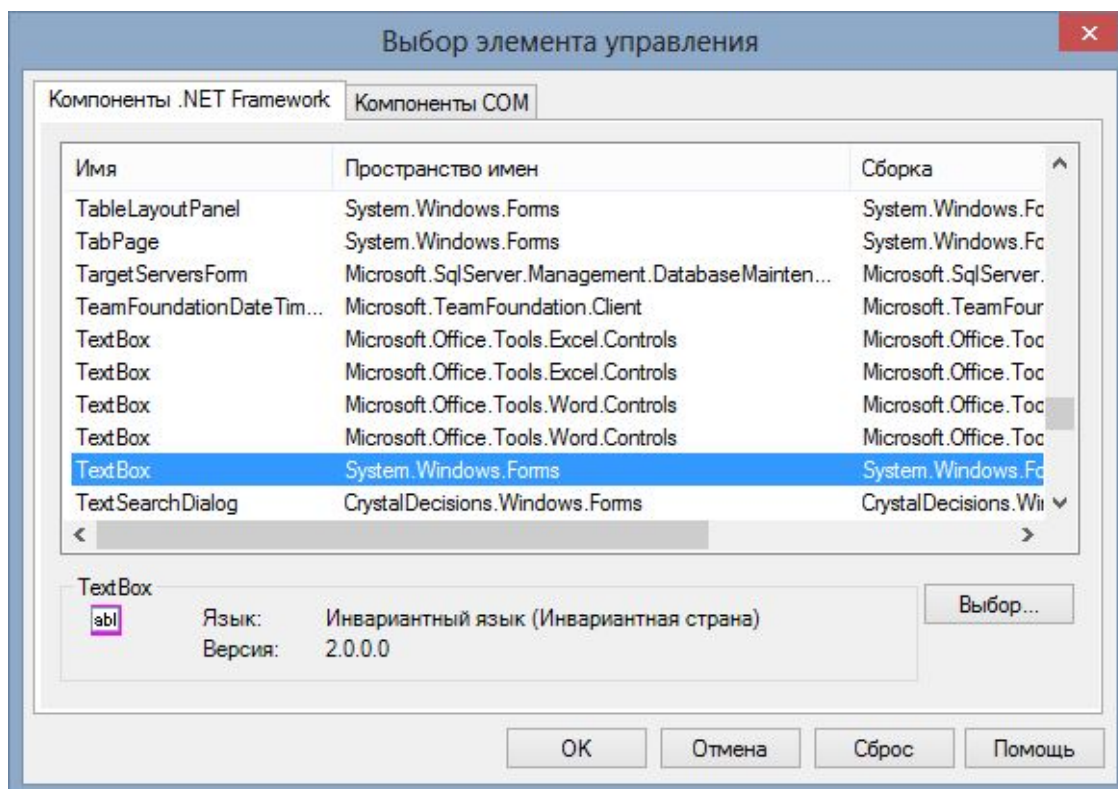
- 1 Поместить на мнемосхему элемент управления



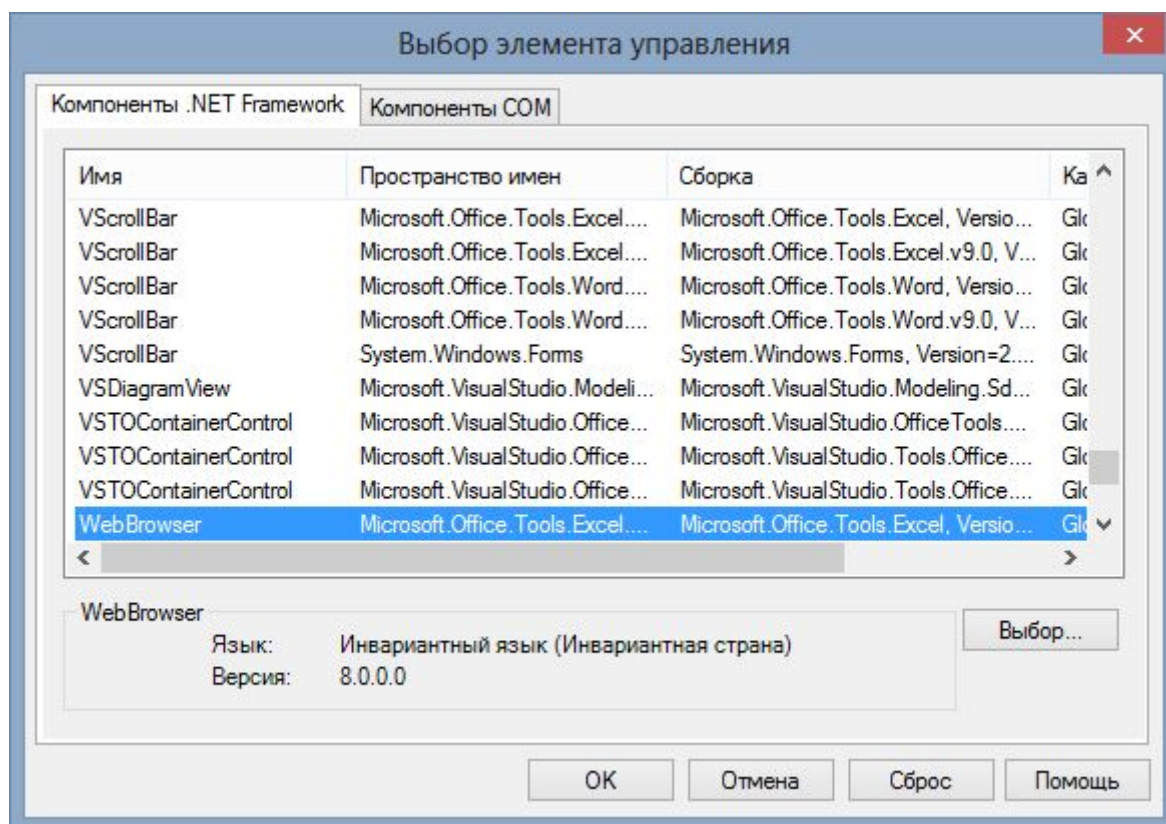
- В окне свойств выбрать **Элемент управления** и нажать на кнопку



- Выбрать вкладку **Компоненты .Net Framework** и найти элемент управления **TextBox** из пространства имен **System. Windows. Forms**



- 4 Повторить пункты 1-3 для размещения на мнемосхеме **.NET** элемента управления **WebBrowser** из пространства имен **System. Windows. Forms**.



- 5 Поместить на мнемосхему примитив **Объемный прямоугольник** (для использования его в качестве кнопки)
- 6 Назначить реакцию **Скрипт** на событие **Двойное нажатие левой кнопки мыши** (как это сделать смотрите в подразделе 3.3 **Реакции**)
- 7 Ввести текст скрипта :
- ```

TextBox textBox =
(TextBox)((ControlHost)Client.Mnemoscheme.GetPrimitive(
"Элемент управления1")).Control);
string URL = textBox.Text;
WebBrowser browser =
(WebBrowser)((ControlHost)Client.Mnemoscheme.GetPrimitive(
"Элемент управления2")).Control;
browser.Navigate(URL);

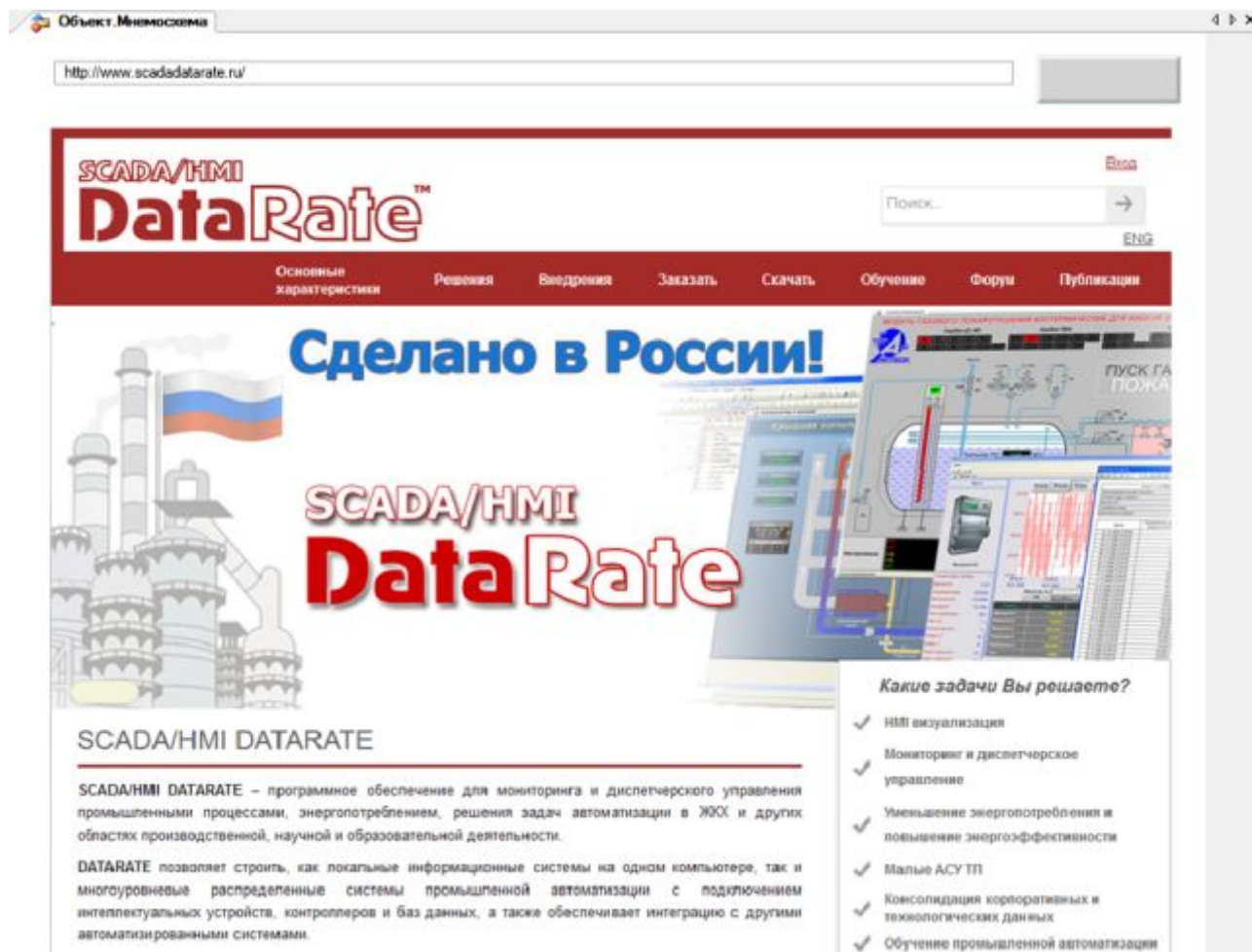
```

где: **Элемент управления1** – имя примитива, содержащего элемент управления **TextBox**  
**Элемент управления2** – имя примитива, содержащего элемент управления **Microsoft Web Browser**

- 8 Выполнить команду **Проверить скрипт на ошибки**
- 9 Убедиться в правильности работы приложения:
- Задать адрес в элементе управления **TextBox**, например: <http://www.scadadatarate.ru/>
  - Выполнить двойной щелчок на примитиве **Объемный прямоугольник**



В результате на мнемосхеме отобразиться окно Web-браузера с главной страницей сайта SCADA DataRate.



### ВНИМАНИЕ!!!

В данной версии отсутствует возможность назначать реакции на события примитива Элемент управления.

Элементы управления, размещенные на мнемосхеме, не будут выводиться при печати мнемосхем, а также при использовании таких мнемосхем при построении отчетов и в Web-контроль



### 3.2 Анимация

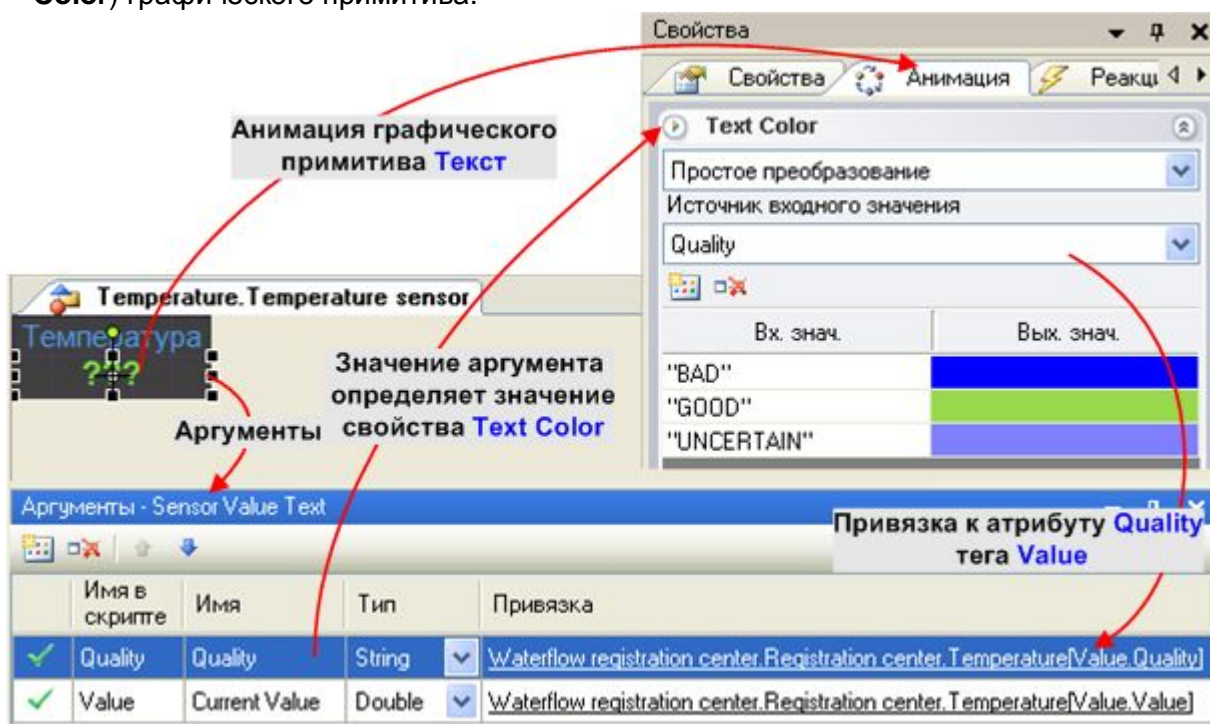
**Анимация** - это зависимость свойства графического примитива от его аргументов по определенному правилу.

Анимировать можно практически любое свойство примитива, настраивая встроенный алгоритм преобразования (простое преобразование) или используя скрипт.

#### Пример

Аргумент **Quality** графического примитива **Текст** изменяется в соответствии со значением атрибута тега, к которому он привязан.

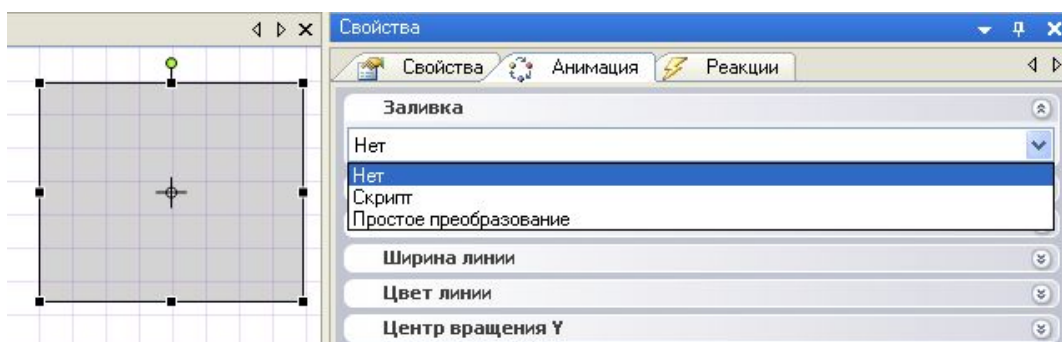
Таким образом, значение тега динамически управляет цветом текста (свойство **Text Color**) графического примитива.



#### Как добавить анимацию?

Для добавления **анимации** к свойству графического примитива следует:

- 1 Выделить графический элемент
- 2 Сформировать аргумент, который определяет свойство графического примитива
- 3 Перейти на вкладку **Анимация** в окне **Свойства**
- 4 Развернуть свиток с именем свойства и выбрать способ преобразования
- 5 Задать правила преобразования.



## 3.2.1 Простое преобразование

**Простое преобразование** - осуществляет преобразование набора входных значений в значения свойства графического примитива.

**Пример 1**

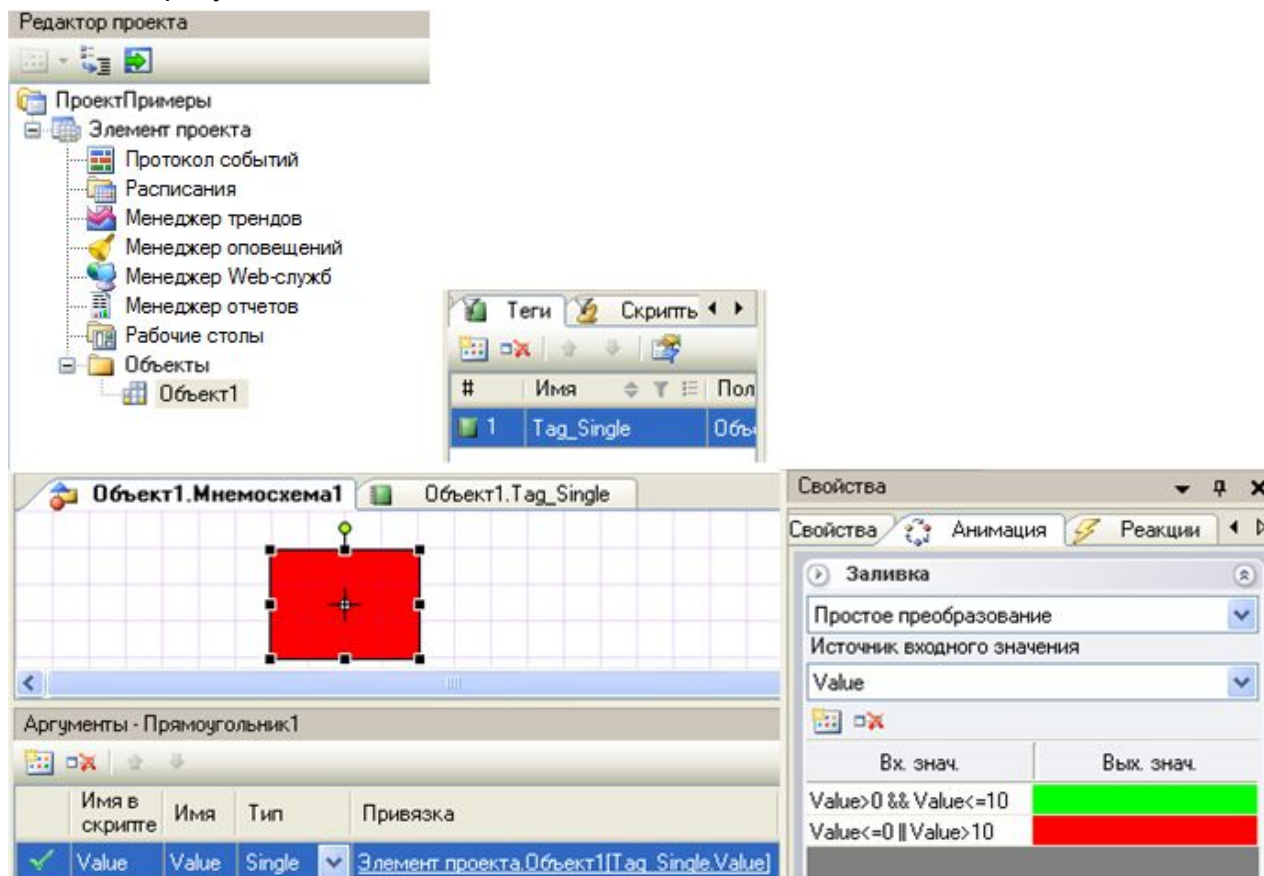
Используем анимацию **Простое преобразование** для изменения цвета графического примитива **Прямоугольник** и значения примитива **Текст** в зависимости от значений атрибута тега.

Создадим простой тестовый проект **DataRate ПроектПримеры**, определим объект **Объект1**, его теги **Tag\_Single**, **Tag\_Boolean** (на основе библиотеки) и вид **Мнемосхема1**.

**Изменение цвета графического примитива Прямоугольник**

Для задания анимации следует:

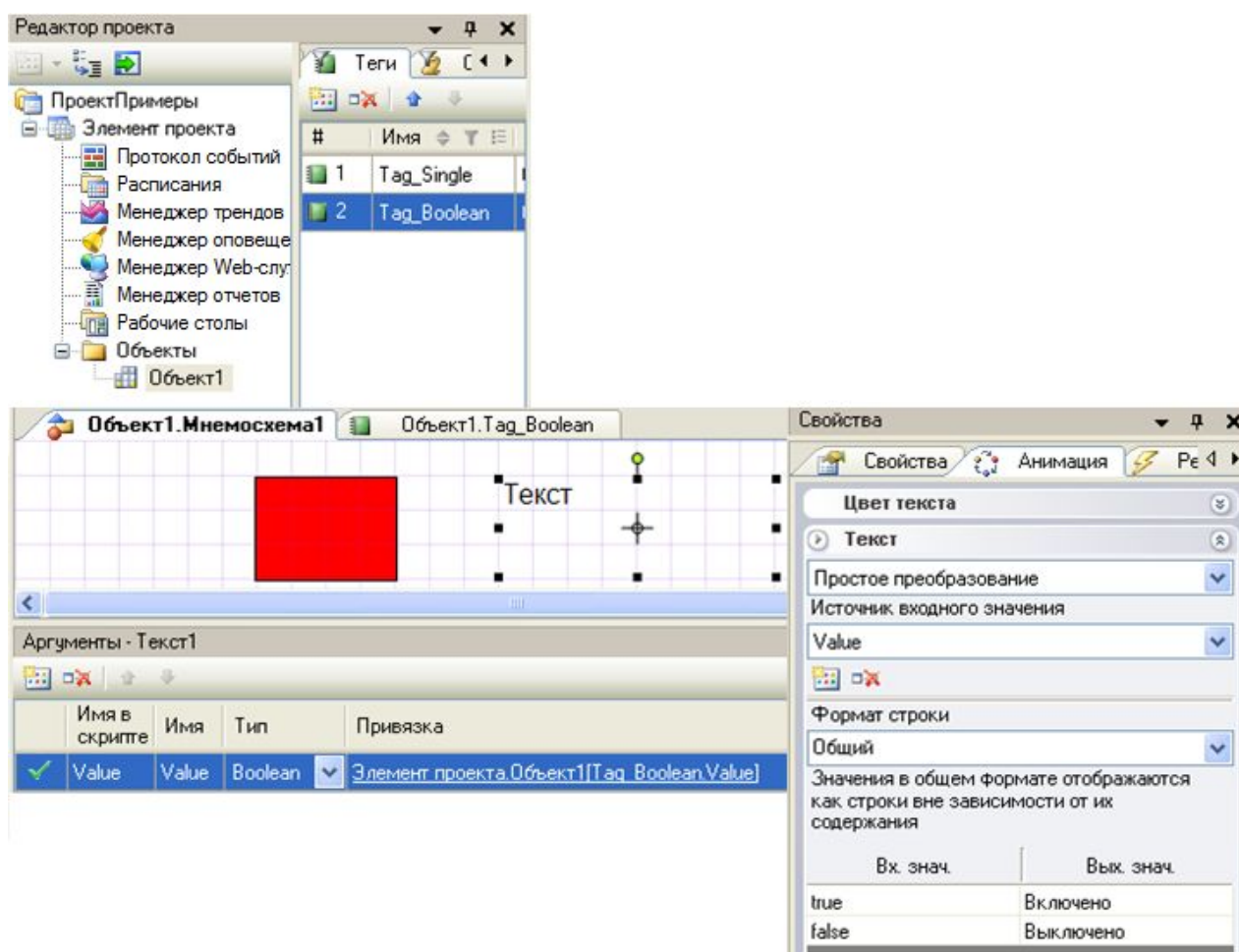
- 1 Разместить на мнемосхеме графический примитив **Прямоугольник**
- 2 Перетащить тег **Tag\_Single** в окно **Аргументы** прямоугольника (укажите при этом в появляющемся окне Редактор связей атрибут **Value**)
- 3 На вкладке **Анимация** выбрать свойство **Заливка** и тип анимации **Простое преобразование**
- 4 Из выпадающего списка **Источник входного значения** выбрать **Value** (в случае единственного источника, он устанавливается автоматически)
- 5 Добавить правила преобразования (**Вх.знач.** и **Вых.знач.**) как показано на рисунке.



### Изменение значения примитива Текст

Для задания анимации следует:

- 1 Разместить на мнемосхеме графический примитив **Текст**
- 2 Перетащить тег **Tag\_Boolean** в окно **Аргументы** примитива **Текст** (укажите при этом в появляющемся окне Редактор связей атрибут **Value**)
- 3 На вкладке **Анимация** выбрать свойство **Текст** и тип анимации **Простое преобразование**
- 4 Из выпадающего списка **Источник входного значения** выбрать **Value** (в случае единственного источника, он устанавливается автоматически)
- 5 Добавить правила преобразования (**Вх.знач.** и **Вых.знач.**) как показано на рисунке.



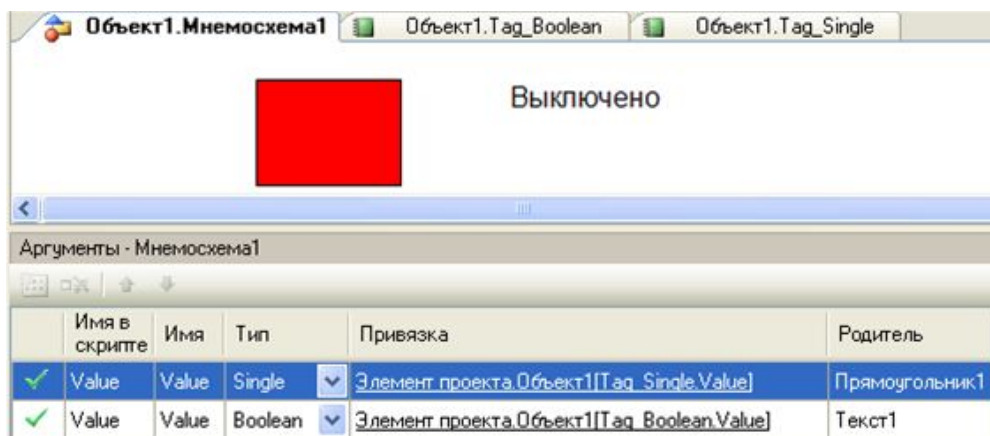
### Тестирование

Запускаем созданный проект на отладку.

#### Тест 1

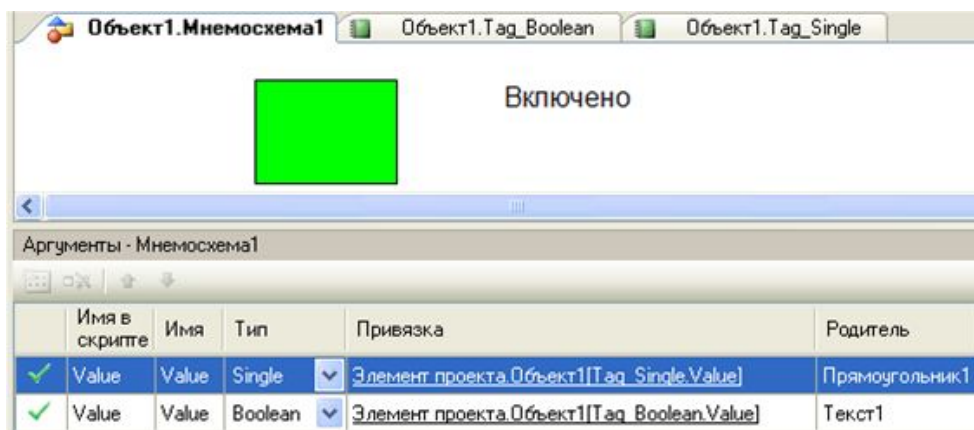
**Исходные данные:** Tag\_Single=0, Tag\_Boolean=false (значения тегов умолчанию).

**Результаты** : Прямоугольник – красный, Текст – **Выключено**

**Тест 2**

**Исходные данные:** Tag\_Single=5, Tag\_Boolean=true. Для задания значений следует двойным щелчком по названию тега открыть в области редактирования таблицу атрибутов тега и задать значение атрибута

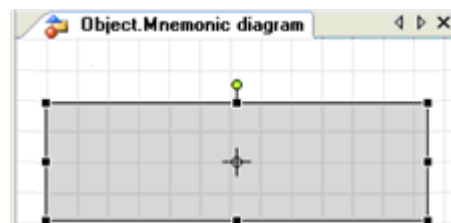
**Результаты** : Прямоугольник – **зеленый**, Текст – **Включено**

**Пример 2**

Используем анимацию **Простое преобразование** для изменения ширины примитива **Прямоугольник** в зависимости от значений атрибута тега. Рассмотрим случаи линейного и дискретного преобразования.

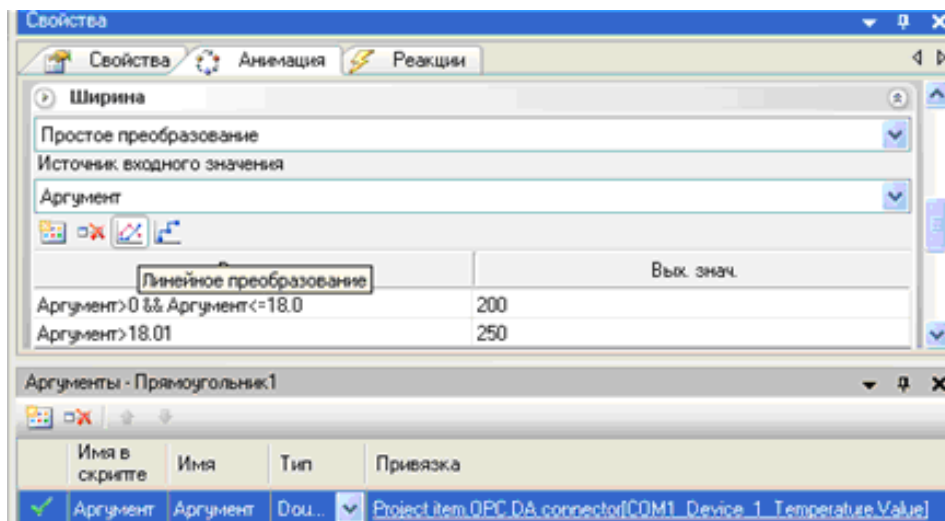
Для задания анимации следует:

- 1 Разместить на мнемосхеме графический примитив **Прямоугольник**
- 2 Перетащить тег в окно **Аргументы** прямоугольника (укажите при этом в появляющемся окне Редактор связей атрибут **Value**)
- 3 На вкладке **Анимация** выбрать свойство **Ширина** и тип анимации **Простое преобразование**
- 4 Из выпадающего списка **Источник входного значения** выбрать **Аргумент** (в случае единственного источника, он устанавливается автоматически)

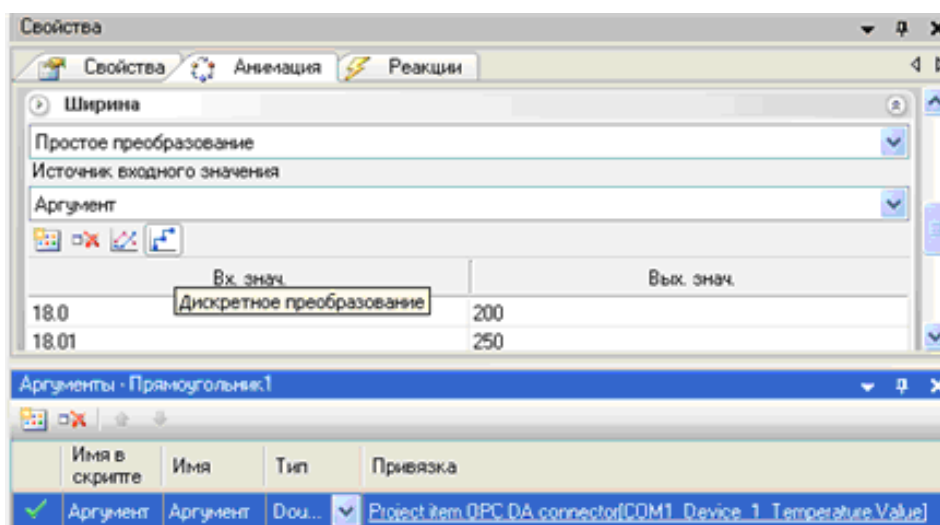




- 5 Добавить правила преобразования (**Вх.знач.** и **Вых.знач.**) как показано на рисунке: **а** – для линейного преобразования, **б** – для дискретного преобразования.



а)



б)

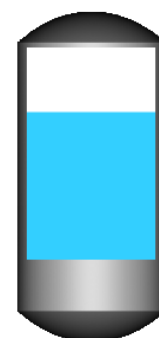
### Тестирование

Запускаем созданный проект на отладку. При изменении значения атрибута **Value** в указанных пределах (для проекта с линейным преобразованием) или для заданных значений **Value** (в случае проекта с дискретным преобразованием) ширина прямоугольника изменяется так, как указано в правилах преобразования (поле **Вых. значение**).

### Пример 3. Использование Графической библиотеки

Используем анимацию **Простое преобразование** для создания барграфа.

Барграф создается на основе примитива **Прямоугольник**, высота которого изменяется в зависимости от значений атрибута тега и библиотечного объекта **Ёмкость\_с\_разрезом**.



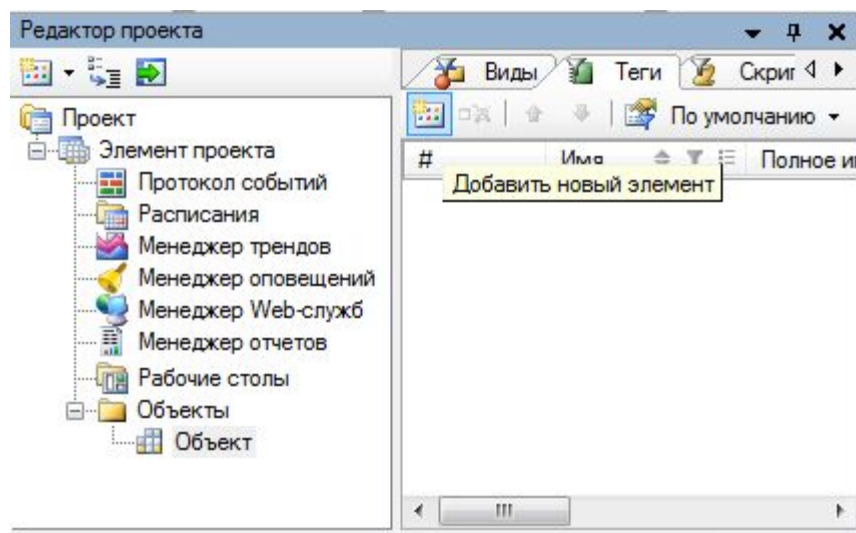
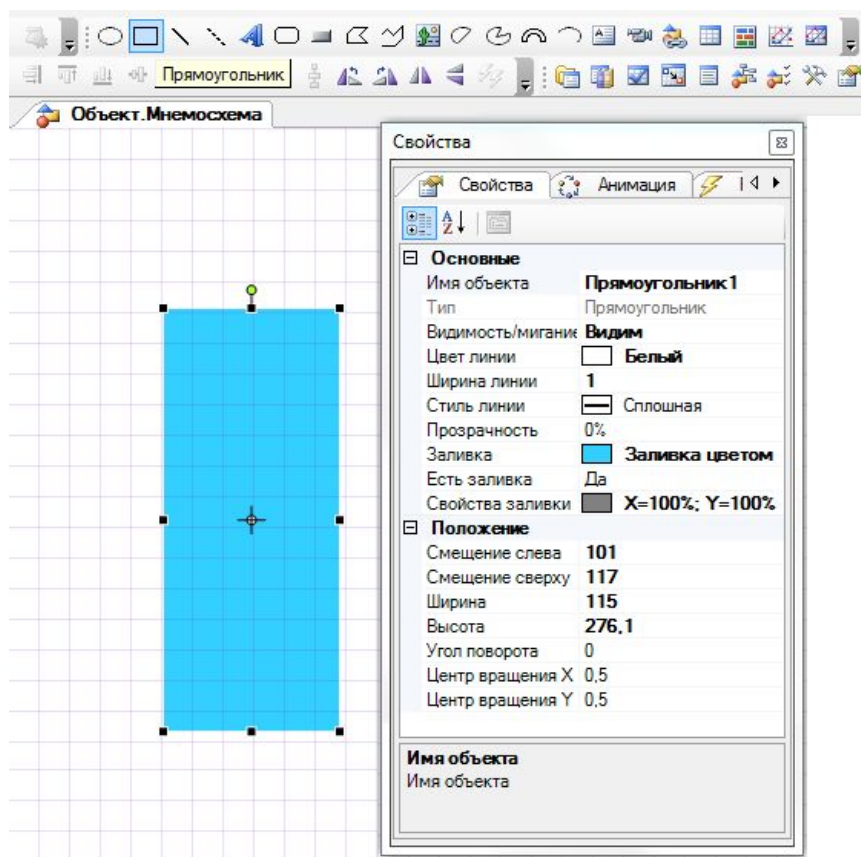


Для создания барграфа следует:

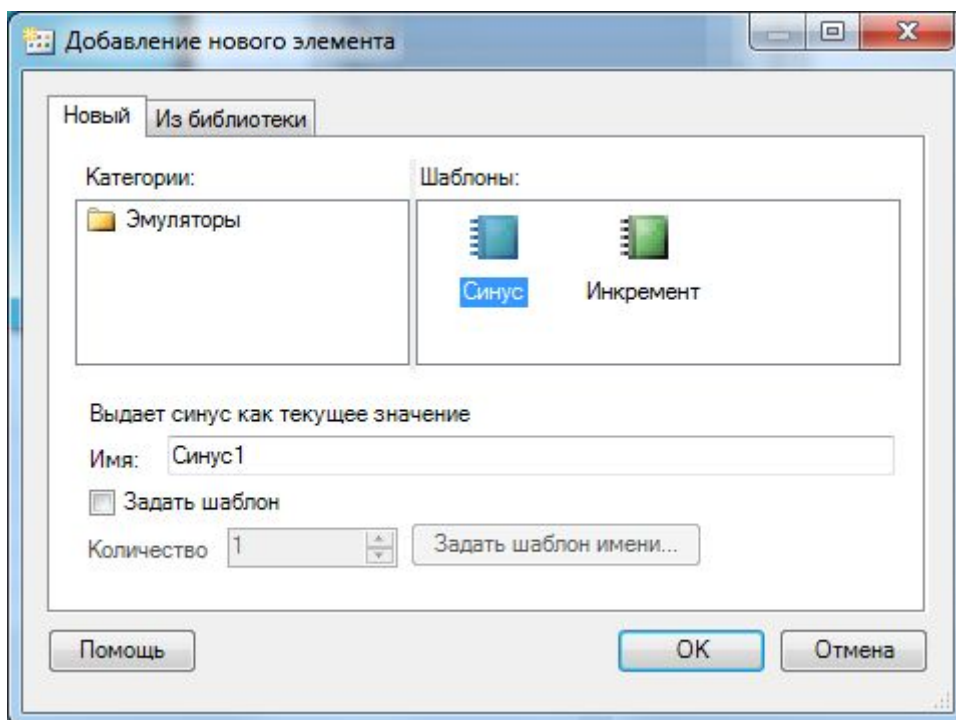
- 1 Разместить на мнемосхеме графический примитив **Прямоугольник** и в окне **Свойства** задать значения свойства **Заливка** и **Цвет линии**
- 2 В качестве тега, задающего высоту прямоугольника, для примера, выбрать тег эмулятора **Синус**.


Для этого:

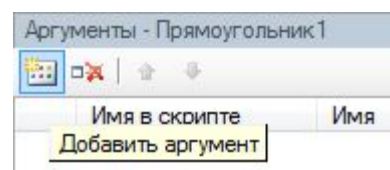
- В окне **Редактор проекта** перейти на вкладку **Теги**, нажать на кнопку **Добавить новый элемент**



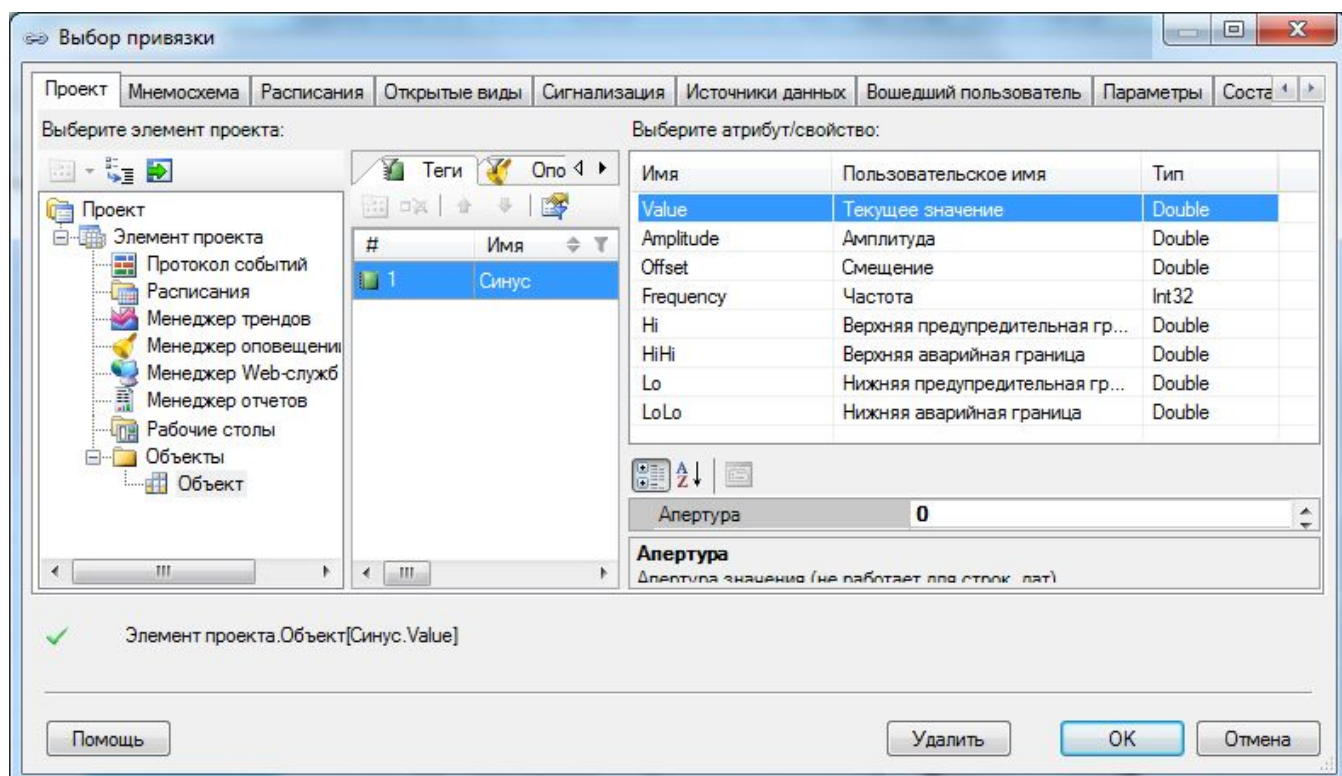
- В появившемся окне **Добавление нового элемента** перейти на вкладку **Новый**, выбрать **Синус**, задать **Имя** и нажать на кнопку **ОК**




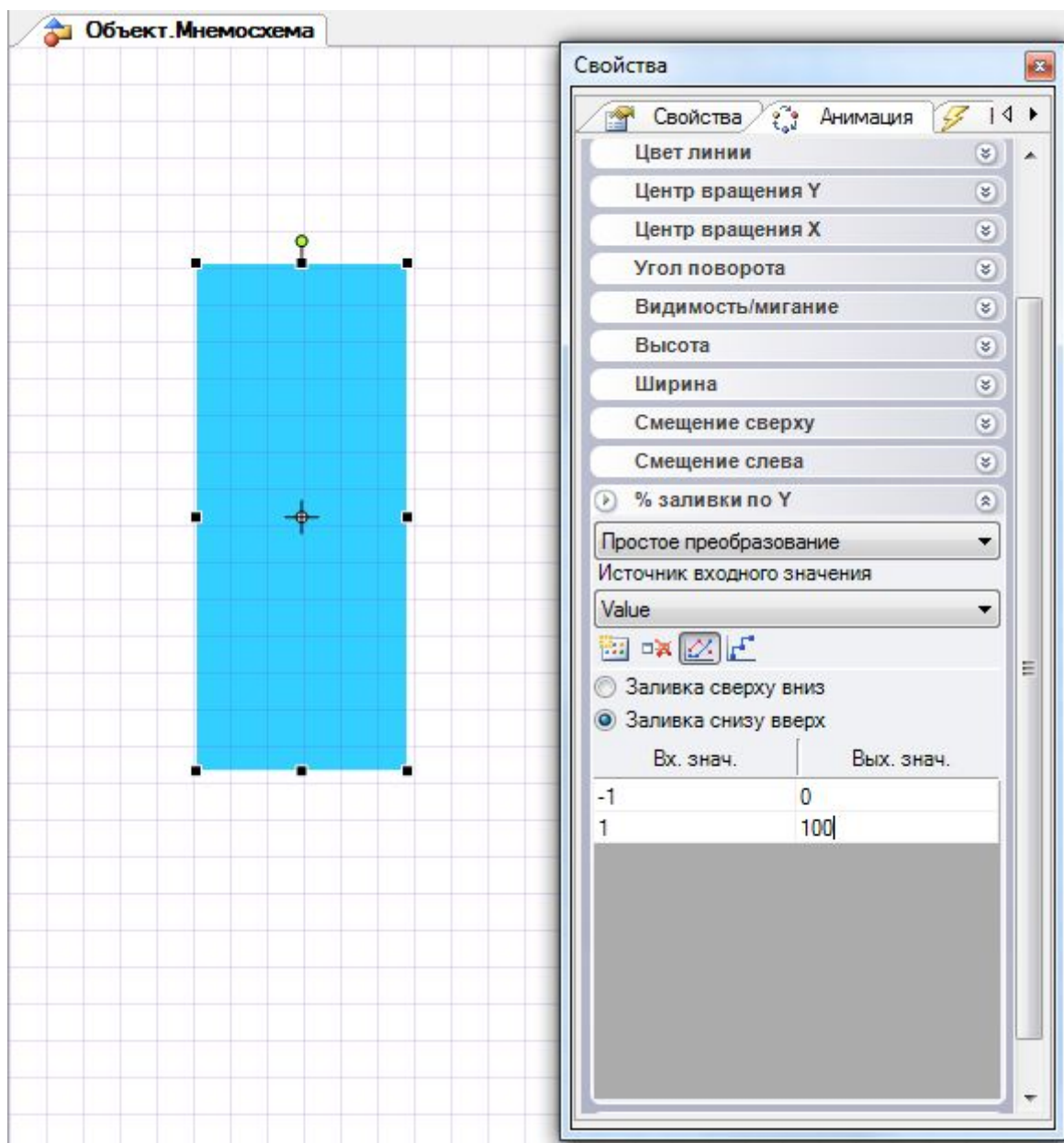
- 3 Открыть окно **Аргументы** (кнопка  на панели инструментов), выделить прямоугольник на мнемосхеме и в окне **Аргументы** нажать на кнопку **Добавить аргумент**




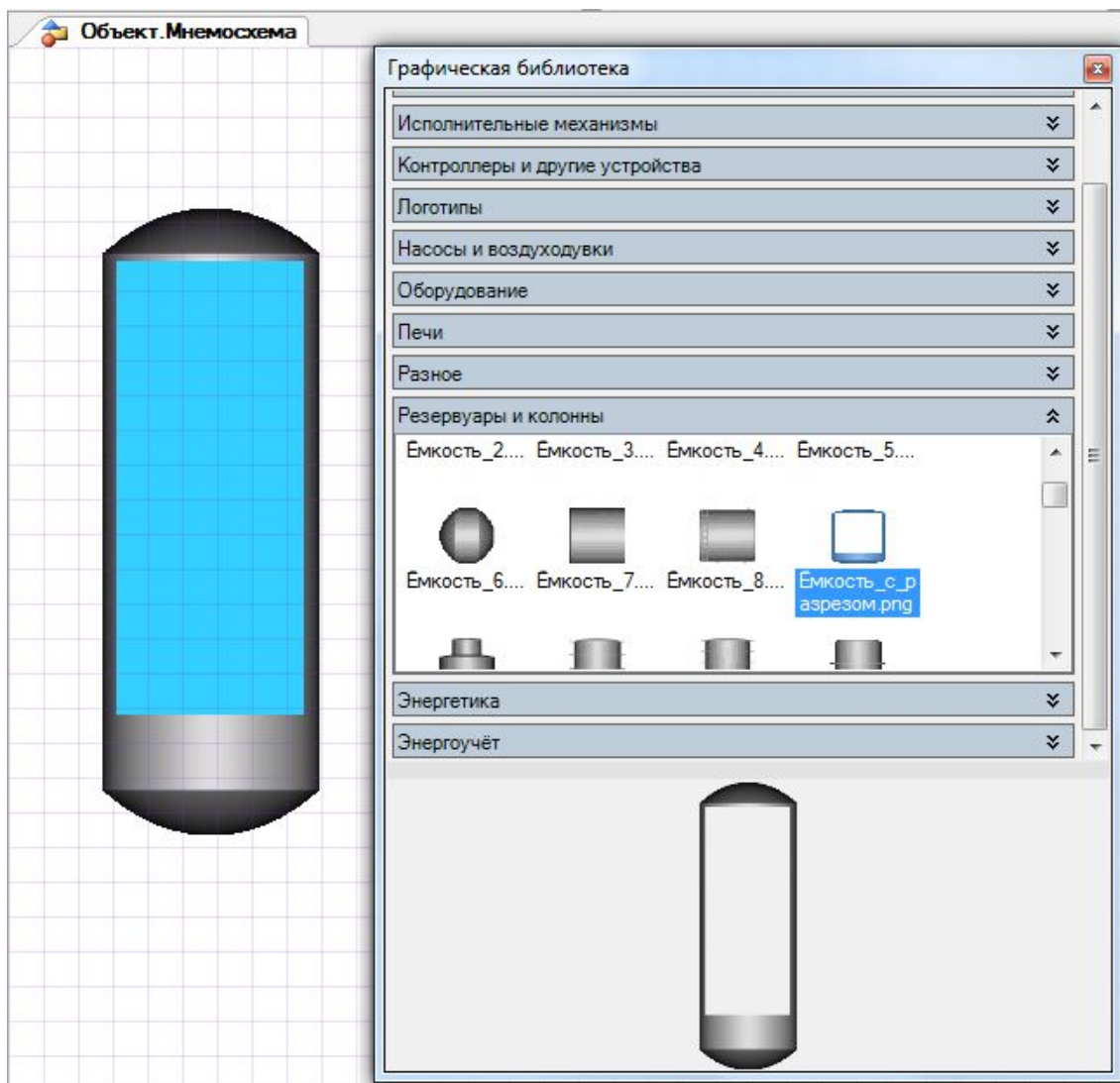
- 4 В окне **Аргументы** в колонке **Привязка** щёлкнуть по значению **NA**
- 5 В открывшемся окне **Выбор привязки** выбрать **Синус** и выделить значение **Value**, затем нажать на кнопку **OK**



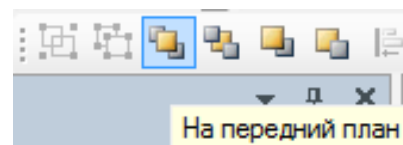
- 6 В окне **Свойства** (примитива Прямоугольник) перейти на вкладку **Анимация**, задать для свойства **% заливки по Y** значение **Простое преобразование**, (**Источник входного значения** подставится автоматически), выбрать **Заливка снизу вверх**, нажать на кнопку **Добавить**  и установить пары значений:  
-1 -> 0; 1 -> 100




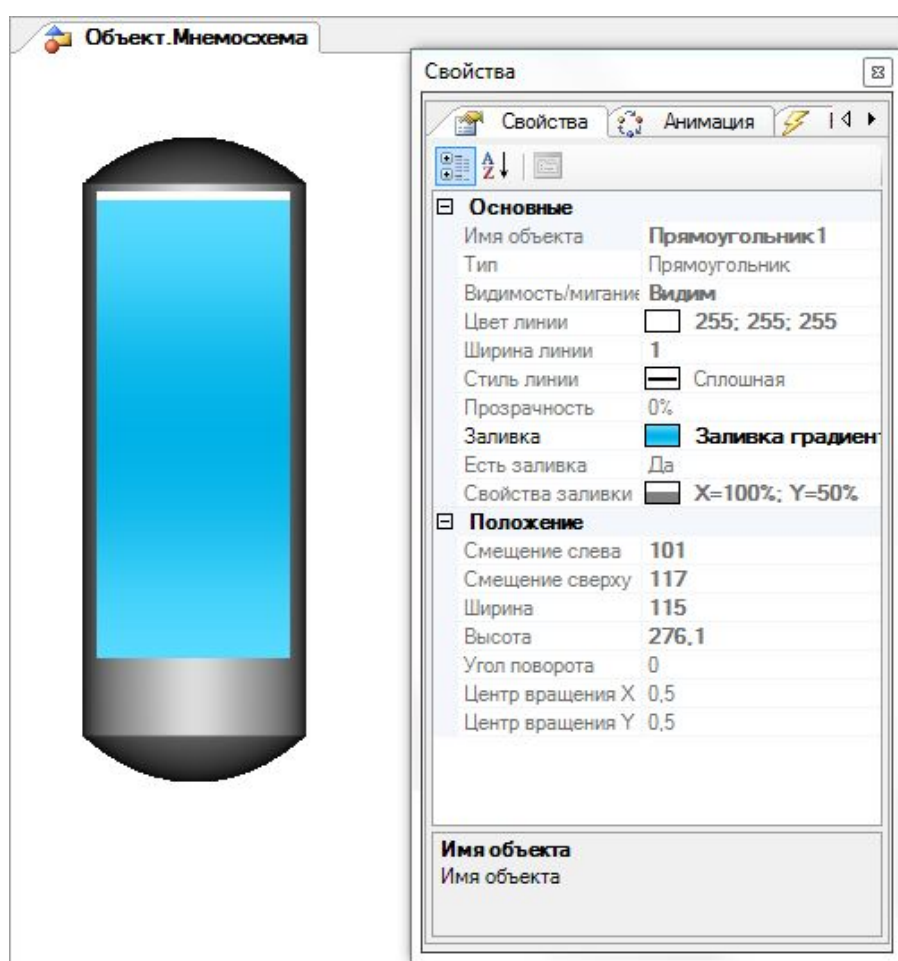
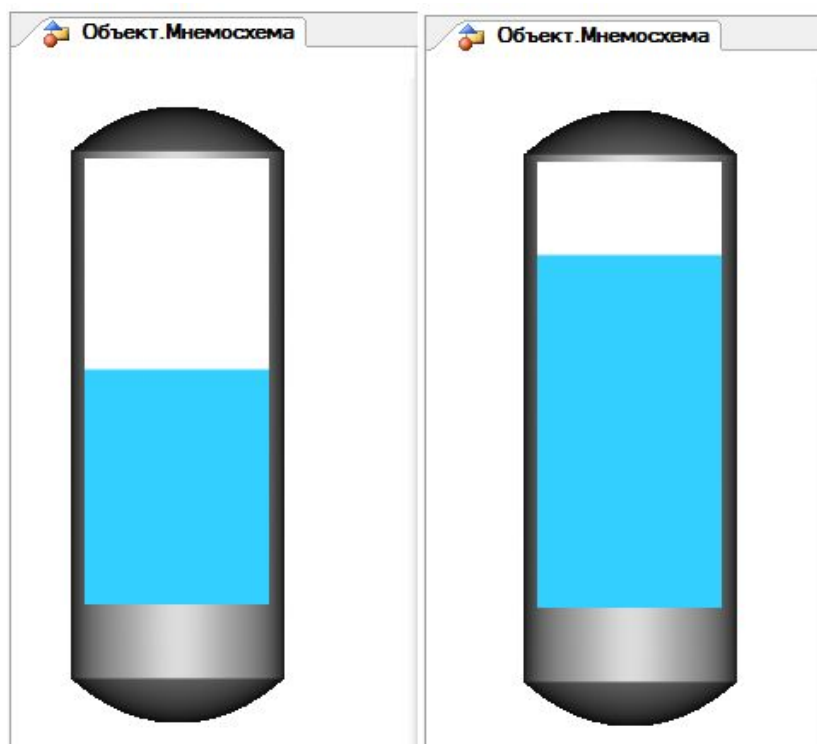
- 7 Открыть окно **Графическая библиотека** (щелчком по иконке  на панели инструментов), выбрать вкладку **Резервуары и колонны**, «кликнуть» по изображению **Ёмкость\_с\_разрезом.png** и перетащить его на мнемосхему поверх прямоугольника.



Используйте кнопки (**На передний план**, **На задний план**) на панели инструментов для расположения изображений



- 8 Запустите проект на исполнение
- 9 Заливку можно делать градиентной, для этого в окне **Свойства** прямоугольника нажать на кнопку  в строке **Заливка**. В открывшемся окне перейти на вкладку **Градиент**, выбрать необходимые цвета и нажать на кнопку **ОК**





### 3.2.2 Преобразование скриптом

**Преобразование скриптом** – устанавливает значение свойства примитива путем выполнения пользовательского скрипта.

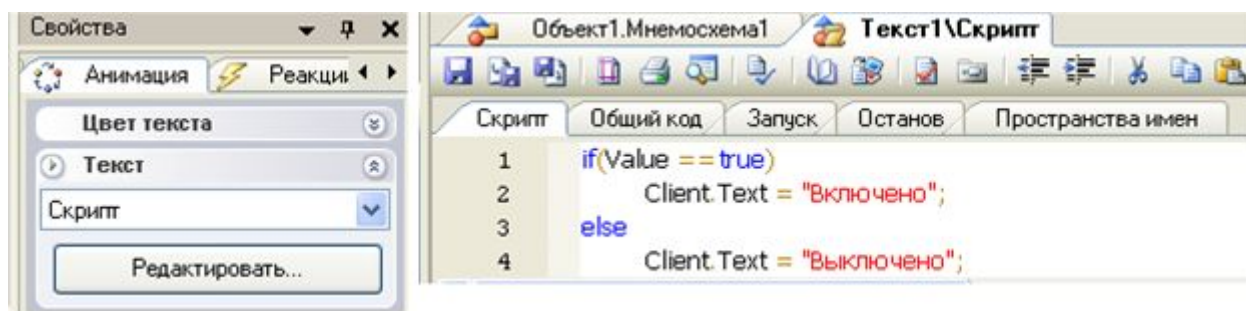
#### Пример

Используем анимацию **Преобразование скриптом** для изменения значения примитива **Текст** в зависимости от значений атрибута тега.

В качестве тестового проекта используем проект из примера анимации **Простое Преобразование**

Для задания анимации следует:

- 1 Разместить на мнемосхеме графический примитив **Текст**
- 2 Перетащить тег **Tag\_Boolean** в окно **Аргументы** примитива **Текст** (укажите при этом в появляющемся окне Редактор связей атрибут **Value**)
- 3 На вкладке **Анимация** выбрать свойство **Текст**, тип анимации **Скрипт** и нажать на кнопку **Редактировать...**
- 4 В редакторе скриптов набрать текст скрипта (как показано на рисунке).



#### Тестирование

Запускаем созданный проект на отладку.

##### Тест 1

**Исходные данные:** значение тега **Tag\_Boolean = false** (по умолчанию)

**Результаты** : текст – **Выключено**

##### Тест 2

**Исходные данные:** значение тега **Tag\_Boolean = true**. Для этого следует двойным щелчком по названию тега открыть в области редактирования таблицу атрибутов тега и задать значение атрибута

**Результаты** : текст – **Включено**

Скриншоты тестов аналогичны скриншотам анимации Простое преобразование

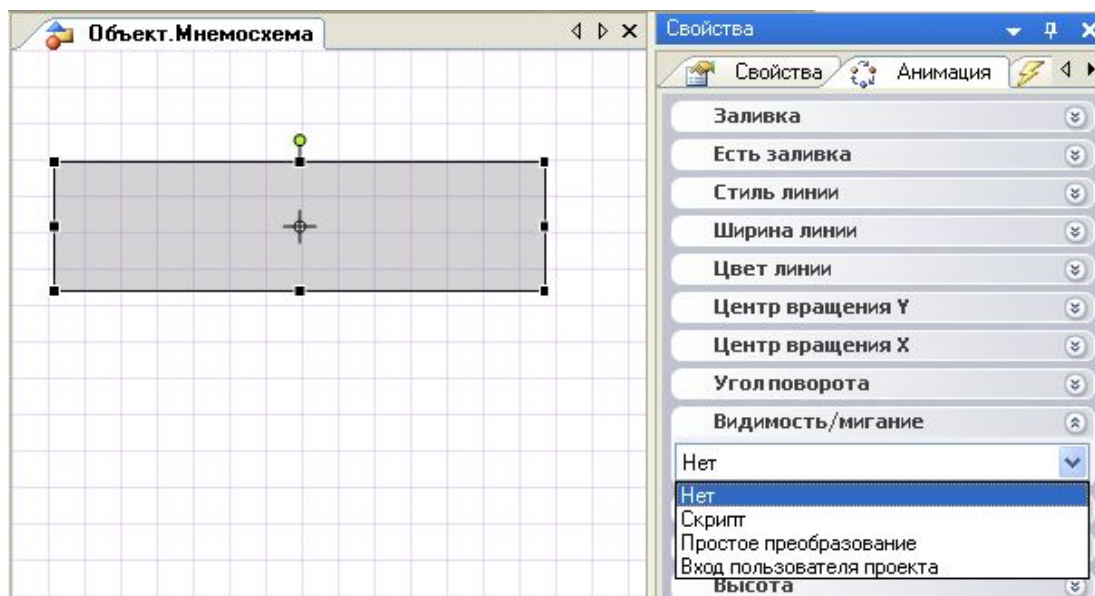
### 3.2.3 Вход пользователя проекта

**Вход пользователя проекта** – управляет видимостью примитивов на мнемосхеме в зависимости от роли пользователя.

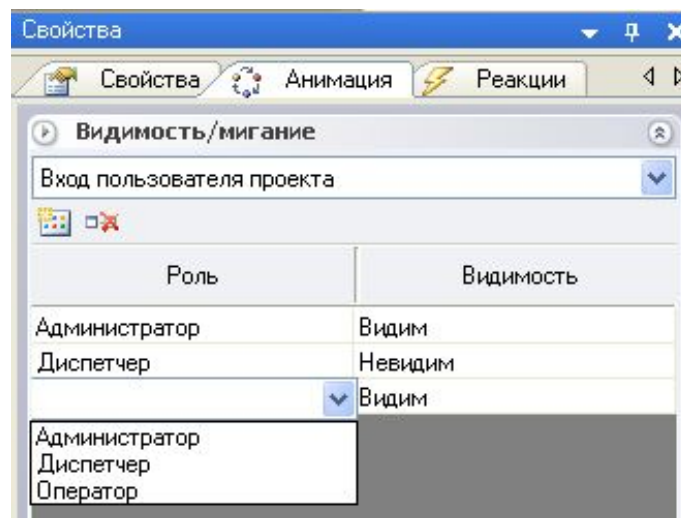
Для задания анимации следует:

- 1 Разместить на мнемосхеме графический примитив **Прямоугольник**

- 2 На вкладке **Анимация** выбрать свойство **Видимость/Мигание** и тип анимации **Вход пользователя проекта**



- 3 Определить видимость примитива в зависимости от роли пользователя (как показано на рисунке).



В режиме исполнения проекта для конкретного пользователя примитив анимируется (видим/невидим) в соответствии с правами доступа (ролью) пользователя.

### 3.3 Реакции

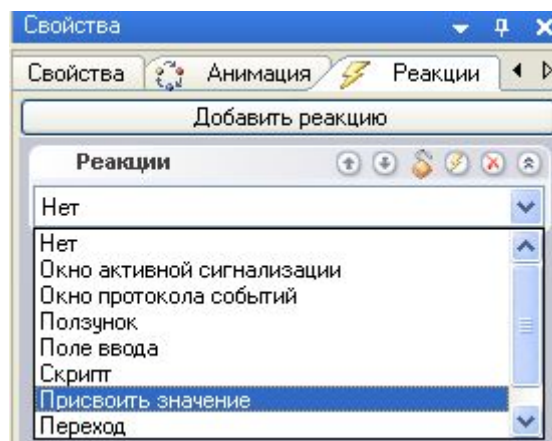
**Реакция объекта** – это действие, автоматически выполняемое **DataRate** в ответ на какое-либо определенное событие.

**Событие:**

- Действие Пользователя (например, нажатие клавиши клавиатуры, нажатие кнопки мыши)
- Попадание курсора в заданную область
- и другие.



**Реакция на событие:**

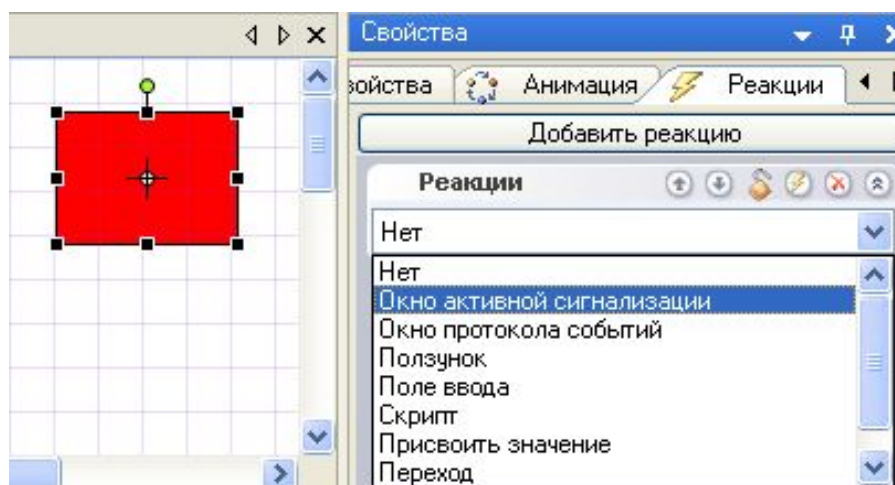
- Изменить значение (**Присвоить значение**)
- Перейти на мнемосхему или на рабочий стол (**Переход**)
- Выполнить скрипт (**Скрипт**)
- Изменить функциональность графического примитива (например, **Поле ввода**, **Ползунок**)
- Вызов диалогового окна (**Окно активной сигнализации**, **Окно протокола событий**, **Окно журнала тревог**)
- Квитирование.



**Как добавить реакцию графическому примитиву?**



Для добавления реакции следует:

- 1 Выделить графический элемент
- 2 В окне **Свойства** перейти на вкладку **Реакции**
- 3 Нажать на кнопку **Добавить реакцию**
- 4 Назначить события реакции – нажать на пиктограмму  в заголовке свитка реакции и в появившемся окне **Редактор событий** выбрать необходимые события
- 5 Нажать на пиктограмму , из выпадающего списка выбрать необходимую реакцию и настроить ее характеристики



**ВНИМАНИЕ!!!**

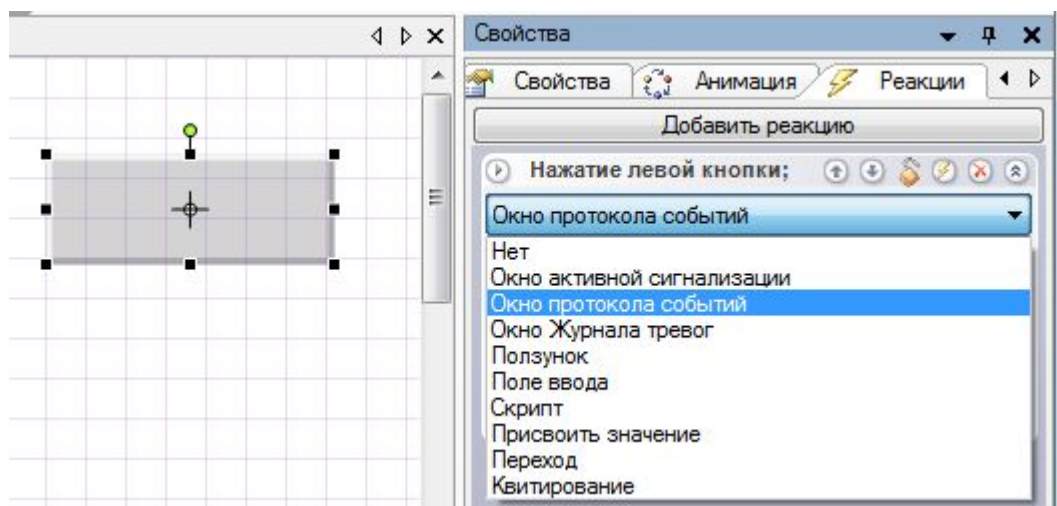
Если на одно и тоже событие назначено несколько реакций, то они будут выполняться в последовательности их определения.

Изменить порядок реакций можно кнопками перемещения  и .



## 3.3.1 Редактирование событий

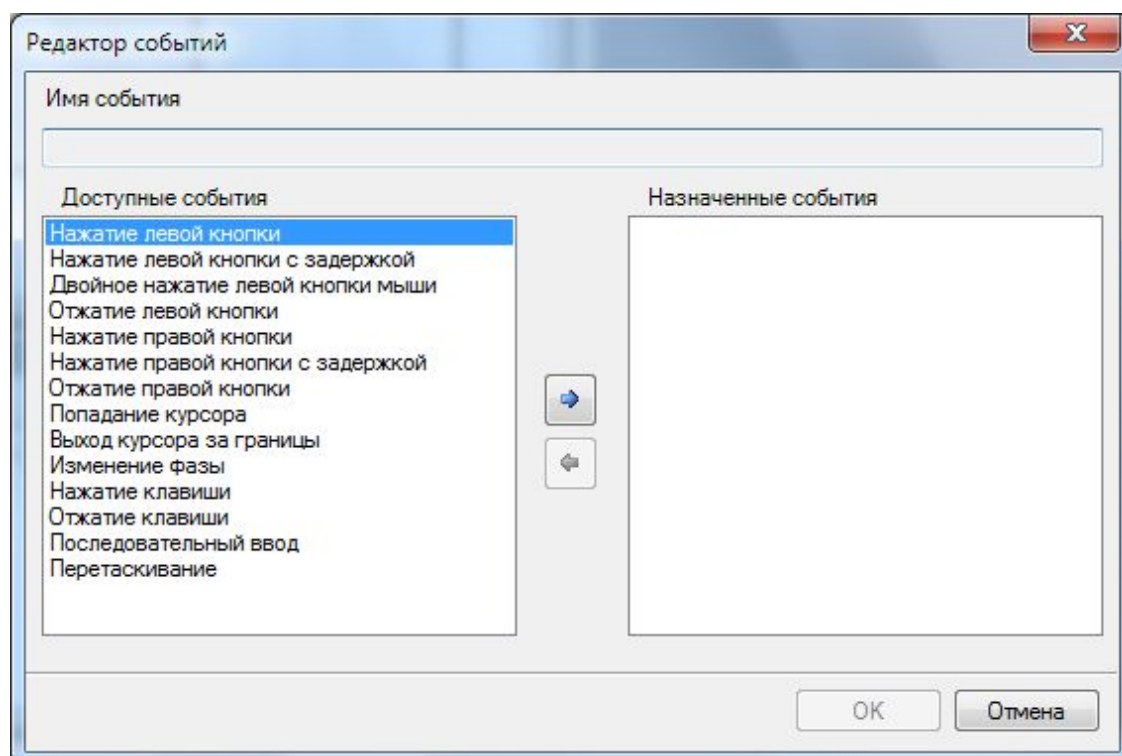
Для назначения событий реакции следует:

- 1 Нажать на пиктограмму  в заголовке свитка реакции.



- 2 В появившемся окне **Редактор событий** отредактировать список **Назначенные события**:

- Для добавления события выделить событие в списке **Доступные события** и нажать на кнопку  или сделать двойной щелчок по имени события
- Для удаления события из списка **Назначенные события** выделить событие в списке **Назначенные события** и нажать на кнопку .

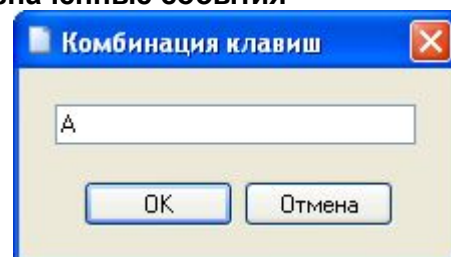


- 3 Для принятия изменений и закрытия окна нажать кнопку **ОК**.

### 3.3.1.1 Редактирование события Нажатие клавиши

Для редактирования события **Нажатие клавиши** следует:

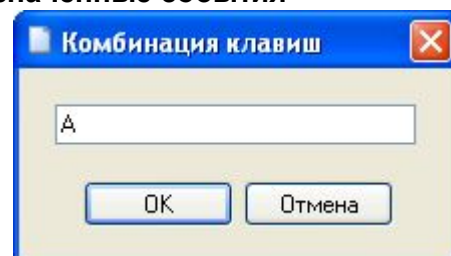
- 1 Добавить событие **Нажатие клавиши** в список **Назначенные события**
- 2 Сделать двойной щелчок по имени события
- 3 В появившемся диалоге **Комбинация клавиш** ввести **клавишу** или **модификатор + клавиша**
- 4 Для принятия изменений нажать кнопку **ОК**, для отмены – кнопку **Отмена**.



### 3.3.1.2 Редактирование события Отжатие клавиши

Для редактирования события **Отжатие клавиши** следует:

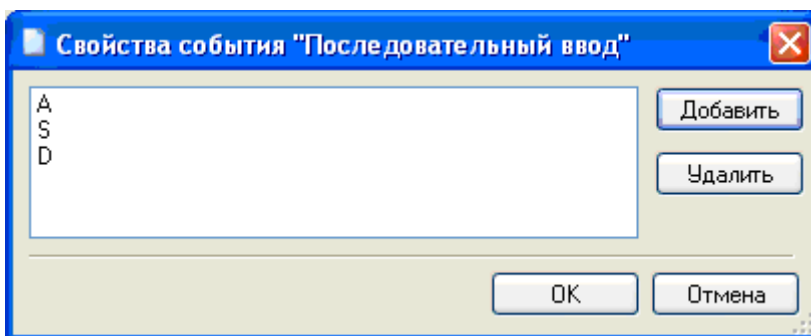
- 1 Добавить событие **Отжатие клавиши** в список **Назначенные события**
- 2 Сделать двойной щелчок по имени события
- 3 В появившемся диалоге **Комбинация клавиш** ввести клавишу или модификатор + клавиша
- 4 Для принятия изменений нажать кнопку **ОК**, для отмены – кнопку **Отмена**.



### 3.3.1.3 Редактирование события Последовательный ввод

Для редактирования события **Последовательный ввод** следует:

- 1 Добавить событие **Последовательный ввод** в список **Назначенные события**
- 2 Сделать двойной щелчок по имени события
- 3 В появившемся окне **Свойства события «Последовательный ввод»** последовательно добавить (нажать на кнопку **Добавить**) необходимые клавиши или комбинации клавиш.
- 4 Для принятия изменений нажать кнопку **ОК**, для отмены – кнопку **Отмена**.



### 3.3.1.4 Редактирование события Нажатие левой клавиши мыши с задержкой

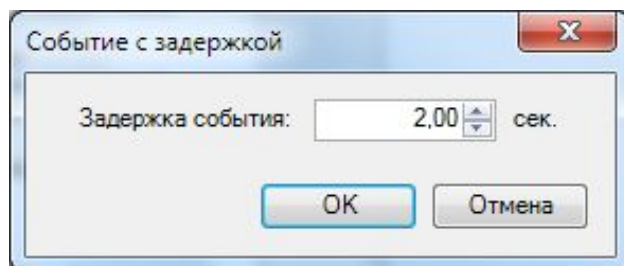
Событие **Нажатие левой клавиши мыши с задержкой** возникает, если после щелчка левой клавиши мыши удерживать ее нажатой определенное время.

Для редактирования события:

- 1 Добавьте событие **Нажатие левой клавиши мыши с задержкой** в список **Назначенные события**, как описано в разделе Редактирование событий
- 2 Сделайте двойной щелчок по имени события



- 3 В появившемся окне **Событие с задержкой** задайте интервал задержки для события.



Интервал задержки события задается в диапазоне от 1 до 60 секунд

- 4 Для принятия изменений нажмите на кнопку **ОК**, для отмены – на кнопку **Отмена**

### 3.3.1.5 Редактирование события Нажатие правой клавиши мыши с задержкой

Событие **Нажатие правой клавиши мыши с задержкой** возникает, если после щелчка правой клавиши мыши удерживать ее нажатой определенное время.

Для редактирования события:

- 1 Добавьте событие **Нажатие правой клавиши мыши с задержкой** в список **Назначенные события**, как описано в разделе Редактирование событий
- 2 Сделайте двойной щелчок по имени события
- 3 В появившемся окне **Событие с задержкой** задайте интервал задержки для события. Интервал задержки события задается в диапазоне от 1 до 60 секунд
- 4 Для принятия изменений нажмите на кнопку **ОК**, для отмены – на кнопку **Отмена**

### 3.3.2 Присвоить значение

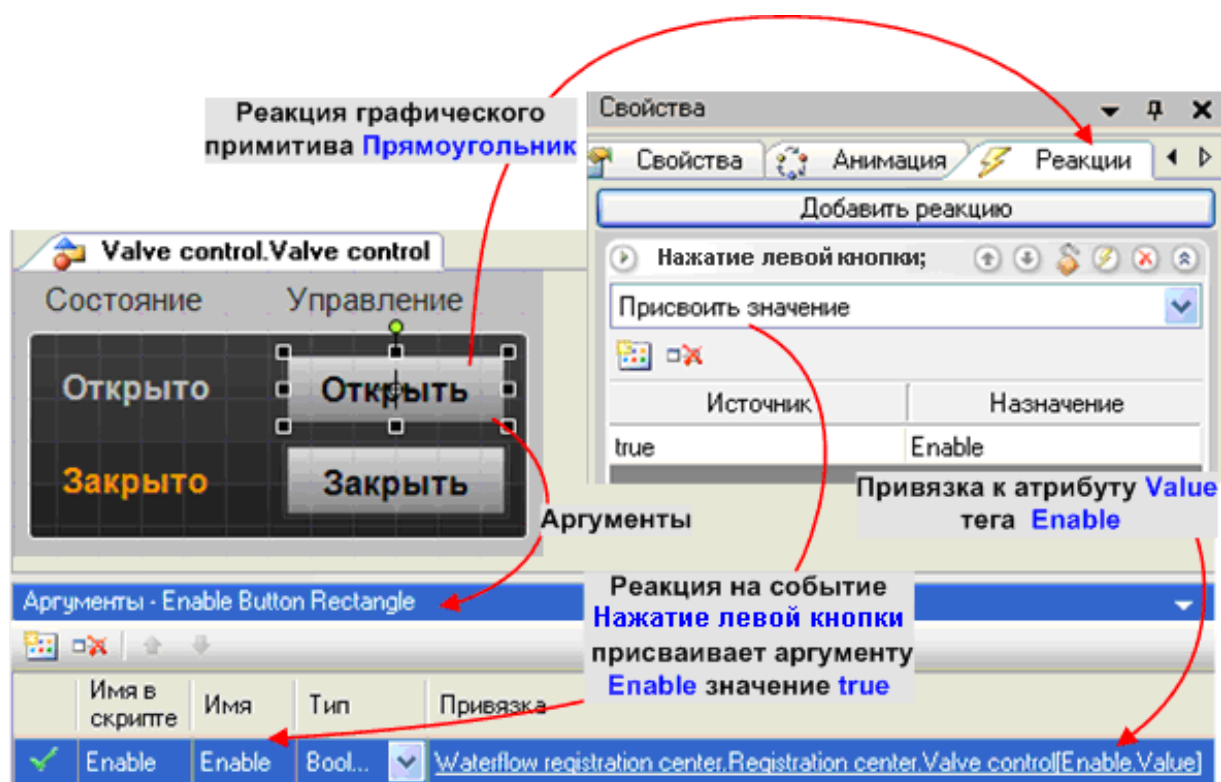
Реакция **Присвоить значение** присваивает значение выражения аргументу.

Одна реакция может присвоить значение нескольким аргументам.

#### Пример

На рисунке показана реакция **Присвоить значение** графического примитива **Прямоугольник** на событие **Нажатие левой кнопки**:

При щелчке по изображению прямоугольника аргументу **Enable** присваивается значение **True**. Это значение «передается» атрибуту тега, к которому привязан данный аргумент.



Для добавления реакции следует:

- 1 Выделить графический примитив
- 2 Добавить аргументы в окно **Аргументы** графического примитива
- 3 В окне **Свойства** перейти на вкладку **Реакции**
- 4 Нажать на кнопку **Добавить реакцию**
- 5 Назначить событие **Нажатие левой кнопки**
- 6 Выбрать реакцию **Присвоить значение**
- 7 Нажать на кнопку **Добавить** на панели реакции
- 8 В столбце **Источник** ввести выражение (на рисунке **true**)
- 9 В выпадающем списке **Назначение** выбрать аргумент, которому будет присвоено значение.

### 3.3.3 Переход

Реакция **Переход** осуществляет открытие выбранного рабочего стола или мнемосхемы.

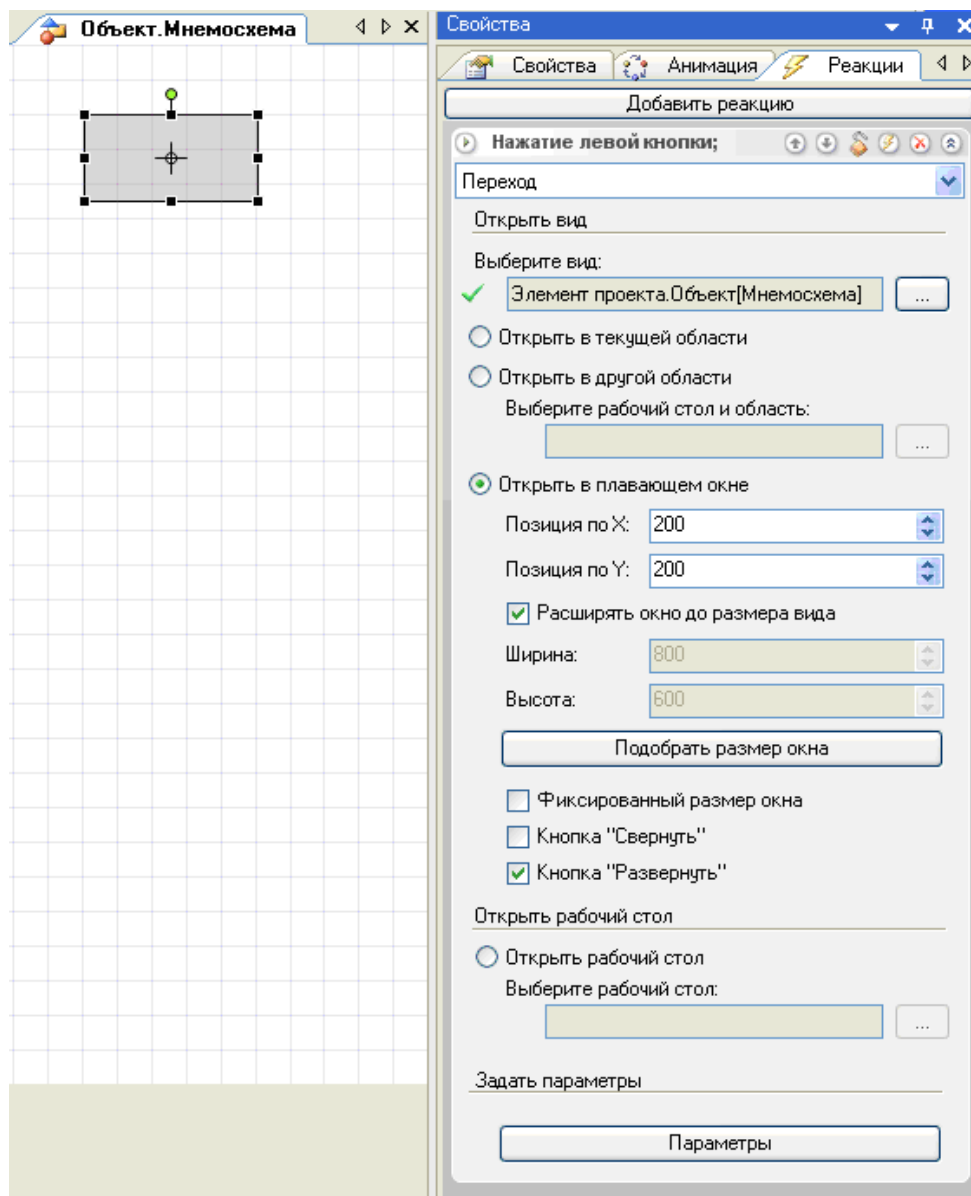
#### **ВНИМАНИЕ!!!**

Если на одно и то же событие назначено несколько реакций, в том числе реакция **Переход**, то она должна быть последней в списке реакций. Реакции, размещённые ниже реакции **Переход**, выполнены не будут.

Для добавления реакции следует:

- 1 Выделить графический примитив
- 2 В окне **Свойства** перейти на вкладку **Реакции**
- 3 Нажать на кнопку **Добавить реакцию**

- 4 Назначить событие
- 5 Выбрать реакцию **Переход**
- 6 На странице свойств выбрать **Открыть вид** или **Открыть рабочий стол**
- 7 Настроить свойства перехода.

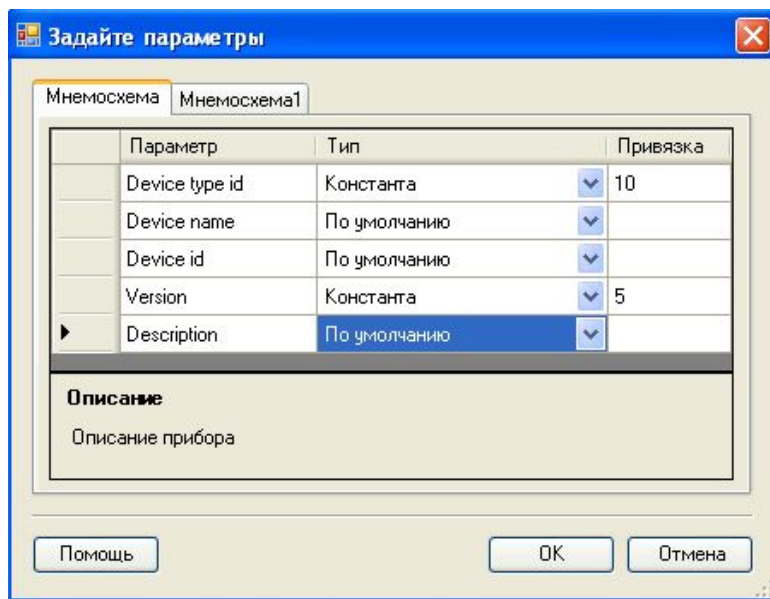


- 8 Настроить параметры

### Редактирование параметров

Для редактирования значений параметров мнемосхем рабочего стола нажмите на кнопку **Параметры**.

Если в настройках задается ссылка на рабочий стол, который содержит более одной мнемосхемы, то окно настроек параметров – **Задать параметры** – открывается с несколькими вкладками, по одной на каждую мнемосхему.



Тип и Привязка параметра могут иметь следующие значения:

- **По умолчанию** – значение по умолчанию
- **Константа** – строковая или числовая константа
- **Аргумент** – имя аргумента из списка аргументов мнемосхемы
- **Поле источника данных** – имя колонки источника данных.

### ВНИМАНИЕ!

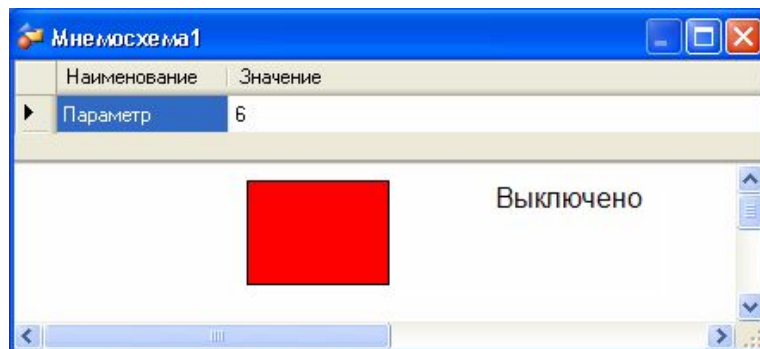
При настройке перехода на шаблон отчёта, необходимо учитывать, что параметры отчёта не могут быть следующих типов: Byte, Decimal, Double, Int16, Int64 и Object!

Для параметров с типами **По умолчанию** и **Поле источника данных** значение привязки не может задаваться пользователем.

У параметра с типом **Константа** в поле ввода **Привязка** может быть введено только значение, соответствующее типу параметра, то есть нельзя, например, ввести текстовую строку в параметр целочисленного типа.

Тип параметра **Поле источника данных** имеет значение только для примитивов **Таблица** и **Табличный тренд**. В момент перехода (в режиме исполнения) значение, находящееся на пересечении данной колонки и выбранной строки, будет передано в данный параметр.

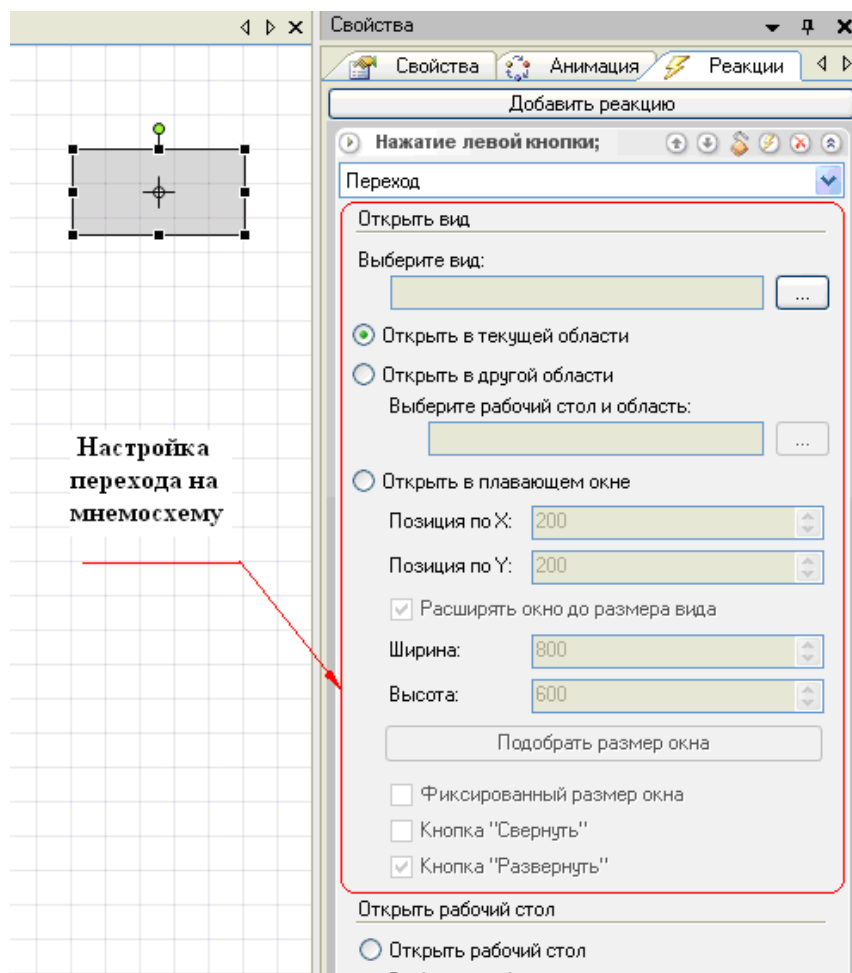
В режиме исполнения при переходе на рабочий стол, мнемосхемы которого имеют параметры, автоматически открывается окно редактирования значений параметров.




## 3.3.3.1 Переход на мнемосхему

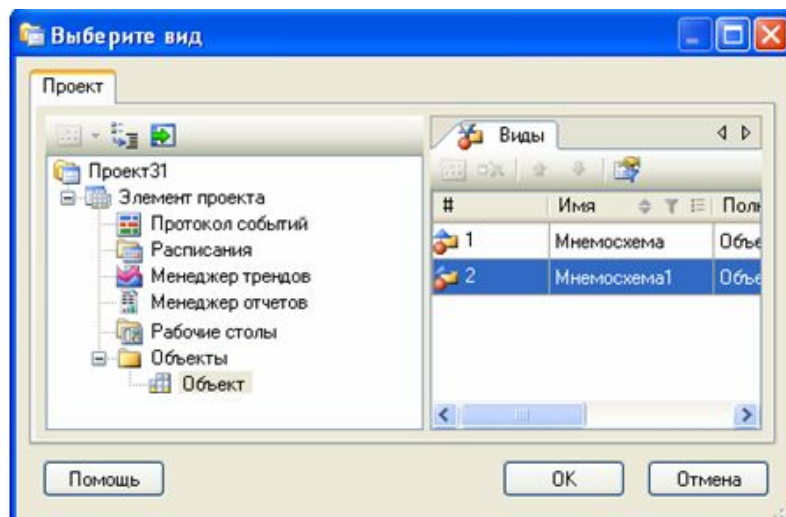
## Как настроить параметры перехода?

Для настройки параметров перехода на мнемосхему предназначена верхняя половина страницы настроек.



## Выбор мнемосхемы

Для назначения перехода в поле **Выберите вид** нажмите на кнопку  и в появившемся окне **Выберите вид** задайте требуемую мнемосхему.




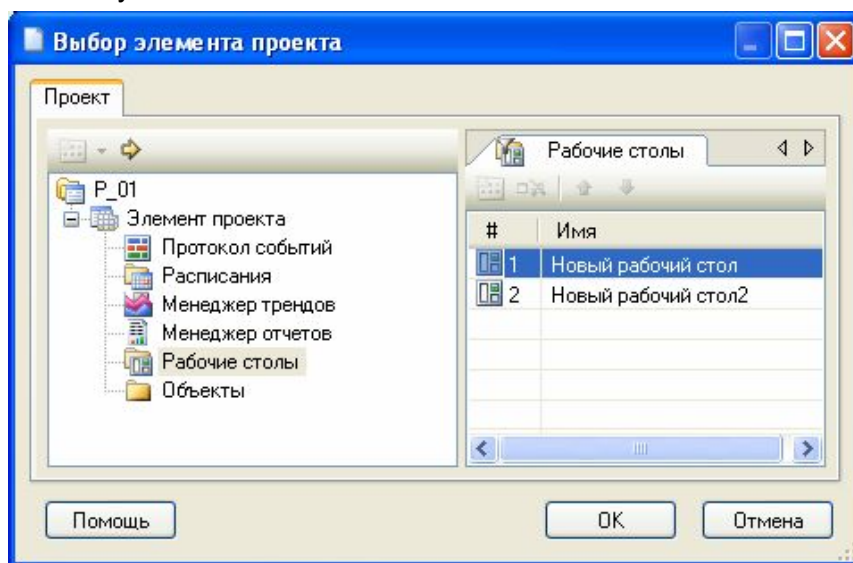


## Выбор области рабочего стола

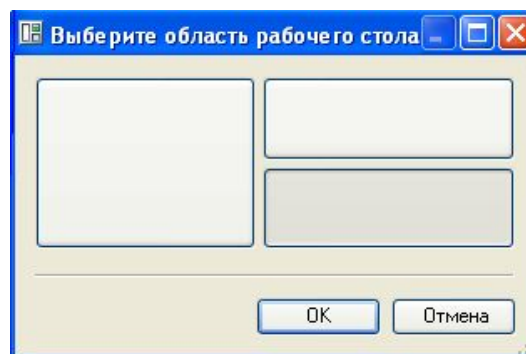
После выбора мнемосхемы необходимо выбрать область рабочего стола, в которой мнемосхема будет открыта. Существует три варианта открытия мнемосхемы:

- **Открыть в текущей области.** Выбранная мнемосхема откроется в той же области, что и примитив, на который назначена реакция **Переход**
- **Открыть в другой области.** Для этого следует:

- 1 Нажать на кнопку  в окне свойств
- 2 В появившемся окне **Выбор элемента проекта** выбрать рабочий стол и нажать на кнопку **ОК**.



- 3 В появившемся окне **Выберите область рабочего стола** указать требуемую область на рабочем столе и нажать на кнопку **ОК**



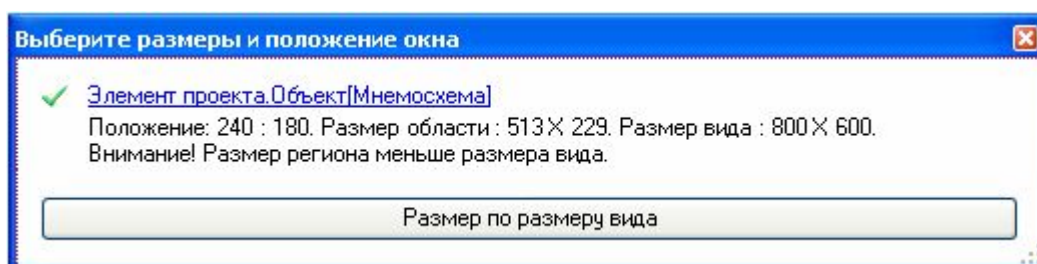
- **Открыть в плавающем окне**


В данном варианте следует задать следующие свойства:

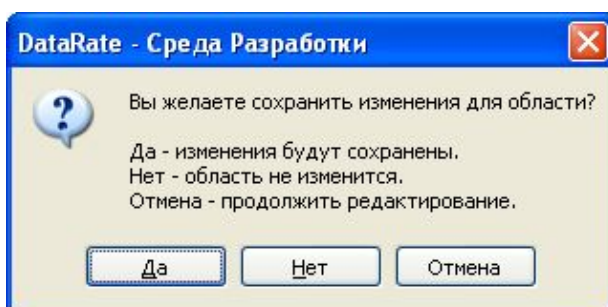
- **Позиция по X, Позиция по Y** – координаты верхнего левого угла плавающего окна относительно левого верхнего угла экрана
- **Расширять окно до размера вида** – установка галочки приведет к тому, что при открытии вида в плавающем окне, размер окна будет подогнан под размер открываемого вида. Если же снять галочку, то в поля **Ширина** и **Высота** следует ввести необходимые размеры открываемого окна в пикселях
- **Фиксированный размер окна** – установка галочки запрещает пользователю изменять размер окна (растягивать или сжимать)
- **Кнопки Свернуть/Развернуть** – установка галочки разрешает сворачивать/разворачивать окно, в котором открывается мнемосхема.

Чтобы упростить подбор позиции окна и его размеров следует использовать кнопку **Подобрать размер окна**. В этом случае следует:

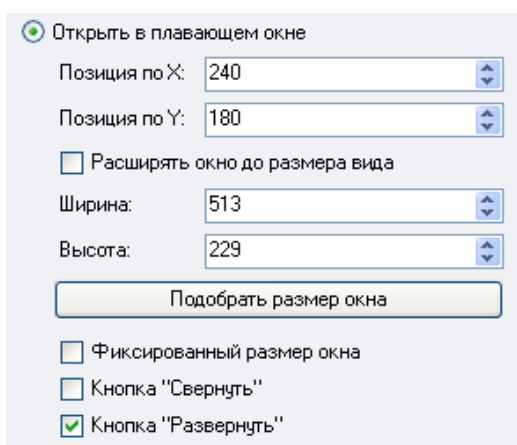
- 1 Нажать на кнопку **Подобрать размер окна**
- 2 В появившемся окне **Выберите размеры и положение окна** нажать на кнопку **Размер по размеру вида**. В результате размер окна будет «приведен» к размеру вида (мнемосхемы).



- 3 Чтобы сохранить изменения нужно нажать на кнопку закрытия окна  и в появившемся окне (**Вы желаете сохранить изменения для области?**) нажать на кнопку **Да**.




Полученные значения настроек отобразятся в соответствующих полях страницы свойств.

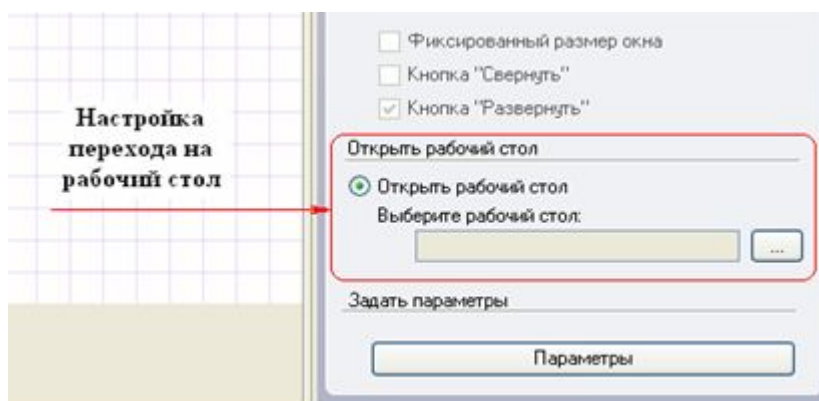


### 3.3.3.2 Переход на рабочий стол

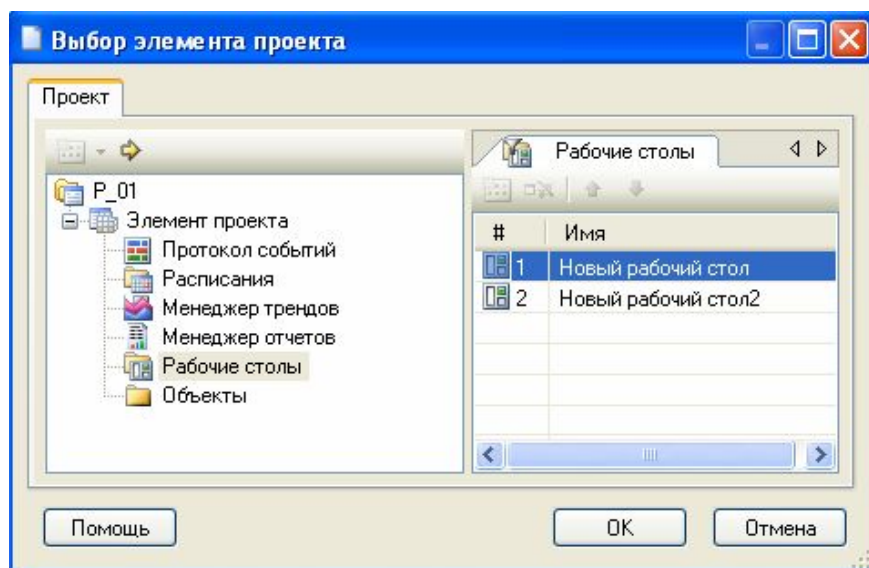
Для настройки параметров перехода на рабочий стол предназначена нижняя половина страницы настроек.

Для выбора рабочего стола следует:

- 1 Нажать на кнопку 



- 2 В появившемся окне **Выбор элемента проекта** выбрать рабочий стол.

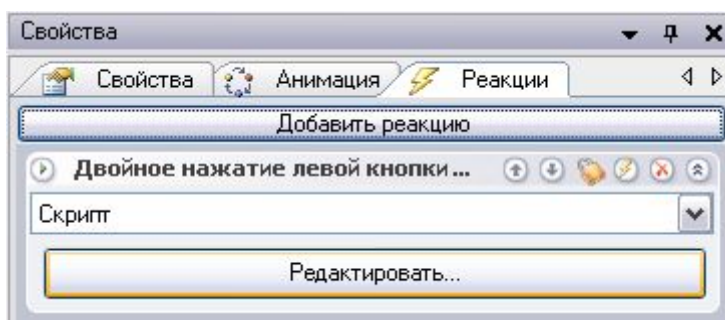


### 3.3.4 Выполнить скрипт

Реакция **Скрипт** предназначена для реакции на события с помощью скрипта.

Для добавления реакции **Скрипт** следует:

- 1 Выделить графический элемент
- 2 Добавить аргументы
- 3 Назначить события реакции
- 4 Выбрать тип реакции **Скрипт**
- 5 Нажать на кнопку **Редактировать...** и в открывшемся редакторе скриптов добавить текст скрипта.

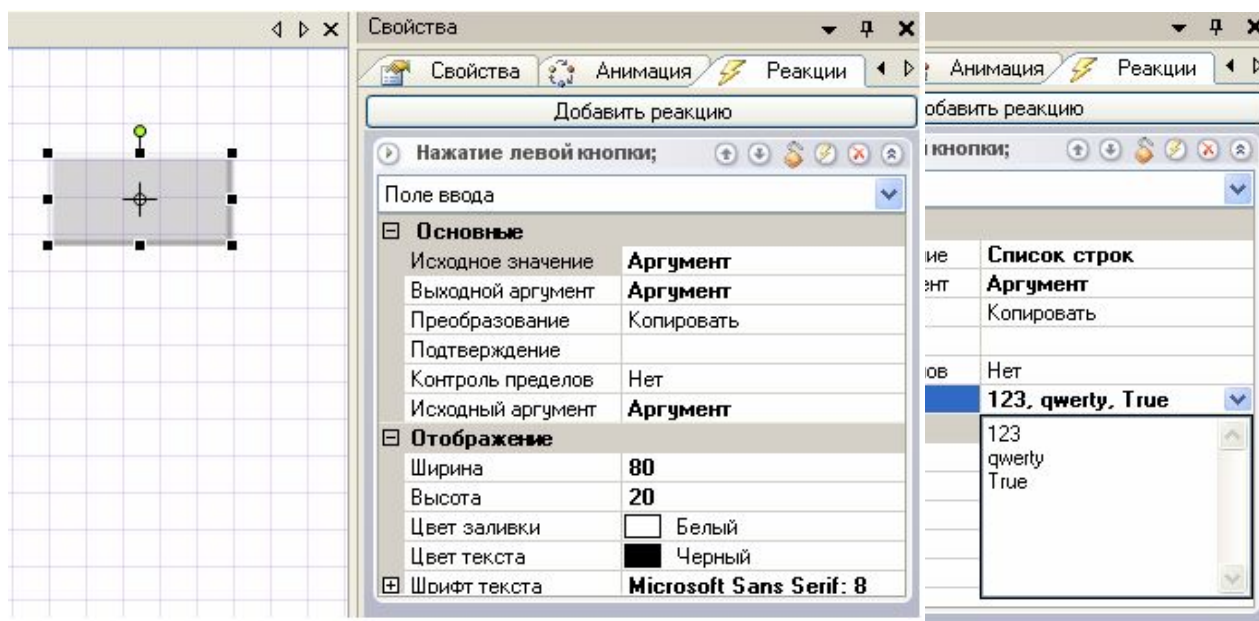


## 3.3.5 Поле ввода

Реакция **Поле ввода** предназначена для ручного ввода значений.

Для создания реакции **Поле ввода** следует:

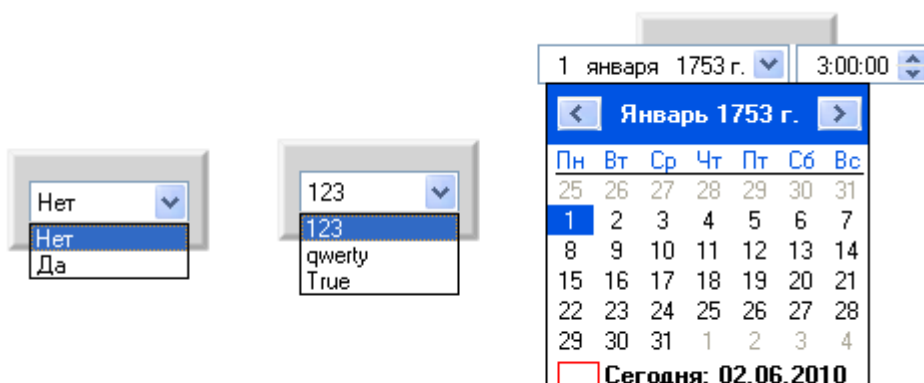
- 1 Выделить графический примитив
- 2 В окне **Свойства** перейти на вкладку **Реакции**
- 3 Нажать на кнопку **Добавить реакцию**
- 4 Назначить событие
- 5 Выбрать реакцию **Поле ввода**



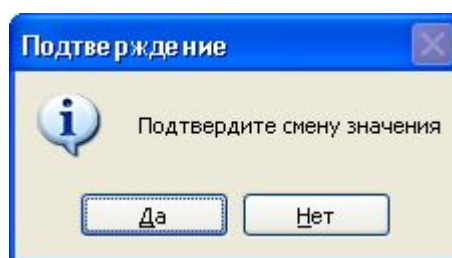
## Свойства

- **Исходное значение** – свойство для задания типа исходного значения. В состоянии **Аргумент** для получения значения будет использоваться свойство **Исходный аргумент**, а в состоянии **Список строк** поле ввода будет отображать поле со списком заданных значений
- **Выходной аргумент** – аргумент, в который будет записываться введенное пользователем значение. Редактируется с помощью выпадающего списка (ComboBox) в котором отображаются все аргументы из окна аргументов

В зависимости от типа данных выходного аргумента (**логическое, дата и время, список**) поле ввода будет представлено соответствующим элементом для ввода значения.



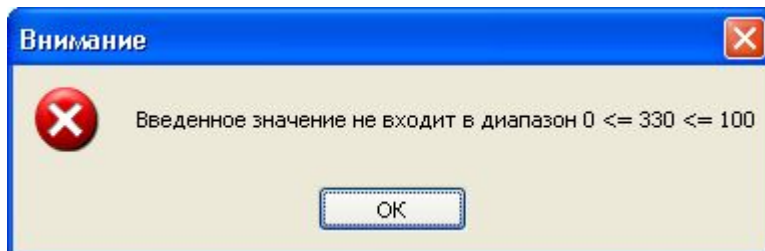
- **Исходный аргумент** – аргумент, из которого будет считано исходное значение для ввода. Аргумент может быть привязан к параметру представления. Тип исходного значения должен быть конвертируемым в тип выходного аргумента
- **Строки** – свойство для задания набора допустимых исходных значений. В редакторе этого свойства каждое из допустимых значений должно быть задано на отдельной строке. Для создания (перехода к) новой строки используется комбинация Ctrl+Enter.
- **Преобразование** – способ записи нового значения в приемник. Возможны следующие значения:
  - **Копировать** – новое значение записывается в приемник
  - **Сложение** – новое значение складывается со значением приемника, сумма записывается в приемник
  - **Вычитать** – новое значение вычитается из значения приемника, разность записывается в приемник
  - **Умножить** – новое значение умножается на значение приемника, результат записывается в приемник
  - **Делить** – значение приемника делится на новое значение, результат записывается в приемник
  - **Инвертировать** – если значение приемника равно нулю, то в приемник записывается новое значение, иначе записывается нуль
  - **Сменить значение** – если значение приемника равно записываемому значению 2, то в приемник записывается новое значение, иначе в приемник записывается записываемое значение 2.
- **Подтверждение** – текст подтверждения. Если текст задан, то после ввода нового значения выводится окно с текстом подтверждения, в котором пользователю предлагается подтвердить ввод нового значения.



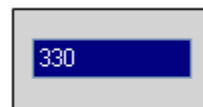
- **Контроль пределов** – признак контроля введенного значения на соответствие диапазону. Если выбрано значение **Да**, то становятся доступными следующие свойства:
  - **Минимальное значение** – нижняя граница диапазона



- **Максимальное значение** – верхняя граница диапазона. Если введенное значение выходит за границы диапазона, то будет выведено сообщение о нарушении границ.



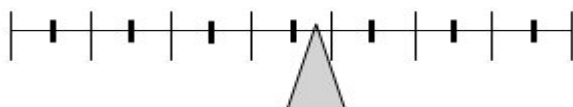
- **Записываемое значение 2** – константа или аргумент, значение которых будет использоваться в преобразовании **Сменить значение**.
- **Высота** – высота поля ввода. Высота может быть скорректирована в зависимости от заданного шрифта.
  - **Ширина** – ширина поля ввода. Эти два свойства определяют размеры поля ввода. Поле ввода всегда выводится по центру примитива.
  - **Шрифт текста** – шрифт текста в поле ввода.
  - **Цвет текста** – цвет текста в поле ввода.
  - **Цвет заливки** – цвет заливки поля ввода.



### 3.3.6 Ползунок

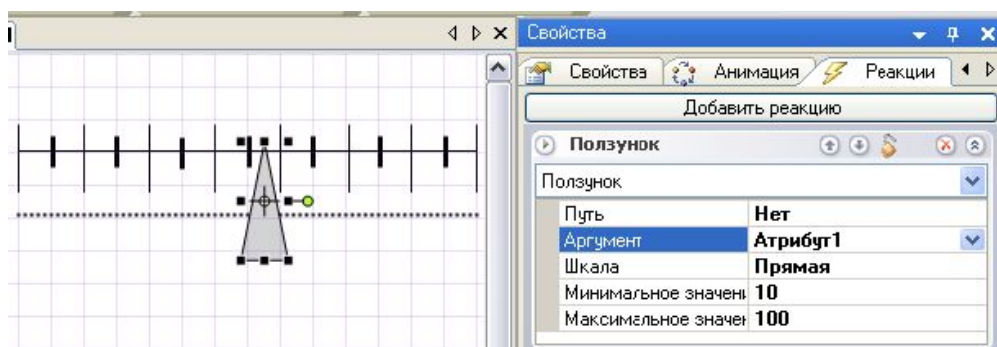
Реакция **Ползунок** добавляет **графическому примитиву** функциональность подобную стандартному компоненту Windows **TrackBar**.

Роль ползунка может выполнять любой **графический примитив**. Для создания стрелок удобнее пользоваться графическим примитивом **Многоугольник**.



Для создания графического примитива с реакцией **Ползунок** следует:

- 1 Поместить на мнемосхему графический примитив, который будет выполнять роль ползунка. Задать ему необходимую форму и размер
- 2 Поместить на мнемосхему графический примитив **Вспомогательная линия**. По этой линии будет перемещаться ползунок. Линия видна только в режиме разработки
- 3 Добавить реакцию **Ползунок**



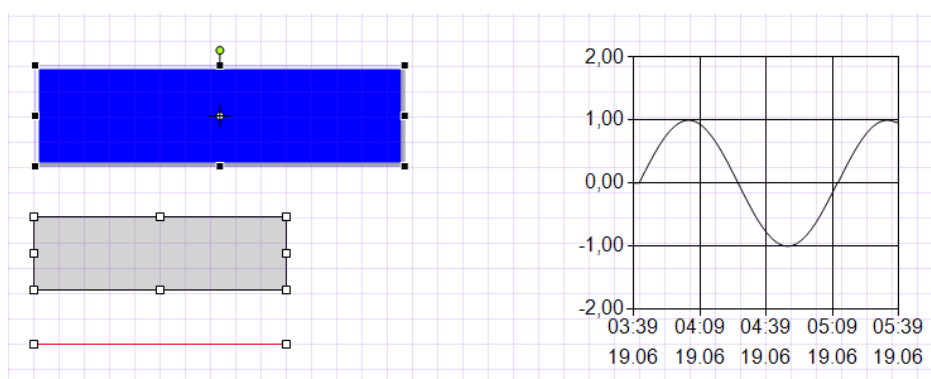
- 4 В свойстве **Путь** указать имя элемента **Вспомогательная линия**
- 5 В свойстве **Привязка** указать аргумент для чтения/записи значений ползунка
- 6 Установить значение свойства **Шкала** (значения **Прямая/Обратная**). Свойство устанавливает на каком конце вспомогательной линии ползунок будет принимать минимальное значение
- 7 В свойстве **Минимальное значение** установите минимальное значение ползунка в крайнем положении
- 8 В свойстве **Максимальное значение** установите максимальное значение ползунка в крайнем положении.


### 3.4 Копирование реакций и анимаций

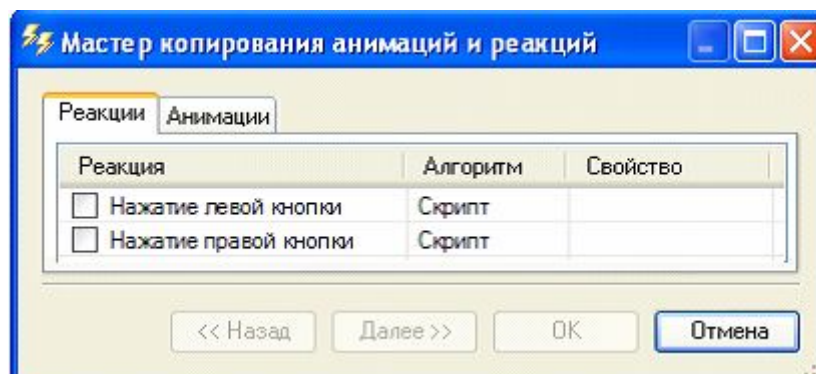
**Мастер копирования реакций и анимаций** позволяет размножить трансформации (реакции или анимации) с одного элемента на несколько других.

Для того чтобы скопировать реакции следует:

- 1 Выделить все элементы, в которые предполагается копирование (на рисунке - линия, серый прямоугольник).  
Последним выделить элемент - источник копирования (сделать это можно выделяя элементы с нажатой клавишей **Shift**). В результате элемент-источник будет выделен как **активный элемент** (на рисунке синий прямоугольник).

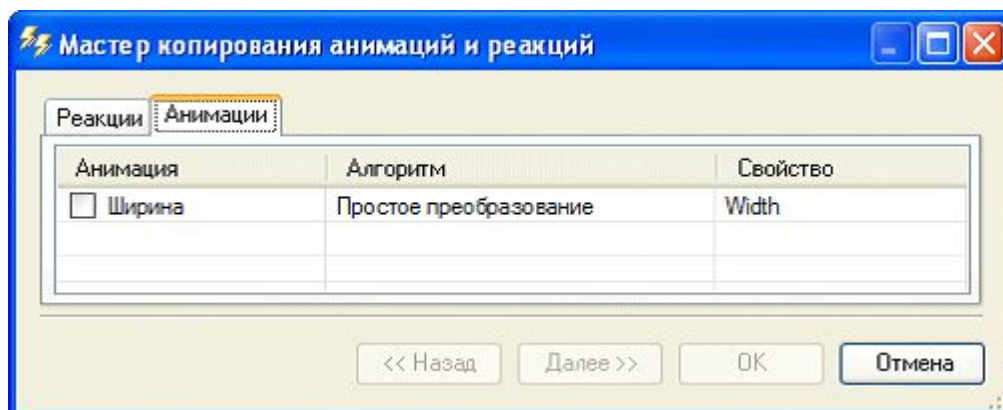


- 2 Нажать на кнопку 
- 3 В появившемся окне **Мастер копирования анимаций и реакций** выбрать реакции и анимации для копирования.  
 Вкладка **Реакции** содержит список реакций и их свойств (имя реакции, алгоритм и свойство). При копировании реакции просто добавляются к реакциям целевого примитива.



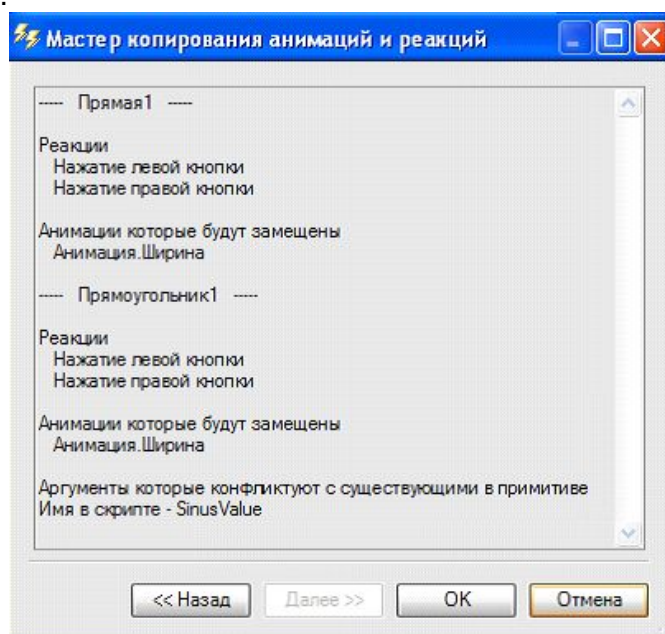
Вкладка **Анимации** отображает список анимаций и их свойства (имя анимации, алгоритм и свойство, к которому применяется анимация).

В список анимаций попадают только те анимации, которые присутствуют во всех выделенных элементах.



4 Нажать на кнопку **Далее>>**

5 Последняя страница мастера отображает суммарную информацию об операции копирования.



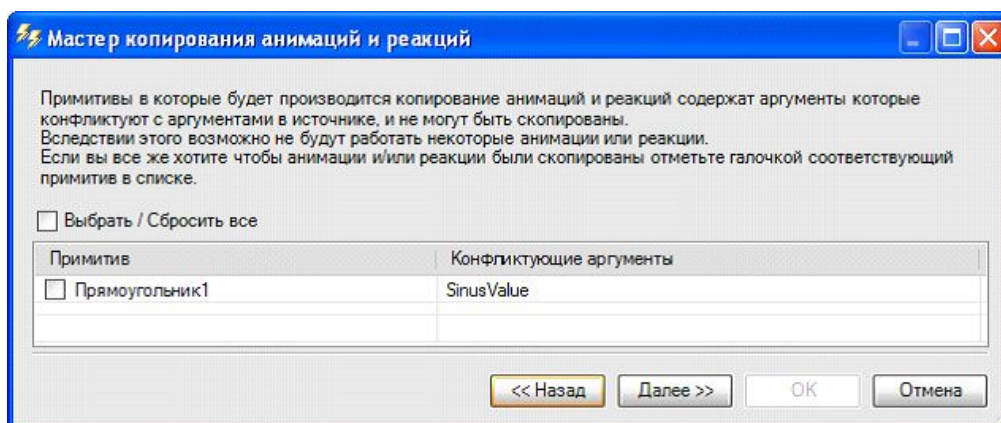
## Конфликты

При копировании трансформаций могут возникнуть конфликты.

### Конфликт аргументов

Аргументы считаются конфликтующими, если у них совпадает имя в скрипте и не совпадают другие параметры (например, тип аргумента).

Если в каком либо элементе присутствует конфликт аргументов, будет показано следующее окно, со списком элементов в которых произошел конфликт аргументов.



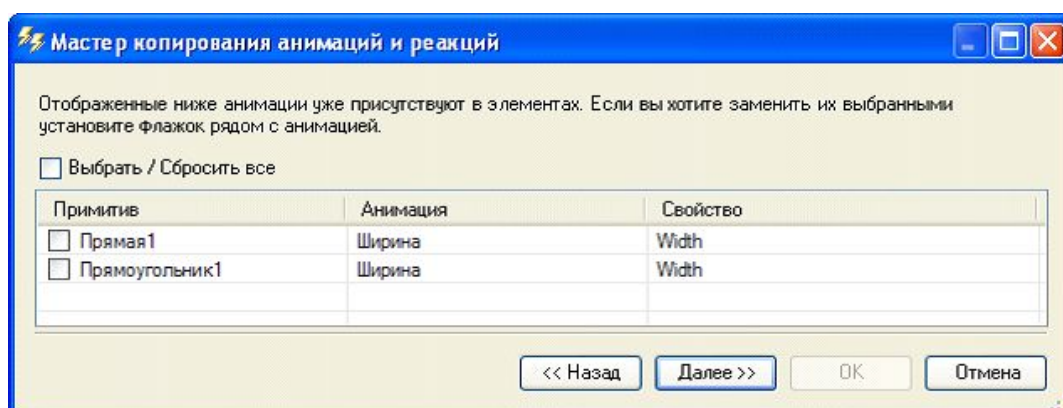
Если все же вы хотите скопировать трансформации, то необходимо выбрать их в списке и нажать кнопку далее.

### **ВНИМАНИЕ!!!**

При копировании "конфликтных" трансформаций возможно, что копии трансформаций не будут работать или будут работать неправильно!

### **Конфликт анимаций**

Анимации считаются конфликтующими, если они совпадают и они не пустые. Если в каком либо элементе присутствуют конфликтующие анимации то будет отображено следующее окно для разрешения конфликтов.



Выбранные анимации будут замещены копируемыми из элемента - источника.

### **ВНИМАНИЕ!!!**

В список конфликтующих анимаций не включаются анимации тех элементов, которые имеют конфликт аргументов и для которых было выбрано копирование в окне разрешения конфликтов аргументов.

## **3.5 Примеры использования реакций и анимаций**

Свойства графических примитивов можно изменять с помощью реакций **Присвоить значение** и **Скрипт**, а также с помощью анимации **Простое преобразование**. Рассмотрим несколько простых примеров использования этих реакций и анимации.

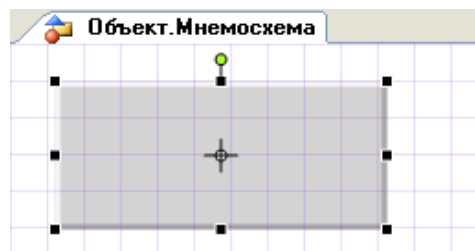
## 3.5.1 Видимость/мигание

**Свойство Видимость/мигание** задается с помощью числовых констант. Описание констант находится в перечислении **VisibleXParam**.

| Название  | Значение                          | Описание                               |
|-----------|-----------------------------------|----------------------------------------|
| None      | 0                                 | Примитив не видим                      |
| Phase1    | 8                                 | Примитив виден во время первой фазы    |
| Phase2    | 4                                 | Примитив виден во время второй фазы    |
| Phase3    | 2                                 | Примитив виден во время третьей фазы   |
| Phase4    | 1                                 | Примитив виден во время четвертой фазы |
| FastBlink | Phase1   Phase3                   | Быстрое мигание                        |
| Blink     | Phase1                            | Мигание                                |
| Always    | Phase1   Phase2   Phase3   Phase4 | Примитив виден постоянно               |

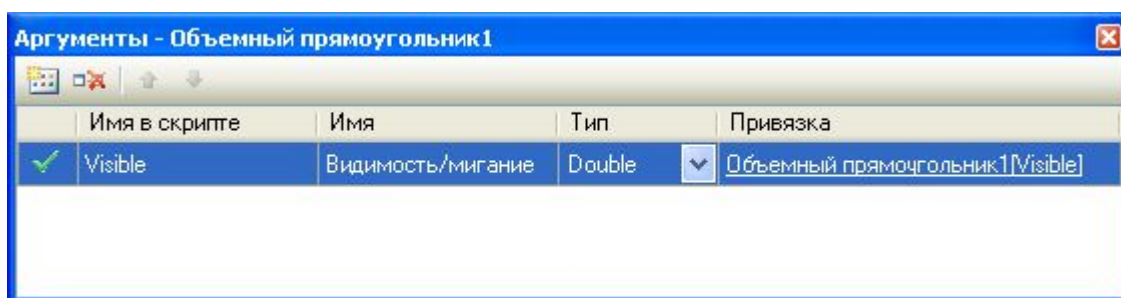
### Пример

Рассмотрим, как с помощью свойства **Видимость/Мигание** и реакции **Присвоить значение** заставить мигать прямоугольник.

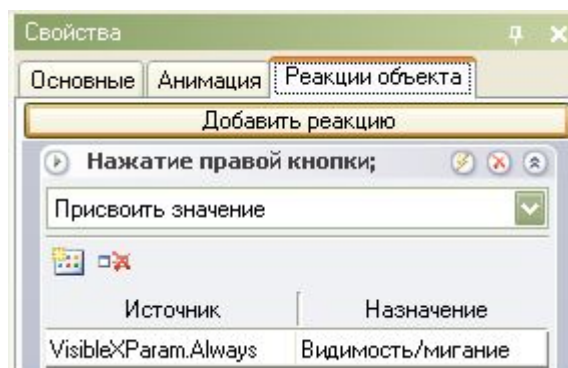


Пусть по нажатию левой кнопки мыши прямоугольник будет мигать, по нажатию правой кнопки мыши прямоугольник перестанет мигать:

- 1 На мнемосхему помещаем прямоугольник.
- 2 В окно аргументов перетаскиваем свойство **Видимость/мигание**.

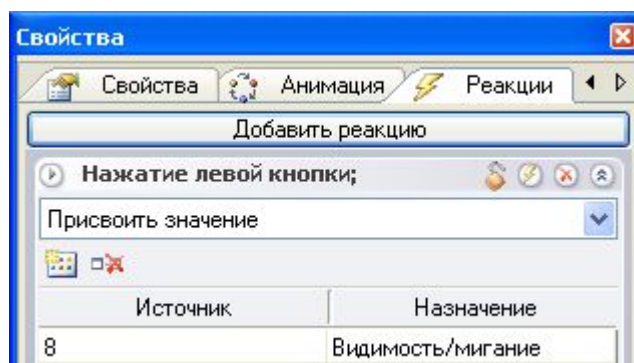


- 3 Прямоугольнику назначаем две реакции:
  - Реакция на нажатие правой кнопки мыши – прямоугольник виден всегда.





- Реакция на нажатие левой кнопки мыши – прямоугольник виден только во время первой фазы. Это создает эффект редкого мигания.



### 3.5.2 Управление открытием/закрытием мнемосхем

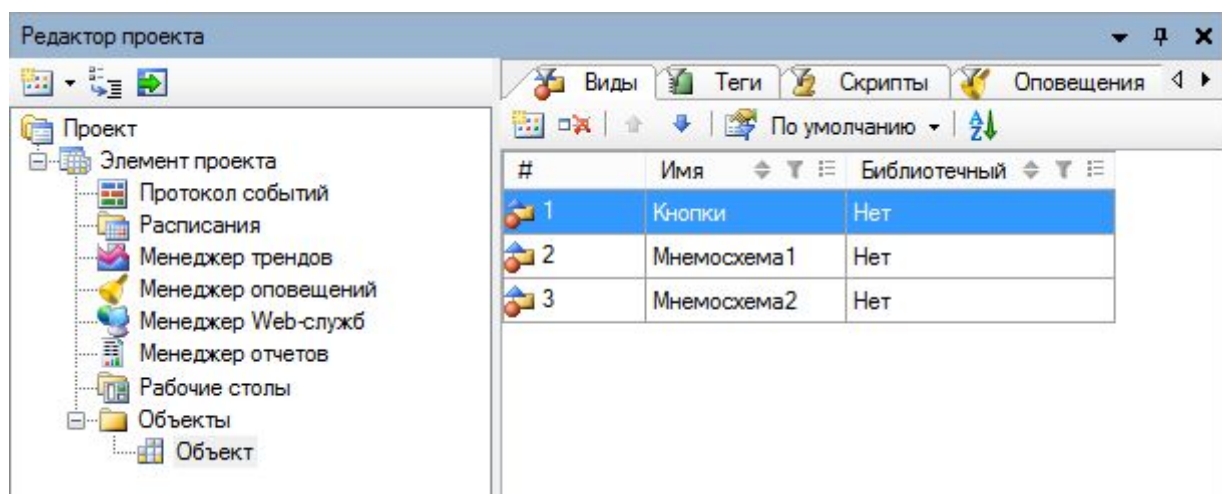
Рассмотрим использование анимации и реакции **Переход** для открытия и закрытия мнемосхем.

#### Пример

Демонстрационный проект будет содержать рабочий стол с двумя областями: область кнопок и область мнемосхем. При нажатии в первой области на кнопку **Кнопка1** во второй области рабочего стола будет открываться мнемосхема с соответствующей надписью – **Мнемосхема 1**; при нажатии в первой области на кнопку **Кнопка2** во второй области рабочего стола будет открываться мнемосхема – **Мнемосхема 2**. При этом активная кнопка будет отображаться нажатой, а другая кнопка – отжатой.

Для демонстрации данного примера следует:

- 1 Создать проект, элемент проекта и объект **Объект**
- 2 Для объекта **Объект** создать три мнемосхемы (вида): **Кнопки**, **Мнемосхема1** и **Мнемосхема2**

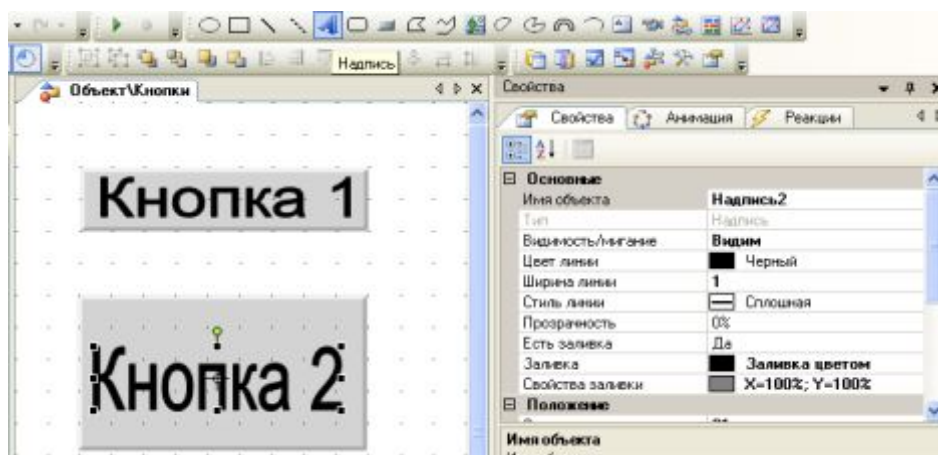


#### Создание мнемосхемы Кнопки

Для создания мнемосхемы **Кнопки** следует:

- 1 Добавить в проект для объекта **Объект** мнемосхему (вид) с именем **Кнопки**

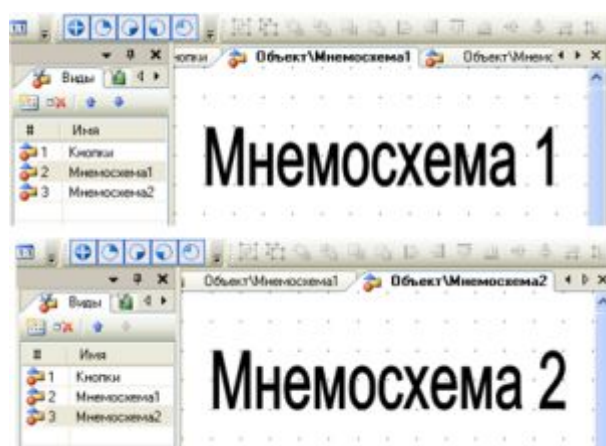
- 2 Разместить на мнемосхеме **Кнопки** графические примитивы **Объемный прямоугольник** (кнопки)
- 3 Добавить графические примитивы **Надпись** на каждый примитив **Объемный прямоугольник** (кнопку): на верхнюю – надпись **Кнопка 1**, на нижнюю – **Кнопка 2**.



### Создание мнемосхем Мнемосхема1 и Мнемосхема2

Для создания мнемосхем следует:

- 1 Добавить в проект для объекта **Объект** мнемосхемы (виды) с именем **Мнемосхема1** и **Мнемосхема2**
- 2 Разместить на мнемосхеме **Мнемосхема1** надпись **Мнемосхема 1**, на **Мнемосхема2** – надпись **Мнемосхема2**



### Настройка рабочего стола

Для отображения мнемосхем следует создать рабочий стол – **Новый рабочий стол** – с двумя областями.

В левой области установить ссылку на мнемосхему **Кнопки**.

### Настройка переходов

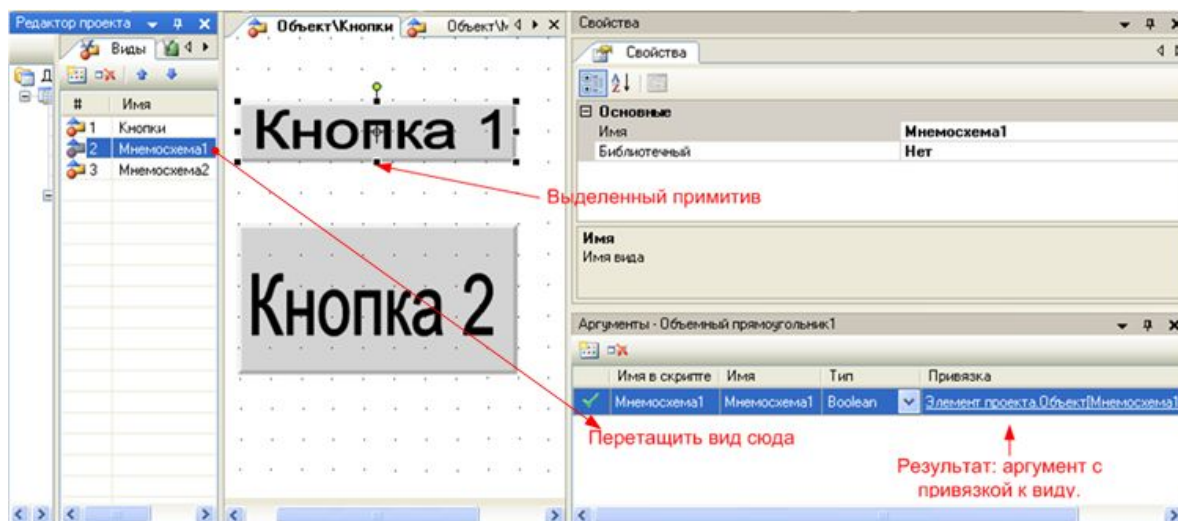
Следующим этапом разработки проекта будет настройка анимации и реакции **Переход** для графических примитивов **Кнопка1**, **Кнопка2** (**объемный прямоугольник**). Необходимо научить наши кнопки трем вещам:

- Реагировать на нажатие левой кнопки мыши
- Отображаться нажатой при нажатии левой кнопки мыши на кнопке и отжатой – в противном случае

- Открывать соответствующую мнемосхему, т.е. выполнять переход на соответствующий вид.

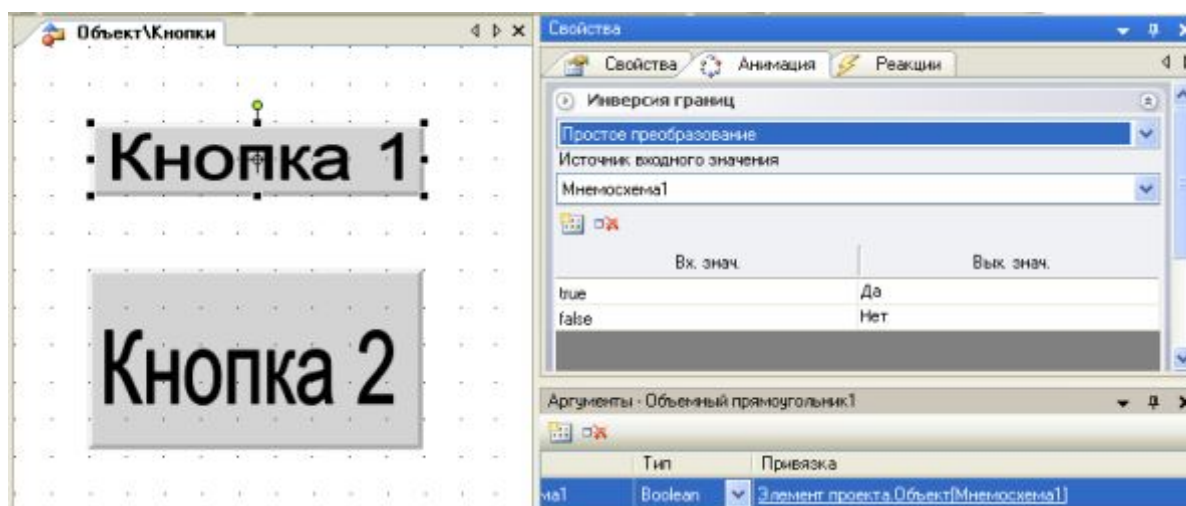
Для этого следует:

- Для каждой кнопки создать аргумент – перетащить в окно **Аргументы** имя соответствующей мнемосхемы (имя вида)





- Для отображения состояния кнопки – нажата/отжата – настроить анимацию **Инверсия границ**:

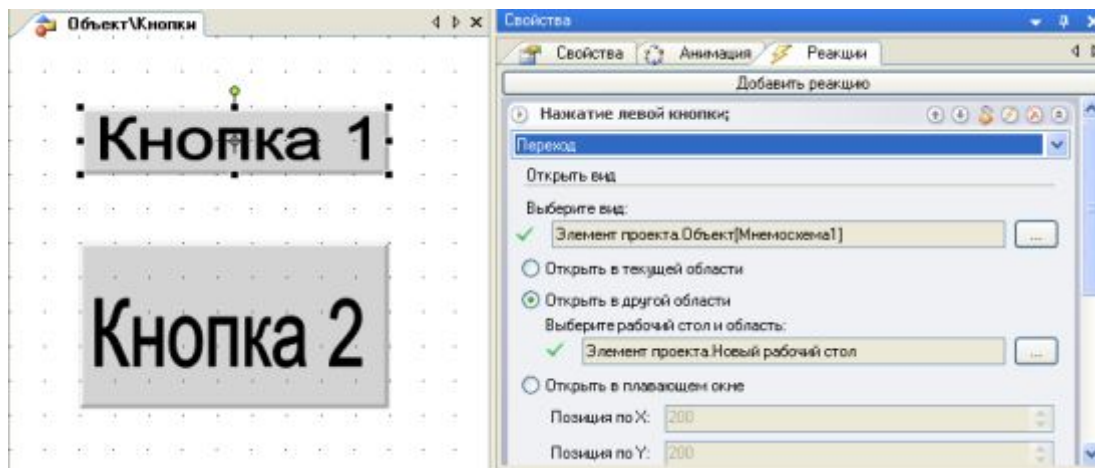
- Выделить графический примитив **Объемный прямоугольник**, например, **Кнопка1**
- В окне **Свойства** выбрать вкладку **Анимация** и свойство **Инверсия границ**
- Выбрать тип анимации **Простое преобразование** и задать правила преобразования – значение аргумента **Мнемосхема1** будет равно **Да**, когда **Мнемосхема1** открыта, и **Нет** – когда закрыта.



Аналогичные действия выполнить и для примитива **Кнопка2**.


- Назначить переход на мнемосхему при щелчке левой клавиши мыши по кнопке:
  - Выделить графический примитив **Объемный прямоугольник**, например, **Кнопка1**

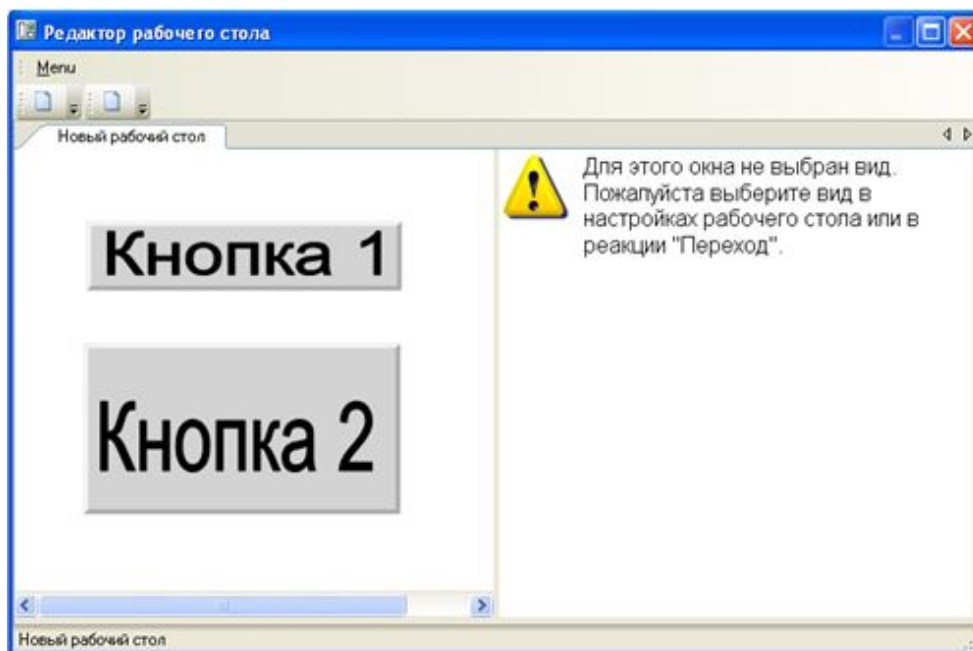
- В окне **Свойства** выбрать вкладку **Реакции** и нажать на кнопку **Добавить реакцию**
- Нажать на кнопку  и в появившемся окне **Редактор событий** выбрать из списка событие **Нажатие левой кнопки**
- Нажать на кнопку , выбрать тип реакции **Переход** и в раскрывшейся странице настроить свойства перехода



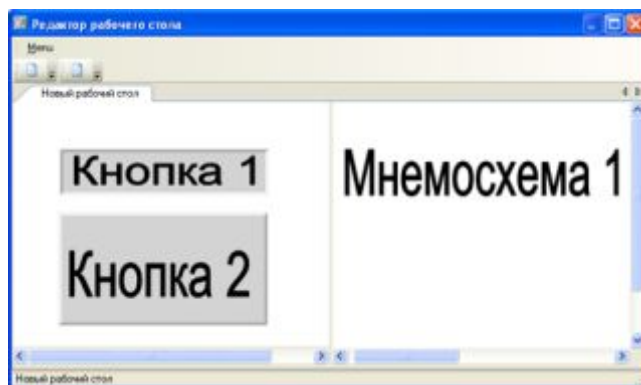
Аналогичные действия выполнить и для примитива **Кнопка2**.

### Демонстрация работы

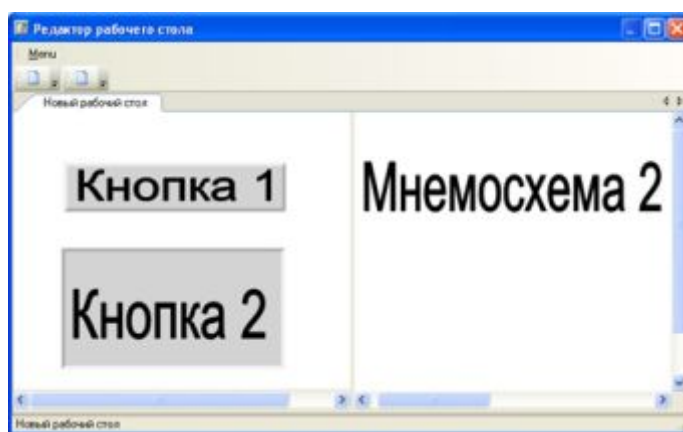
- 1 Закрыть все открытые виды проекта
- 2 Запустить проект на имитацию (кнопка  на панели инструментов) и открыть рабочий стол.



- 3 Щелкнуть по кнопке **Кнопка 1**. В правой области открылась **Мнемосхема1**. **Кнопка 1** отображается нажатой



- 4 Щелкнуть по кнопке **Кнопка2**. **Кнопка2** отображается нажатой, а **Кнопка1** – отжатой. В правой части рабочего стола открывается мнемосхема с надписью **Мнемосхема 2**.



### 3.5.3 Цвет

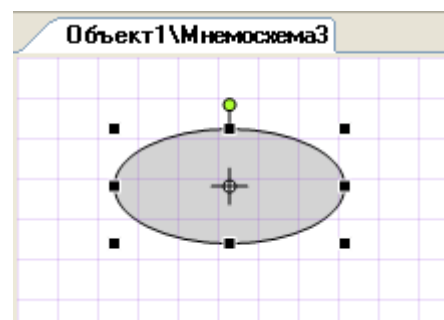
**Свойство Цвет** задается с помощью объекта **Color**.

Объект **Color** имеет несколько predefined констант для задания цвета, а также может задавать цвет с помощью значений RGB (красный, зеленый, синий).

#### Пример

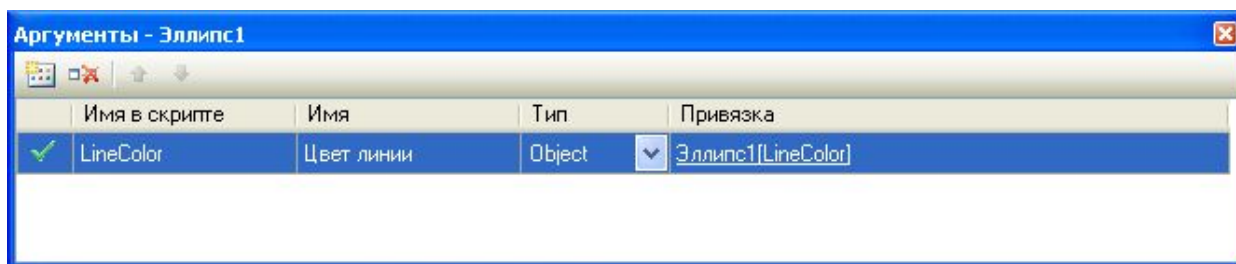
Путь по щелчку правой кнопкой мыши цвет линии овала становится зеленым, а по щелчку левой кнопкой мыши – красным:

- 1 На мнемосхему помещаем овал.



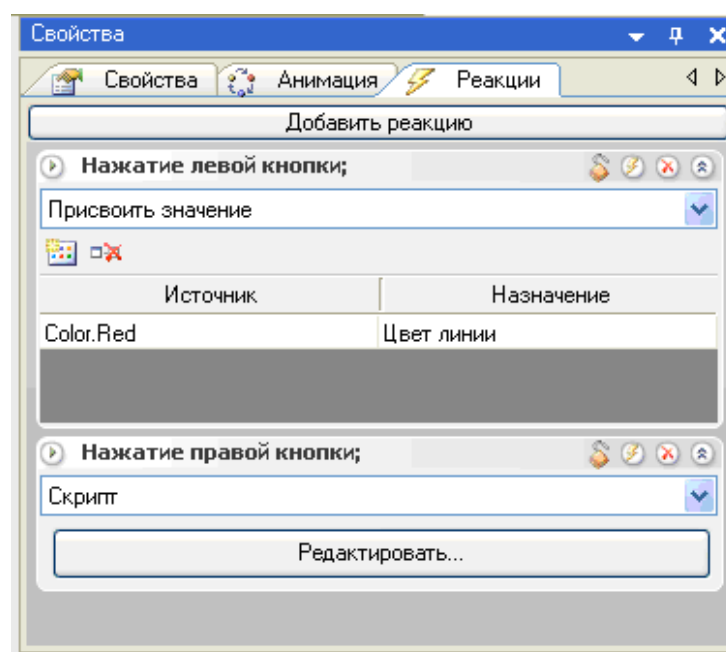


- 2 В окно аргументов перетаскиваем свойство **Цвет линии**.

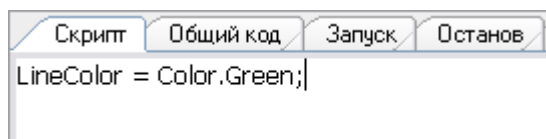


- 3 Создаем две реакции:

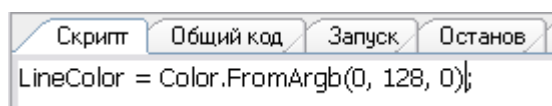
- Нажатие левой кнопки мыши (реакция **Присвоить значение**) меняет цвет линии на красный.



- Нажатие правой кнопки (реакция **Скрипт**) меняет цвет линии на зеленый.



или можно так



- 4 Запускаем проект на имитацию (кнопка  на панели инструментов):

- Щелчок левой кнопки мыши по овалу устанавливает красный цвет линии.



- Щелчок правой кнопки мыши по овалу устанавливает зеленый цвет линии.



В этом примере продемонстрировали два способа воздействия на свойства примитива – с помощью реакции **Присвоить значение** и реакции **Скрипт**.

### 3.5.4 Заливка

Свойство **Заливка** задается с помощью объекта **BrushX**.

#### Методы объекта BrushX

- `static BrushX CreateSolid(Color color)` – метод создает кисть сплошной заливки.

Аргументы:

- `color` – цвет, которым будет заливаться примитив.

#### Пример

//создание сплошной кисти красного цвета

```
BrushX brush = BrushX.CreateSolid(Color.Red);
```

- `static BrushX CreateHatch(HatchStyle hatchStyle, Color foreColor, Color backColor)` – метод создает кисть с заливкой в виде решетки.

Аргументы:

- `hatchStyle` – тип решетки
- `foreColor` – цвет решетки
- `backColor` – цвет фона.

#### Пример

//создание кисти с черным цветом решетки и серым цветом фона

```
BrushX brush = BrushX.CreateHatch(HatchStyle.BackwardDiagonal, Color.Black, Color.Gray);
```

- `static BrushX CreateGradient(GradientBrushInfo info)` – метод создает кисть с градиентной заливкой.

Аргументы:

`info` – свойства градиента.

#### Пример

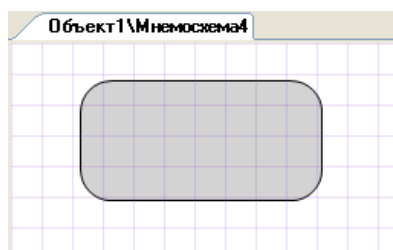
//создание кисти с градиентом

```
BrushX brush = BrushX.CreateGradient(new GradientBrushInfo(Color.Red, Color.Green);
```

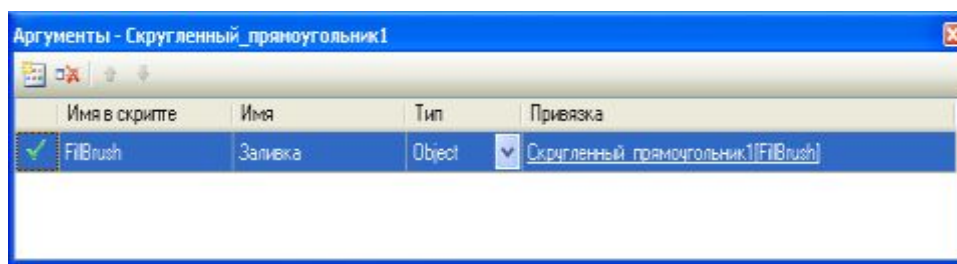
**Пример использования объекта Brush**

Для того чтобы изменить заливку графического примитива **Скругленный прямоугольник** следует:

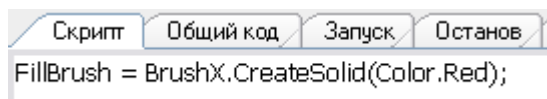
- 1 Разместить на мнемосхеме скругленный прямоугольник



- 2 В окно аргументов помещаем свойство **Заливка**.



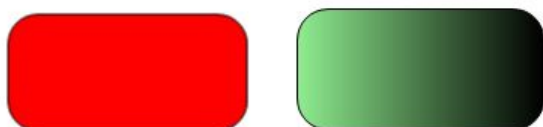
- 3 Создаем реакцию **Скрипт** на событие нажатие левой кнопки мыши. При нажатии на левую кнопку мыши прямоугольник будет становиться красным.



- 4 Создаем реакцию **Скрипт** на событие нажатие правой кнопки мыши. При нажатии на правую кнопку мыши прямоугольник будет заливаться градиентом от зеленого до черного.



















































- 5 Запускаем проект на имитацию. Вот что получилось при нажатии на кнопку мыши: на левую кнопку и на правую кнопку



## 3.5.4.1 HatchStyle

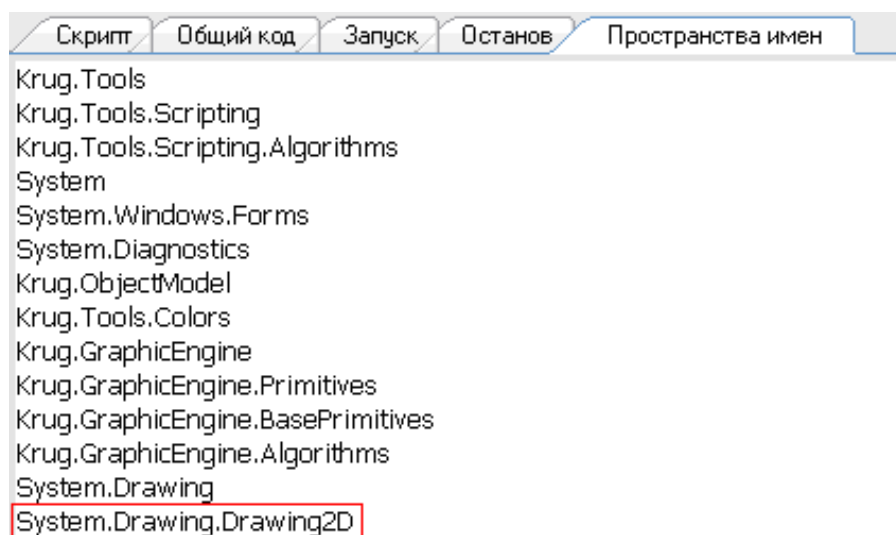
Объект **HatchStyle** представляет собой перечисление констант, определяющих способ штриховки.

| Название               | Штриховка                                                                           | Название             | Штриховка                                                                             |
|------------------------|-------------------------------------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------|
| Horizontal             |    | Vertical             |    |
| ForwardDiagonal        |    | BackwardDiagonal     |    |
| Cross                  |    | DiagonalCross        |    |
| Percent05              |    | Percent10            |    |
| Percent20              |    | Percent25            |    |
| Percent30              |    | Percent40            |    |
| Percent50              |    | Percent60            |    |
| Percent70              |    | Percent75            |    |
| Percent80              |    | Percent90            |    |
| LightDownwardDiagonal  |   | LightUpwardDiagonal  |   |
| DarkDownwardDiagonal   |  | DarkUpwardDiagonal   |  |
| WideDownwardDiagonal   |  | WideUpwardDiagonal   |  |
| LightVertical          |  | LightHorizontal      |  |
| NarrowVertical         |  | NarrowHorizontal     |  |
| DarkVertical           |  | DarkHorizontal       |  |
| DashedDownwardDiagonal |  | DashedUpwardDiagonal |  |
| DashedHorizontal       |  | DashedVertical       |  |
| SmallConfetti          |  | LargeConfetti        |  |
| ZigZag                 |  | Wave                 |  |
| DiagonalBrick          |  | HorizontalBrick      |  |
| Weave                  |  | Plaid                |  |
| Divot                  |  | DottedGrid           |  |
| DottedDiamond          |  | Shingle              |  |
| Trellis                |  | Sphere               |  |

## 3.5.4.2 GradientBrushInfo

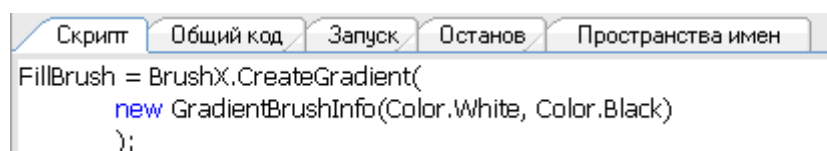
**ВНИМАНИЕ!!!**

Чтобы использовать режимы заливки нужно добавить ссылку на сборку **System.Drawing.Drawing2D**.

**Двухцветная градиентная заливка**

Объект **GradientBrushInfo** представляет собой структуру, описывающую способ градиентной заливки.

**GradientBrushInfo(Color color1, Color color2)** – конструктор создает заливку от цвета **color1** к цвету **color2** слева направо.

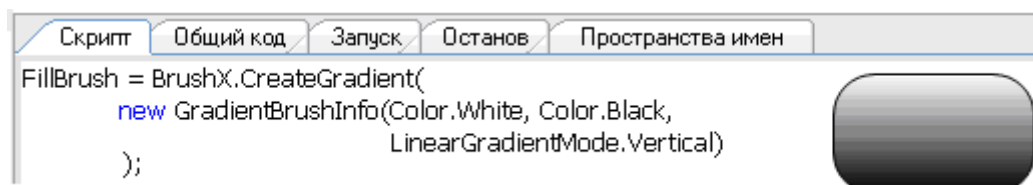
**Пример**

Объект **LinearGradientMode** представляет собой перечисление констант, управляющих направлением заливки.

|                         |                                                                     |
|-------------------------|---------------------------------------------------------------------|
| <b>BackwardDiagonal</b> | Направление заливки от правого верхнего угла к левому нижнему углу. |
| <b>ForwardDiagonal</b>  | Направление заливки от левого верхнего угла к правому нижнему углу. |
| <b>Horizontal</b>       | Направление заливки слева направо.                                  |
| <b>Vertical</b>         | Направление заливки сверху вниз.                                    |

**GradientBrushInfo(Color color1, Color color2, LinearGradientMode mode)** – конструктор создает заливку от цвета **color1** к цвету **color2**, направление заливки определяется параметром **mode**



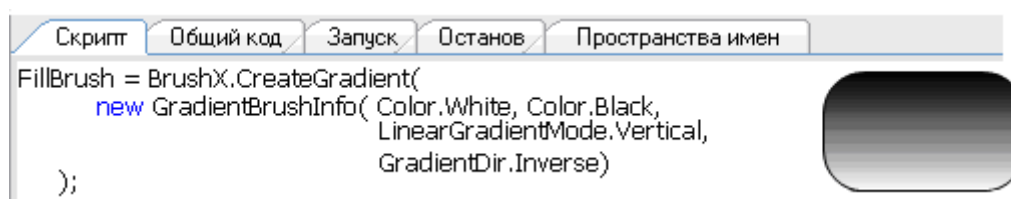
**Пример**

Объект **GradientDir** представляет собой перечисление констант, управляющих направление градиента.

**Normal** Заливка от цвета **color1** к цвету **color2**

**Inverse** Заливка от цвета **color2** к цвету **color1**. Практически означает замену цветов местами.

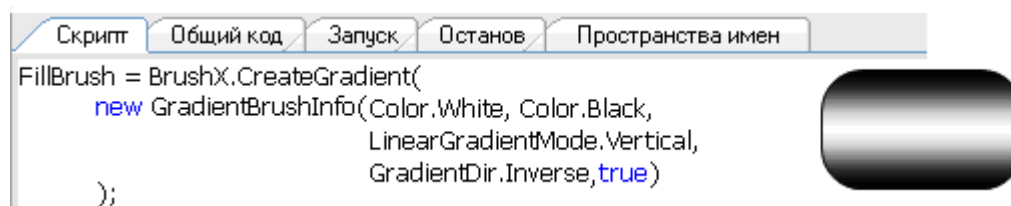
`GradientBrushInfo(Color color1, Color color2, LinearGradientMode mode, GradientDir direction)` – конструктор создает заливку от цвета **color1** к цвету **color2**, направление заливки определяется параметром **mode**. Параметр **direction** позволяет инвертировать направление заливки.

**Пример**

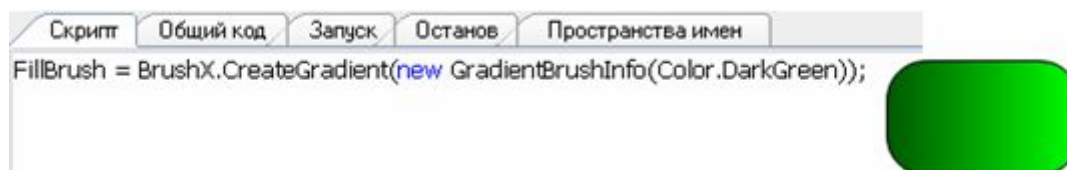
Сравните результат работы данного скрипта с результатом предыдущего примера

Параметр **fromCenter** задает способ заливки от центра.

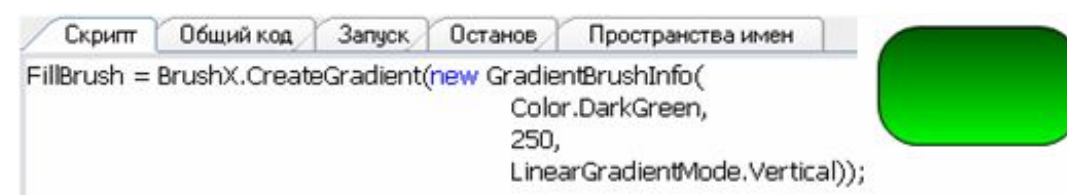
`GradientBrushInfo(Color color1, Color color2, LinearGradientMode mode, GradientDir direction, bool fromCenter)` – конструктор создает заливку от цвета **color1** к цвету **color2**, направление заливки определяется параметром **mode**. Параметр **direction** позволяет инвертировать направление заливки. Параметр **fromCenter** задает способ заливки: **true** – заливка от центра, **false** – заливка от краев.

**Пример****Одноцветная градиентная заливка**

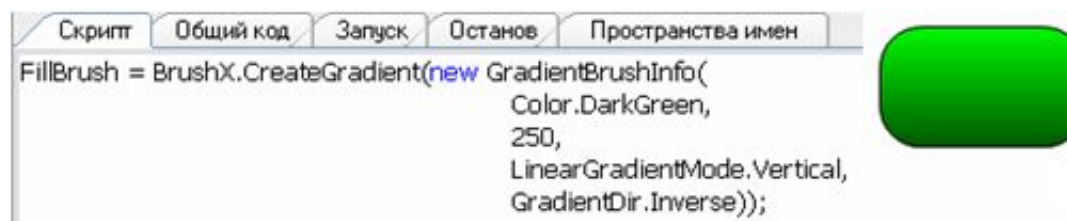
`GradientBrushInfo(Color color1)` - конструктор создает одноцветный градиент цветом **color1**.

**Пример**

`GradientBrushInfo(Color color1, int brightness, LinearGradientMode mode)` – конструктор создает одноцветный градиент цветом `color1`. Параметр `brightness` определяет яркость цвета. Параметр `mode` задает режим заливки.

**Пример**

`GradientBrushInfo(Color color1, int brightness, LinearGradientMode mode, GradientDir direction)` - конструктор создает одноцветный градиент цветом `color1`. Параметр `brightness` определяет яркость цвета. Параметр `mode` задает режим заливки. Параметр `direction` позволяет инвертировать направление заливки.

**Пример**

`GradientBrushInfo(Color color1, int brightness, LinearGradientMode mode, GradientDir direction, bool fromCenter)` – конструктор создает одноцветный градиент цветом `color1`. Параметр `brightness` определяет яркость цвета. Параметр `mode` задает режим заливки. Параметр `direction` позволяет инвертировать направление заливки. Параметр `fromCenter` задает способ заливки: `true` - заливка от центра, `false` - заливка от краев.

**Пример**