

ПРОГРАММНЫЙ КОМПЛЕКС ASTRAREGUL



РГДП.58.29.14.000-001-05 РП

# Astra.HMI

---

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Редакция 1

# ОГЛАВЛЕНИЕ

СПИСОК ИЗМЕНЕНИЙ .....	3
1. Разработка ППО .....	4
1.1. Разработка проекта автоматизации .....	6
1.1.1. Правила именования тегов .....	7
1.1.2. Astra.IDE .....	8
1.1.3. Astra.AStudio .....	218
1.1.4. Astra.HMI .....	427
1.2. Разработка пользовательской библиотеки .....	5641
1.2.1. Правила именования тегов .....	5642
1.2.2. Разработка библиотеки алгоритмов в Astra.IDE .....	5643
1.2.3. Разработка библиотеки типов в Astra.AStudio .....	5676
1.2.4. Разработка библиотеки мнемосимволов в Astra.HMI .....	5758
1.3. Языки программирования .....	5864
Язык Astra.Om .....	5866
1.4. Командная разработка .....	6115
1.5. Решение проблем .....	6116
1.5.1. Не применяются атрибуты при развертывании конфигурации на серверах .....	6117
1.5.2. Пустые сообщения о деактивации в Astra.HMI.Alarms .....	6119



# СПИСОК ИЗМЕНЕНИЙ

Редакция	Список изменений
Редакция 1	<ul style="list-style-type: none"><li>- Добавлена описание настройки пользовательского столбца в Astra.HMI.Alarms.</li><li>- Актуализирована информация по группе элементов AP в Astra.HMI.</li><li>- Актуализирована информация по обработчикам событий в Astra.HMI.</li><li>- Добавлена информация по ограничениям именованя тегов.</li><li>- Добавлен раздел <a href="#">Решение проблем</a>.<ul style="list-style-type: none"><li>- Добавлен раздел <a href="#">Ограничения по работе с энергонезависимой памятью</a>.</li></ul></li></ul>

# 1. Разработка ППО

Для разработки проекта необходимо предварительно скачать и установить программный комплекс AstraRegul.



Пошаговая инструкция приведена в разделе "Установка и удаление ПК AstraRegul" документа "Администрирование. Руководство пользователя".

Для разработки проекта необходимо установить следующие компоненты:

Компонент	Описание
Astra.IDE	Среда разработки
Пакет обновления Astra.IDE	Пакет обновления для среды разработки Astra.IDE
Плагин AstraRegul	Для автоматизации разработки проекта
Astra.AStudio	Конфигуратор сервера ввода/вывода (расширение Astra.IDE)
Astra.Server	Сервер ввода/вывода (служба)
Astra.Domain	Домен (служба)
Astra.Historian	Сервер истории (служба)
Astra.Tools	Набор утилит для диагностики и отладки
Astra.Security	Подсистема безопасности (служба)
Astra.HMI	Графический редактор и среда исполнения
Astra.HMI.CommonLib	Библиотека Astra.HMI
Astra.HMI.Charts	Библиотека Astra.HMI
Astra.HMI.Tables	Библиотека Astra.HMI
Astra.HMI.Security	Библиотека Astra.HMI
Astra.HMI.Trends	Инструмент для просмотра графиков

Astra.HMI.Alarms	Инструмент для просмотра журнала событий
Astra.HMI.Explorer	Утилита для диагностики и отладки
Astra.HMI.Security.Configurator	Инструмент для разграничения прав доступа пользователей
Astra.HMI.IntegrityControl	Инструмент для контроля целостности файлов

Для корректной работы ПК AstraRegul необходимо выполнить первичную настройку ОС Windows.



Пошаговая инструкция приведена в разделе "Первичная настройка ОС" документа "Администрирование. Руководство пользователя".

# 1.1. Разработка проекта автоматизации

Основные правила разработки ППО:

1. Названия узлов (ПЛК, коммутаторы, сервера) должны соответствовать структурной схеме проекта.
2. Учитывайте [Правила именования тегов](#).
3. Присваивайте подробные комментарии всем объектам.

## 1.1.1. Правила именования тегов

1. Используйте названия, которые будут максимально понятны инженерам АСУ ТП. Например, KKS-коды или позиционные обозначения.
2. Не используйте для именования тегов спецсимволы (пробелы, тире, точки и т.д.).



Допустимым спецсимволом является только символ подчеркивания "\_" ("земля").

3. Используйте при именвании тегов только английский алфавит.



При использовании русских букв в именвании тегов могут возникнуть проблемы при обращении к его свойствам.

## 1.1.2. Astra.IDE

Разработка проекта в среде Astra.IDE состоит из двух этапов:

1. [Создание топологии системы АСУ ТП](#)
2. [Разработка технологических алгоритмов](#)

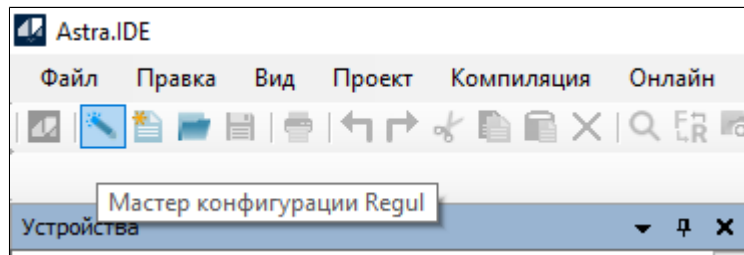
## 1.1.2.1. Создание проекта в среде разработки Astra.IDE

Создание проекта в среде Astra.IDE состоит из двух этапов:

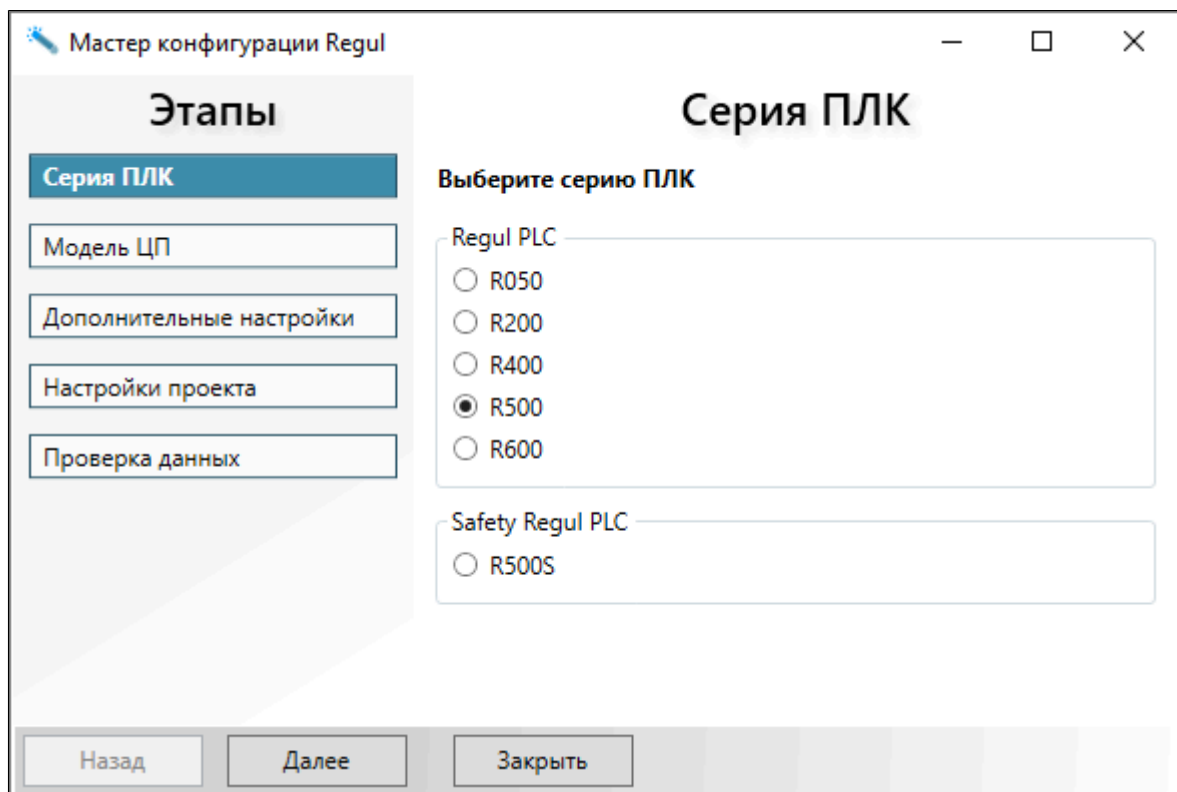
1. [Создание проекта для ПЛК REGUL R500](#)
2. [Создание проекта для ПЛК SAFETY REGUL R500S](#)

## 1.1.2.1.1. ПЛК REGUL R500

1. Чтобы создать новый проект контроллера REGUL R500 в среде Astra.IDE запустите приложение Astra.IDE (ярлык на рабочем столе).
2. Нажмите кнопку "Мастер конфигурации Regul" на панели инструментов.

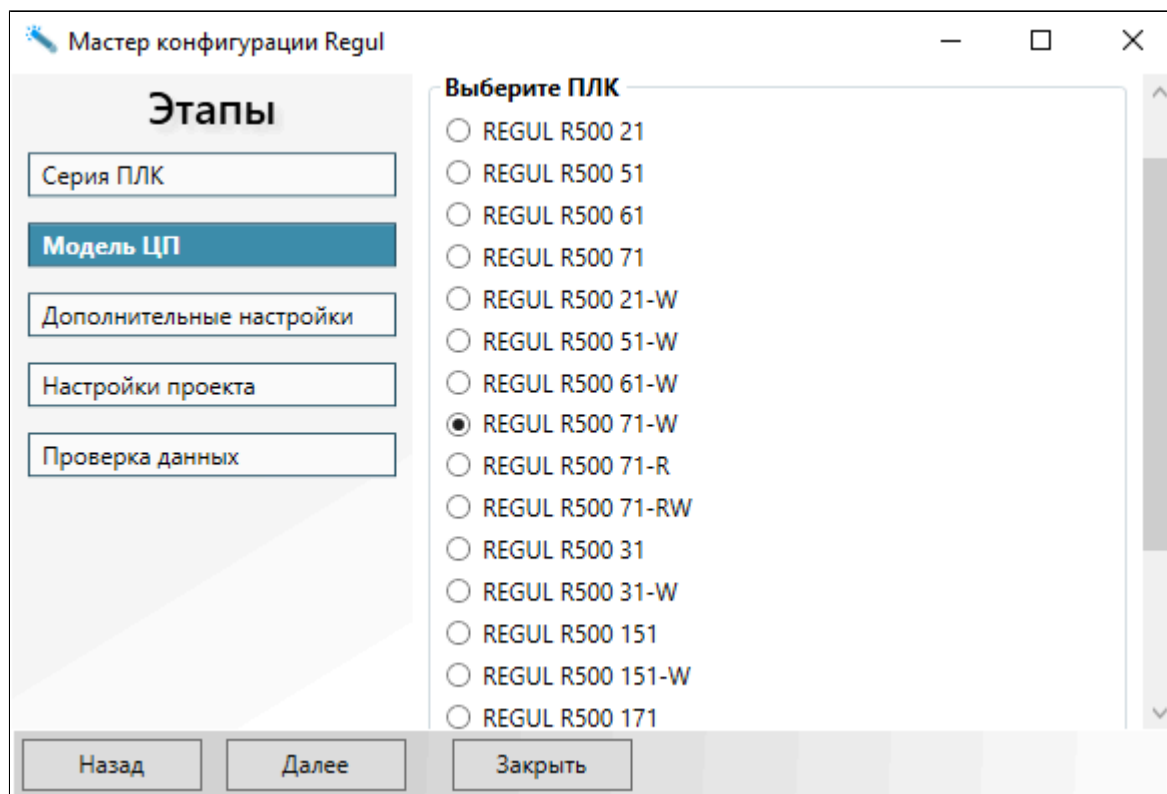


3. В открывшемся окне выберите серию контроллера "R500" и нажмите кнопку "Далее".

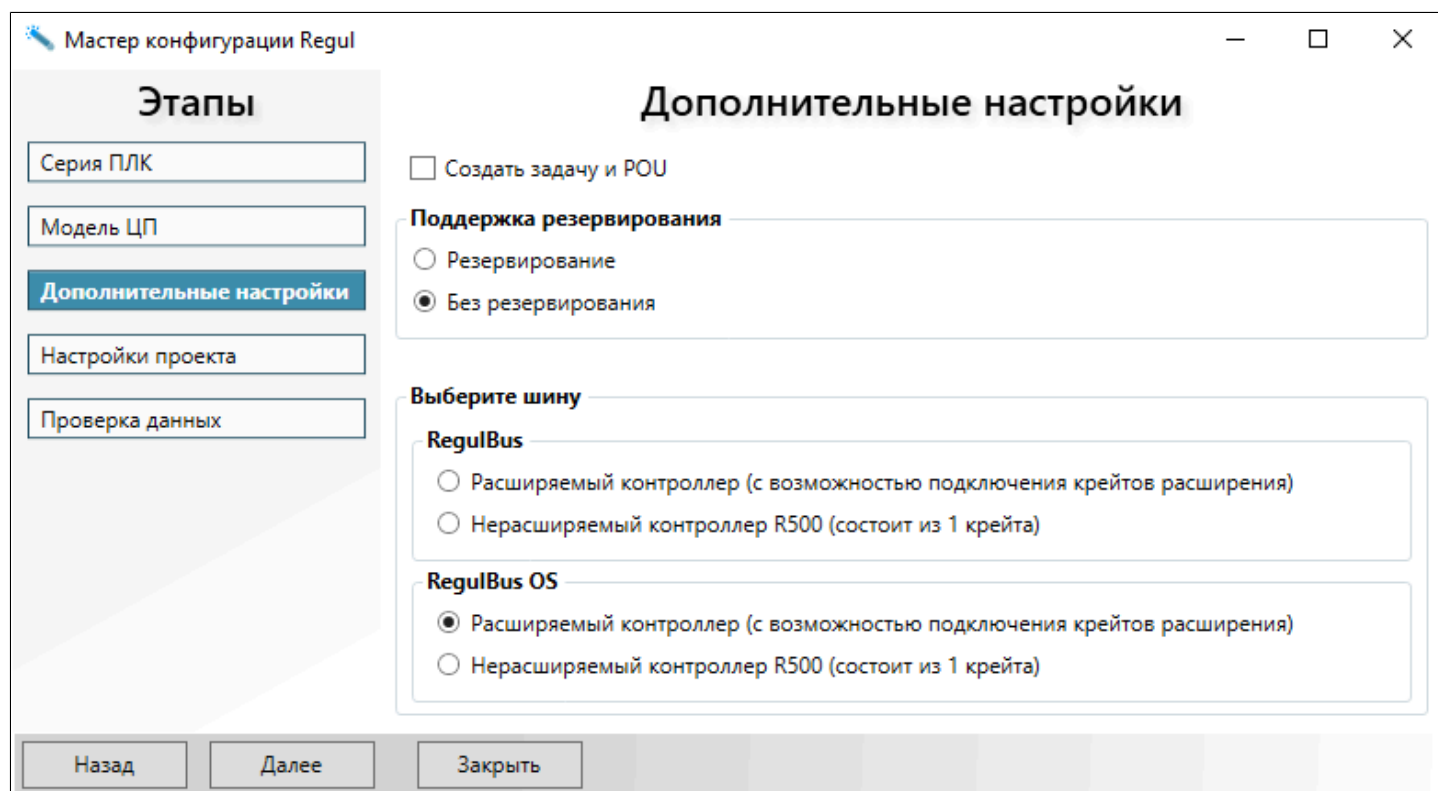




4. Выберите "Модель ЦП" – "REGUL R500 71-W" и нажмите кнопку "Далее".



5. Снимите флаг "Создать задачу и POU". В разделе "Поддержка резервирования" выберите "Без резервирования". Выберите шину RegulBus OS с расширяемым контроллером. Нажмите кнопку "Далее".



6. Выберите пункт "Создать новый проект", укажите папку хранения проекта и название проекта - "QUICK\_START". Нажмите кнопку "Далее".

Мастер конфигурации Regul

### Этапы

- Серия ПЛК
- Модель ЦП
- Дополнительные настройки
- Настройки проекта**
- Проверка данных

## Настройки проекта

Для работы с мастером конфигурации regul вам надо определить проект

Использовать существующий проект

Создать новый проект

Имя:

Путь:

Назад    Далее    Закрыть

7. Проверьте настройки и нажмите кнопку "Завершить".

Мастер конфигурации Regul

### Этапы

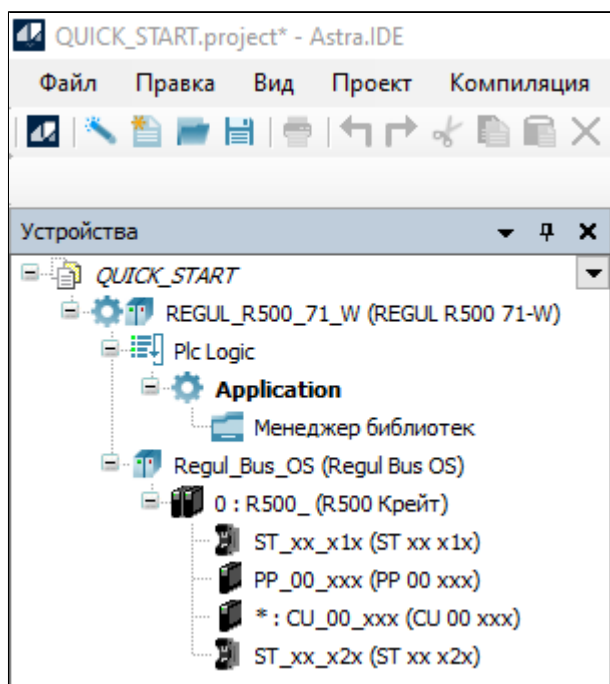
- Серия ПЛК
- Модель ЦП
- Дополнительные настройки
- Настройки проекта
- Проверка данных**

## Проверка данных

Серия:	R500
Модель ЦП:	REGUL R500 71-W
Шина Regul:	Расширяемый контроллер (с возможностью подключения крейтов расширения)
Резервирование:	<input type="checkbox"/>
Дублирование:	<input type="checkbox"/>
Создать проект:	<input checked="" type="checkbox"/>
Создать задачу и POU:	<input type="checkbox"/>

Назад    Завершить    Закрыть

8. Откроется окно среды разработки Astra.IDE с шаблоном проекта контроллера REGUL R500.



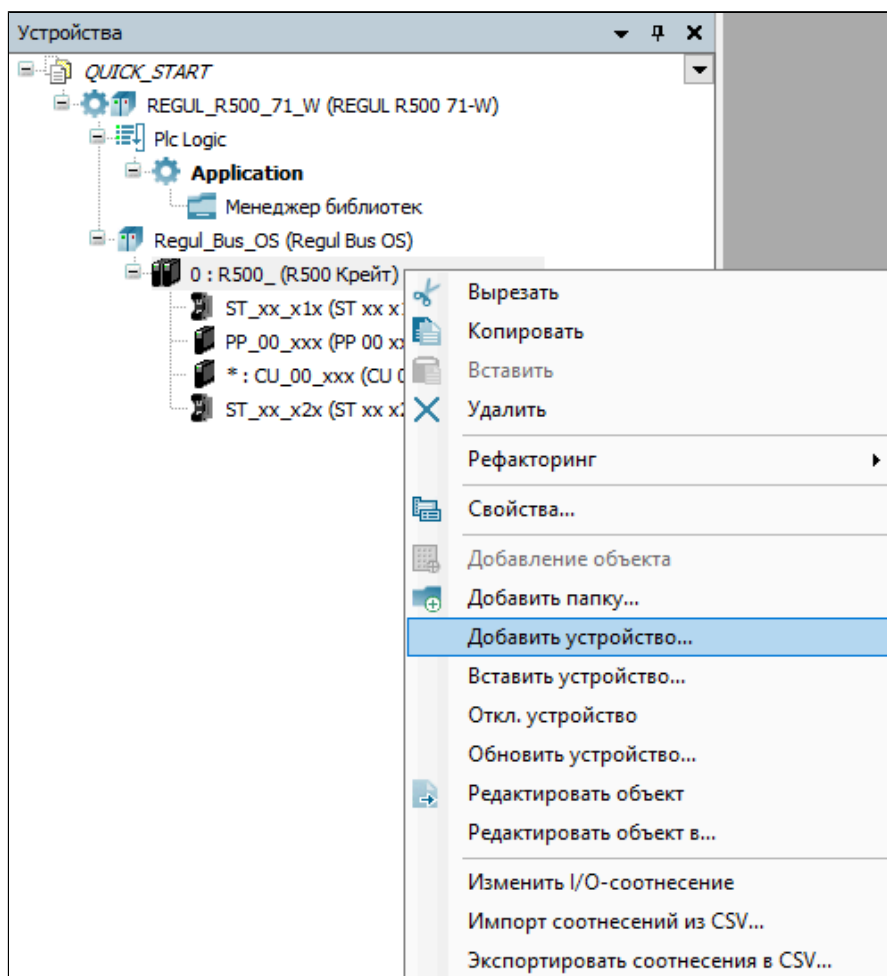
Для унификации в автоматизации разработки проектов запрещается изменять название приложения "Application".

# 1.1.2.1.1.1. Создание конфигурации крейта ПЛК REGUL R500

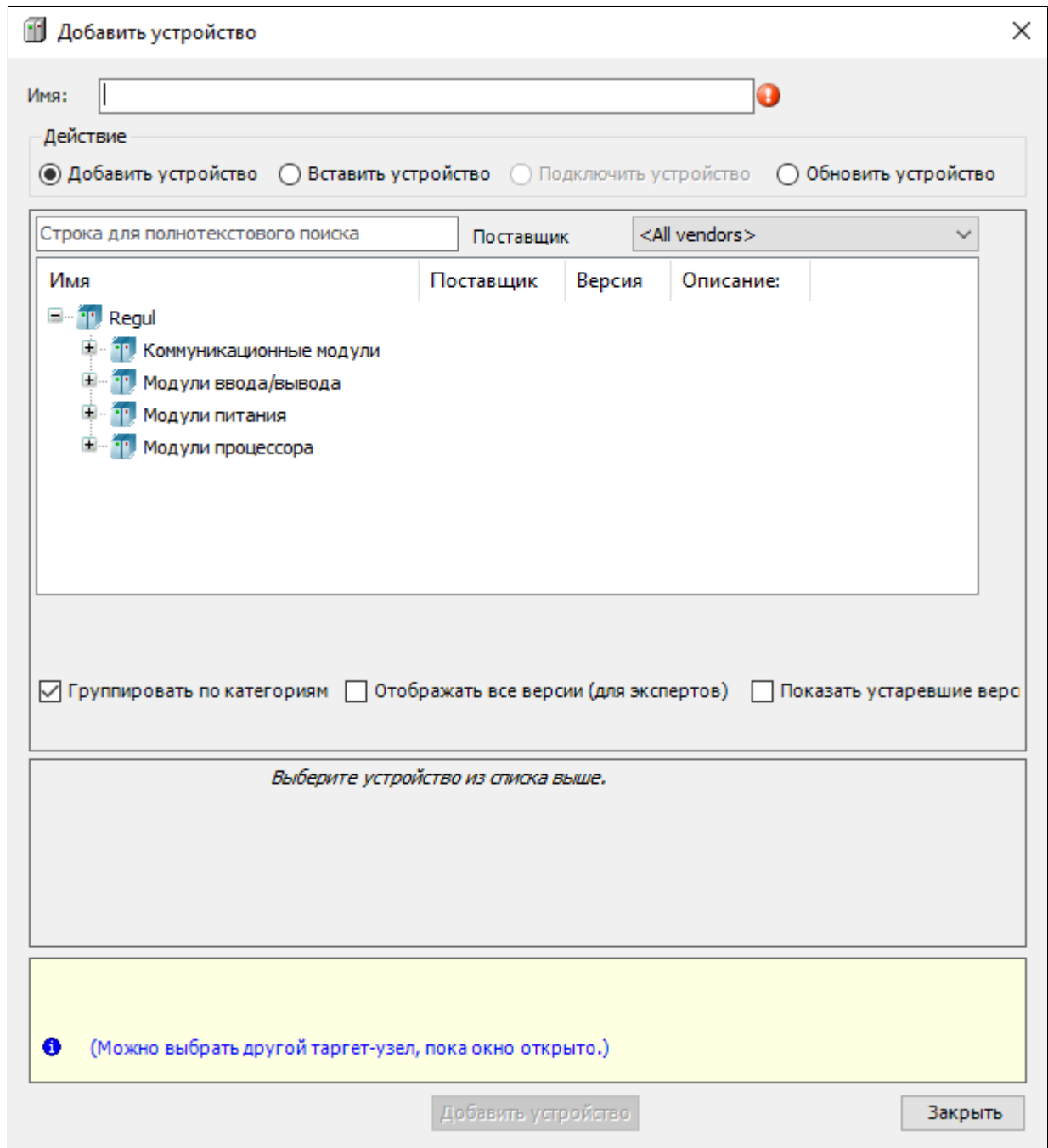
Конфигурация крейта ПЛК R500:

1. Модуль оконечный R500 ST 02 012.
2. Модуль источника питания R500 PP 00 011.
3. Центральный процессор R500 CU 00 071.
4. Модуль аналогового ввода R500 AI 08 041.
5. Модуль аналогового вывода R500 AO 08 011.
6. Модуль дискретного ввода R500 DI 32 011.
7. Модуль дискретного вывода R500 DO 32 011.
8. Модуль оконечный R500 ST 02 022.

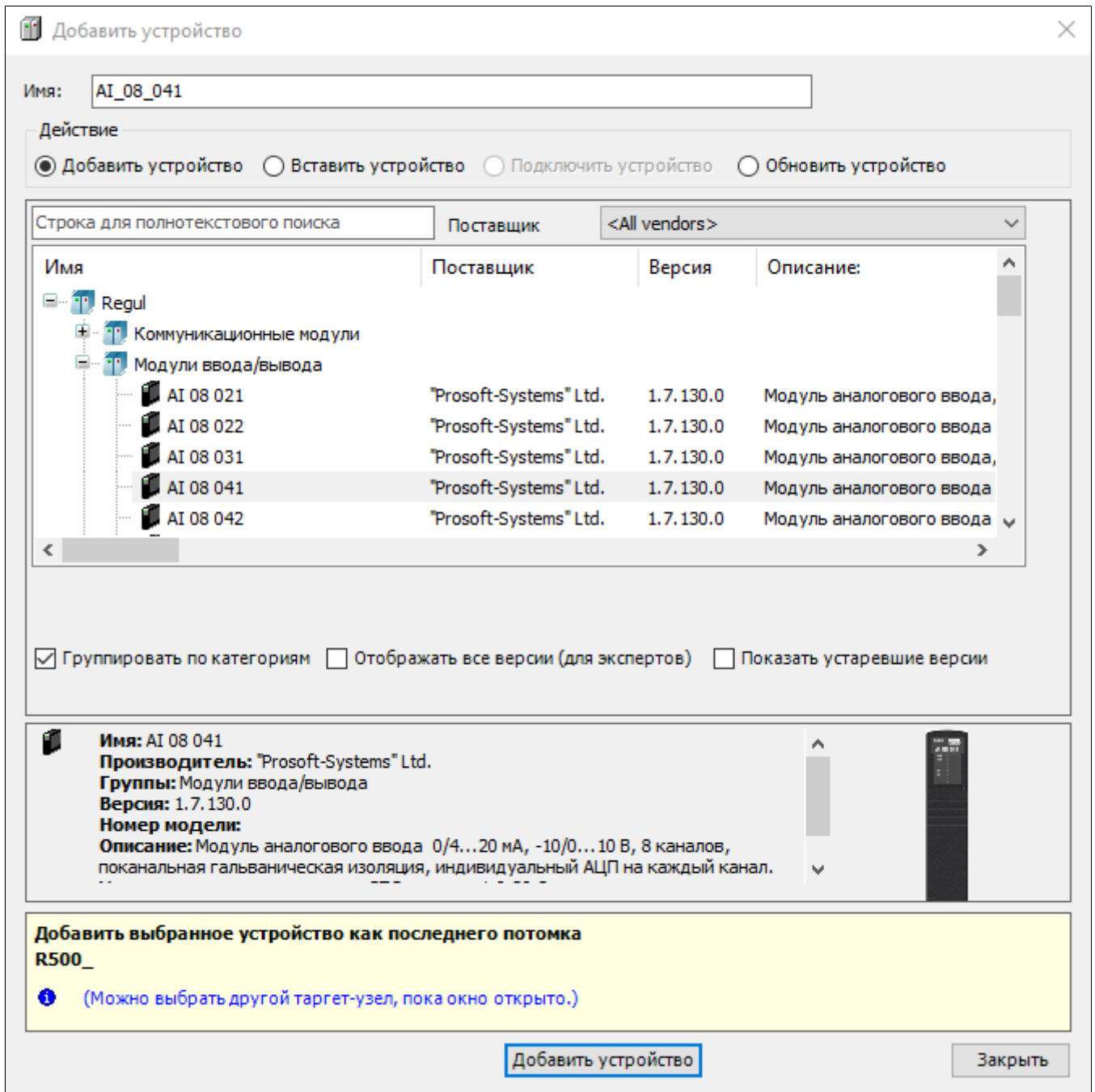
1. Для добавления модулей кликните правой кнопкой мыши по крейту и в выпадающем меню выберите пункт "Добавить устройство".



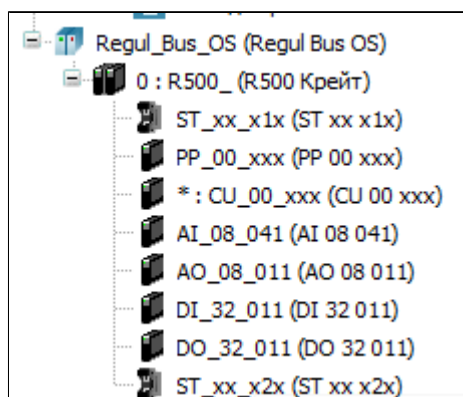
Откроется диалоговое окно:



2. Выберите нужный модуль и нажмите кнопку "Добавить устройство".



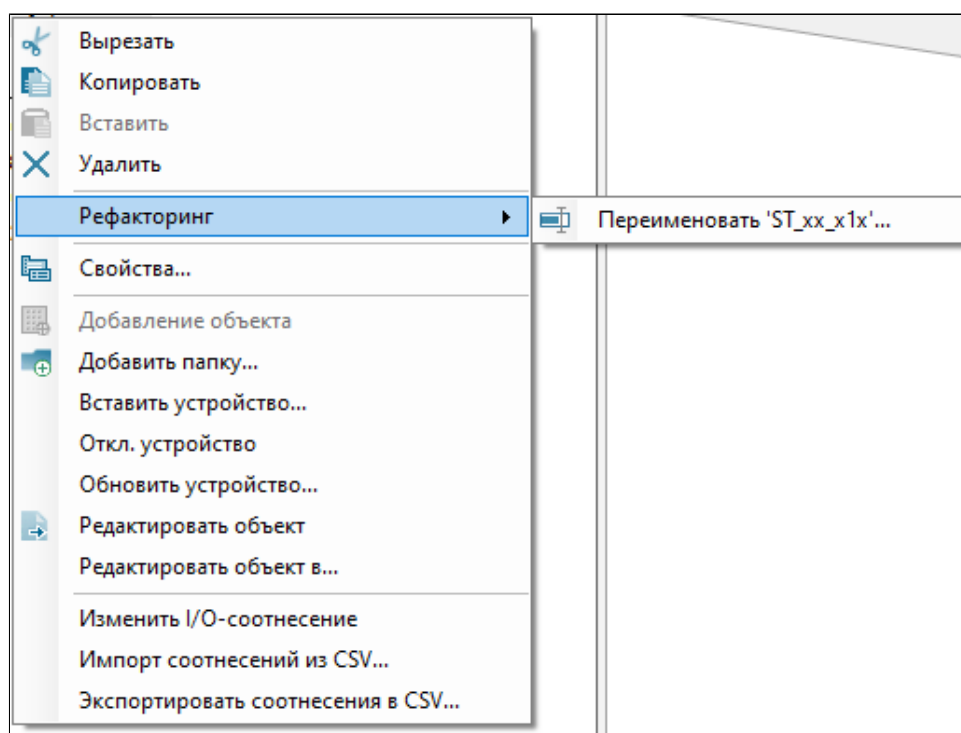
3. Аналогичным образом добавьте все модули. В итоге должна получиться следующая конфигурация:



4. Чтобы переименовать модуль в контекстном меню выберите "Рефакторинг" → "Переименовать...".

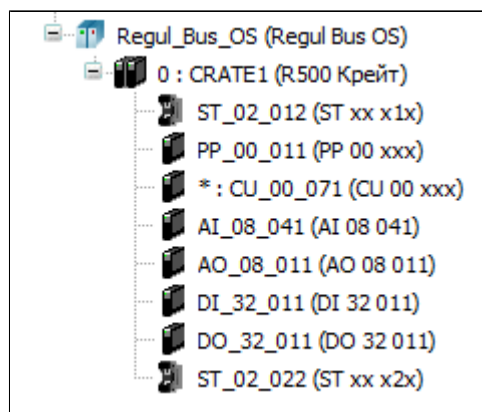


Рекомендуется всем объектам в проекте давать названия, которые отражают их функциональное назначение.

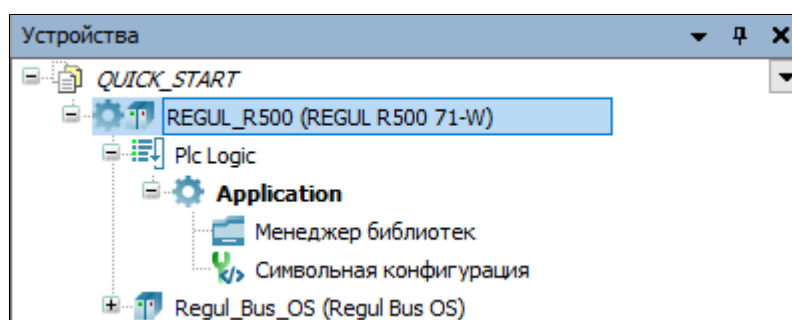




Должна получиться следующая конфигурация:



Аналогично переименуйте контроллер на "REGUL\_R500".



5. Чтобы открыть окно редактора крейта, кликните по нему два раза левой кнопкой мыши.

QUICK\_START.project\* - Astra.IDE

Файл Плавка Вид Проект Компиляция Онлайн Отладка Инструменты Окно Справка

Application [REGUL\_R500: Plc Logic]

Устройства

- QUICK\_START
  - REGUL\_R500 (REGUL R500 71-W)
    - Plc Logic
      - Application
        - Менеджер библиотек
        - Символьная конфигурация
      - Regul\_Bus\_OS (Regul Bus OS)
        - 0 : CRATE1 (R500 Крейт)
          - ST\_02\_012 (ST xx x1x)
          - PP\_00\_011 (PP 00 xxx)
          - \* : CU\_00\_071 (CU 00 xxx)
          - AI\_08\_041 (AI 08 041)
          - AO\_08\_011 (AO 08 011)
          - DI\_32\_011 (DI 32 011)
          - DO\_32\_011 (DO 32 011)
          - ST\_02\_022 (ST xx x2x)

CRATE1 x

Редактор крейта Regul R500

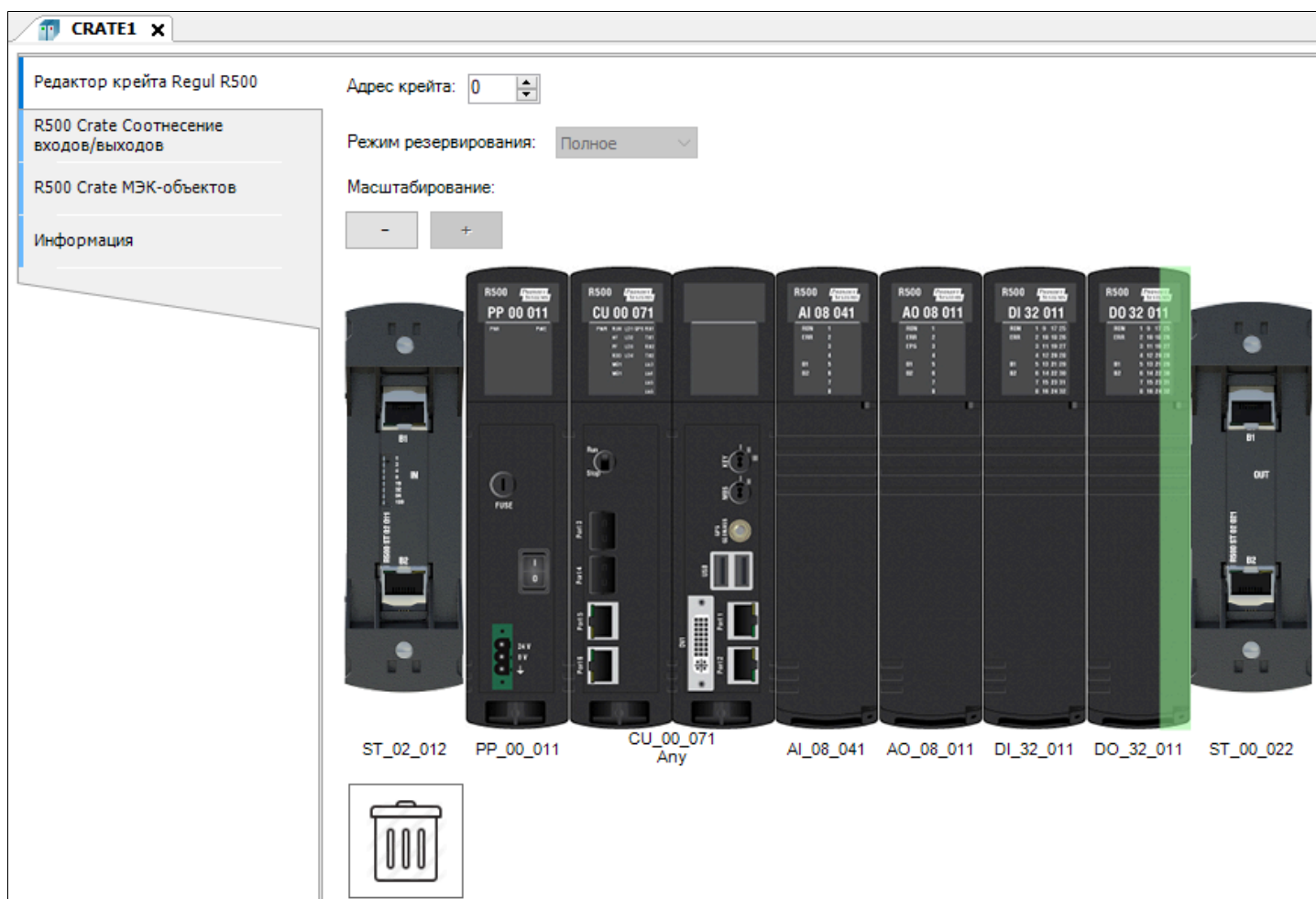
Адрес крейта: 0

Режим резервирования: Полное

Масштабирование: - +

ST\_02\_012 PP\_00\_011 CU\_00\_071 Any AI\_08\_041 AO\_08\_011 DI\_32\_011 DO\_32\_011 ST\_02\_022

6. Для добавления модуля выделите место, в которое необходимо вставить модуль. Область для вызова окна "Вставить устройство" подсветится зеленым цветом.

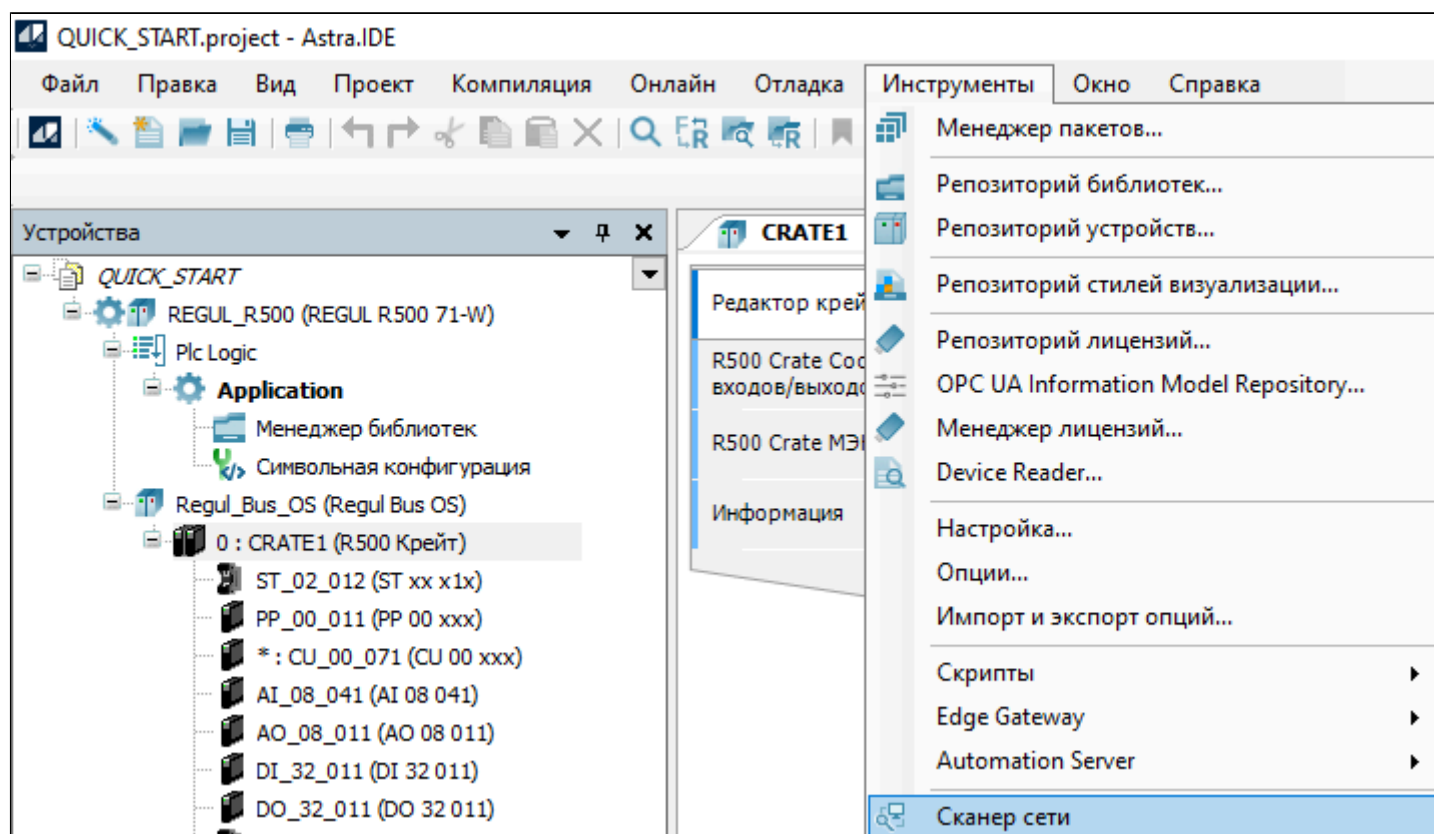


7. Для удаления модуля перетащите его в "Корзину"  .

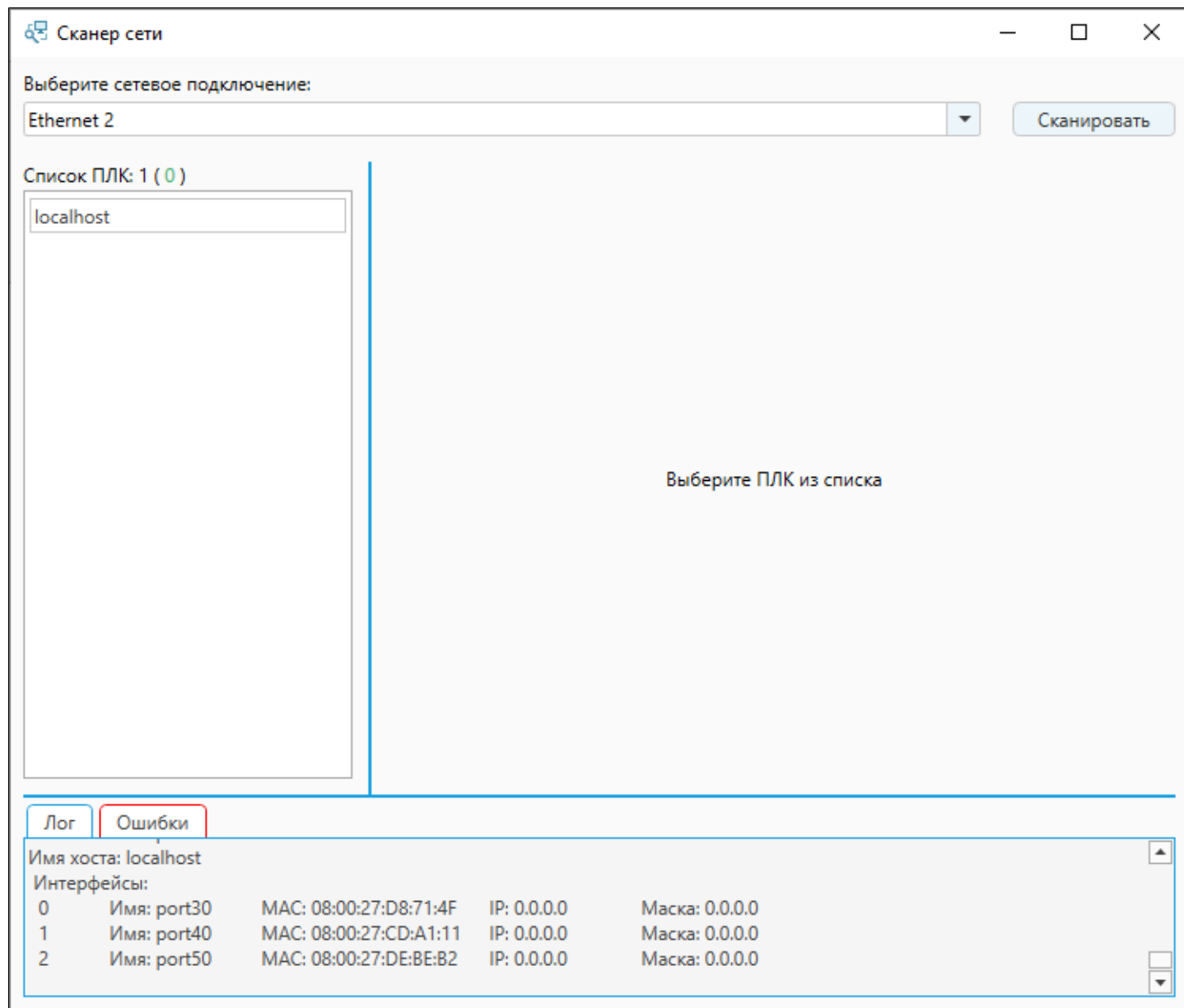
## 1.1.2.1.1.2. Настройка подключения к ПЛК REGUL

Чтобы загрузить разработанный проект в ПЛК, необходимо настроить его сетевые параметры:

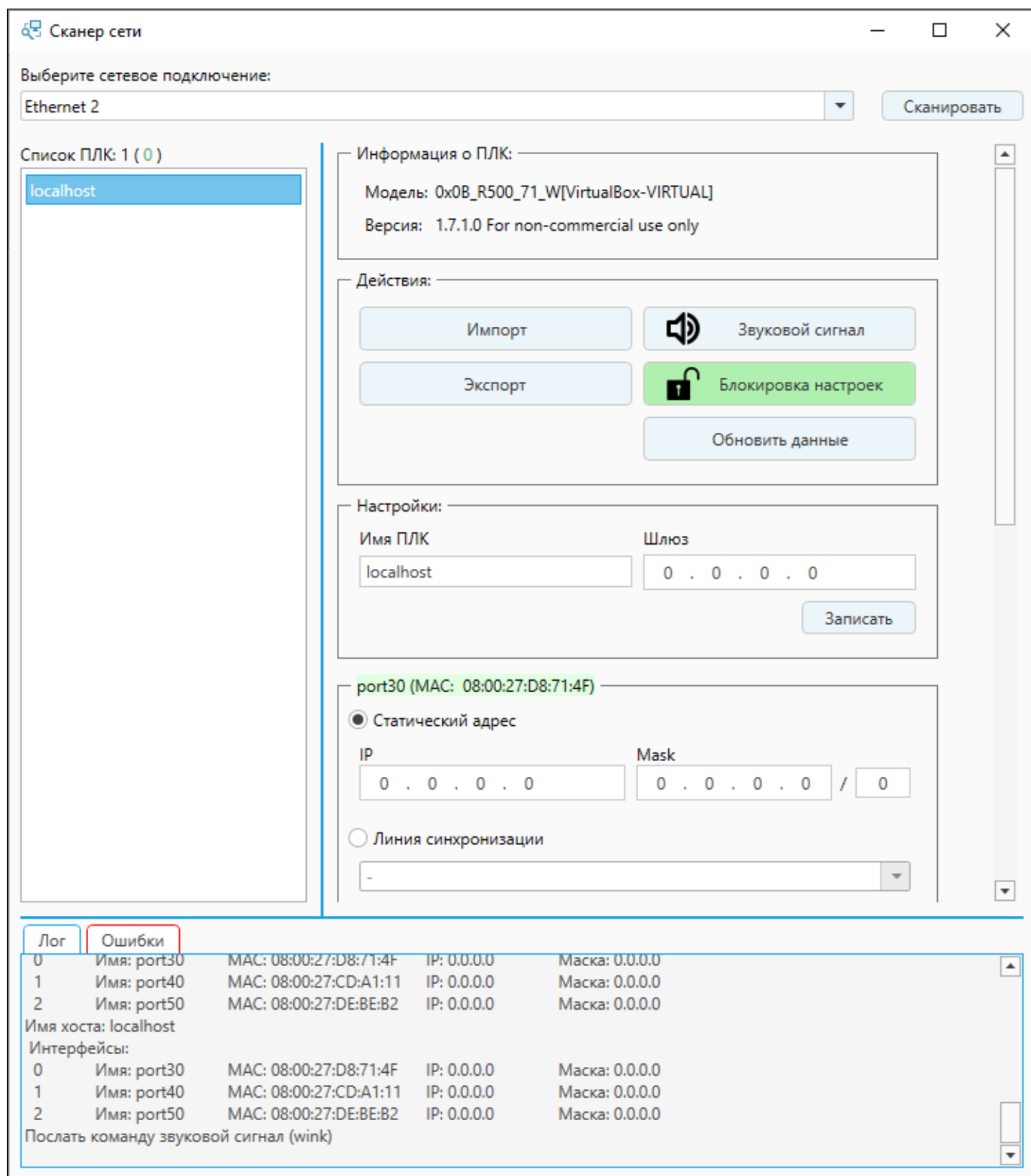
1. Перейдите в меню "Инструменты" и выберите пункт "Сканер сети".



2. В открывшемся диалоговом окне "Сканер сети" выберите сетевое подключение из выпадающего списка и нажмите кнопку "Сканировать". В результате сканирования в списке ПЛК появятся все контроллеры, подключенные к выбранной сети.



3. Выберите контроллер, с которым требуется установить соединение. Чтобы убедиться, что выбран нужный контроллер, нажмите кнопку "Звуковой сигнал". Контроллер издаст кратковременный звуковой сигнал.



#### 4. Настройте ПЛК А:

› Имя ПЛК: REGUL\_R500\_A.

› Шлюз: 0.0.0.0.

› port30:

port30 (MAC: 08:00:27:26:23:1F)

Статический адрес

IP: 192 . 168 . 98 . 1      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

Резервирование CPU A канал 1

DHCP

Очистить      Записать

› port40:

port40 (MAC: 08:00:27:53:B2:06)

Статический адрес

IP: 192 . 168 . 99 . 1      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

Резервирование CPU A канал 2

DHCP

Очистить      Записать

› port50:

port50 (MAC: 00:BD:C7:95:DB:23)

Статический адрес

IP: 192 . 168 . 1 . 11      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

-

DHCP

Очистить      Записать

› port60:

port60 (MAC: 00:FB:2D:C5:85:C4)

Статический адрес

IP: 192 . 168 . 2 . 11      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

DHCP



Чтобы сохранить изменения, необходимо нажать кнопку "Записать", иначе введенные значения не будут применены.

После нажатия кнопки "Записать" в журнале отобразится сообщение об успешной записи введенных значений.

Лог    Ошибки

Задать новое имя: REGUL\_R500\_A  
Имя успешно изменено  
Задать Gateway: 0.0.0.0  
Шлюз успешно задан.  
Задать IP: 192.168.1.11 && Маску: 255.255.255.0  
IP и маска успешно изменены.  
Задать IP: 192.168.2.11 && Маску: 255.255.255.0  
IP и маска успешно изменены.

5. Настройте ПЛК Б:

- › Имя ПЛК: REGUL\_R500\_B.
- › Шлюз: 0.0.0.0.



> port30:

port30 (MAC: 08:00:27:26:23:1F)

Статический адрес

IP: 192 . 168 . 98 . 2      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

Резервирование CPU В канал 1

DHCP

Очистить      Записать

> port40:

port40 (MAC: 08:00:27:53:82:06)

Статический адрес

IP: 192 . 168 . 99 . 2      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

Резервирование CPU В канал 2

DHCP

Очистить      Записать

> port50:

port50 (MAC: 00:BD:C7:95:DB:23)

Статический адрес

IP: 192 . 168 . 1 . 12      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

-

DHCP

Очистить      Записать

> port60:

port60 (MAC: 00:FB:2D:C5:85:C4)

Статический адрес

IP: 192 . 168 . 2 . 12      Mask: 255 . 255 . 255 . 0 / 24

Линия синхронизации

-

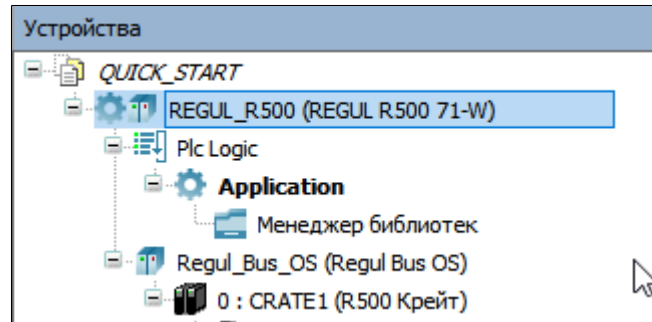
DHCP

Очистить      Записать

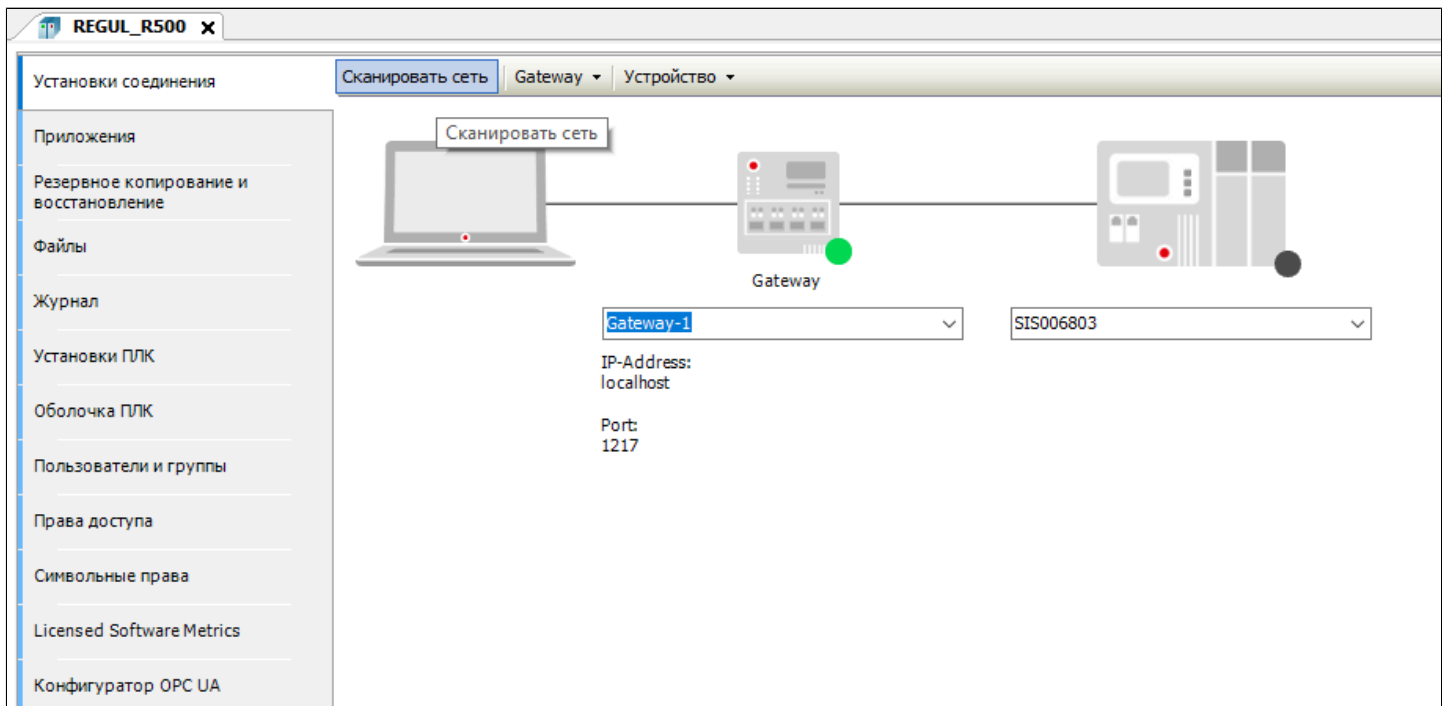


Компьютер и все ПЛК должны находиться в одной подсети.

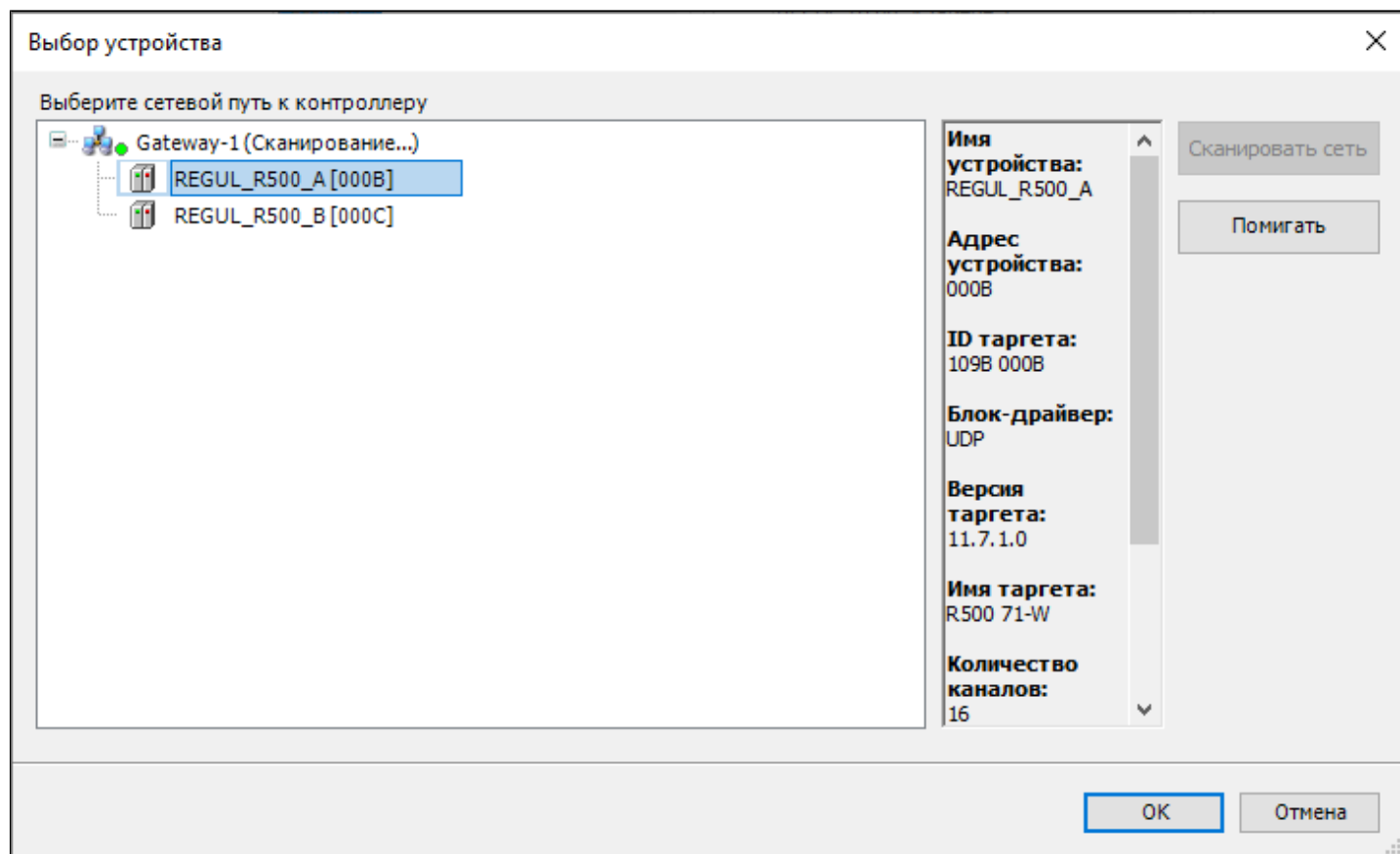
6. Выполните двойной клик мыши по ПЛК в дереве устройств.



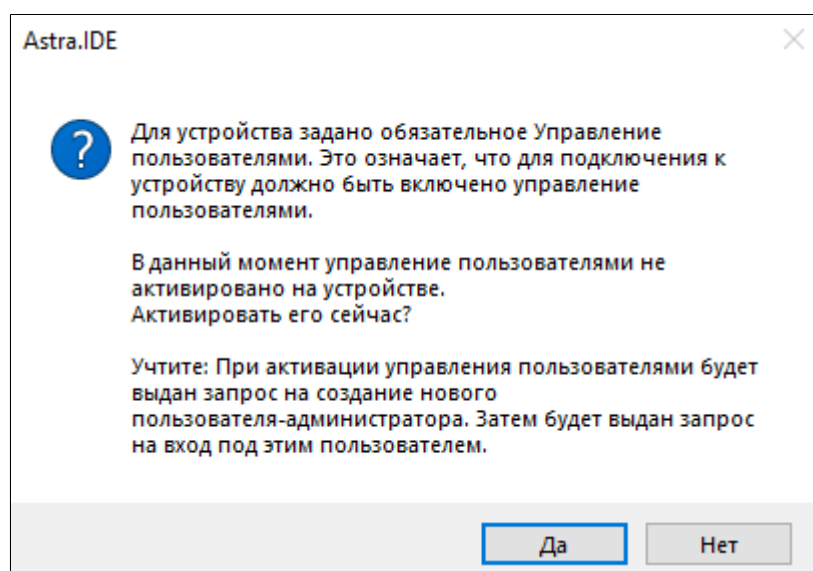
7. В меню "Установки соединения" нажмите кнопку "Сканировать сеть".



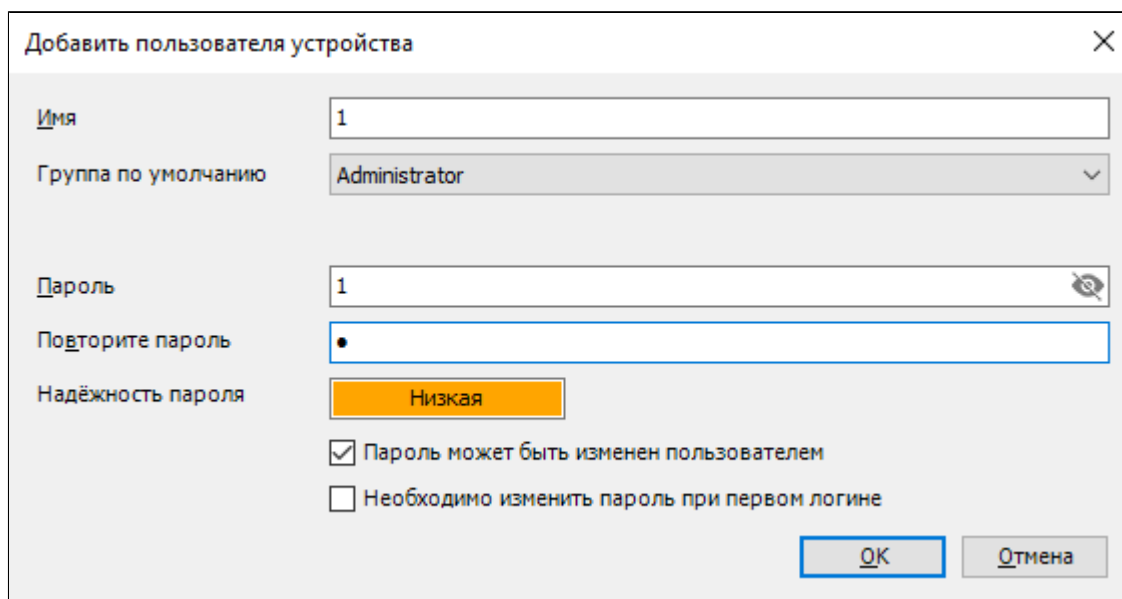
8. Выберите "REGUL\_R500\_A" и нажмите кнопку "OK".



Появится всплывающее окно. Нажмите кнопку "Да".



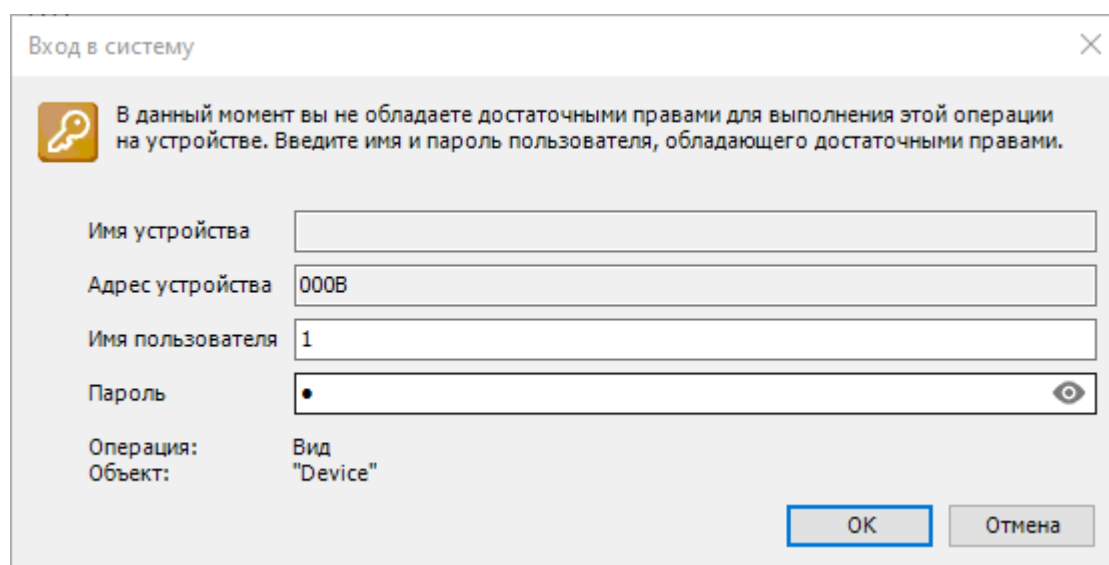
9. Задайте имя пользователя "1", группу по умолчанию "Administrator" и пароль "1". Нажмите кнопку "OK".



The screenshot shows a dialog box titled "Добавить пользователя устройства" (Add user to device). It contains the following fields and options:

- Имя** (Name): Text box containing "1".
- Группа по умолчанию** (Default group): Dropdown menu showing "Administrator".
- Пароль** (Password): Text box containing "1" with a visibility icon on the right.
- Повторите пароль** (Repeat password): Text box containing a black dot.
- Надёжность пароля** (Password strength): A yellow button labeled "Низкая" (Low).
- Two checkboxes:
  - Пароль может быть изменен пользователем (Password can be changed by user)
  - Необходимо изменить пароль при первом входе (Must change password on first login)
- Buttons: "OK" and "Отмена" (Cancel).

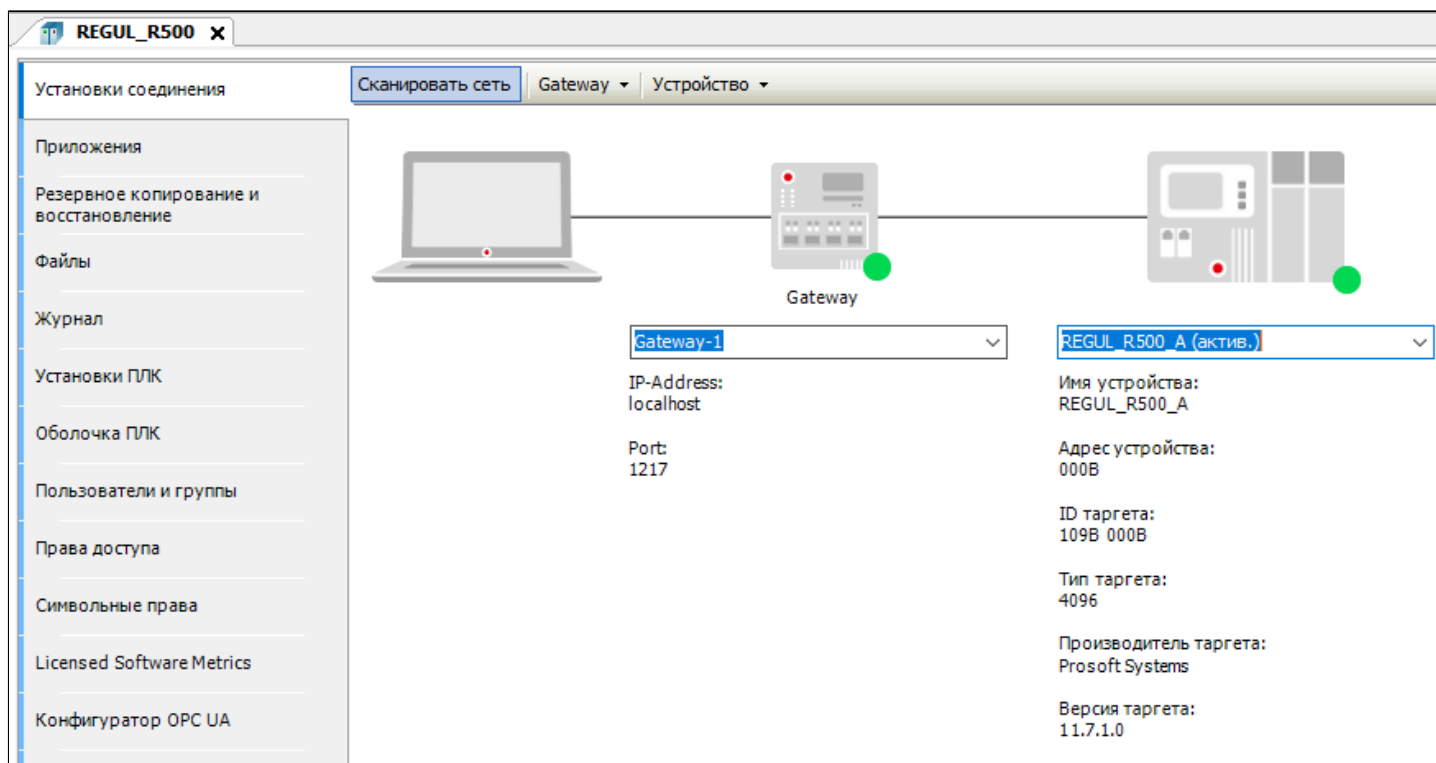
10. Откроется окно "Вход в систему". Оно необходимо для исключения несанкционированного доступа к ПЛК. Введите имя пользователя и пароль, которые были заданы ранее, и нажмите кнопку "OK".



The screenshot shows a dialog box titled "Вход в систему" (Sign in to system). It contains the following fields and options:

- A warning icon and text: "В данный момент вы не обладаете достаточными правами для выполнения этой операции на устройстве. Введите имя и пароль пользователя, обладающего достаточными правами." (At this moment you do not have sufficient rights to perform this operation on the device. Enter the name and password of the user who has sufficient rights.)
- Имя устройства** (Device name): Empty text box.
- Адрес устройства** (Device address): Text box containing "000B".
- Имя пользователя** (User name): Text box containing "1".
- Пароль** (Password): Text box containing a black dot with a visibility icon on the right.
- Операция:** Вид "Device" (Operation: View "Device")
- Buttons: "OK" and "Отмена" (Cancel).

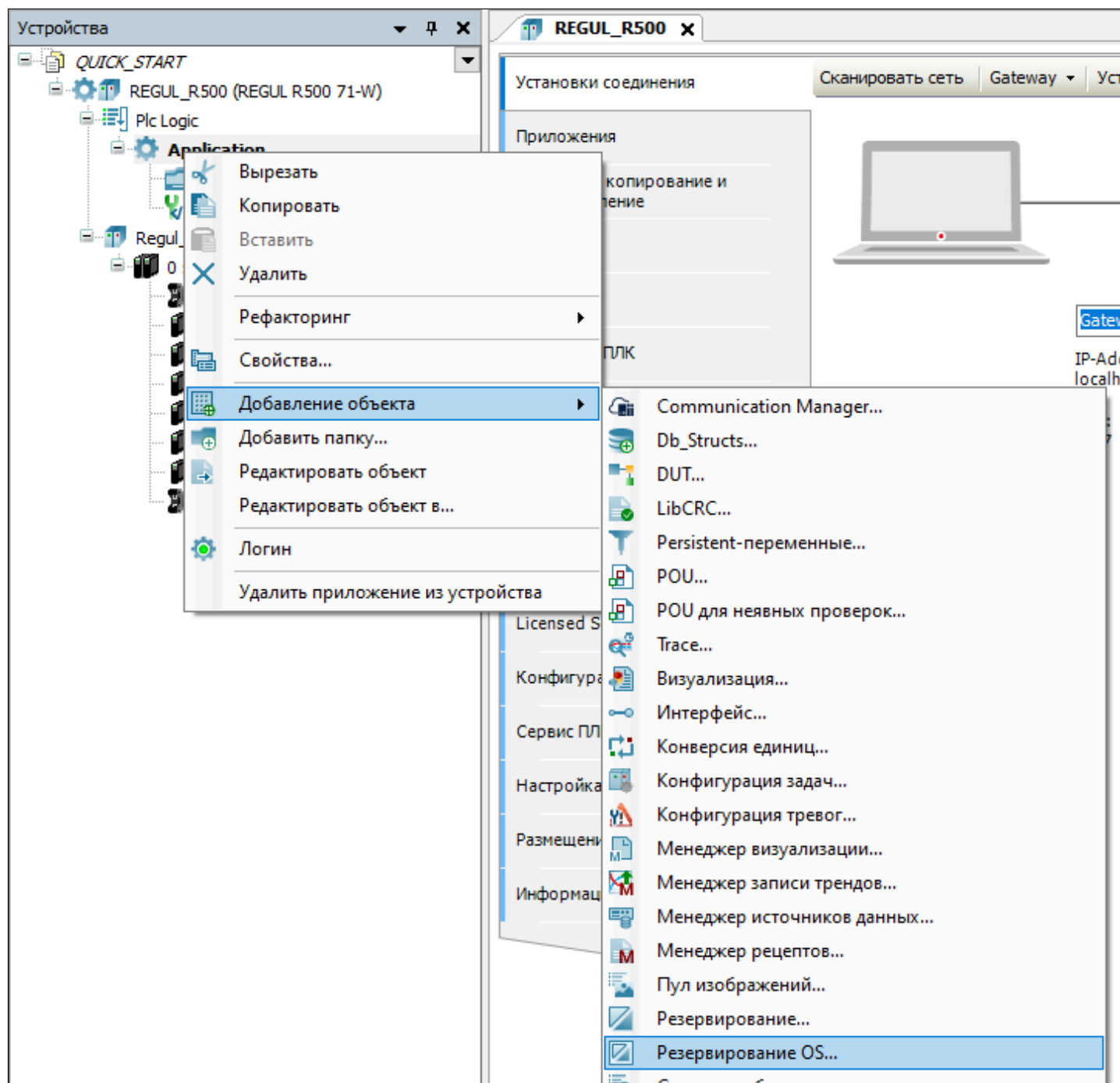
## 11. Подключение к REGUL\_R500\_A выполнено.



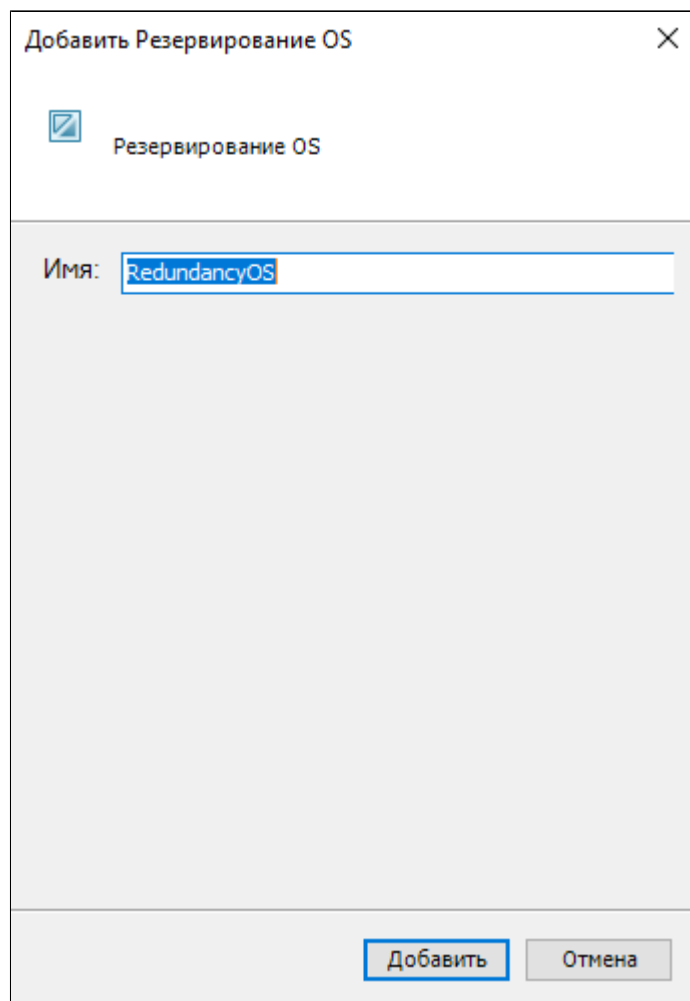
Аналогичным образом можно подключиться к REGUL\_R500\_B.

# 1.1.2.1.1.3. Настройка резервирования ПЛК REGUL R500

1. В контекстном меню приложения "Application" выберите "Добавление объекта" → "Резервирование OS...".



2. В открывшемся диалоговом окне нажмите кнопку "Добавить".



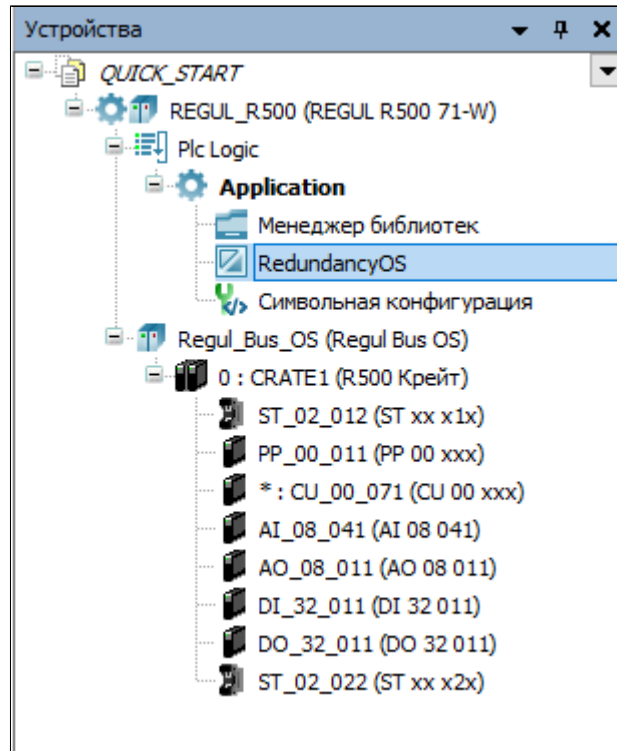
Добавить Резервирование OS

Резервирование OS

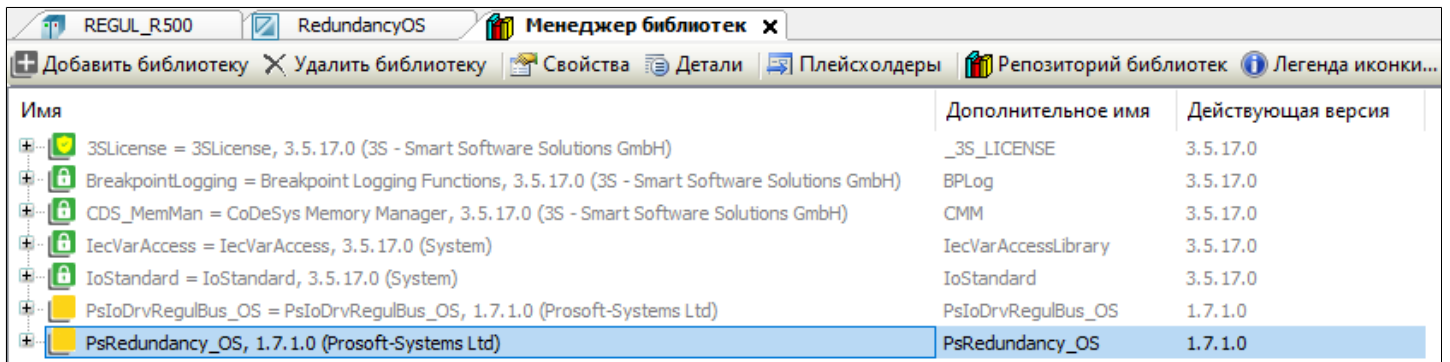
Имя:

Добавить Отмена

Компонент "Резервирование OS" будет добавлен в дерево устройств.

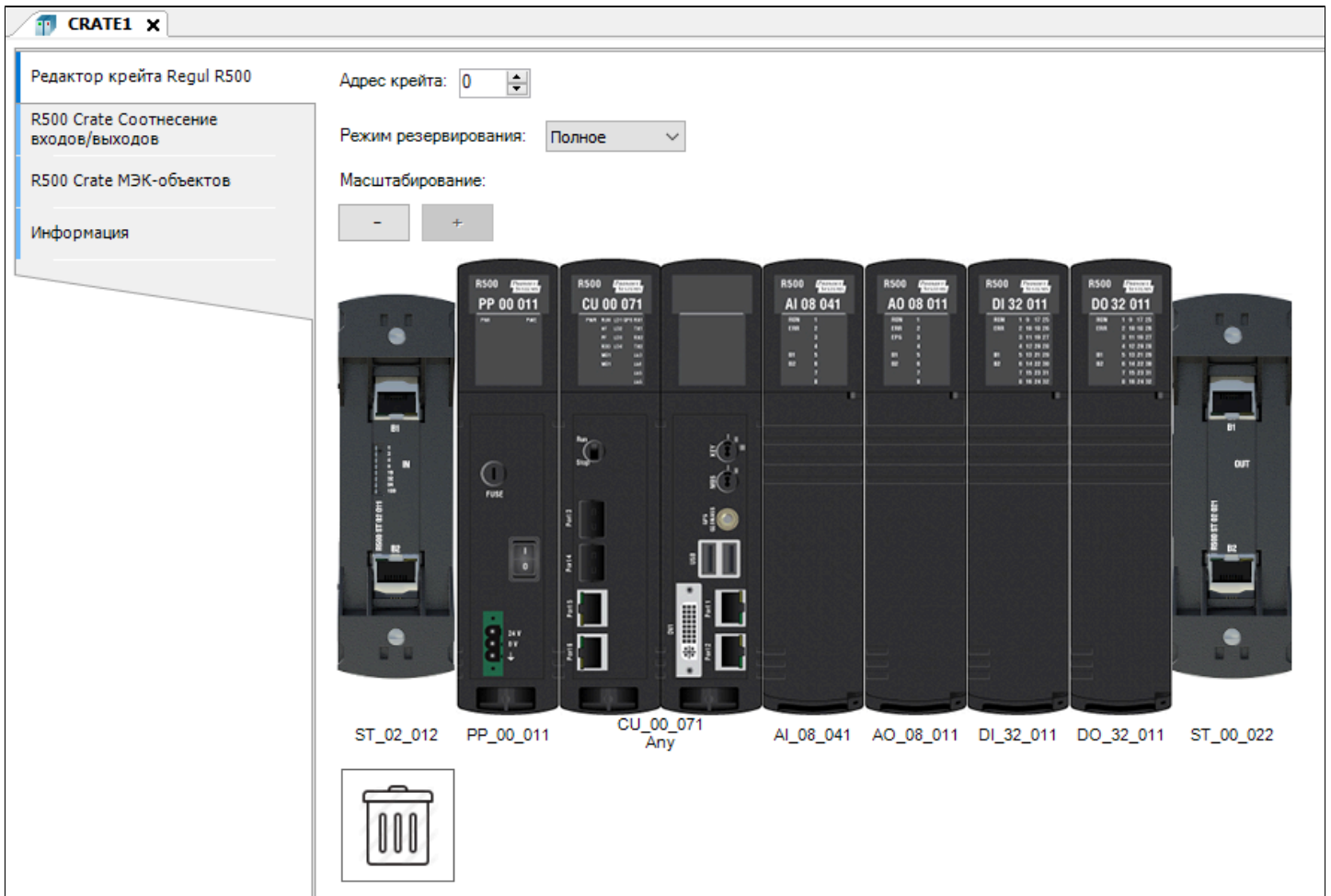


В проект также автоматически добавится библиотека "PsRedundancy\_OS", которая предоставляет диагностическую информацию по резервированию.

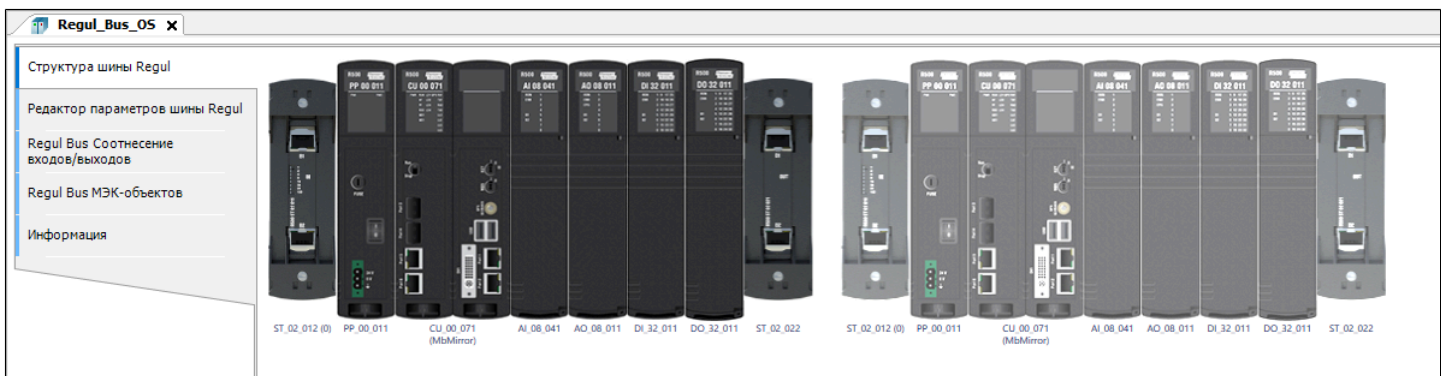




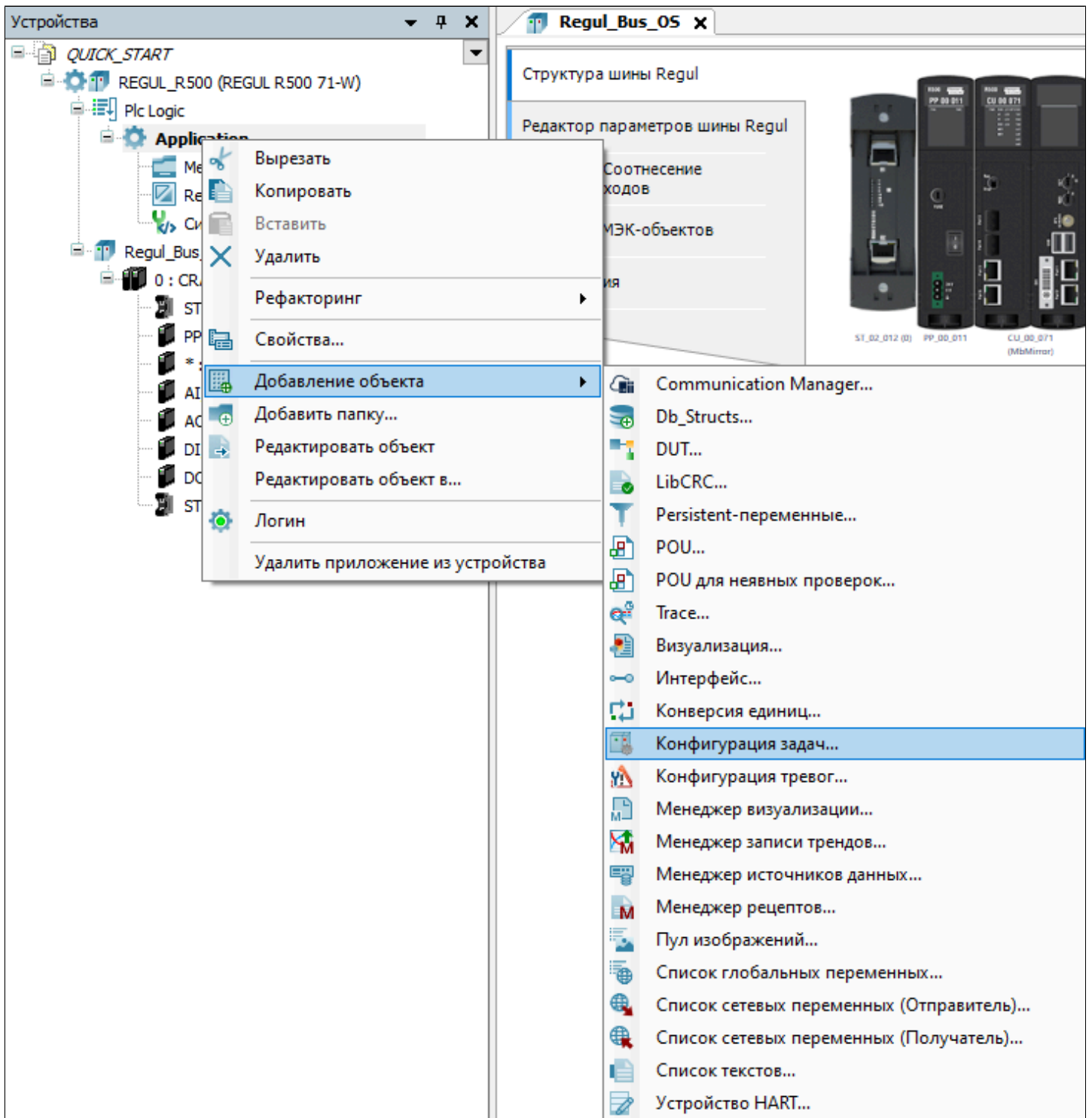
3. Перейдите в редактор крейта двойным кликом левой кнопки мыши. В поле "Режим резервирования" в выпадающем списке выберите "Полное".



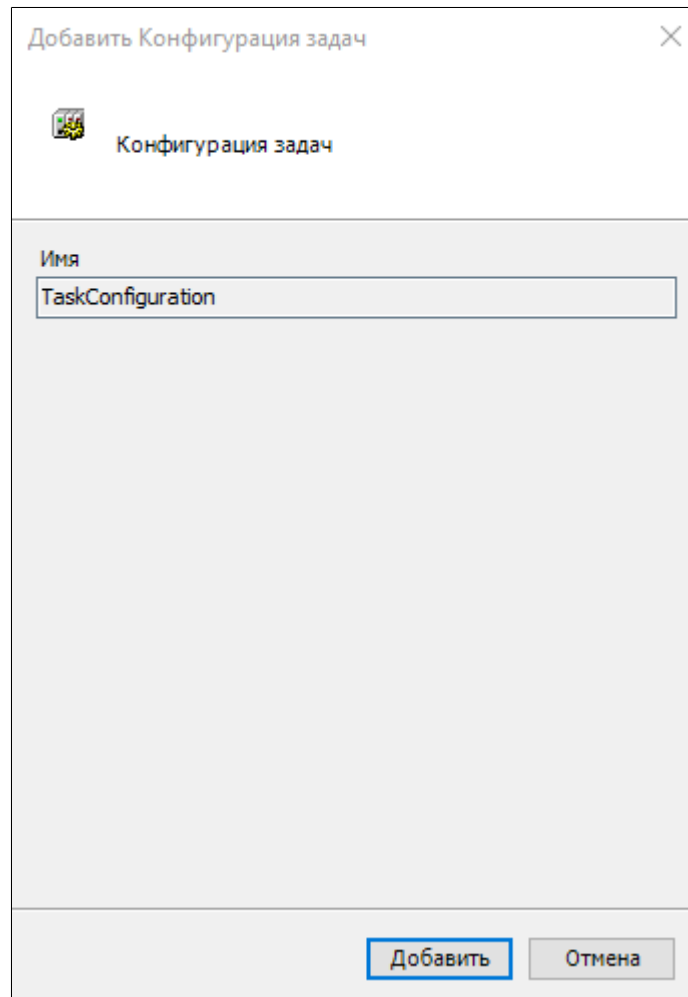
4. Перейдите в редактор шины "Regul\_Bus\_OS". При выборе режима резервирования "Полное" отобразится крейт для резервирования.



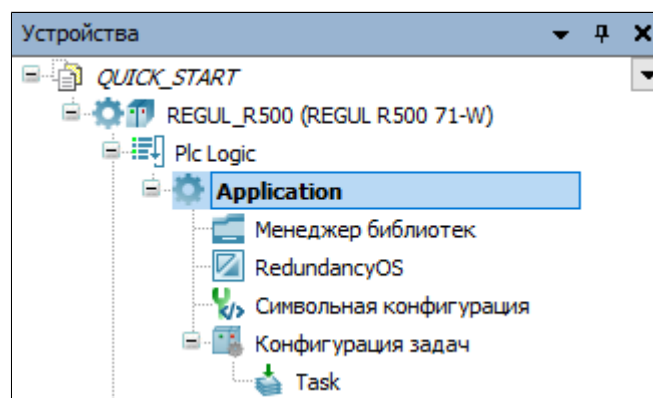
5. Добавьте в проект объект "Конфигурация задач". Для этого в контекстном меню приложения "Application" выберите "Добавление объекта" → "Конфигурация задач...".



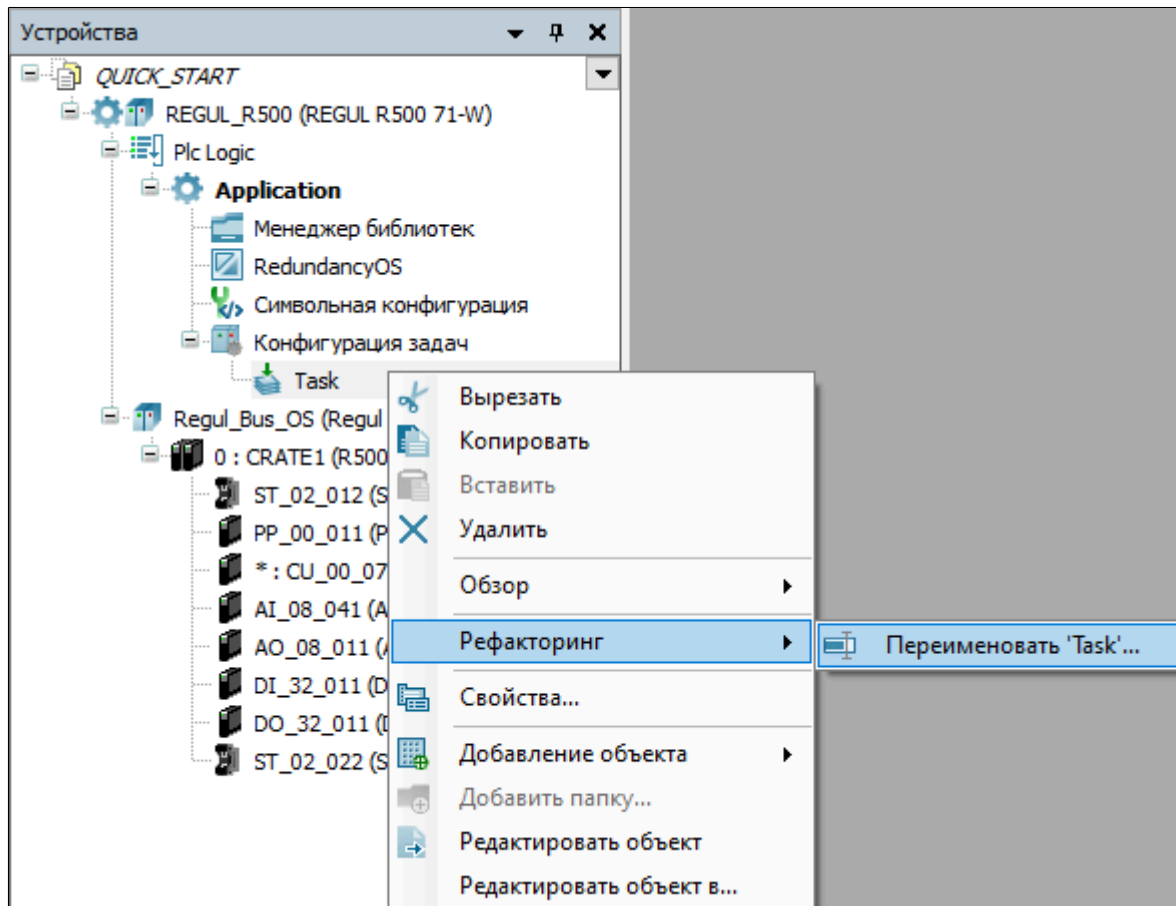
6. В открывшемся окне нажмите кнопку "Добавить".



В проект будет добавлен конфигуратор задач и задача "Task".

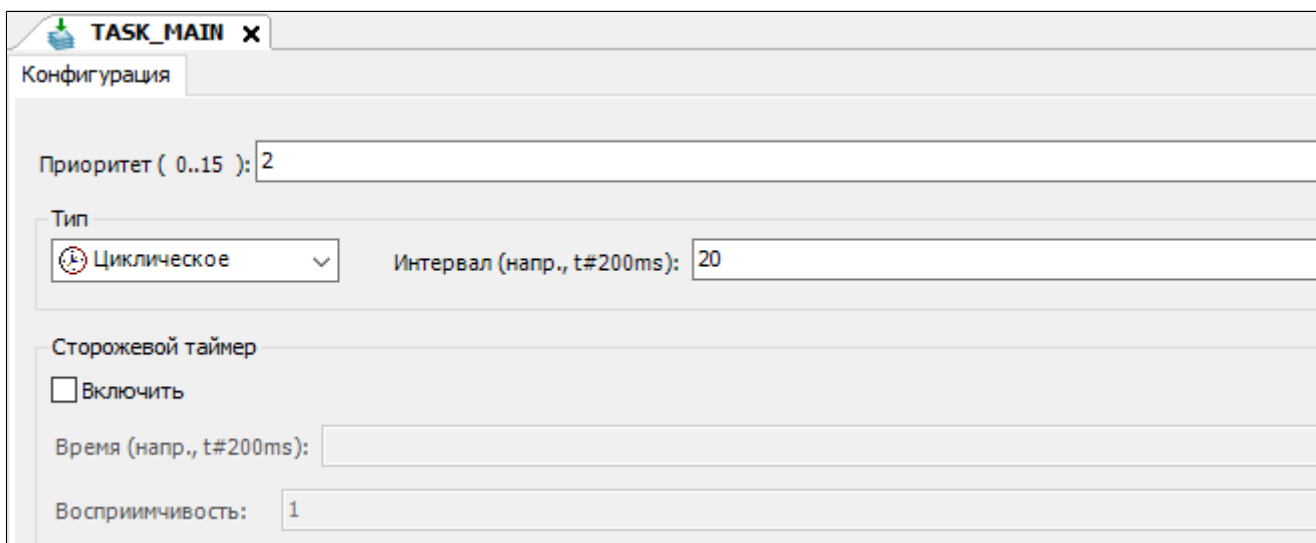


7. Нажмите правой кнопкой мыши по объекту "Task" и в контекстном меню выберите "Рефакторинг" → "Переименовать 'Task'..."

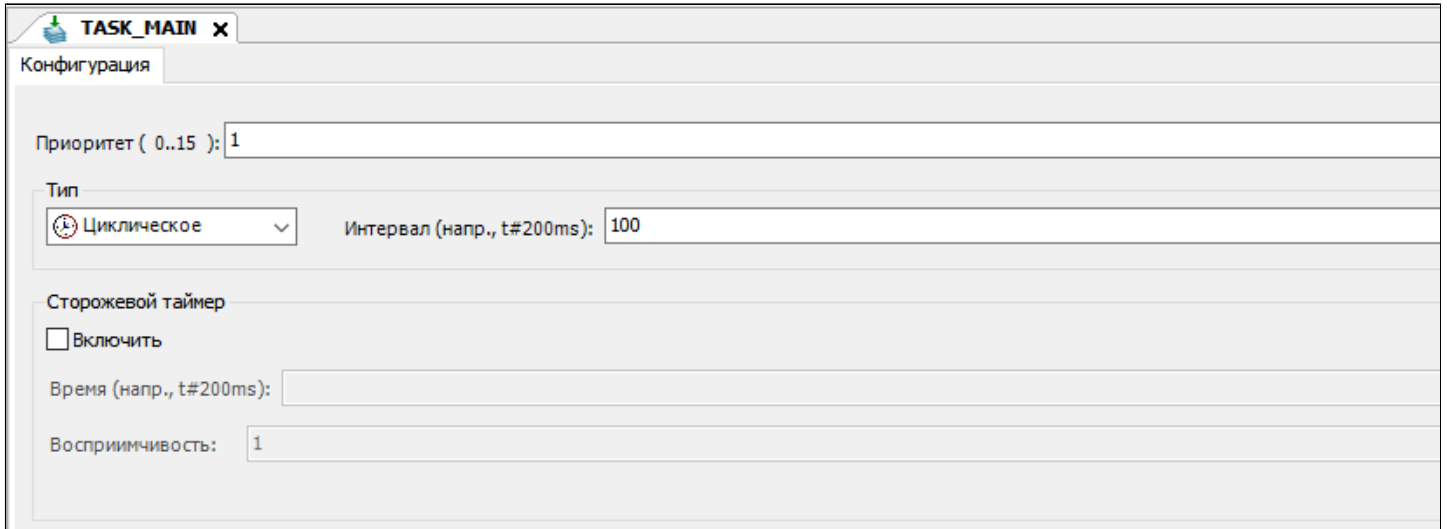


Задайте имя "TASK\_MAIN" и нажмите кнопку "ОК". Подтвердите переименовывание объекта.

8. Откройте задачу "TASK\_MAIN" двойным кликом левой кнопки мыши. Задача имеет следующие параметры: приоритет – 2, тип исполнения – "Циклическое" и интервал – 20 мс.



В конфигурации задачи задайте приоритет – 1 и интервал – 100 мс.



TASK\_MAIN x

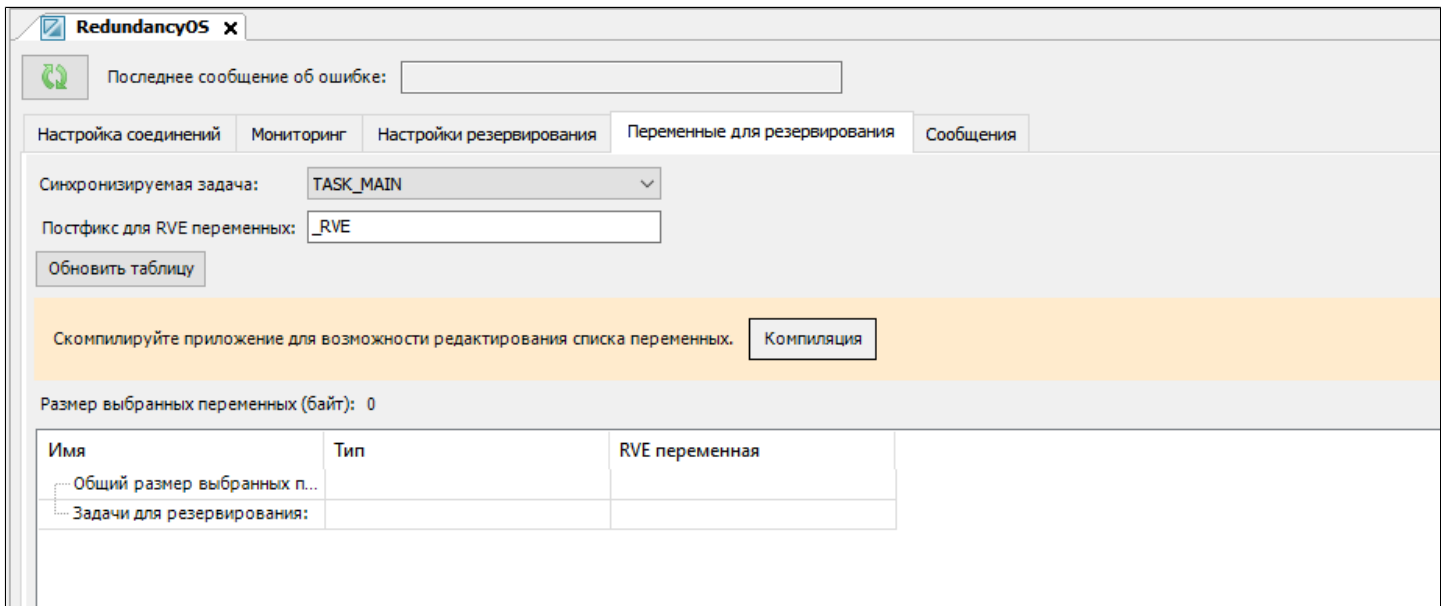
Конфигурация

Приоритет ( 0..15 ): 1

Тип  
Циклическое    Интервал (напр., t#200ms): 100

Сторожевой таймер  
 Включить  
Время (напр., t#200ms):  
Восприимчивость: 1

9. Перейдите в компоненте "RedundancyOS" во вкладку "Переменные для резервирования" и выберите в выпадающем списке синхронизируемую задачу "TASK\_MAIN", нажмите кнопку "Компиляция".



RedundancyOS x

Последнее сообщение об ошибке:

Настройка соединений    Мониторинг    Настройки резервирования    **Переменные для резервирования**    Сообщения

Синхронизируемая задача: TASK\_MAIN

Постфикс для RVE переменных: \_RVE

Обновить таблицу

Скомпилируйте приложение для возможности редактирования списка переменных. **Компиляция**

Размер выбранных переменных (байт): 0

Имя	Тип	RVE переменная
Общий размер выбранных п...		
Задачи для резервирования:		

# 1.1.2.1.1.4. Конфигурирование сервера OPC UA

1. Проверьте, что вы подключены к ПЛК А и перейдите во вкладку "Сервис ПЛК".

The screenshot shows the configuration interface for a REGUL\_R500 device. The left sidebar contains a menu with the following items: "Установки соединения", "Приложения", "Резервное копирование и восстановление", "Файлы", "Журнал", "Установки ПЛК", "Оболочка ПЛК", "Пользователи и группы", "Права доступа", "Символьные права", "Licensed Software Metrics", "Конфигуратор OPC UA", and "Сервис ПЛК". The "Сервис ПЛК" item is currently selected.

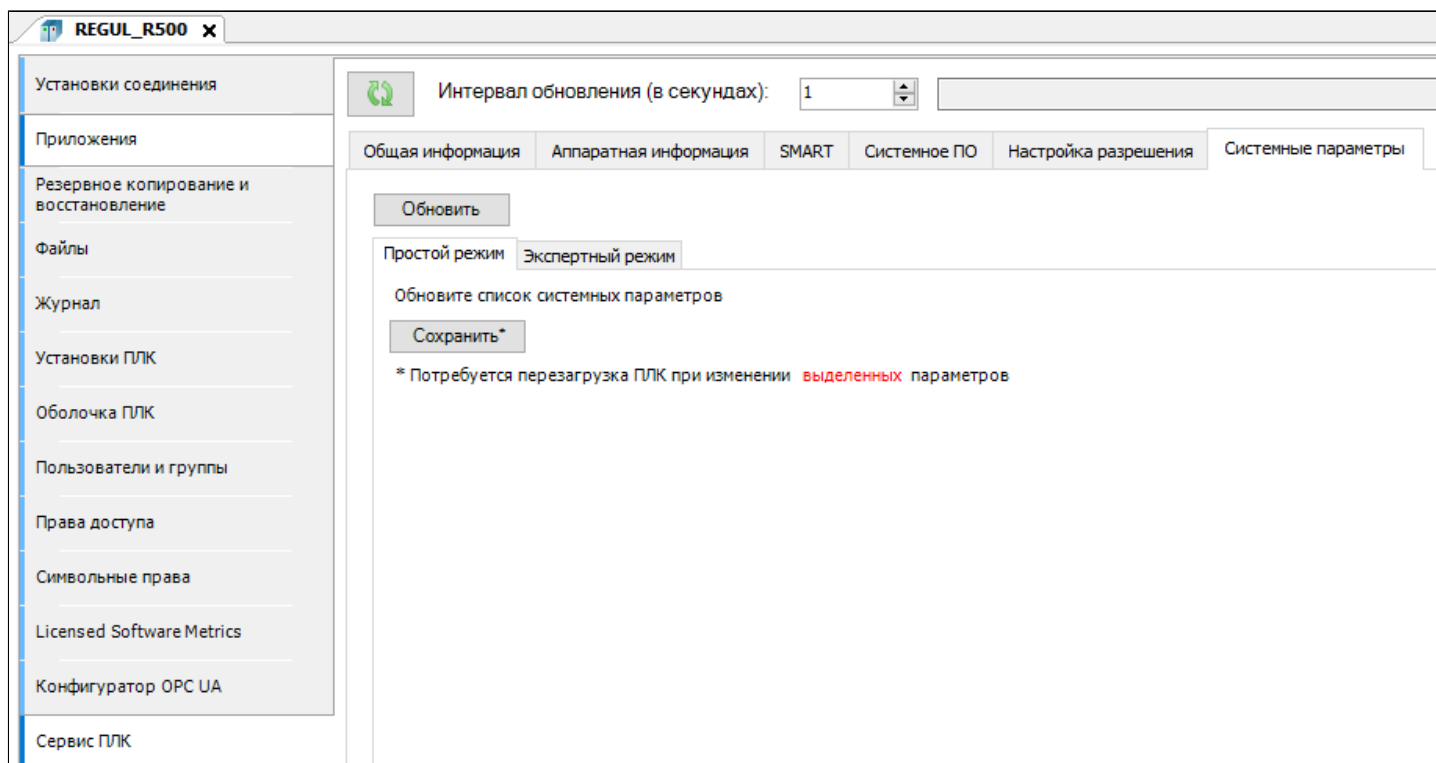
The main content area is titled "Интервал обновления (в секундах):" with a value of 1. Below this, there are several tabs: "Общая информация", "Аппаратная информация", "SMART", "Системное ПО", "Настройка разрешения", and "Системные параметры". The "Общая информация" tab is active, displaying the following information:

- Версия системного ПО: н/д
- Среда исполнения: н/д
- Версия среды исполнения: н/д

Below this information is a dropdown menu labeled "сеть и хранение данных" with a downward arrow. Underneath is a section titled "Сетевые интерфейсы" which contains a table with the following columns: "Имя", "Линк", "Получено пакетов", "Получено байт", "Передано пакетов", "Передано байт", "Ошибки выравнивания", and "Ошибки Fcs". The table is currently empty.

At the bottom of the main content area is a section titled "Хранение данных" which contains a table with the following columns: "Раздел", "Всего (Мб)", and "Занято (Мб)". This table is also empty.

2. Откройте вкладку "Системные параметры" и нажмите кнопку "Обновить".




3. На экран будет выведена информация о текущем состоянии доступных параметров. Убедитесь, что флаг "Запуск сервера OPC UA" установлен.

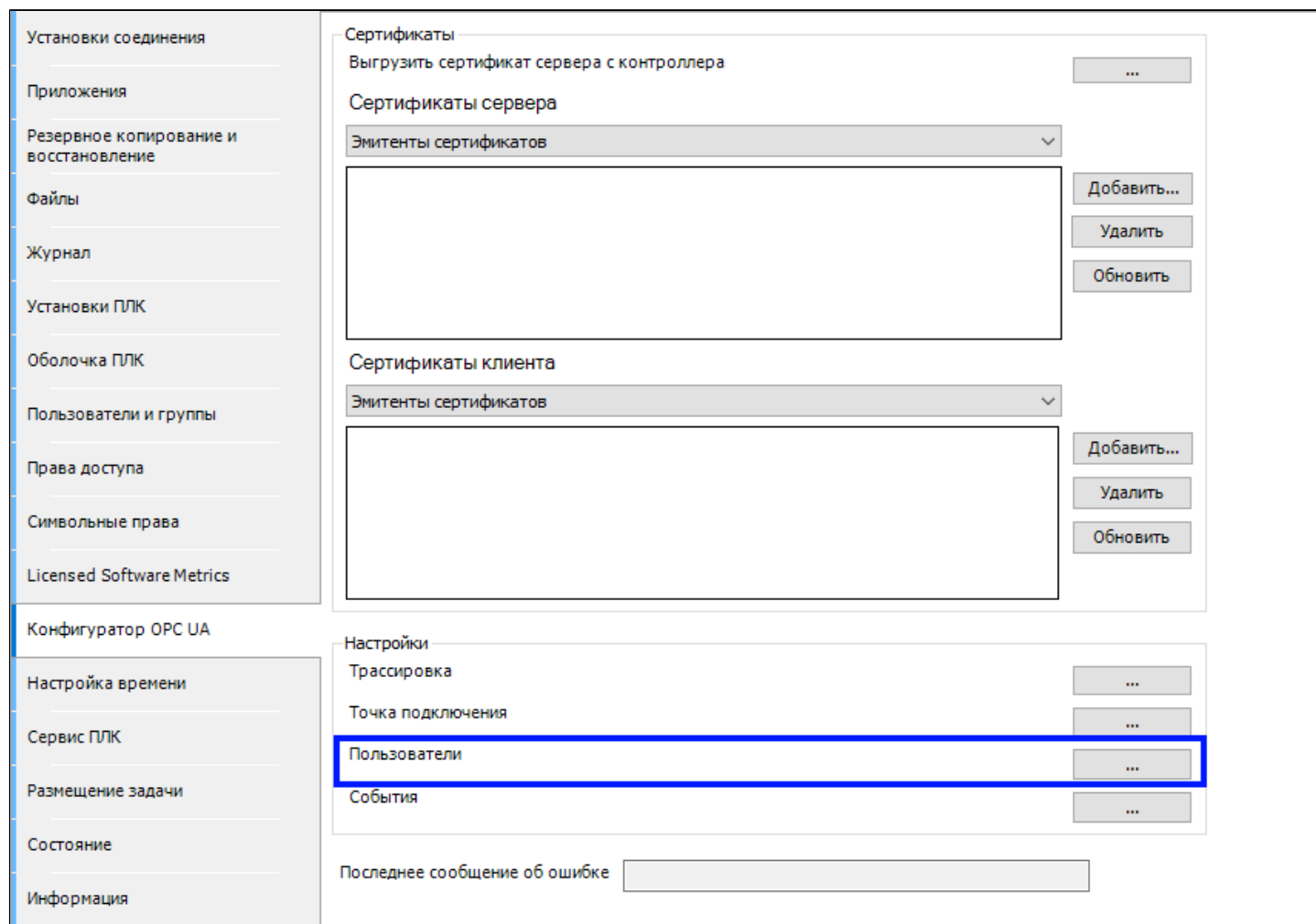


Если флаг "Запуск сервера OPC UA" снят, его нужно установить и нажать кнопку "Сохранить".

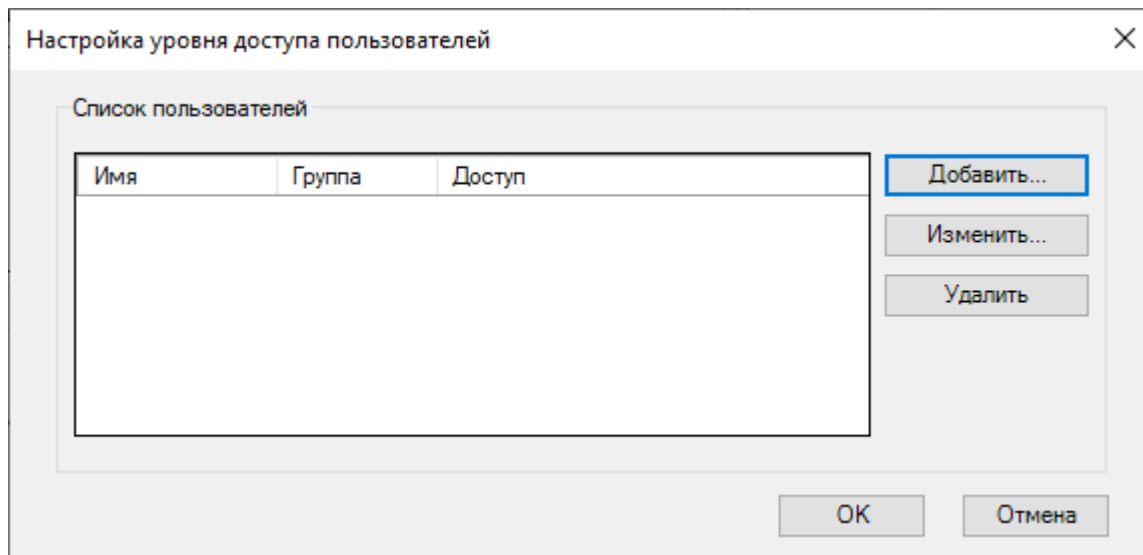
The screenshot shows the 'REGUL\_R500' configuration window. The left sidebar contains a menu with items like 'Установки соединения', 'Приложения', 'Резервное копирование и восстановление', 'Файлы', 'Журнал', 'Установки ПЛК', 'Оболочка ПЛК', 'Пользователи и группы', 'Права доступа', 'Символьные права', 'Licensed Software Metrics', 'Конфигуратор OPC UA', 'Сервис ПЛК', and 'Настройка времени'. The main area is titled 'Интервал обновления (в секундах): 1' and has tabs for 'Общая информация', 'Аппаратная информация', 'SMART', 'Системное ПО', 'Настройка разрешения', and 'Системные параметры'. The 'Системные параметры' tab is active, showing a 'Обновить' button and two modes: 'Простой режим' and 'Экспертный режим'. Under 'PlcServices', the following options are checked: 'Запуск сервера SQL', 'Предоставление сетевого доступа к БД MySQL', and 'Запуск сервера OPC UA'. Other options like 'Запуск сервера FTP', 'Разрешение на подключение внешних накопителей', and 'Запуск службы печати LPD' are unchecked. Under 'PsOpcUaServer', 'Запуск сервера OPC UA' is checked. Under 'global', 'Запрет на изменение сетевых настроек' is unchecked. A 'Сохранить\*' button is at the bottom. A note states: '\* Потребуется перезагрузка ПЛК при изменении выделенных параметров'.



4. Перейдите на вкладку "Конфигуратор OPC UA" и в разделе настройки нажмите на кнопку  напротив строчки "Пользователи".



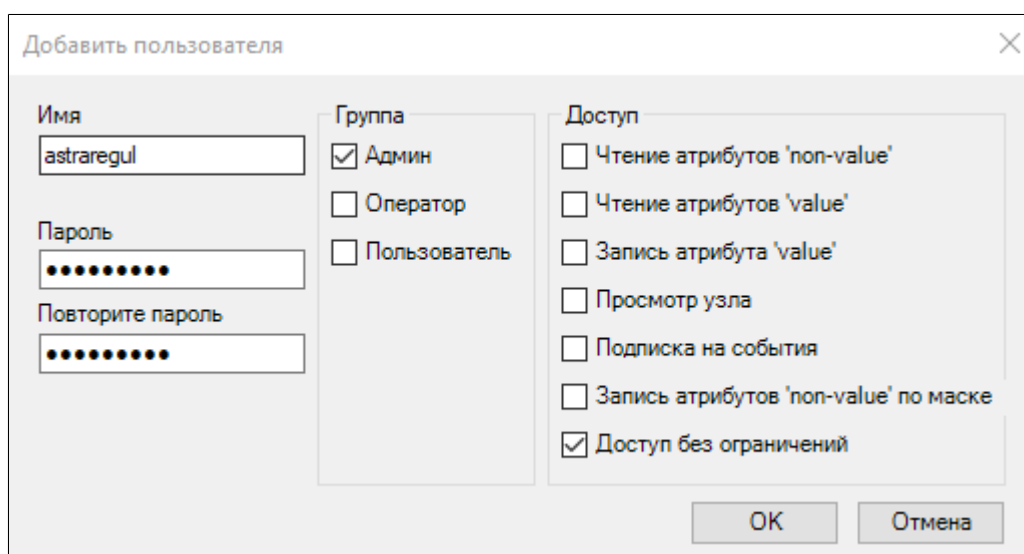
5. Откроется окно "Настройка уровня доступа пользователей". Нажмите кнопку "Добавить...".



6. В открывшемся окне установите следующие параметры:

- › Имя: astraregul;
- › Пароль: astraregul;
- › Группа "Админ";
- › Установите флаг "Доступ без ограничений".

Нажмите кнопку "ОК".



После ввода нового пароля выйдет уведомление о смене пароля, подтвердите изменение текущего пароля.

Должен получиться следующий пользователь с доступом "Full access":

Настройка уровня доступа пользователей



Список пользователей

Имя	Группа	Доступ
astraregul	Admin	Full access

Добавить...

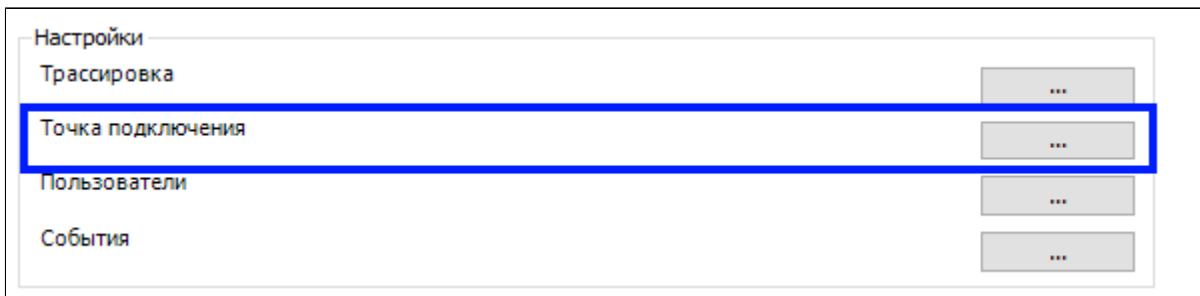
Изменить...

Удалить

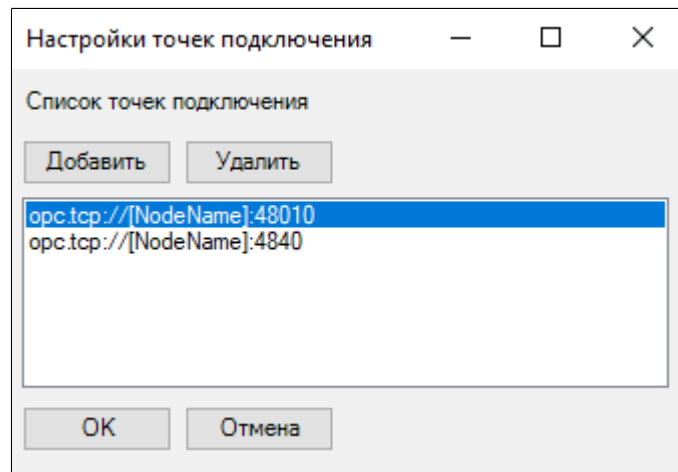
OK

Отмена

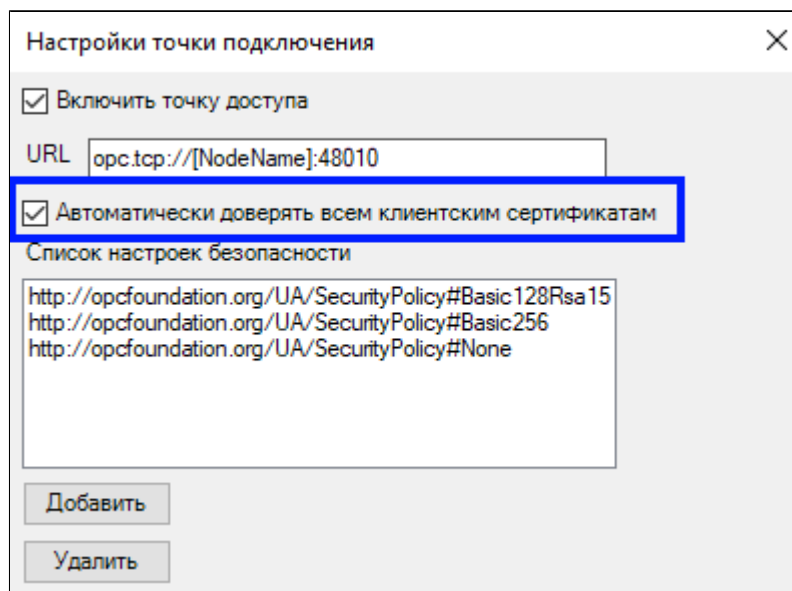
7. Нажмите на кнопку  напротив строки "Точка подключения".



8. В открывшемся окне двойным щелчком левой кнопкой мыши перейдите в настройки точки доступа по порту 48010.



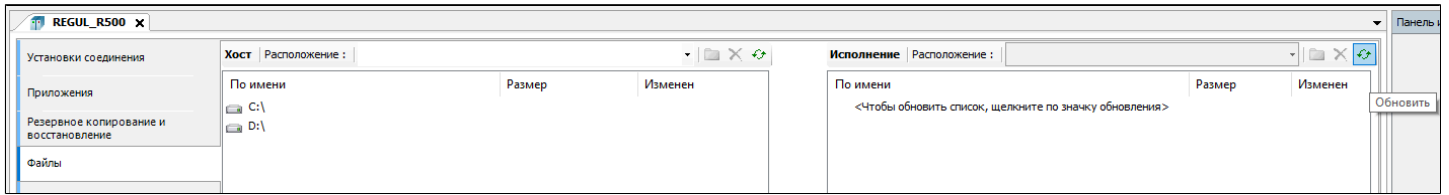
9. В настройках точки доступа убедитесь, что установлен флаг "Автоматически доверять всем клиентским сертификатам".



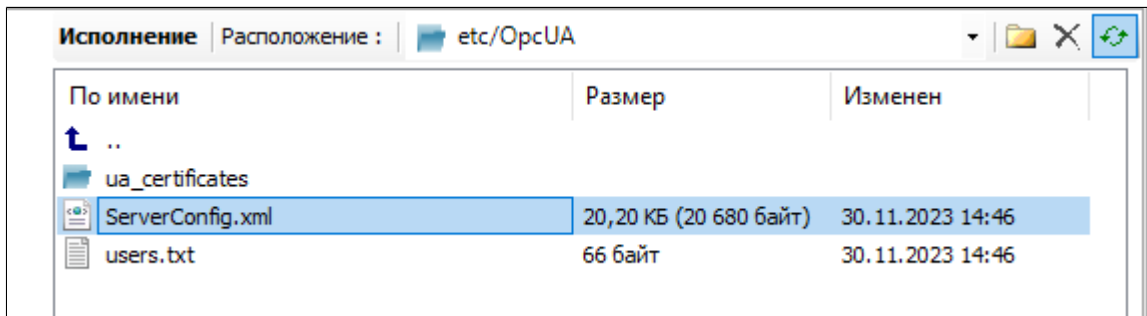


Если флаг "Автоматически доверять всем клиентским сертификатам" снят, его нужно установить.

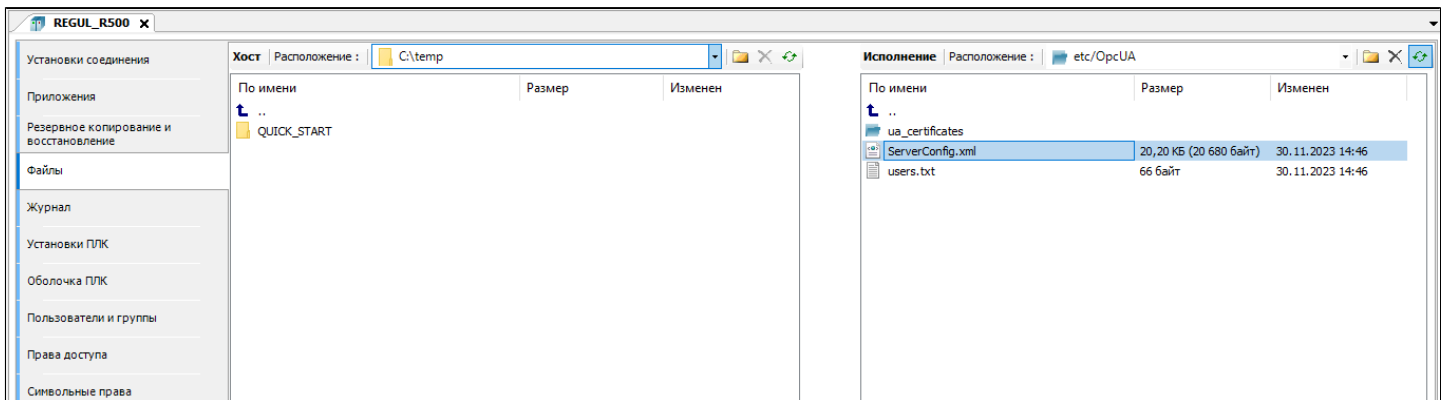
10. Перейдите во вкладку "Файлы" и обновите информацию о файловом расположении ПЛК нажав кнопку "Обновить".




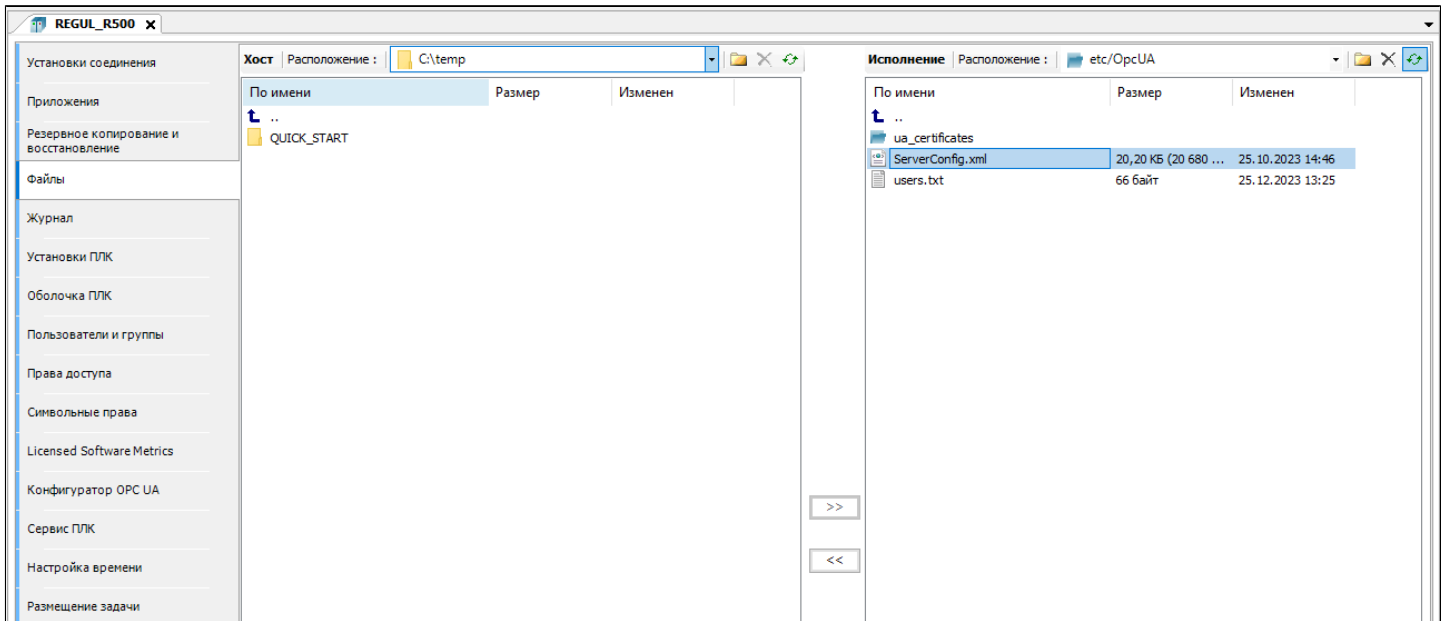
11. Перейдите по пути "etc/OpcUa" и выделите файл "ServerConfig.xml".



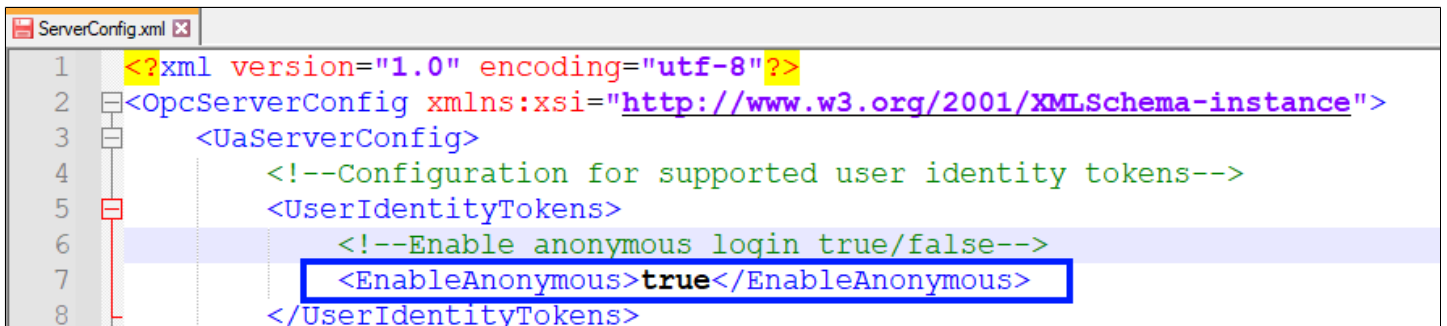
12. Выберите папку в левом окне для сохранения файла с ПЛК на компьютер, например "C:\temp".



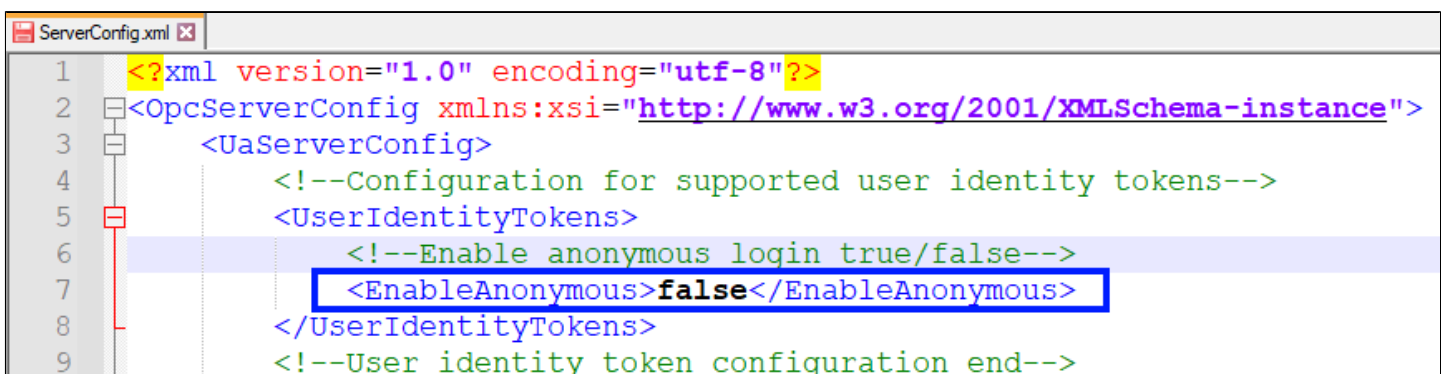
13. Нажмите на клавишу , чтобы переместить файл "ServerConfig.xml" в выбранную папку.



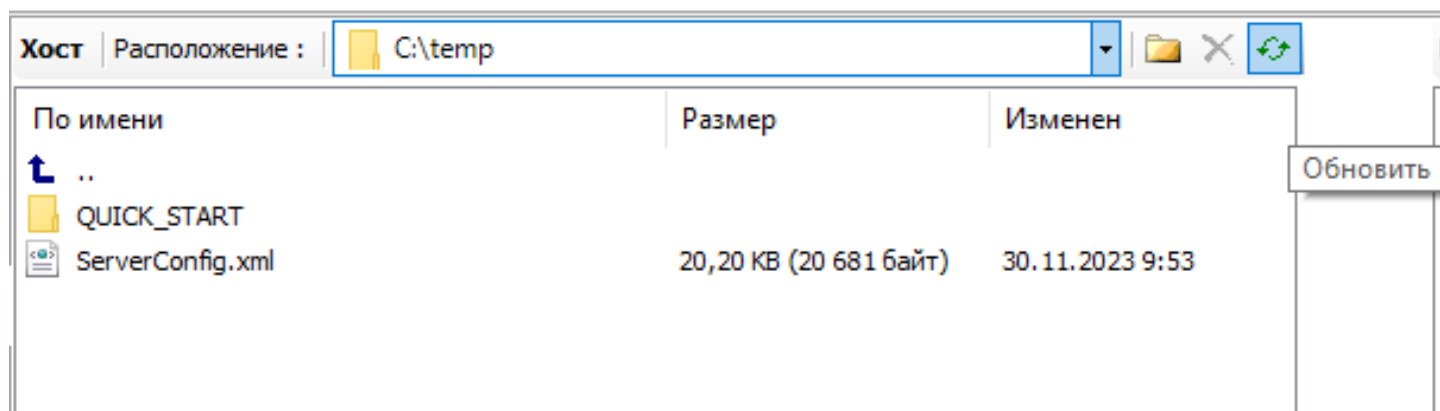
14. Перейдите в папку с файлом и откройте его при помощи редактора Notepad ++.




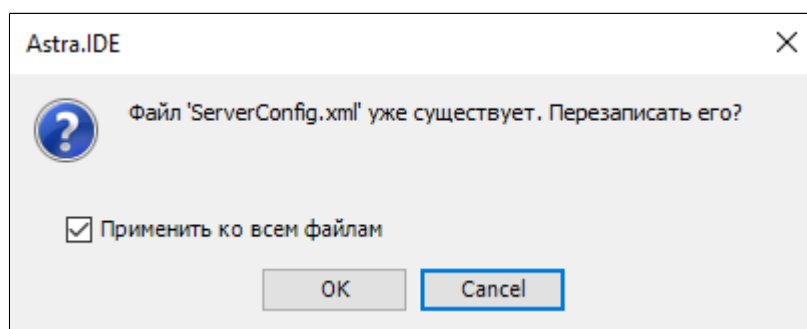
15. Измените значение "true" на "false" в строке <EnableAnonymous> и сохраните измененный файл.



16. Перейдите в Astra.IDE и обновите файловую конфигурацию ОС нажав клавишу "Обновить".



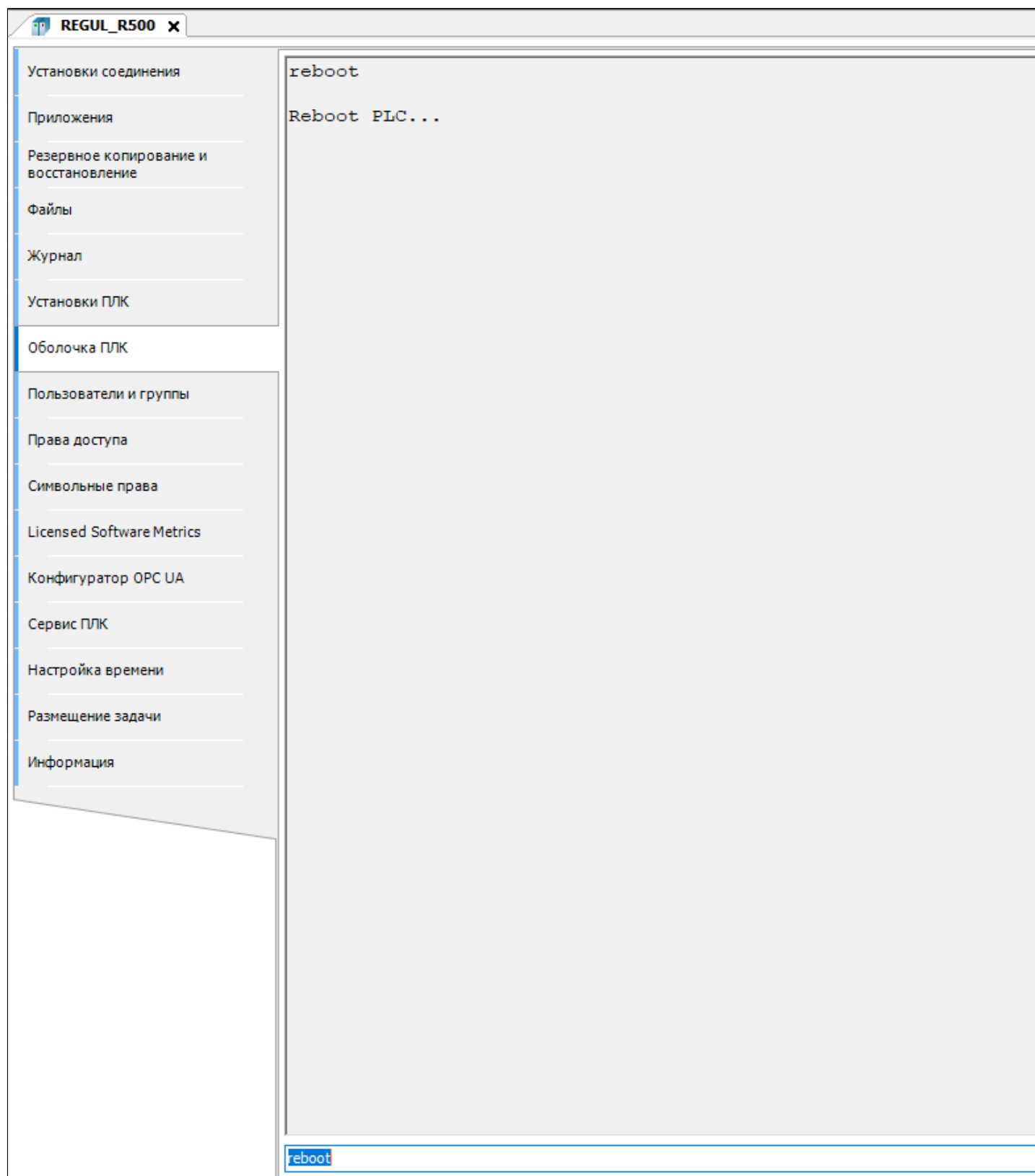
17. Передайте измененный файл в ПЛК, выделите его и нажмите на кнопку . Появится следующее всплывающее окно, нажмите кнопку "OK".



После проделанных действий необходимо перезагрузить ПЛК.

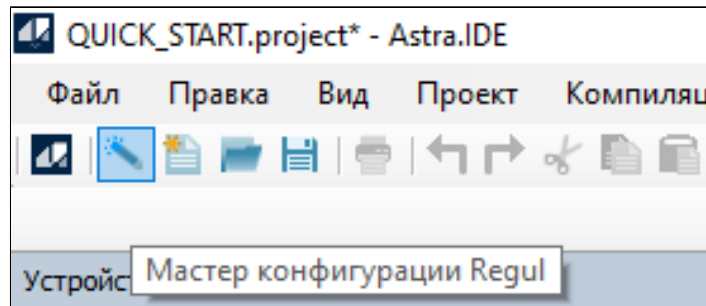


18. Перейдите во вкладку "Оболочка ПЛК". Введите в поле ввода команду "reboot" и нажмите клавишу "Enter". ПЛК будет перезагружен.

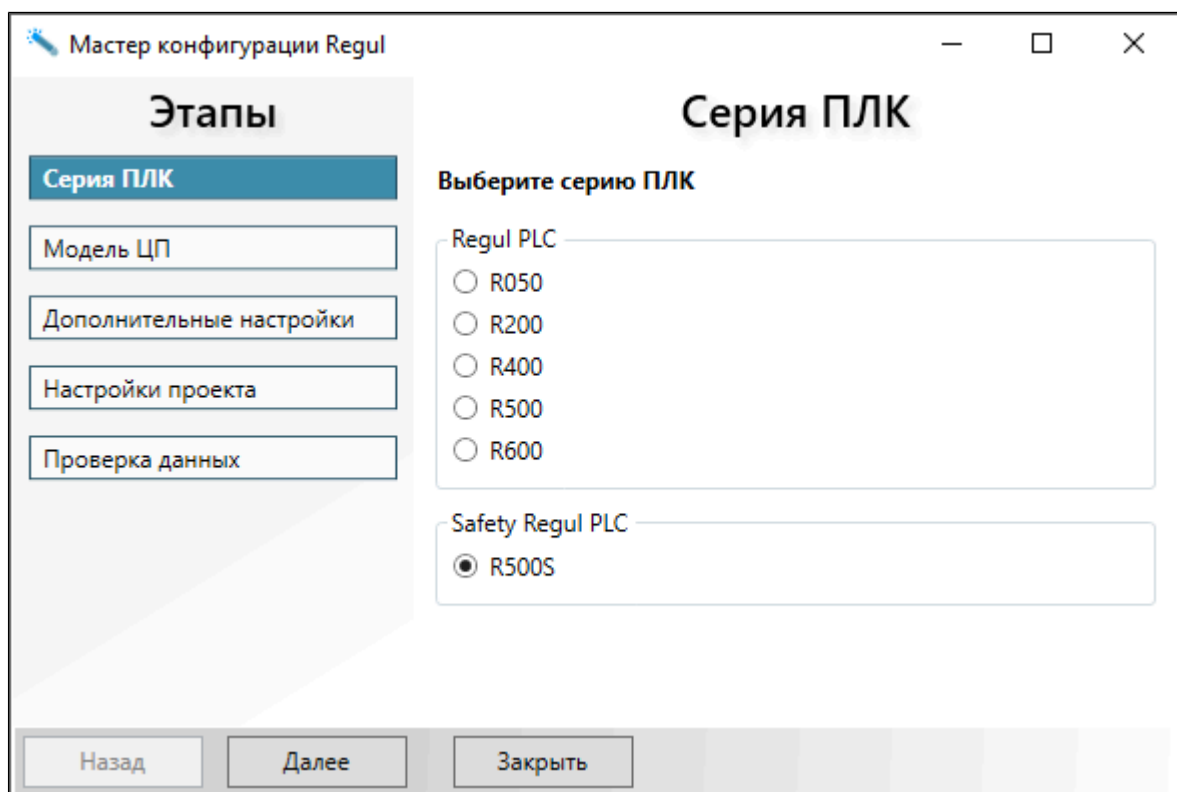


## 1.1.2.1.2. ПЛК SAFETY REGUL R500S

1. Для добавления в созданный проект контроллера SAFETY REGUL R500S, нажмите кнопку "Мастер конфигурации Regul" на панели инструментов.

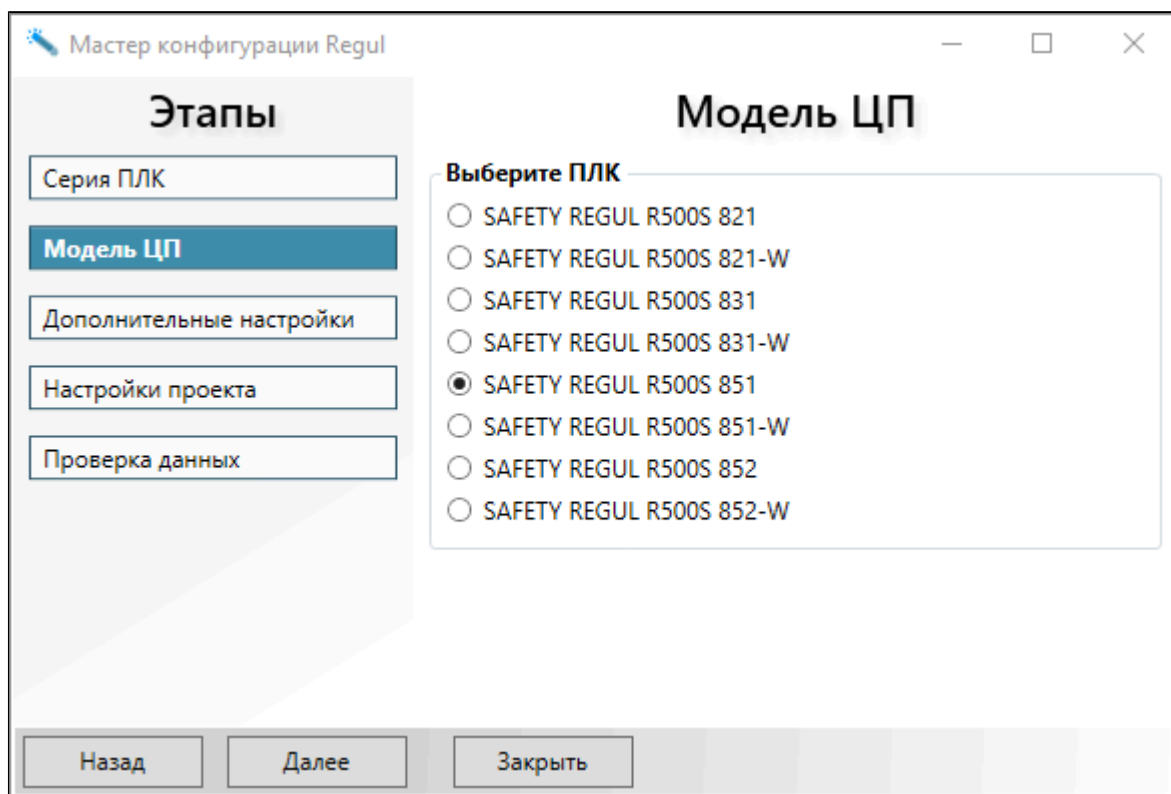


2. В открывшемся окне выберите серию контроллера "R500S" и нажмите кнопку "Далее".



Для отображения серии ПЛК R500S необходимо установить пакет Safety в Astra.IDE.

3. Выберите "Модель ЦП" – "SAFETY REGUL R500S 851" и нажмите кнопку "Далее".



4. Снимите флаг "Создать задачу и POU". В разделе "Поддержка дублирования" выберите "Без дублирования". Выберите шину "RegulBus OS" с расширяемым контроллером. Нажмите кнопку "Далее".

## Этапы

Серия ПЛК

Модель ЦП

**Дополнительные настройки**

Настройки проекта

Проверка данных

## Дополнительные настройки

Создать задачу и ROU

### Поддержка дублирования

- Дублирование
- Без дублирования

### Выберите шину

#### RegulBus OS

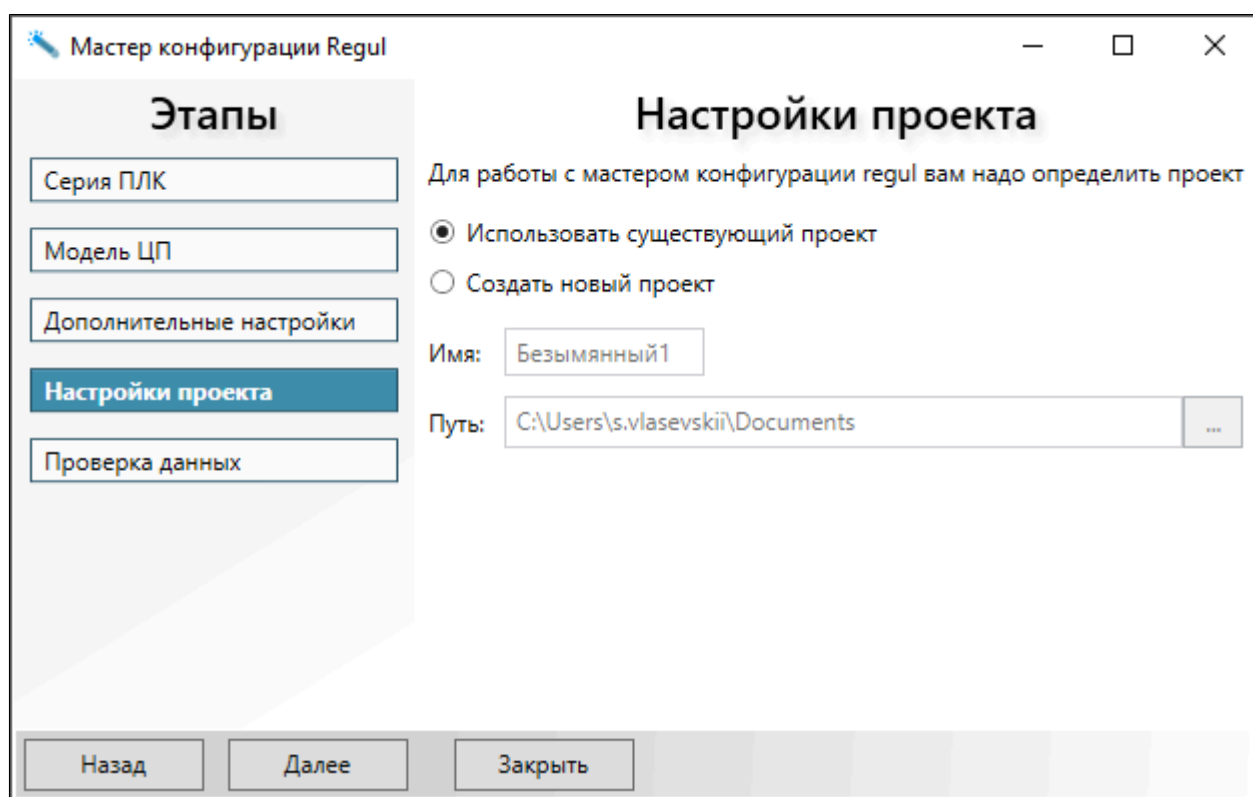
- Расширяемый контроллер (с возможностью подключения крейтов расширения)
- Нерасширяемый контроллер R500 (состоит из 1 крейта)

Назад

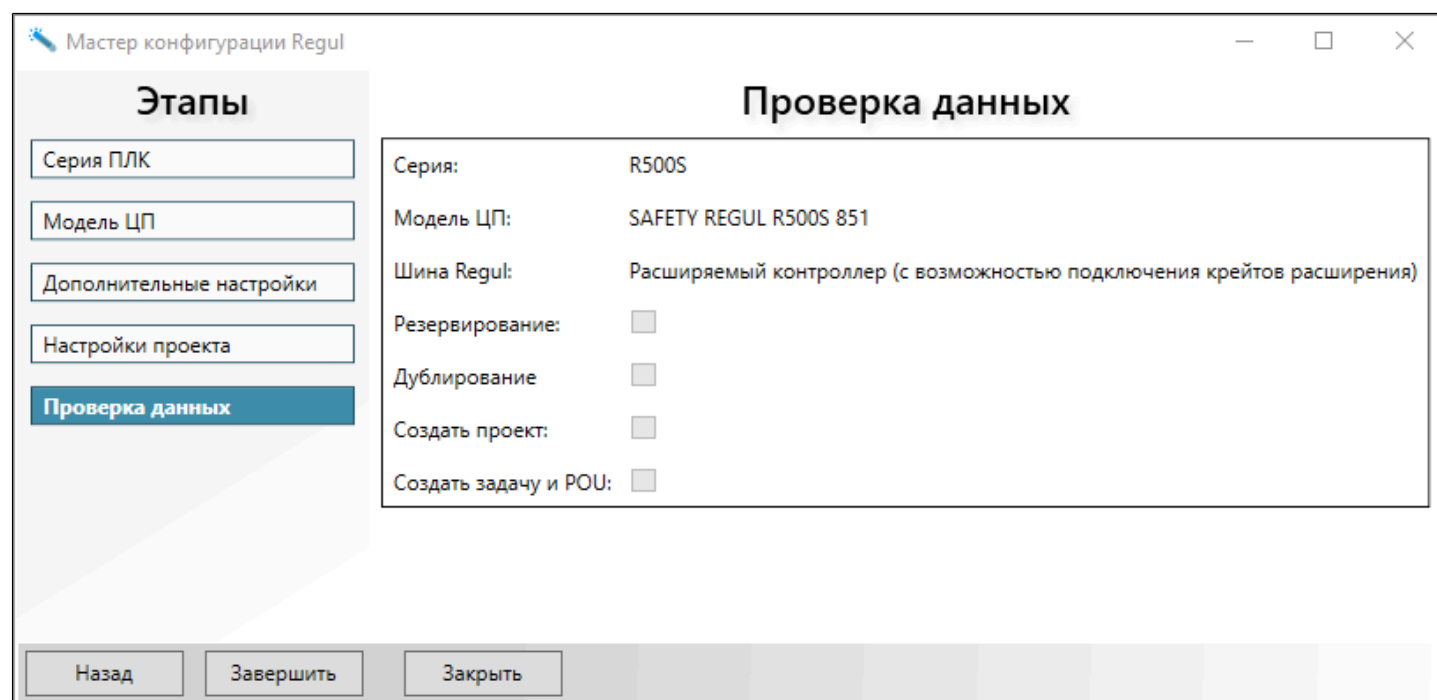
Далее

Заккрыть

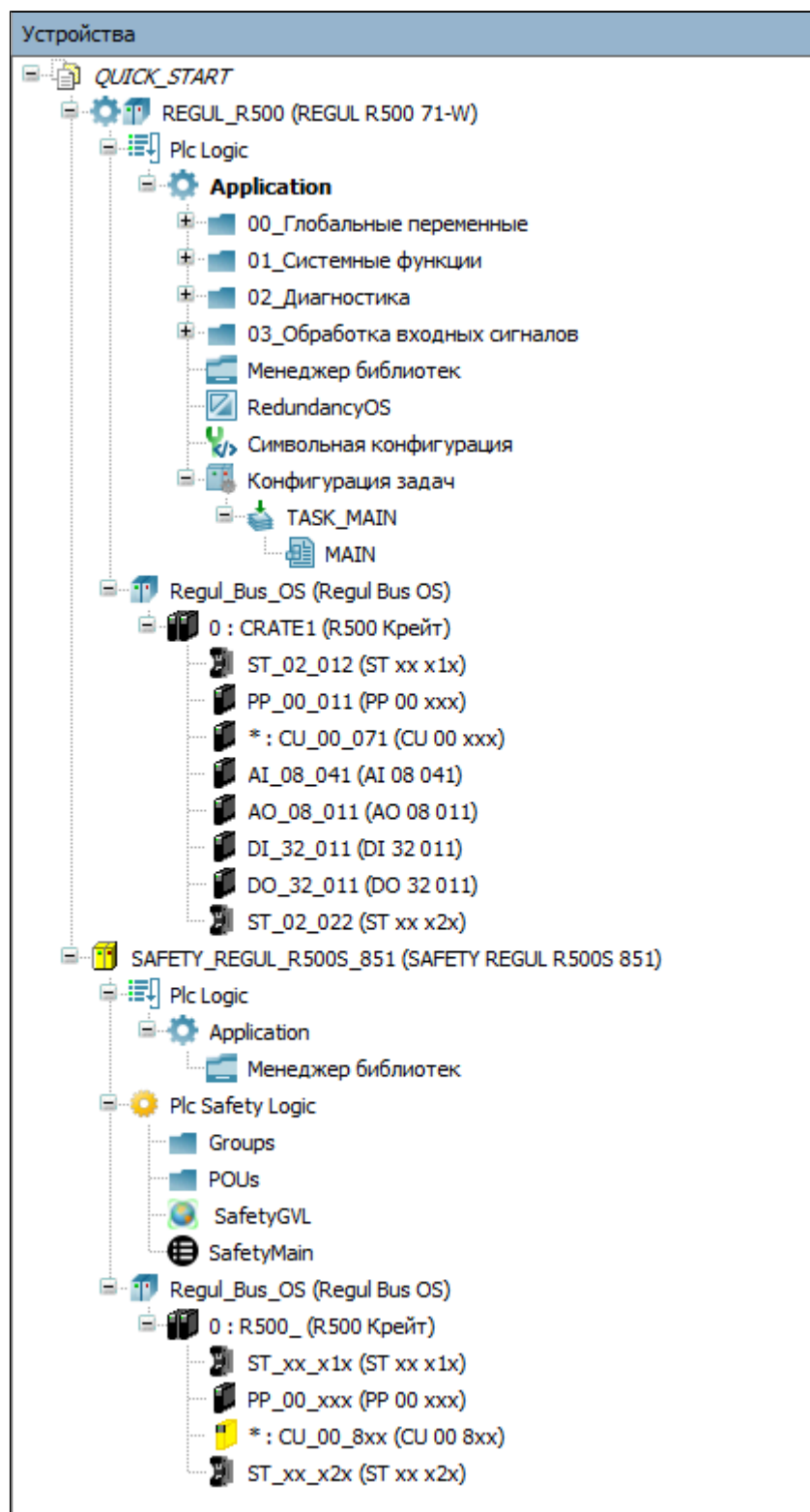
5. Выберите пункт "Использовать существующий проект", для создания шаблона проекта контроллера SAFETY REGUL R500S в проекте. Нажмите кнопку "Далее".



6. Проверьте настройки и нажмите кнопку "Завершить".



7. В окне среды разработки Astra.IDE добавится шаблон проекта контроллера REGUL R500S.



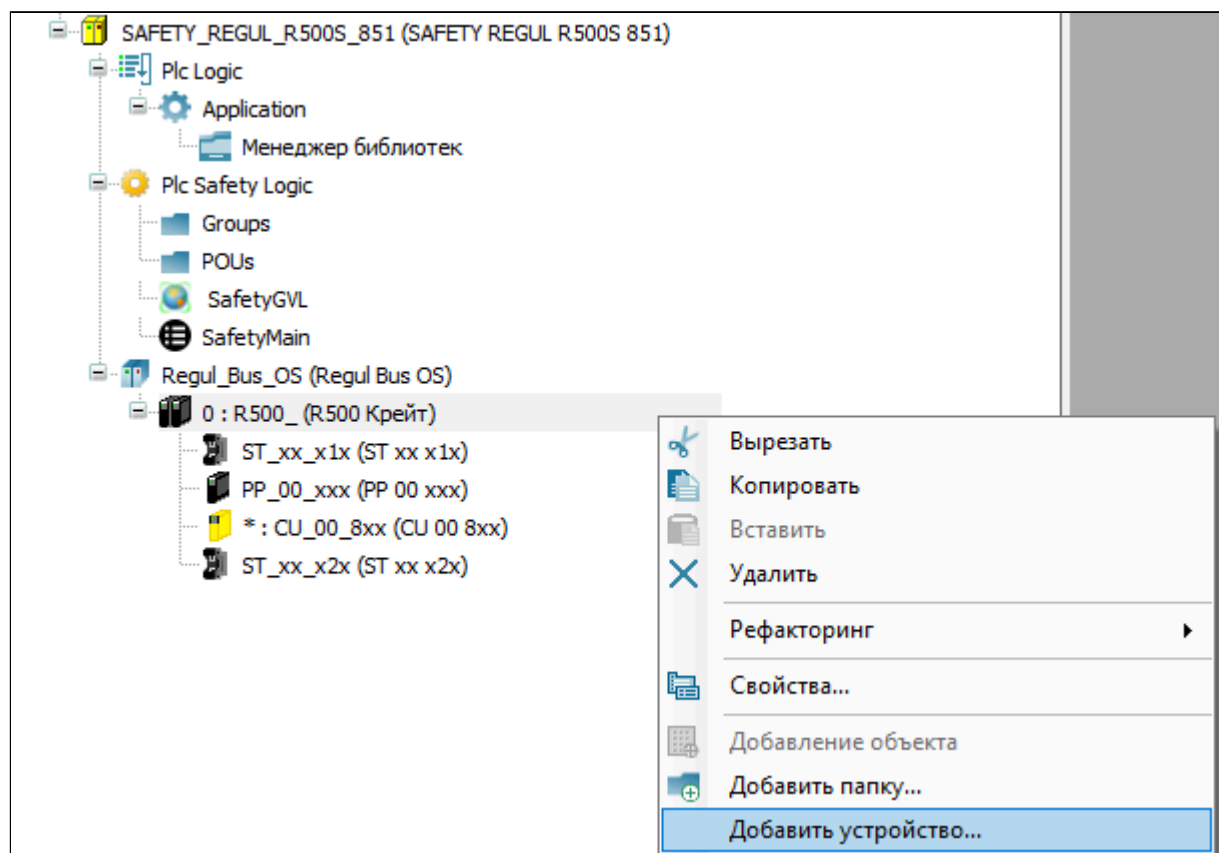
Для унификации в автоматизации разработки проектов запрещается изменять название приложения "Application".

## 1.1.2.1.2.1. Создание конфигурации крейта ПЛК SAFETY REGUL R500S

Конфигурация крейта ПЛК R500S:

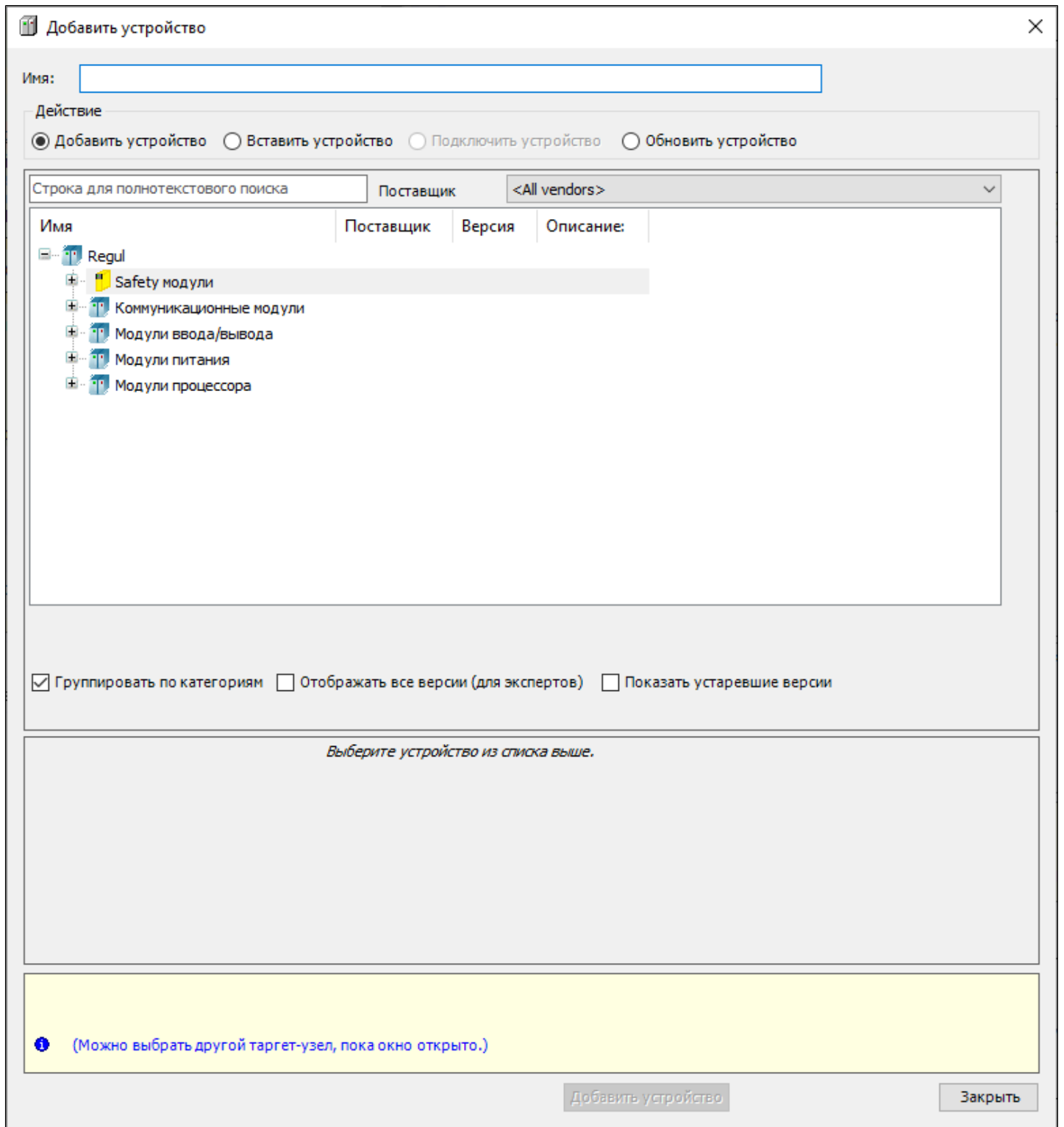
1. Модуль оконечный R500 ST 02 012.
2. Модуль источника питания R500 PP 00 011.
3. Центральный процессор R500S CU 00 851.
4. Модуль аналогового ввода R500S AI 08 811.
5. Модуль аналогового вывода R500S AO 04 831.
6. Модуль дискретного ввода R500S DI 28 811.
7. Модуль дискретного вывода R500S DO 16 811.
8. Модуль оконечный R500 ST 02 022.

1. Для добавления модулей кликните правой кнопкой мыши по крейту и в выпадающем меню выберите пункт "Добавить устройство".



Откроется диалоговое окно:





2. Выберите нужный модуль и нажмите кнопку "Добавить устройство".

Добавить устройство



Имя: AI\_08\_881

Действие

Добавить устройство
  Вставить устройство
  Подключить устройство
  Обновить устройство

Строка для полнотекстового поиска

Поставщик

<All vendors>

Имя	Поставщик	Версия	Описание:
Regul			
Safety модули			
AI 04 841	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль аналогового ввода, 0...20, 4...20 мА, 4 ка
AI 04 861	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль аналогового ввода, 0...+5 В, -5 В...+5 В,
AI 08 851	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль аналогового ввода, 0...+5 В, -5 В...+5 В,
AI 08 881	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль аналогового ввода, 0...20, 4...20 мА, 8 ка
AO 04 831	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль аналогового вывода, 4...20 мА, 0...+10 В,
CU 00 8xx	"Prosoft-Systems" Ltd.	2.0.2.0	Модуль центрального процессора
DI 16 831	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль дискретного ввода NAMUR, 16 каналов. М
DI 28 811	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль дискретного ввода, 24 VDC, 28 каналов (4
DO 08 821	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль дискретного вывода, 220 VDC, 6 А, 8 кана
DO 16 811	"Prosoft-Systems" Ltd.	2.0.230.0	Модуль дискретного вывода, 24 VDC, 0.6 А, 16 ка
Коммуникационные модули			

Группировать по категориям
  Отображать все версии (для экспертов)
  Показать устаревшие версии

**Имя:** AI 08 881  
**Производитель:** "Prosoft-Systems" Ltd.  
**Группы:** Safety модули  
**Версия:** 2.0.230.0  
**Номер модели:**  
**Описание:** Модуль аналогового ввода, 0...20, 4...20 мА, 8 каналов, Hart, поканальная гальваническая изоляция. Два АЦП (работают в паре) на все каналы. С функцией питания датчиков от модуля. Минимально допустимая версия СПО модуля: 2.0.0.0 и выше



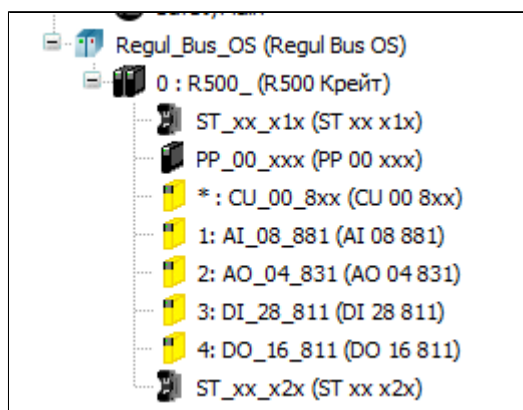
Добавить выбранное устройство как последнего потомка R500\_

(Можно выбрать другой таргет-узел, пока окно открыто.)

Добавить устройство

Закреть

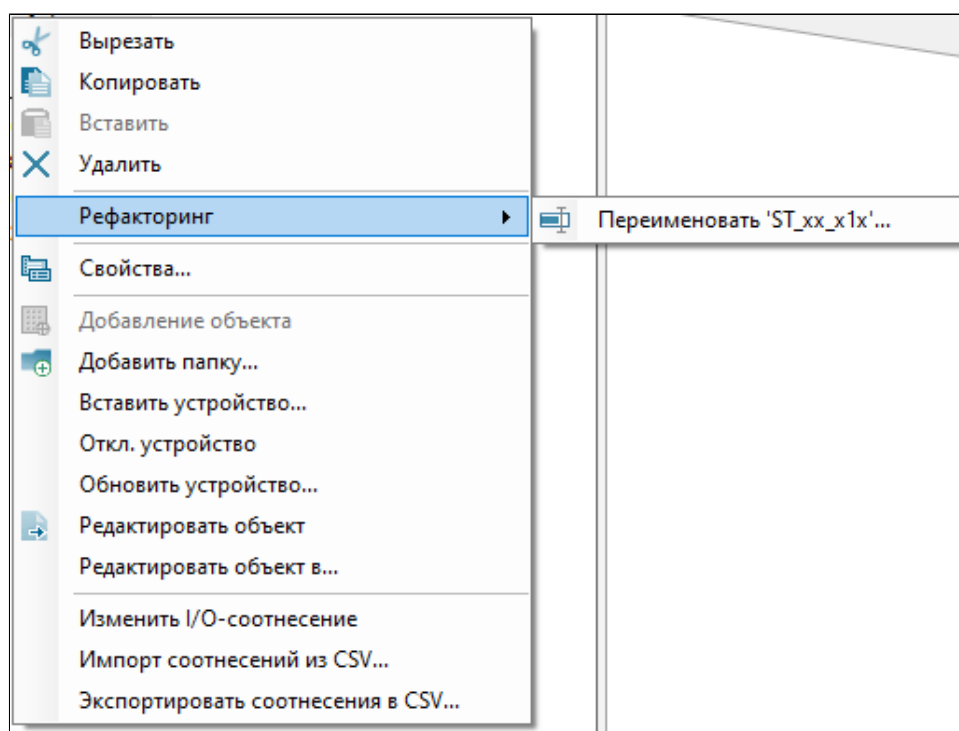
3. Аналогичным образом добавьте все модули. В итоге должна получиться следующая конфигурация:



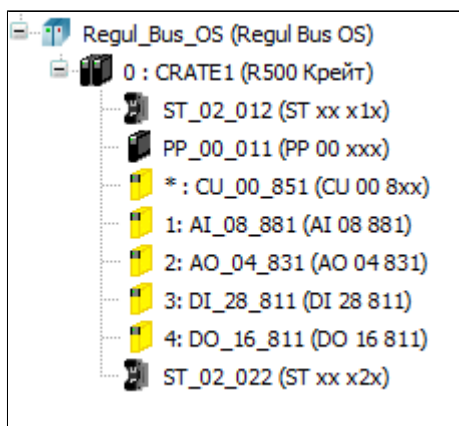
4. Чтобы переименовать модуль в контекстном меню выберите "Рефакторинг" → "Переименовать...".



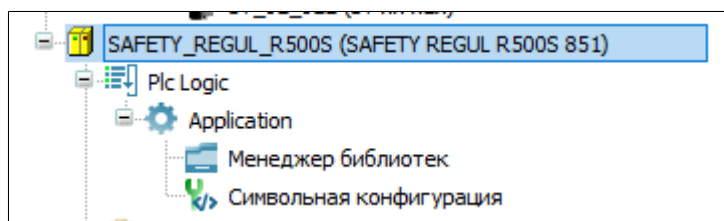
Рекомендуется всем объектам в проекте давать названия, которые отражают их функциональное назначение.



Должна получиться следующая конфигурация:

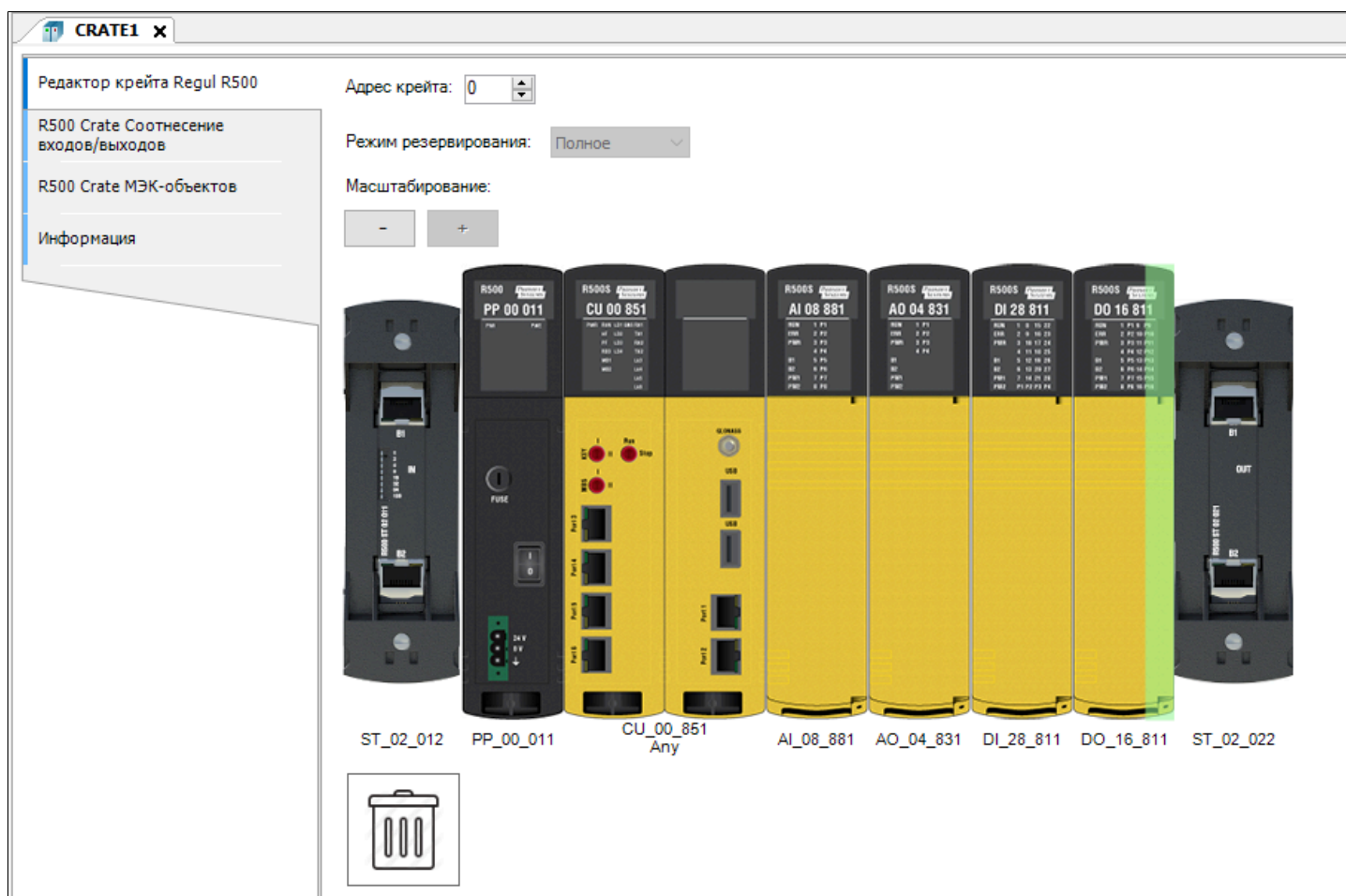


Аналогично переименуйте контроллер на "SAFETY\_REGUL\_R500S".



5. Чтобы открыть окно редактора крейта, кликните по нему два раза левой кнопкой мыши.

6. Для добавления модуля выделите место, в которое необходимо вставить модуль. Область для вызова окна "Вставить устройство" подсветится зеленым цветом.

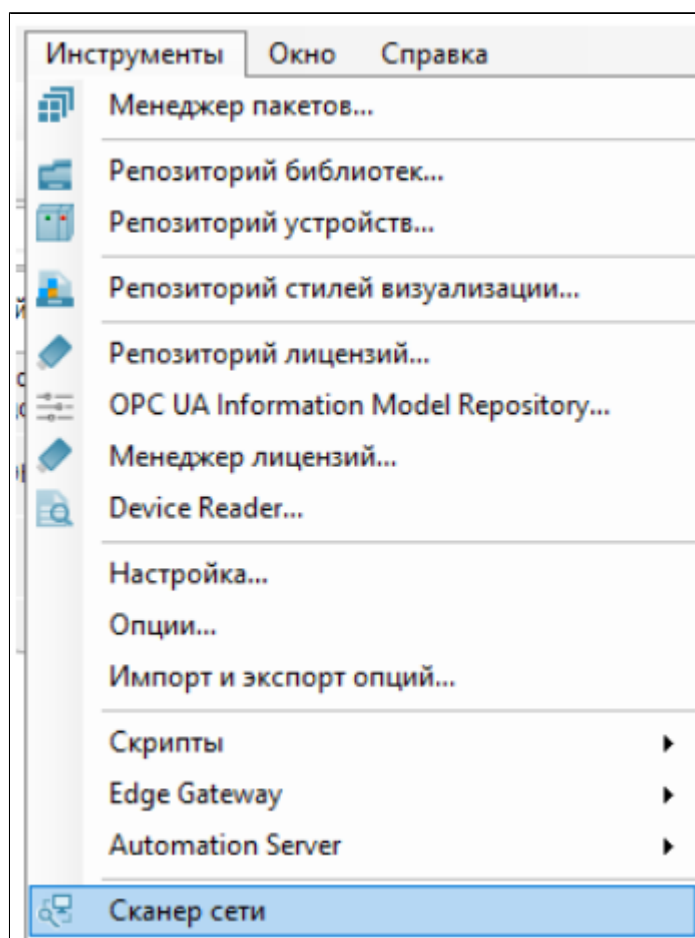


7. Для удаления модуля перетащите его в "Корзину" .

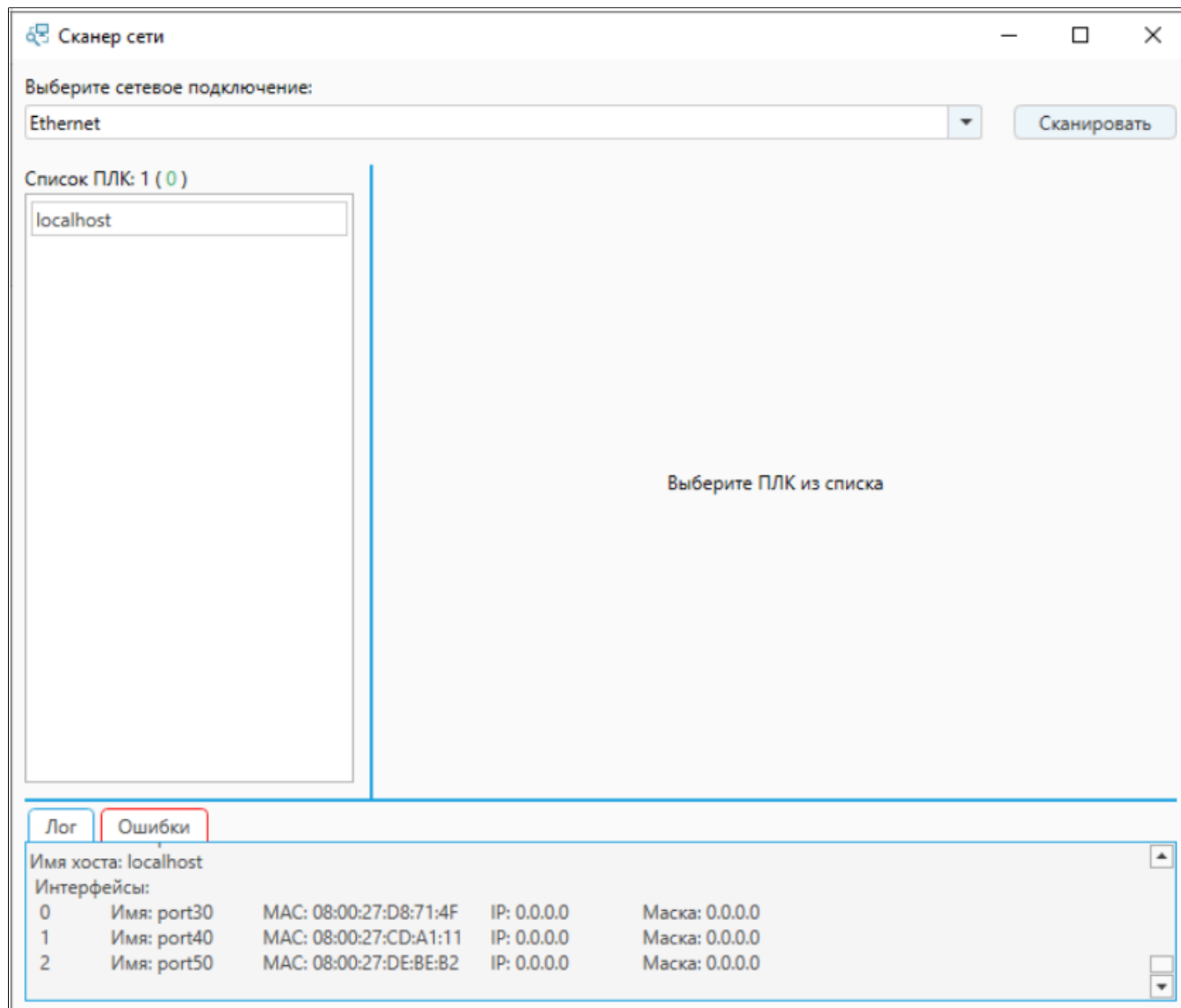
## 1.1.2.1.2.2. Настройка подключения к ПЛК REGUL

Чтобы загрузить разработанный проект в ПЛК, необходимо настроить его сетевые параметры:

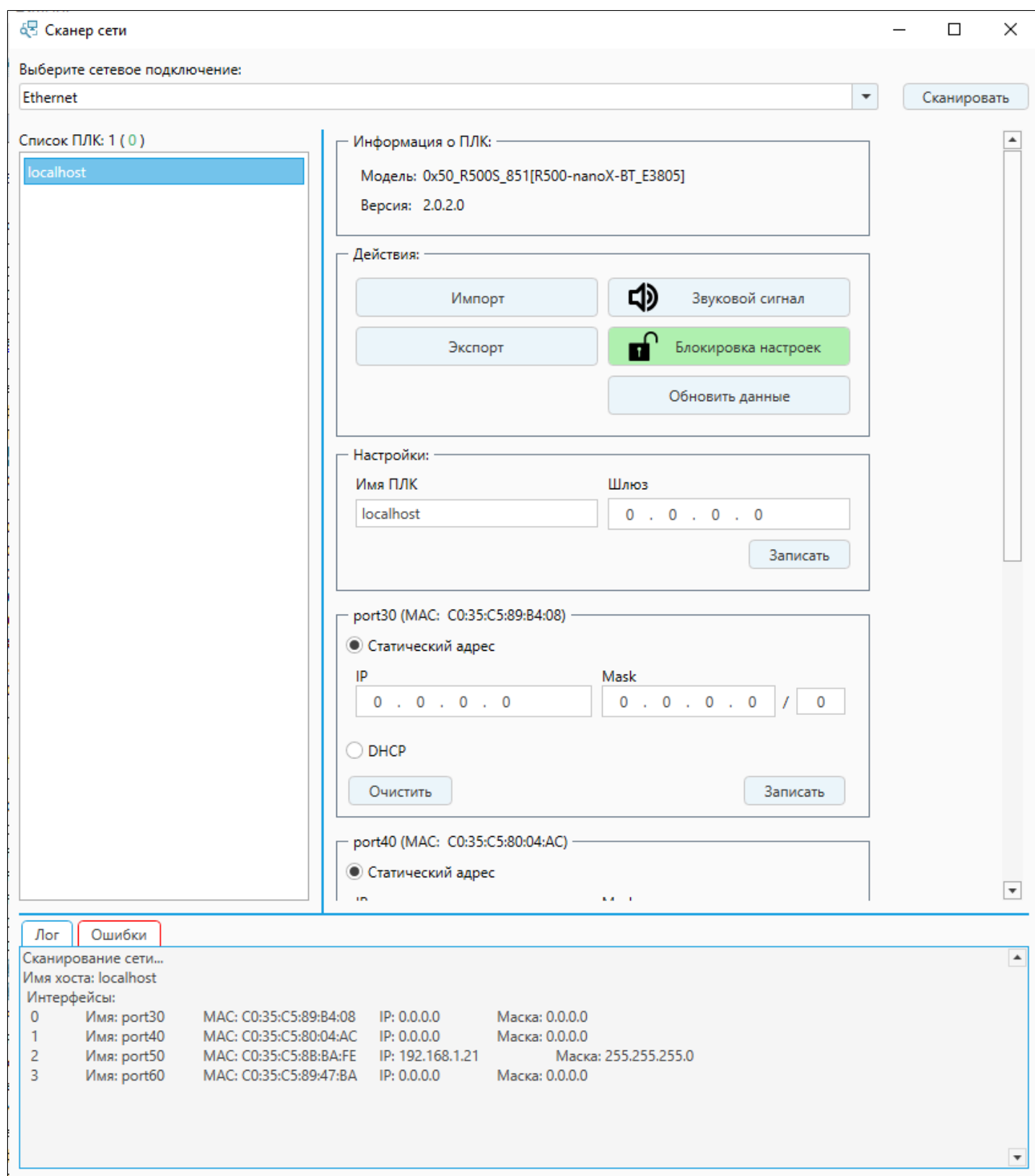
1. Перейдите в меню "Инструменты" и выберите пункт "Сканер сети".



2. В открывшемся диалоговом окне "Сканер сети" выберите сетевое подключение из выпадающего списка и нажмите кнопку "Сканировать". В результате сканирования в списке ПЛК появятся все контроллеры, подключенные к выбранной сети.



3. Выберите контроллер, с которым требуется установить соединение. Чтобы убедиться, что выбран нужный контроллер, нажмите кнопку "Звуковой сигнал". Контроллер издаст кратковременный звуковой сигнал.





#### 4. Настройте ПЛК А:

› Имя ПЛК: SAFETY\_REGUL\_R500S\_A.

› Шлюз: 0.0.0.0.

› port30:

port30 (MAC: C0:35:C5:89:B4:08)

Статический адрес

IP: 0 . 0 . 0 . 0      Mask: 0 . 0 . 0 . 0 / 0

DHCP

› port40:

port40 (MAC: C0:35:C5:80:04:AC)

Статический адрес

IP: 0 . 0 . 0 . 0      Mask: 0 . 0 . 0 . 0 / 0

DHCP

› port50:

port50 (MAC: C0:35:C5:8B:BA:FE)

Статический адрес

IP: 192 . 168 . 1 . 13      Mask: 255 . 255 . 255 . 0 / 24

DHCP

› port60:

port60 (MAC: C0:35:C5:89:47:BA)

Статический адрес

IP:  Mask:  /

DHCP



Чтобы сохранить изменения, необходимо нажать кнопку "Записать", иначе введенные значения не будут применены.

После нажатия кнопки "Записать" в журнале отобразится сообщение об успешной записи введенных значений.

Лог	Ошибки			
0	Имя: port30	MAC: C0:35:C5:89:B4:08	IP: 0.0.0.0	Маска: 0.0.0.0
1	Имя: port40	MAC: C0:35:C5:80:04:AC	IP: 0.0.0.0	Маска: 0.0.0.0
2	Имя: port50	MAC: C0:35:C5:8B:BA:FE	IP: 192.168.1.21	Маска: 255.255.255.0
3	Имя: port60	MAC: C0:35:C5:89:47:BA	IP: 0.0.0.0	Маска: 0.0.0.0
Задать IP: 192.168.1.13 && Маску: 255.255.255.0				
IP и маска успешно изменены.				
Задать новое имя: SAFETY_REGUL_R500S_A				
Имя успешно изменено				
Задать Gateway: 0.0.0.0				
Шлюз успешно задан.				

## 5. Настройте ПЛК Б:

- › Имя ПЛК: SAFETY\_REGUL\_R500S\_B.
- › Шлюз: 0.0.0.0.

> port30:

port30 (MAC: C0:35:C5:89:B4:08)

Статический адрес

IP: 0 . 0 . 0 . 0      Mask: 0 . 0 . 0 . 0 / 0

DHCP

Очистить      Записать

> port40:

port40 (MAC: C0:35:C5:80:04:AC)

Статический адрес

IP: 0 . 0 . 0 . 0      Mask: 0 . 0 . 0 . 0 / 0

DHCP

Очистить      Записать

> port50:

port50 (MAC: C0:35:C5:8B:BA:FE)

Статический адрес

IP: 192 . 168 . 1 . 14      Mask: 255 . 255 . 255 . 0 / 24

DHCP

Очистить      Записать

> port60:

port60 (MAC: C0:35:C5:89:47:BA)

Статический адрес

IP: 192 . 168 . 2 . 14      Mask: 255 . 255 . 255 . 0 / 24

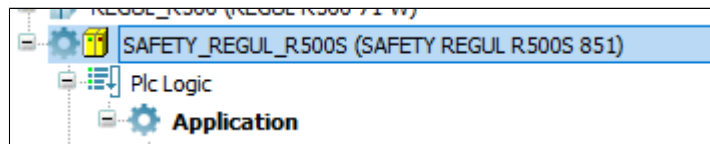
DHCP

Очистить      Записать

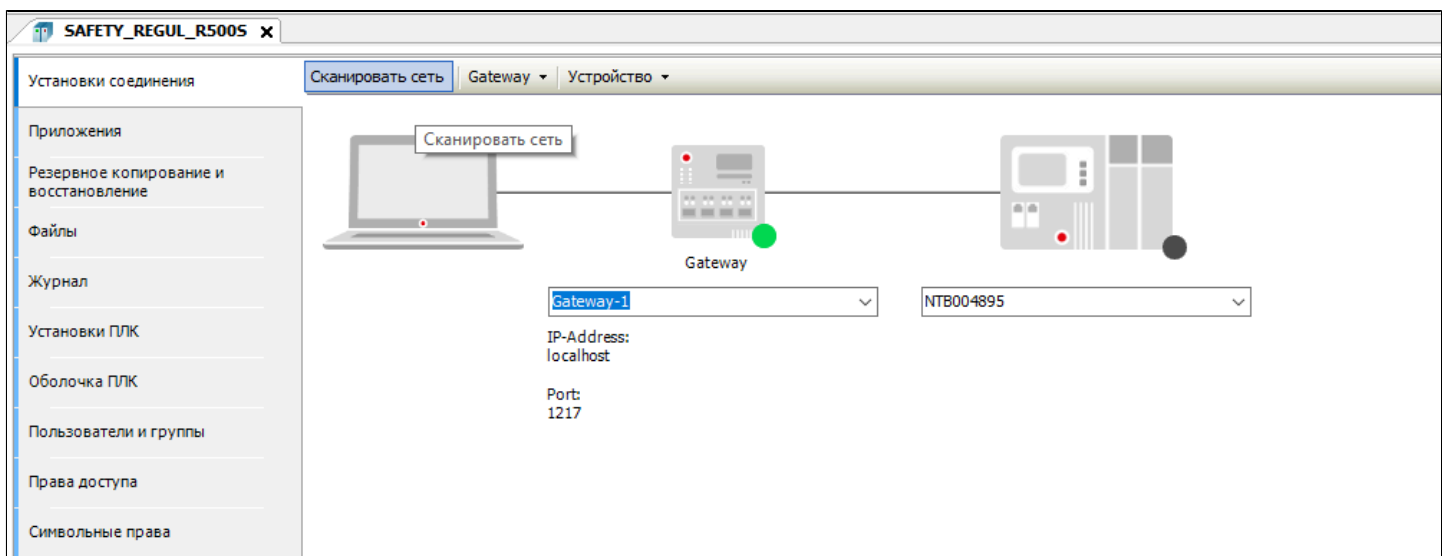


Компьютер и все ПЛК должны находиться в одной подсети.

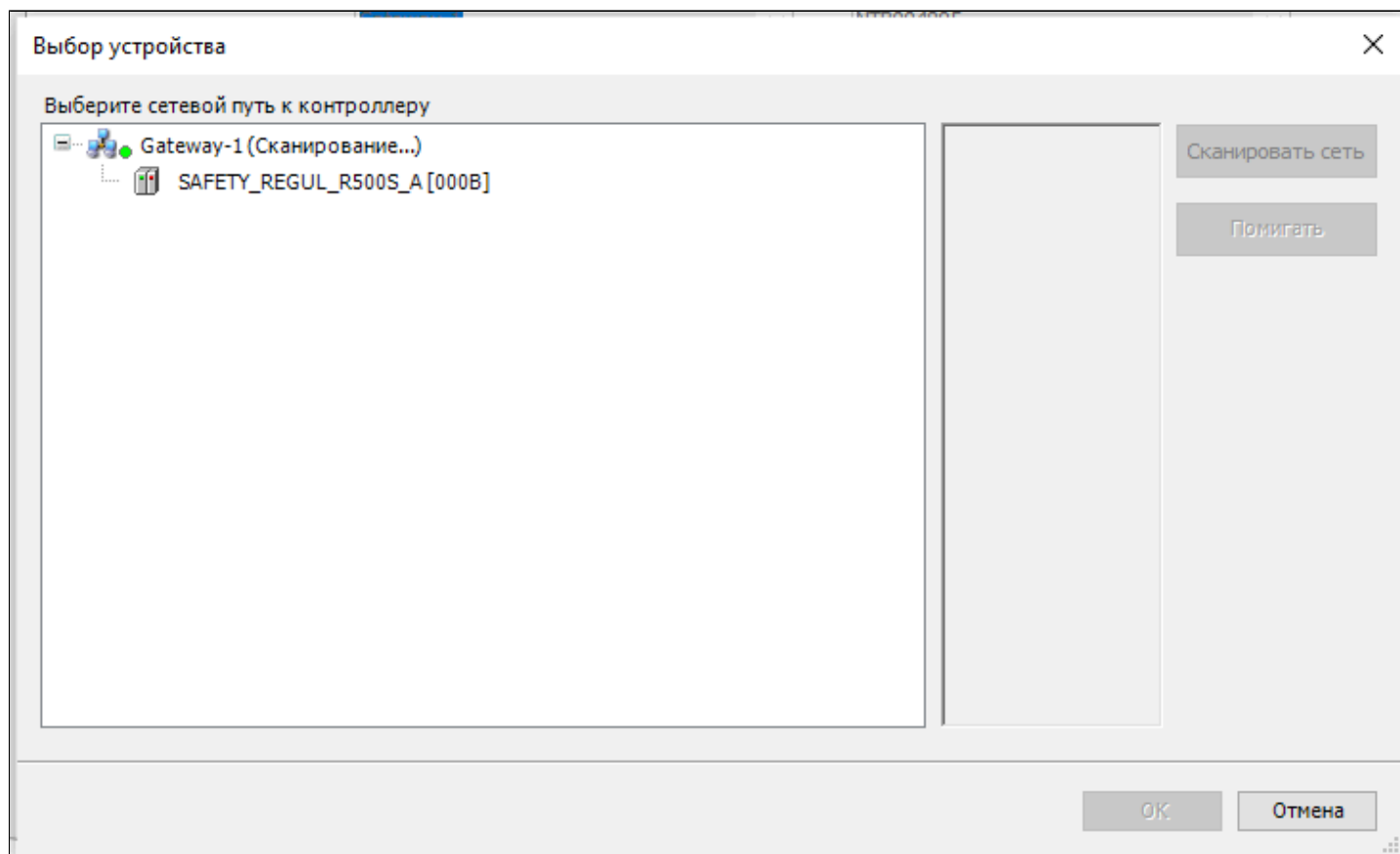
6. Выполните двойной клик мыши по ПЛК в дереве устройств.



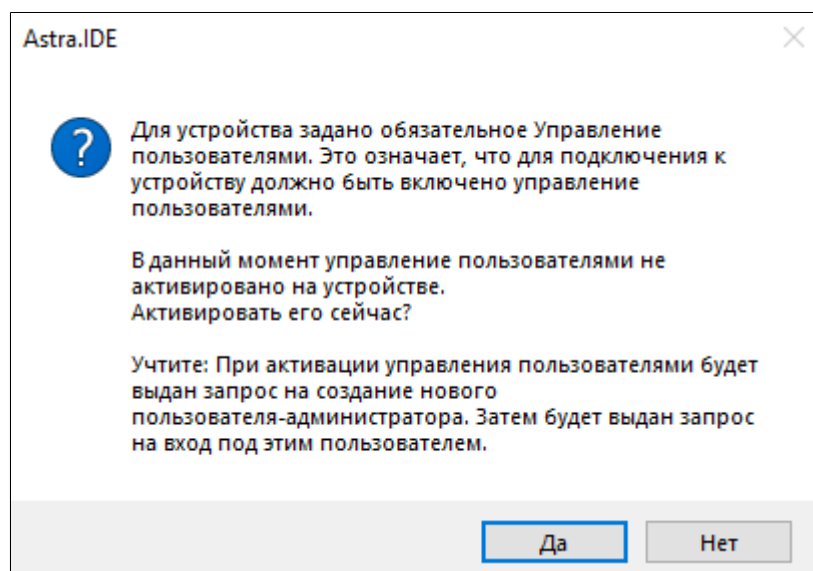
7. В меню "Установки соединения" нажмите кнопку "Сканировать сеть".



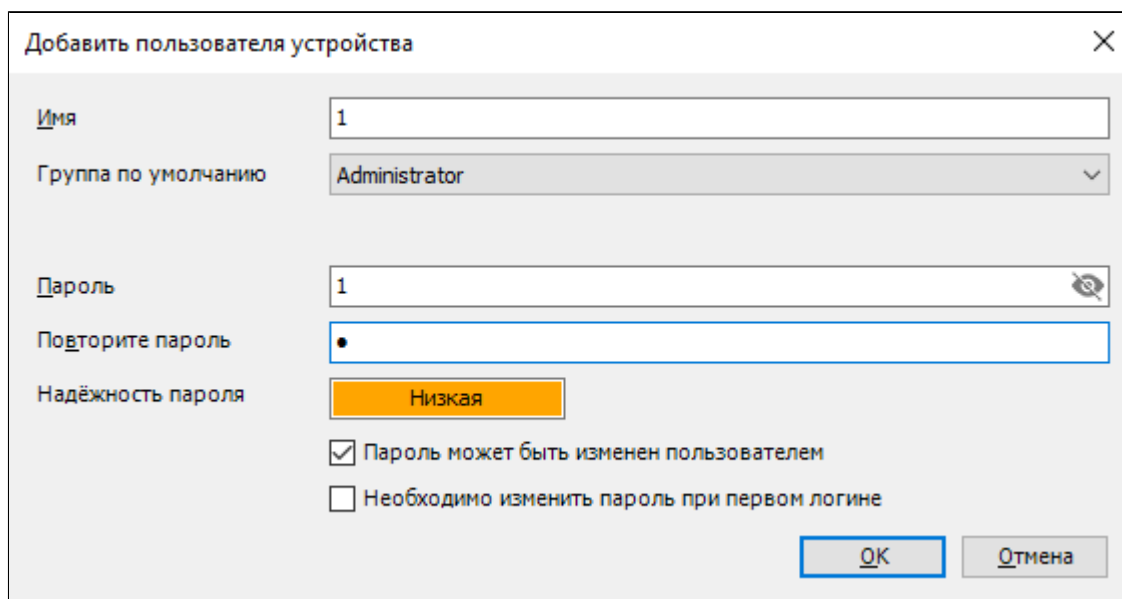
8. Выберите "SAFETY\_REGUL\_R500S\_A" и нажмите кнопку "OK".



Появится всплывающее окно. Нажмите кнопку "Да".



9. Задайте имя пользователя "1", группу по умолчанию "Administrator" и пароль "1". Нажмите кнопку "OK".



Добавить пользователя устройства

Имя: 1

Группа по умолчанию: Administrator

Пароль: 1

Повторите пароль: •

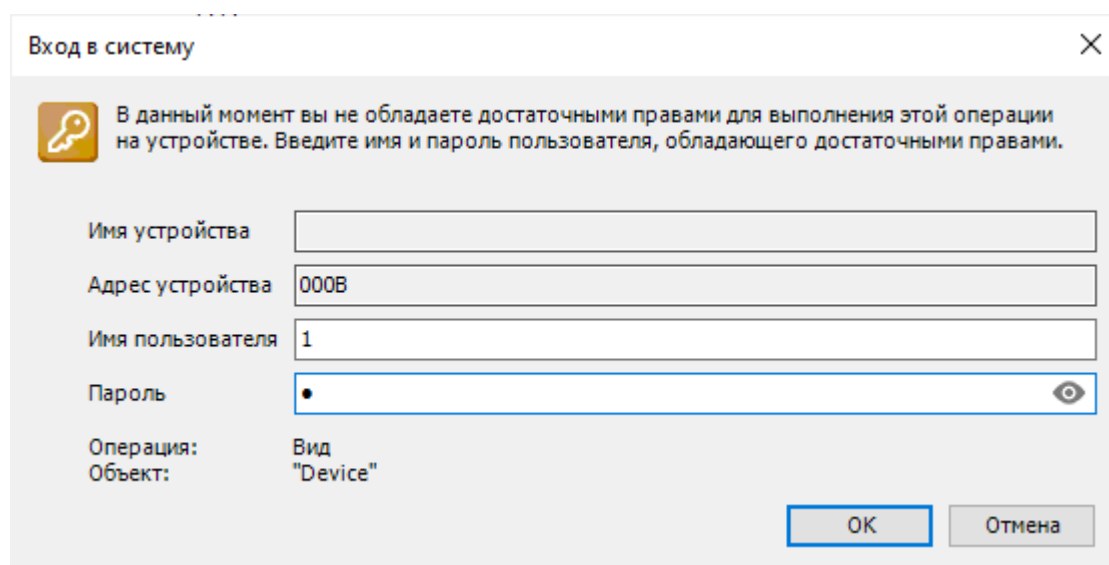
Надёжность пароля: Низкая

Пароль может быть изменен пользователем

Необходимо изменить пароль при первом логине

OK Отмена

10. Откроется окно "Вход в систему". Оно необходимо для исключения несанкционированного доступа к ПЛК. Введите имя пользователя и пароль, которые были заданы ранее, и нажмите кнопку "OK".



Вход в систему

В данный момент вы не обладаете достаточными правами для выполнения этой операции на устройстве. Введите имя и пароль пользователя, обладающего достаточными правами.

Имя устройства:

Адрес устройства: 000B

Имя пользователя: 1

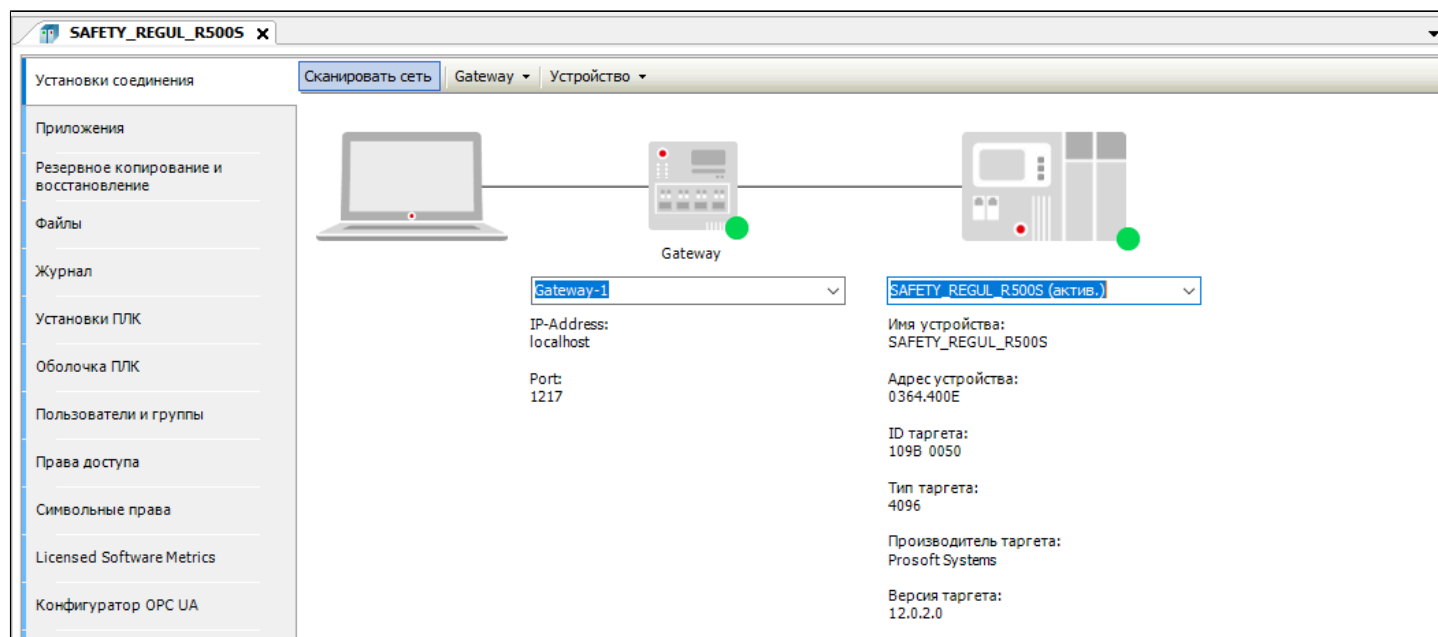
Пароль: •

Операция: Вид

Объект: "Device"

OK Отмена

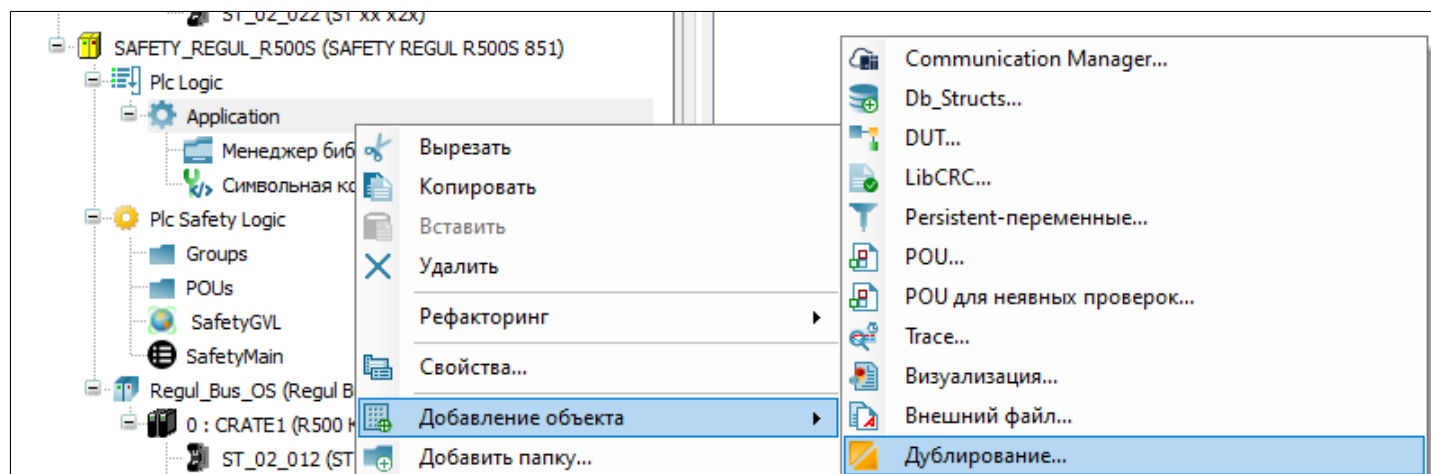
## 11. Подключение к SAFETY\_REGUL\_R500S\_A выполнено.



Аналогичным образом можно подключиться к SAFETY\_REGUL\_R500S\_B.

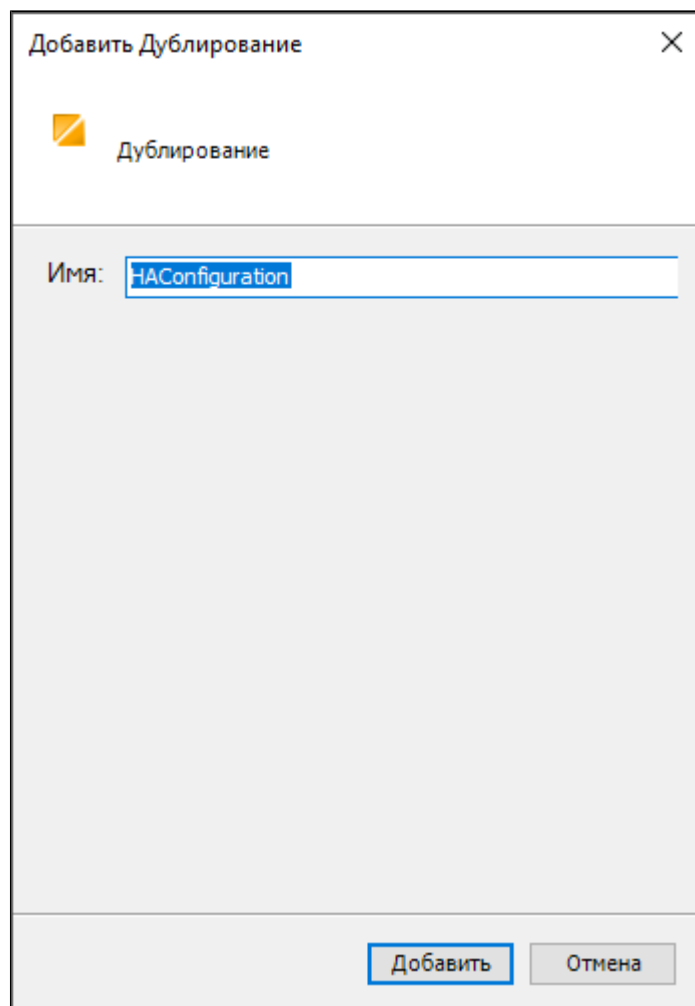
## 1.1.2.1.2.3. Настройка резервирования ПЛК SAFETY REGUL R500S

1. В контекстном меню приложения "Application" выберите "Добавление объекта" → "Дублирование...".

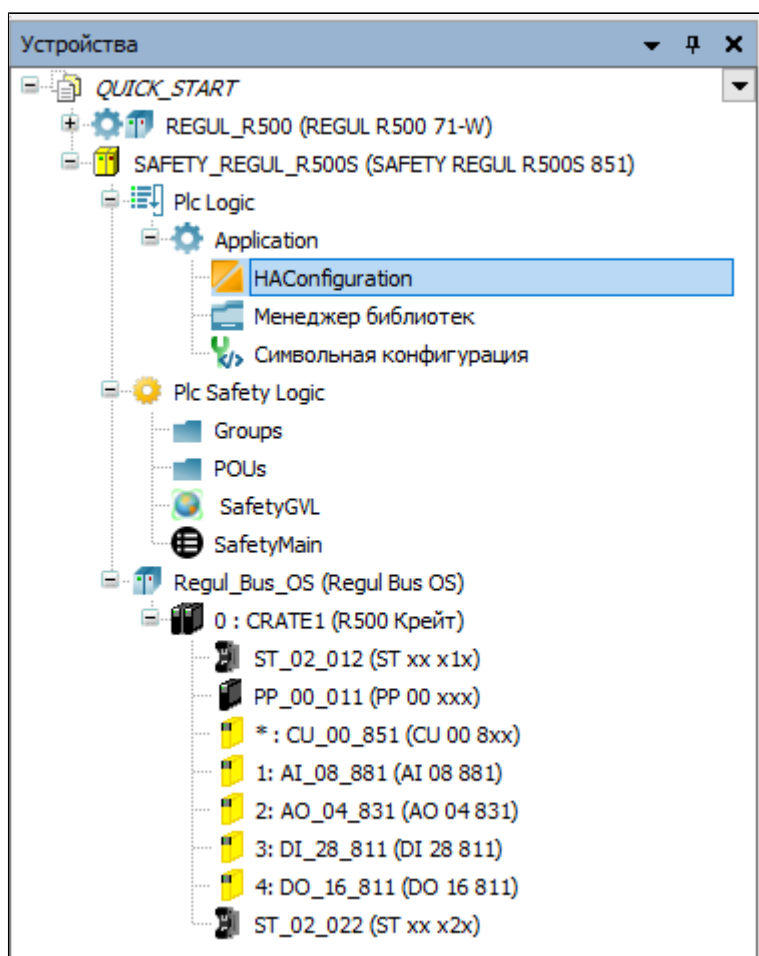




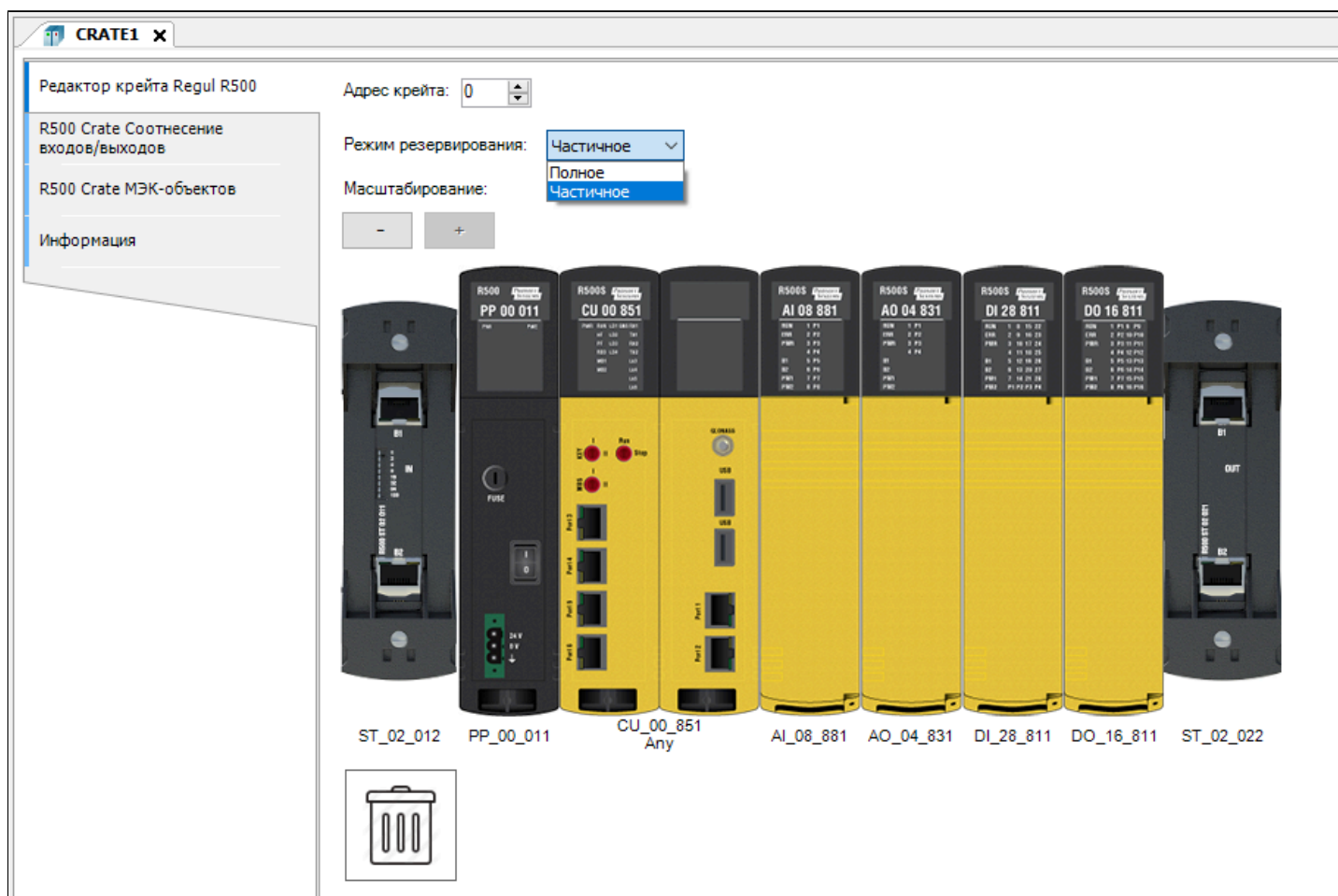
2. В открывшемся диалоговом окне нажмите кнопку "Добавить".



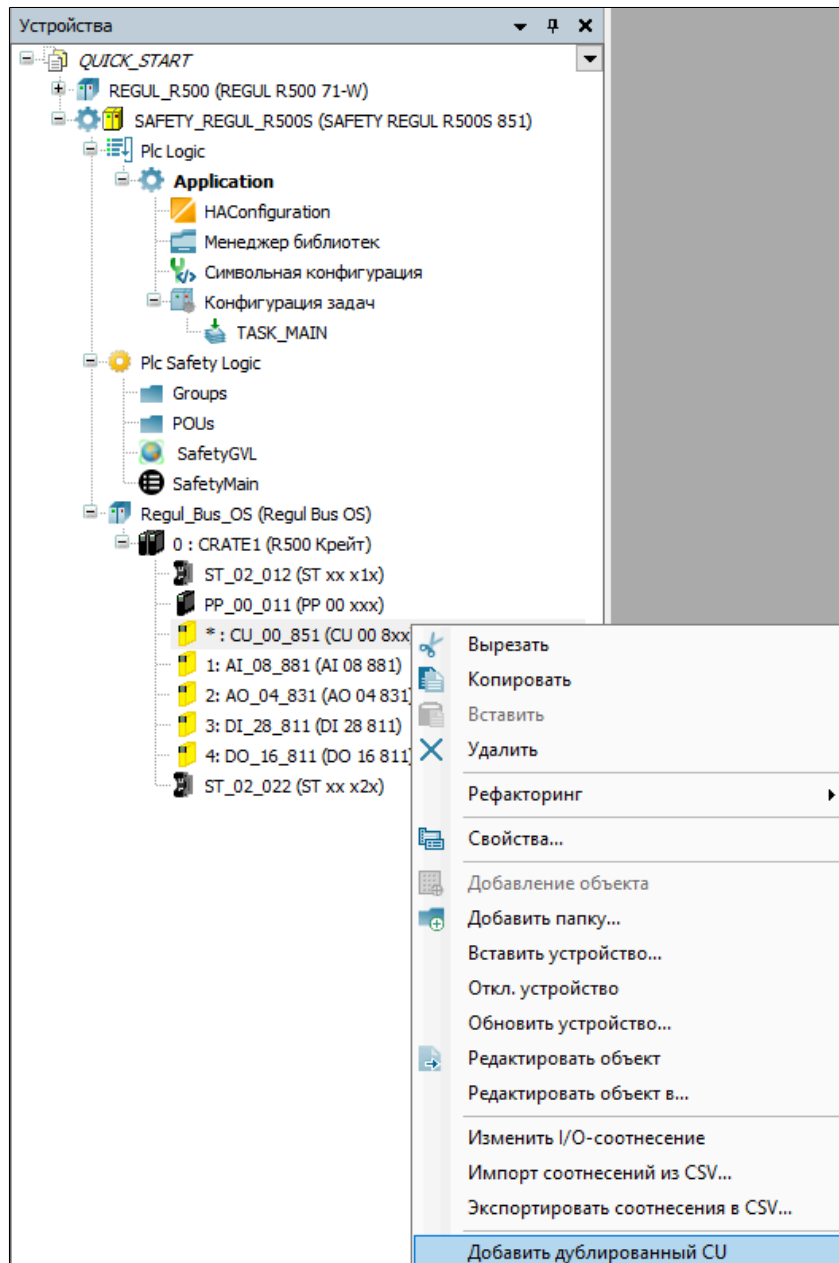
Компонент "Дублирование" будет добавлен в дерево устройств.



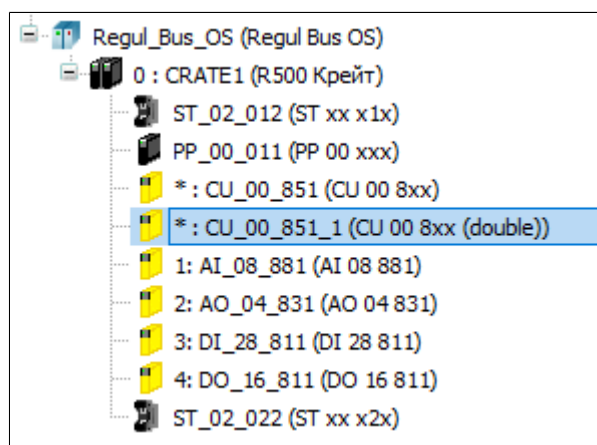
3. Перейдите в редактор крейта двойным кликом левой кнопки мыши. В поле "Режим резервирования" в выпадающем списке выберите "Частичное".



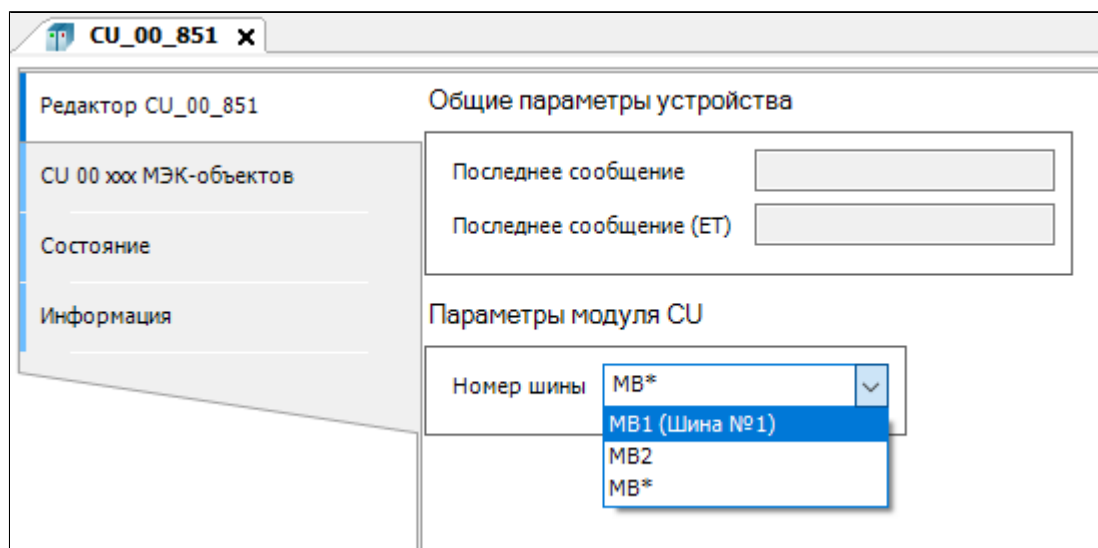
4. В контекстном меню контроллера выберите "Добавить дублированный CU".



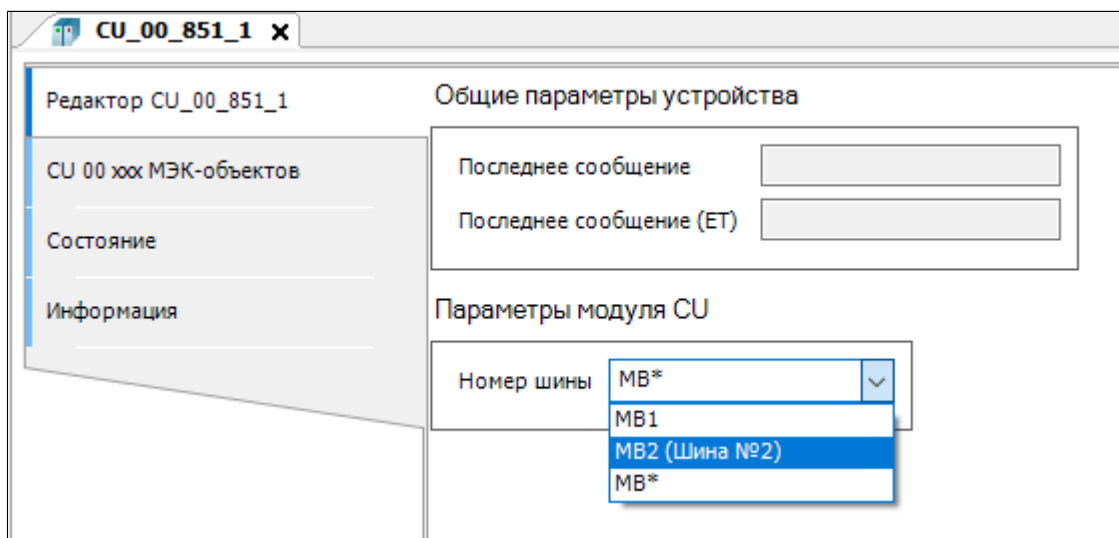
Конфигурация крейта будет выглядеть следующим образом:



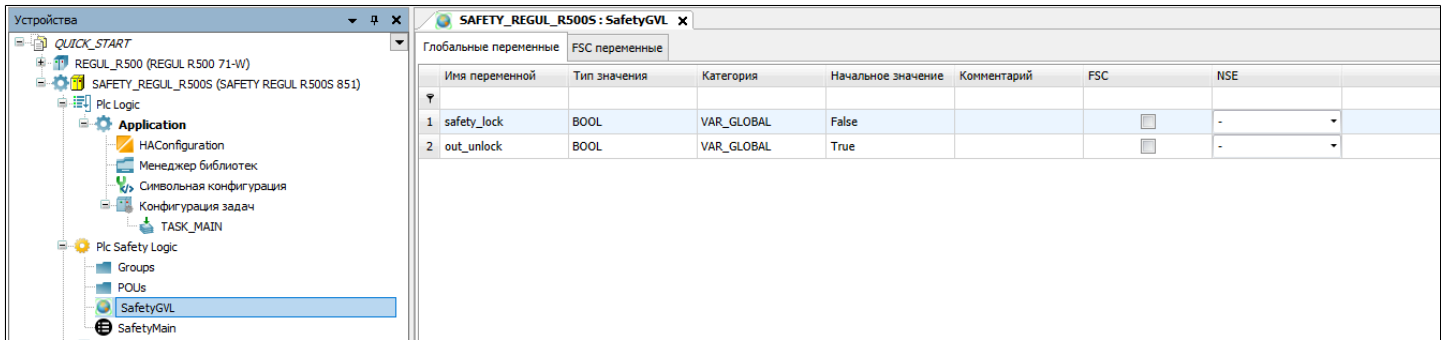
5. Перейдите в основной ПЛК и во вкладке "Редактор CU\_00\_851" выберите номер шины из выпадающего списка "MB1 (Шина №1)".



6. Перейдите в дублированный ПЛК и во вкладке "Редактор CU\_00\_851\_1" выберите номер шины из выпадающего списка "MB2 (Шина №2)".

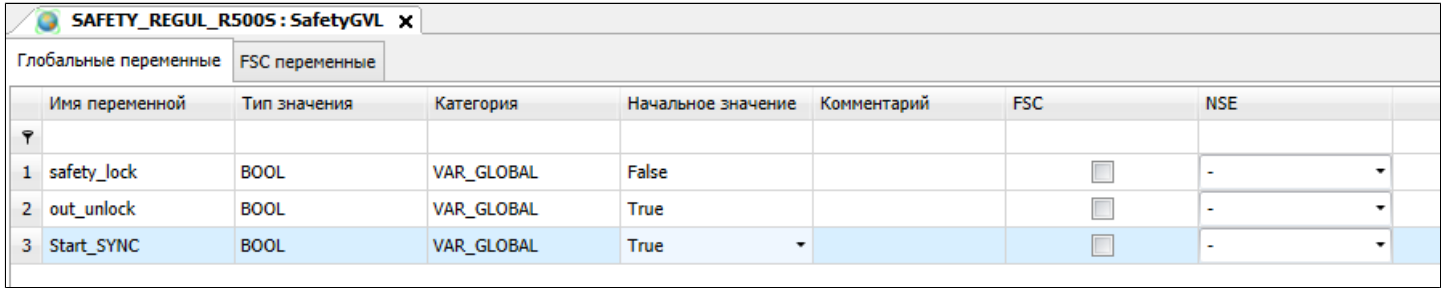
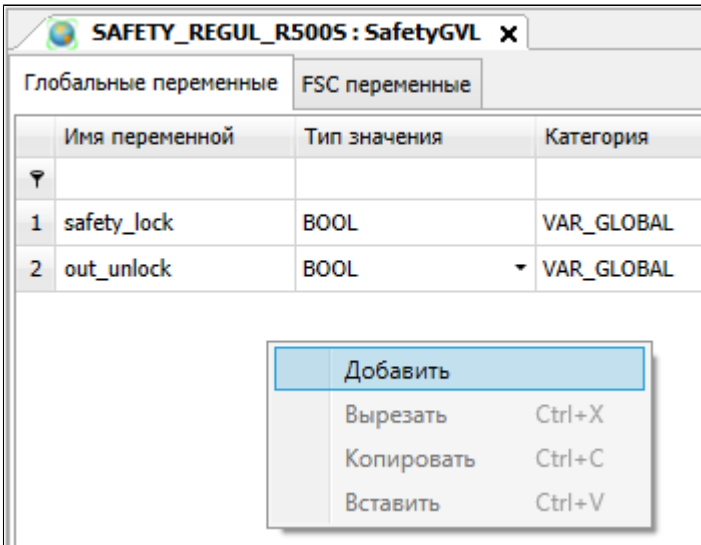


7. Для корректной работы дублирования необходимо создать глобальную переменную наличия стартовой синхронизации. Перейдите в редактор глобальных переменных "SafetyGVL".

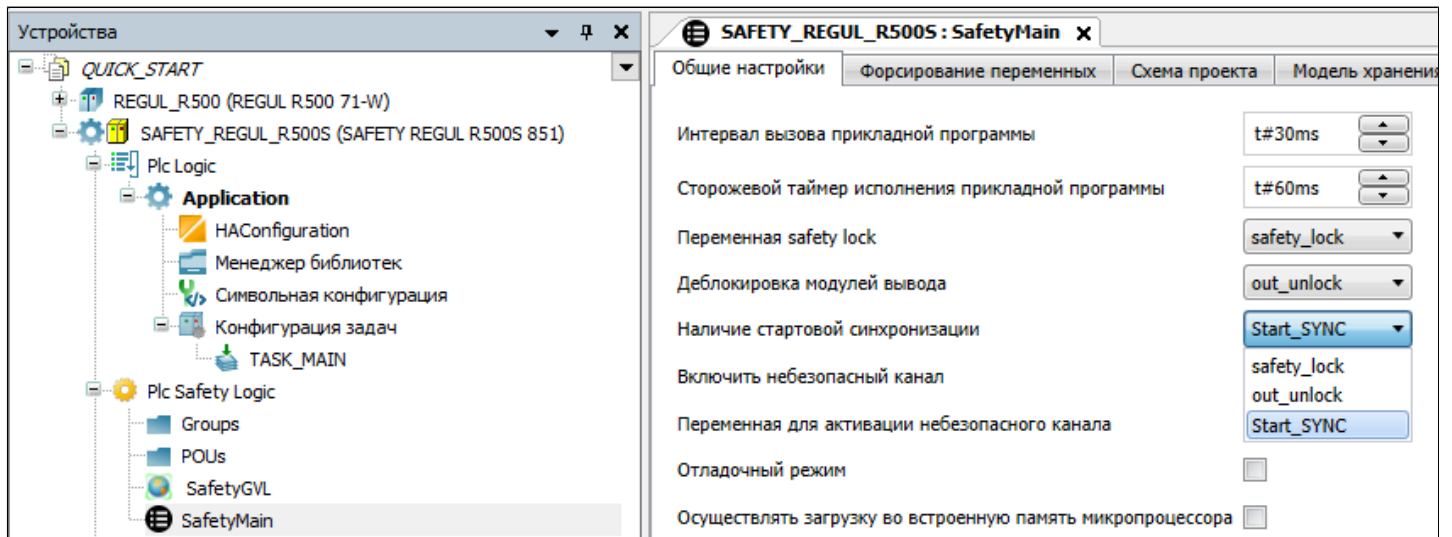


**i** При создании проекта автоматически формируются переменные "safety\_lock" и "out\_unlock", которые используются для параметров "Переменная Safety lock" и "Деблокировка модулей вывода" соответственно в "SafetyMain".

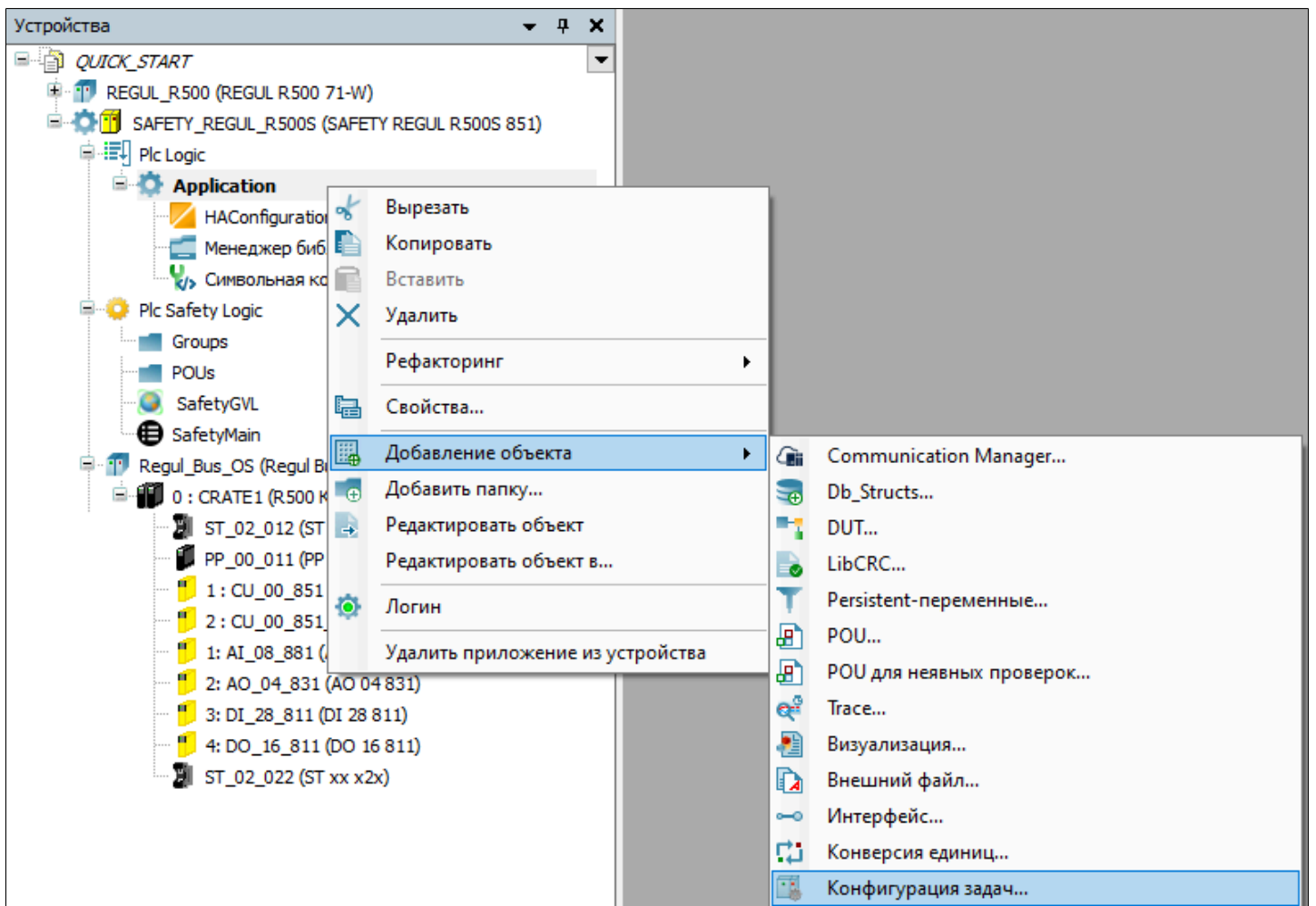
8. Добавьте переменную "Start\_SYNC" типа "BOOL" и начальным значением "True". Для этого в контекстном меню выберите "Добавить".



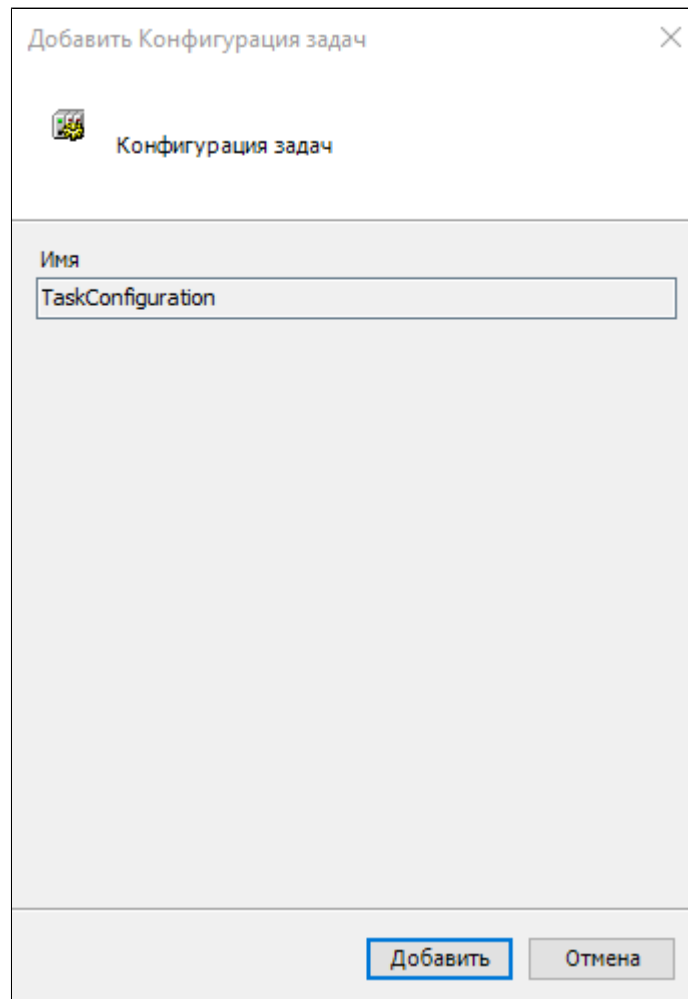
9. Перейдите в "SafetyMain" и выберите созданную переменную в выпадающем списке параметра "Наличие стартовой синхронизации".



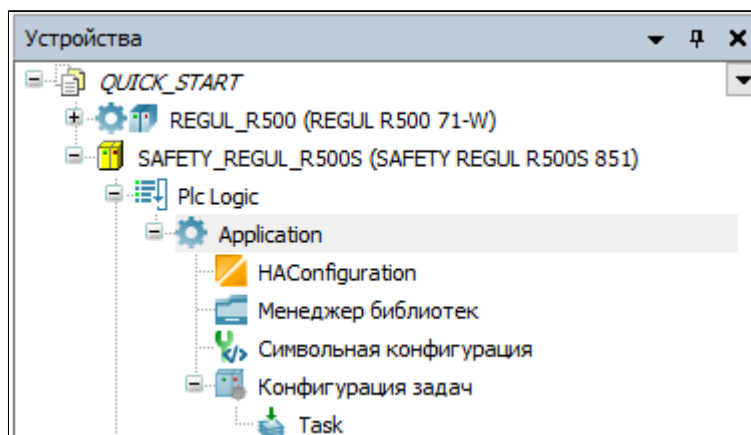
10. Добавьте в проект объект "Конфигурация задач". Для этого в контекстном меню приложения "Application" выберите "Добавление объекта" → "Конфигурация задач...".



11. В открывшемся окне нажмите кнопку "Добавить".

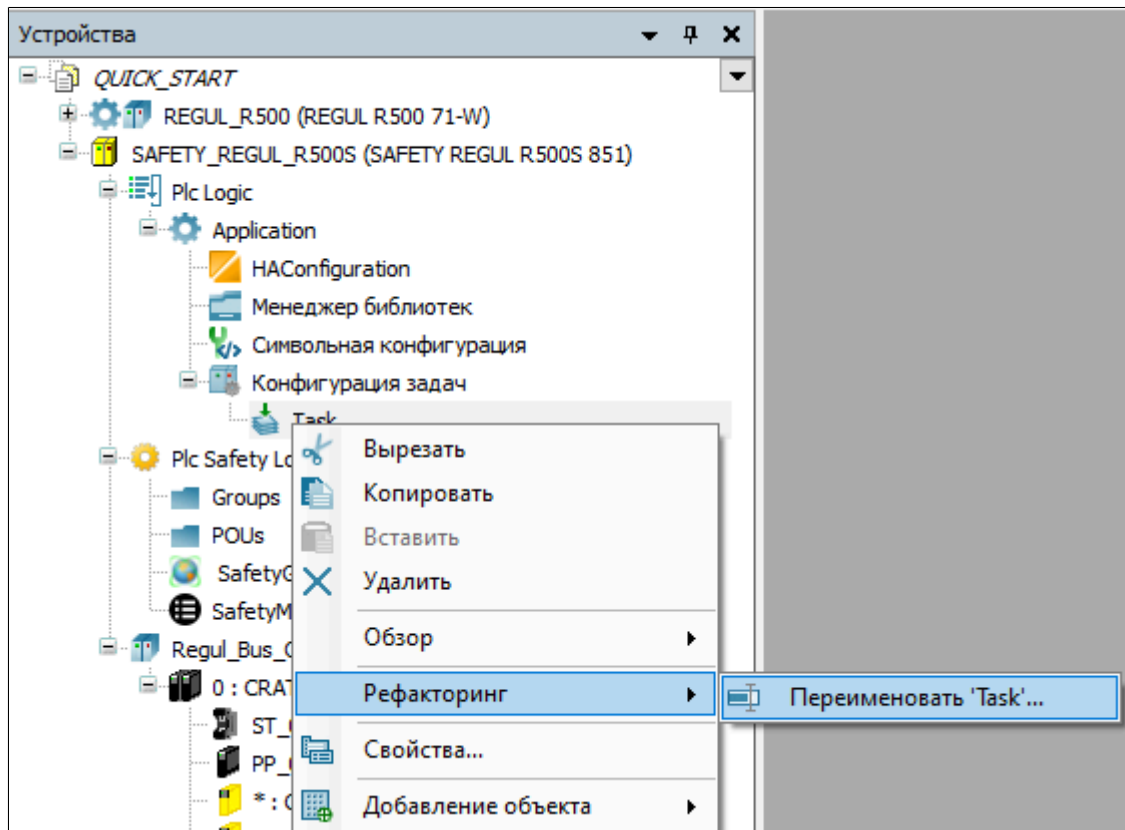


В проект будет добавлен конфигуризатор задач и задача "Task".



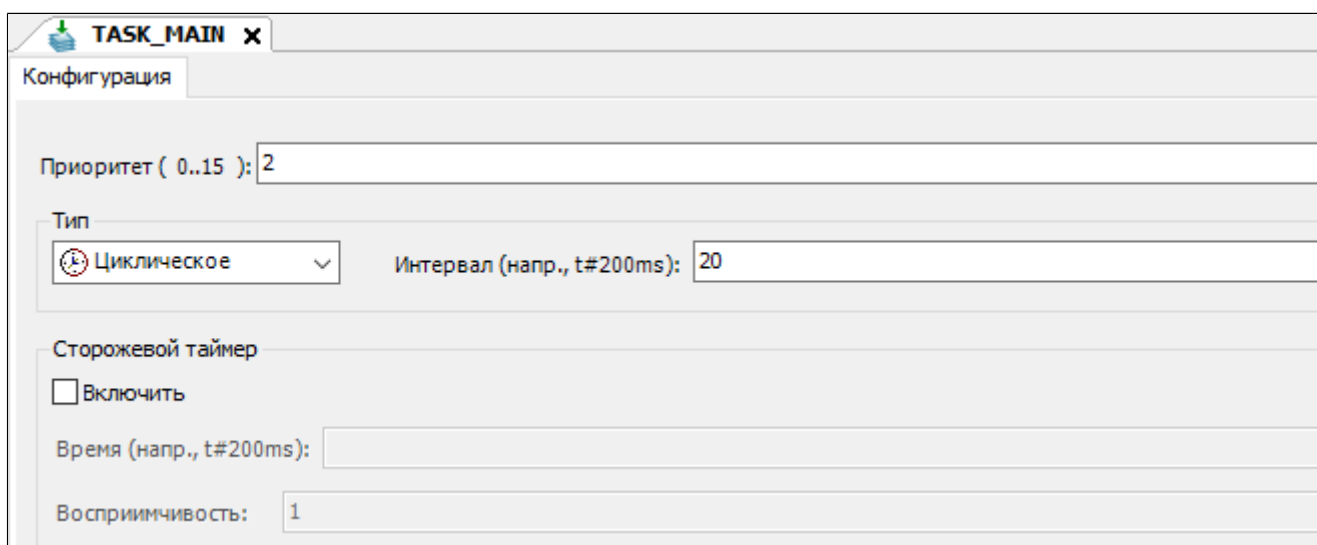
12. Нажмите правой кнопкой мыши по объекту "Task" и в контекстном меню выберите "Рефакторинг" → "Переименовать 'Task'...".





Задайте имя "TASK\_MAIN" и нажмите кнопку "OK". Подтвердите переименовывание объекта.

13. Откройте задачу "TASK\_MAIN" двойным кликом левой кнопки мыши. Задача имеет следующие параметры: приоритет – 2, тип исполнения – "Циклическое" и интервал – 20 мс. Оставьте значения по умолчанию.



# 1.1.2.1.2.4. Конфигурирование сервера OPC UA

1. Проверьте, что вы подключены к ПЛК А и перейдите во вкладку "Сервис ПЛК".

Интервал обновления (в секундах): 1

Общая информация | Аппаратная информация | SMART | Системное ПО | Настройка разрешения | Системные параметры | Статус safety

Версия системного ПО: н/д  
Среда исполнения: н/д  
Версия среды исполнения: н/д

сеть и хранение данных

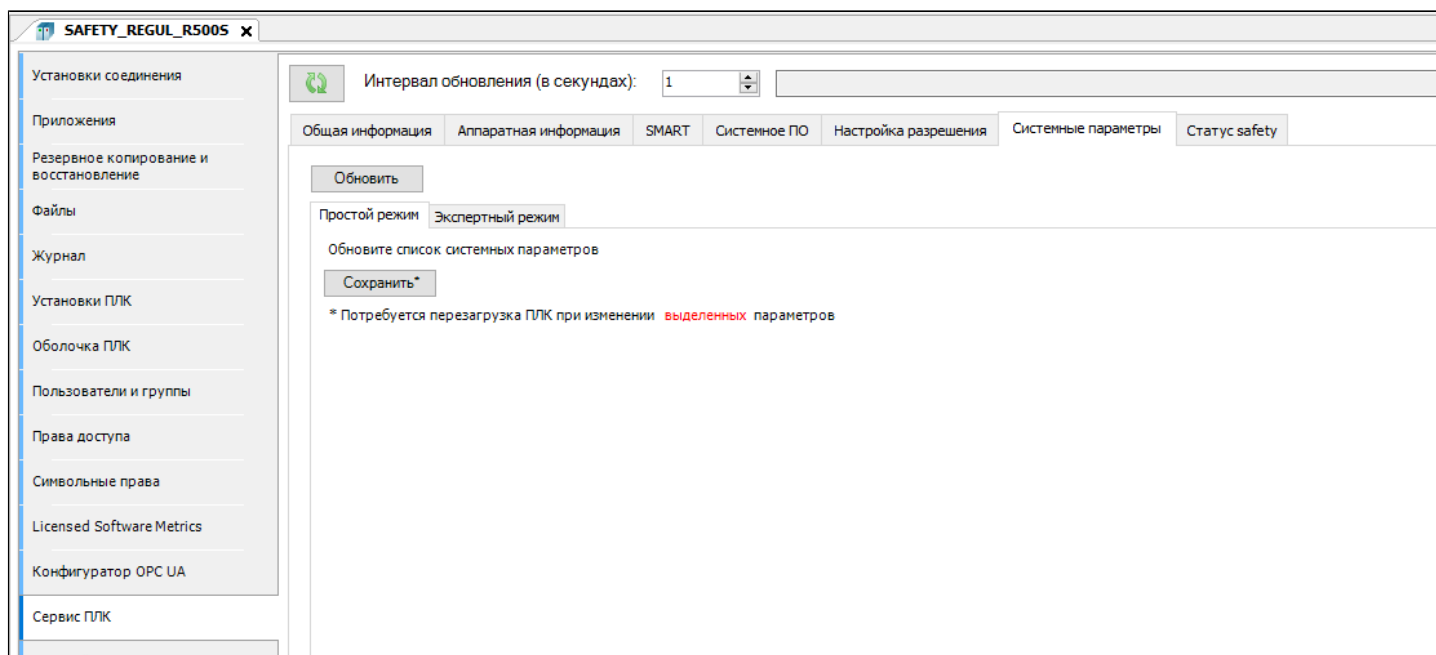
Сетевые интерфейсы

Имя	Линк	Получено пакетов	Получено байт	Передано пакетов	Передано байт	Ошибки выравнивания	Ошибки Fcs
-----	------	------------------	---------------	------------------	---------------	---------------------	------------

Хранение данных

Раздел	Всего (Мб)	Занято (Мб)
--------	------------	-------------

2. Откройте вкладку "Системные параметры" и нажмите кнопку "Обновить".



3. На экран будет выведена информация о текущем состоянии доступных параметров. Убедитесь, что флаг "Запуск сервера OPC UA" установлен.



Если флаг "Запуск сервера OPC UA" снят, его нужно установить и нажать кнопку "Сохранить".

Интервал обновления (в секундах): 1

Общая информация | Аппаратная информация | SMART | Системное ПО | Настройка разрешения | Системные параметры | Статус safety

Обновить

Простой режим | Экспертный режим

PlcServices

- Запуск сервера FTP
- Запуск сервера SQL
- Предоставление сетевого доступа к БД MySQL
- Разрешение на подключение внешних накопителей
- Запуск службы печати LPD

PsOpcUaServer


- Запуск сервера OPC UA

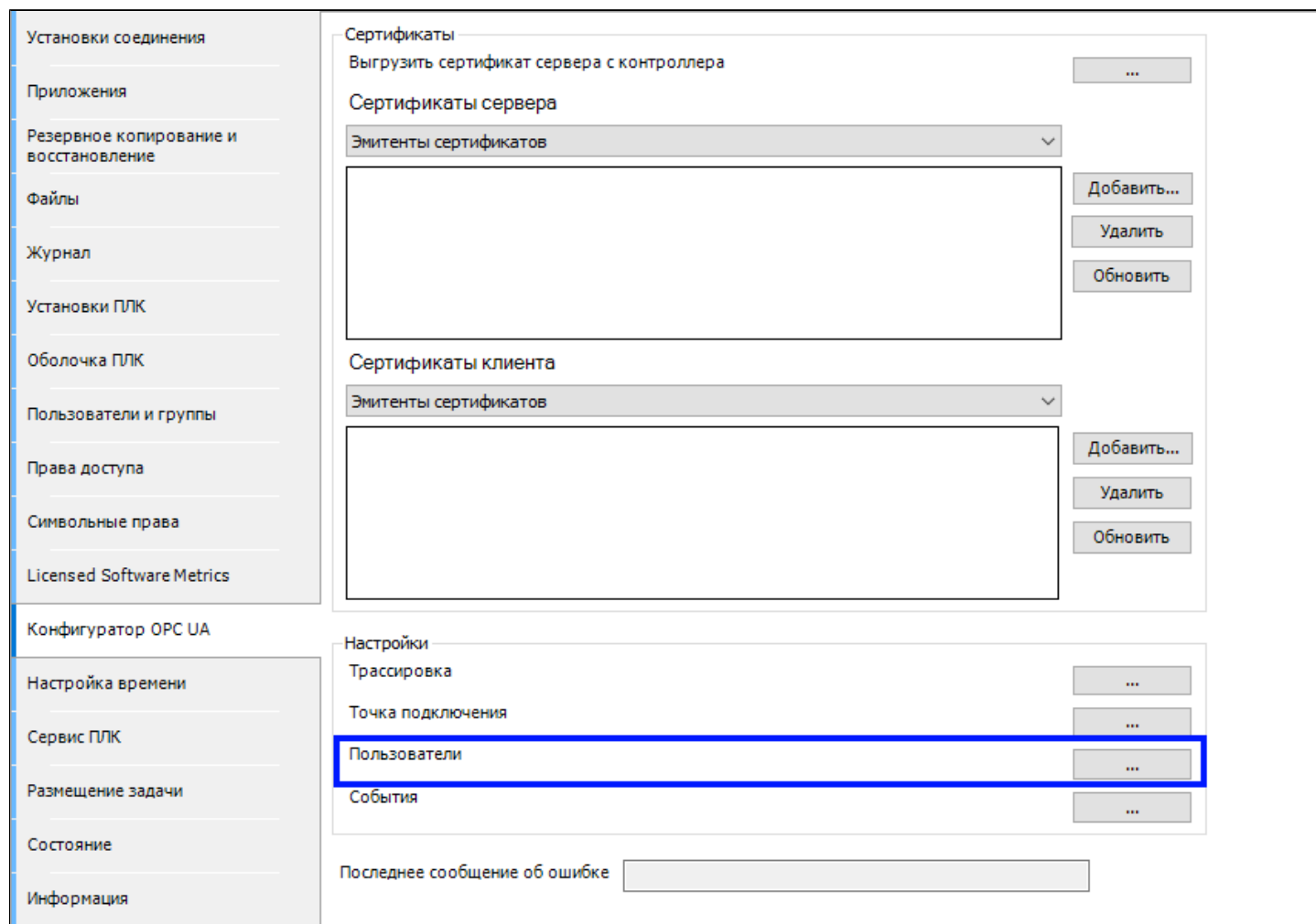
global

- Запрет на изменение сетевых настроек

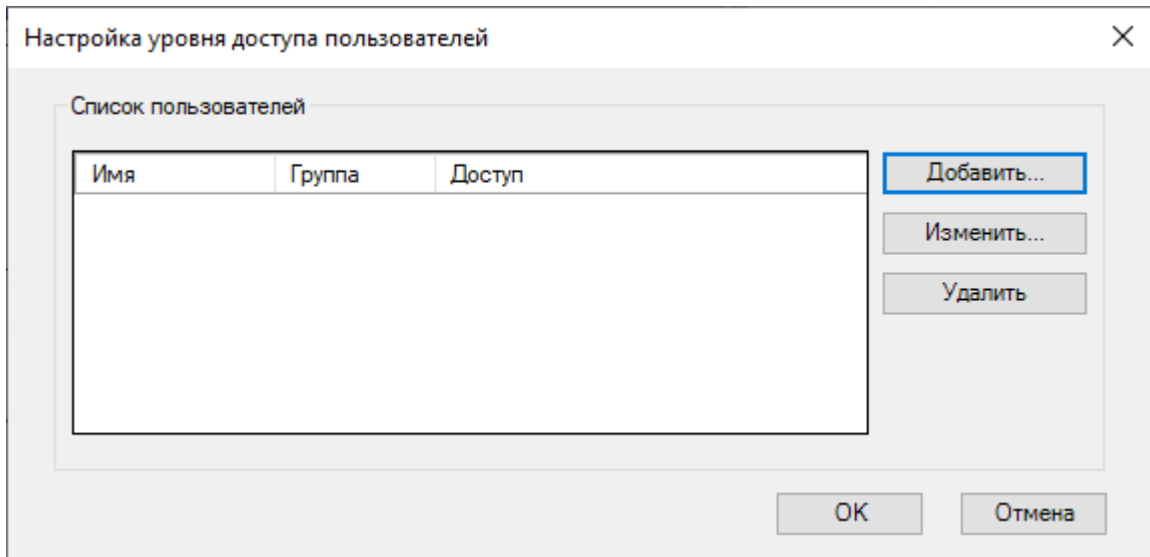
Сохранить\*

\* Потребуется перезагрузка ПЛК при изменении выделенных параметров

4. Перейдите на вкладку "Конфигуратор OPC UA" и в разделе настройки нажмите на кнопку  напротив строчки "Пользователи".



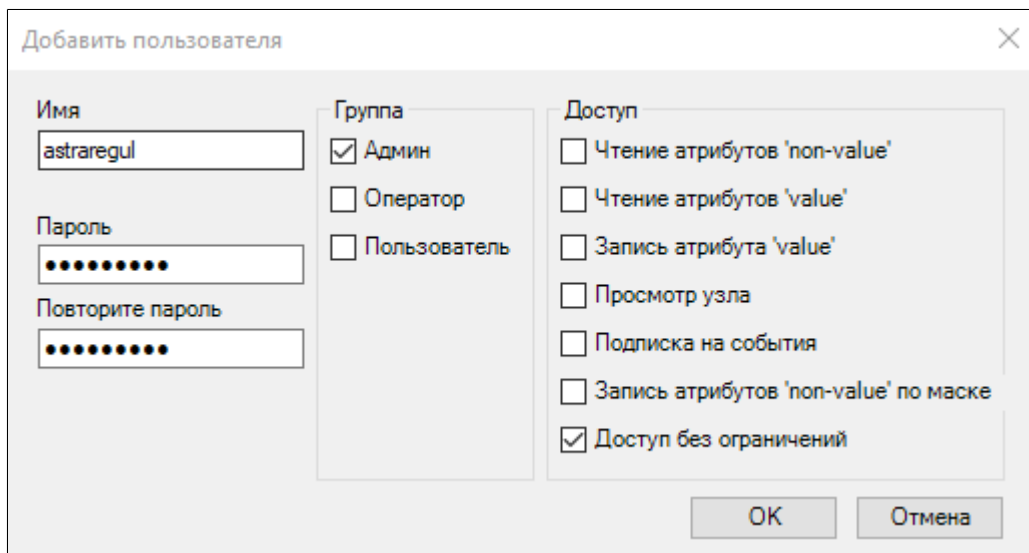
5. Откроется окно "Настойка уровня доступа пользователей". Нажмите кнопку "Добавить...".



6. В открывшемся окне установите следующие параметры:

- › Имя: astraregul;
- › Пароль: astraregul;
- › Группа "Админ";
- › Установите флаг "Доступ без ограничений".

Нажмите кнопку "OK".



После ввода нового пароля выйдет уведомление о смене пароля, подтвердите изменение текущего пароля.

Должен получиться следующий пользователь с доступом "Full access":

Настройка уровня доступа пользователей



Список пользователей

Имя	Группа	Доступ
astraregul	Admin	Full access

Добавить...

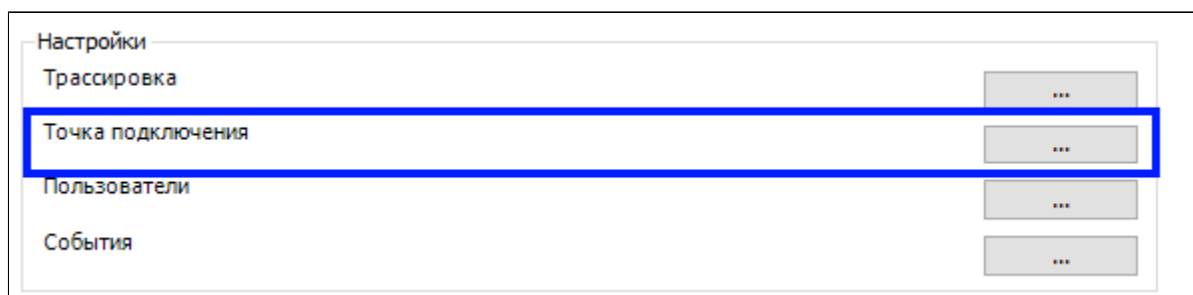
Изменить...

Удалить

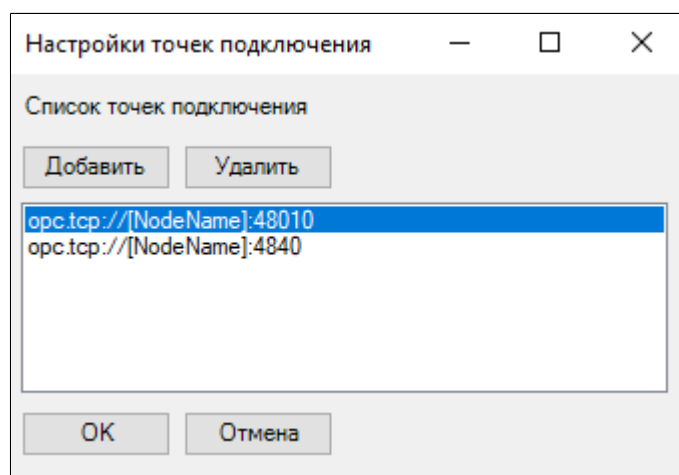
OK

Отмена

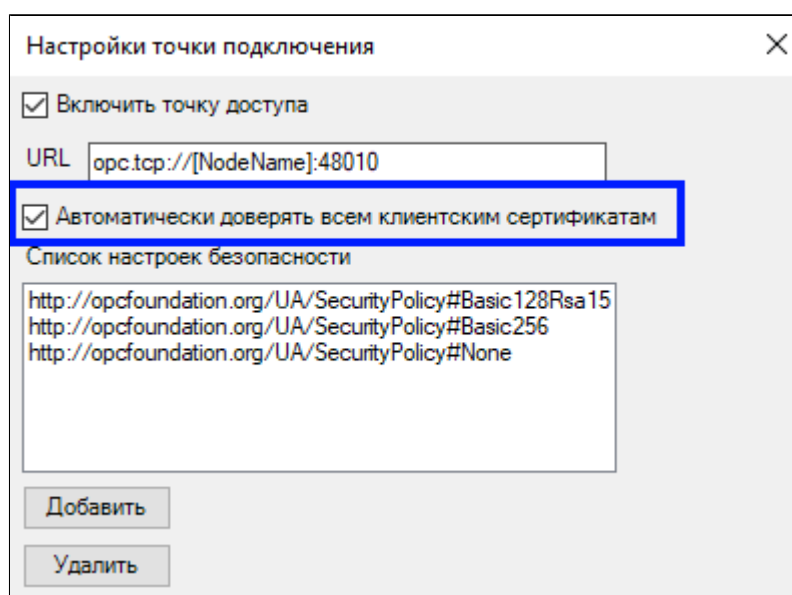
7. Нажмите на кнопку  напротив строки "Точка подключения".



8. В открывшемся окне двойным щелчком левой кнопкой мыши перейдите в настройки точки доступа по порту 48010.



9. В настройках точки доступа убедитесь, что установлен флаг "Автоматически доверять всем клиентским сертификатам".

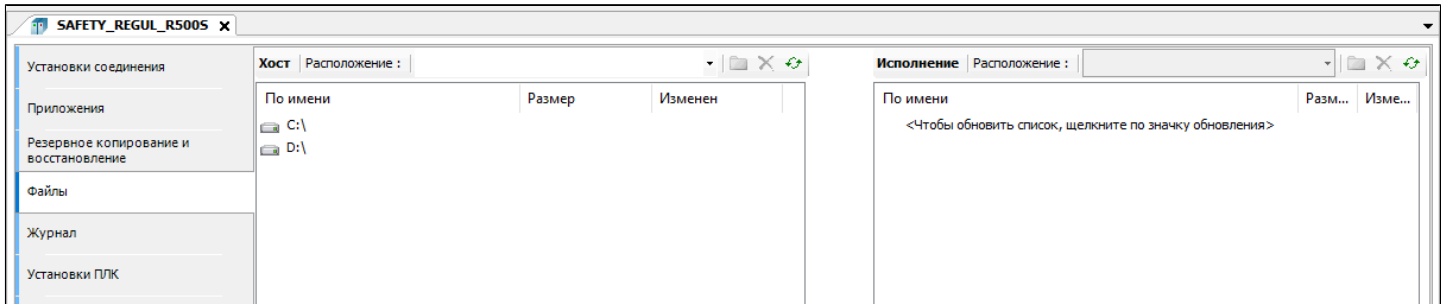




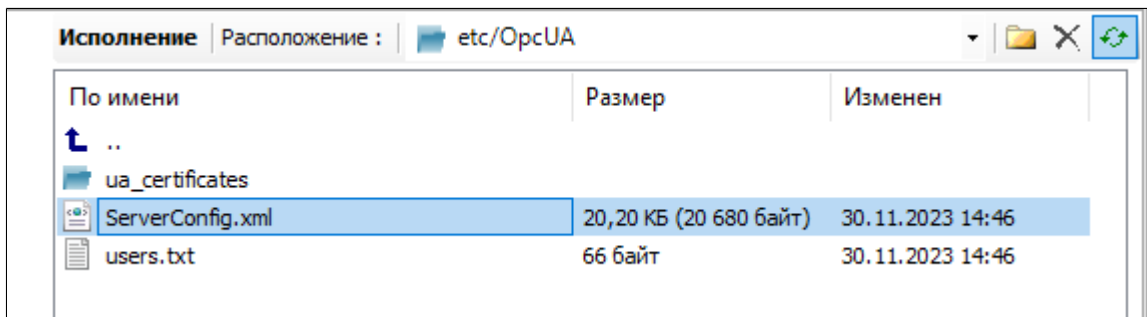


Если флаг "Автоматически доверять всем клиентским сертификатам" снят, его нужно установить.

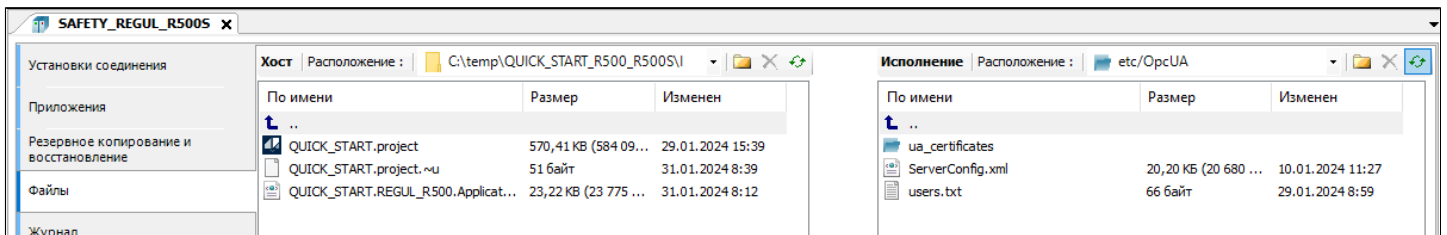
10. Перейдите во вкладку "Файлы" и обновите информацию о файловом расположении ПЛК нажав кнопку "Обновить".




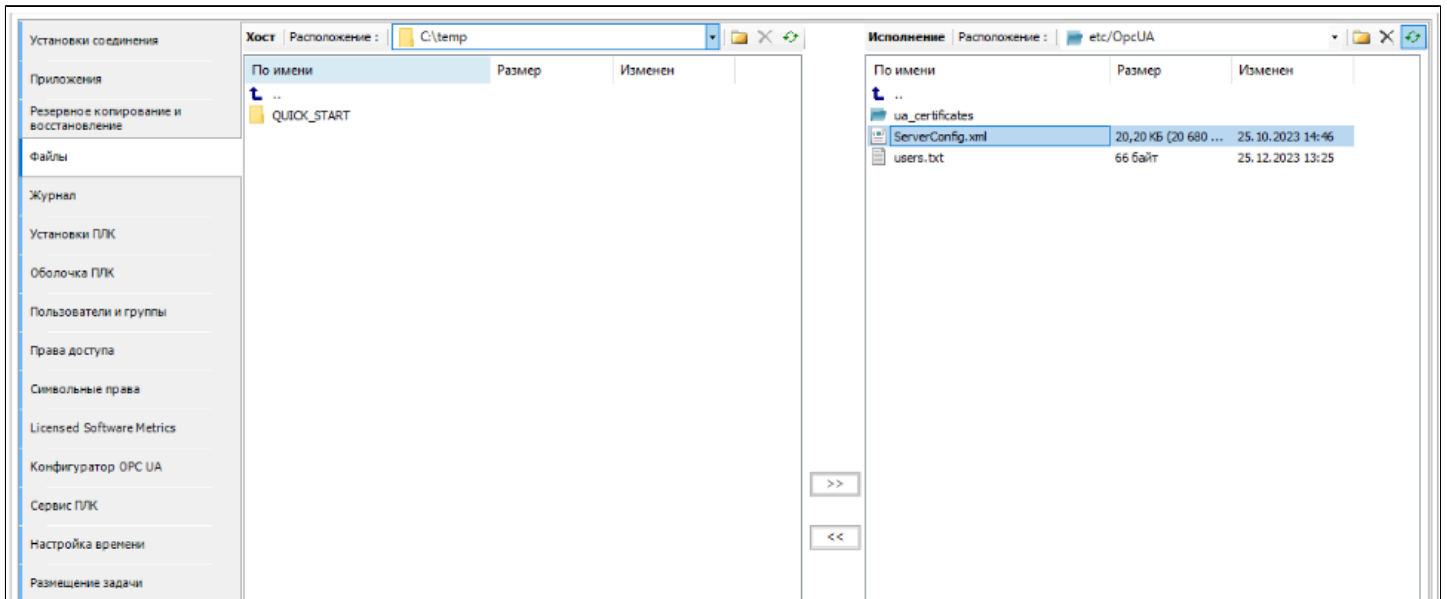
11. Перейдите по пути "etc/OpсUa" и выделите файл "ServerConfig.xml".



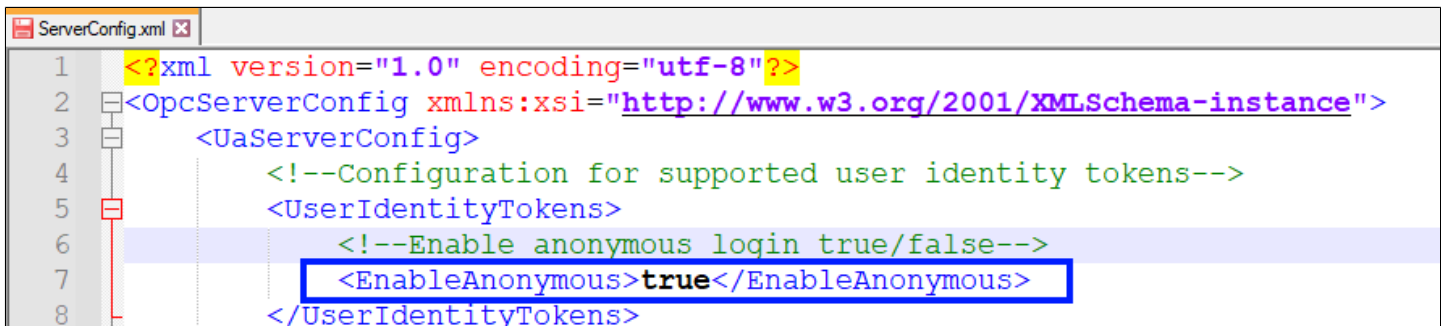
12. Выберите папку в левом окне для сохранения файла с ПЛК на компьютер, например "C:\temp".



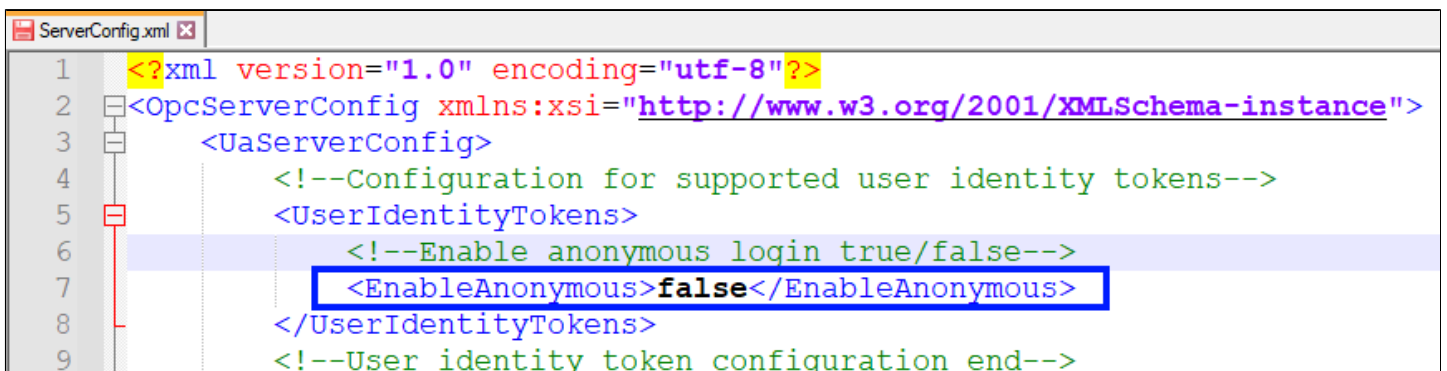
13. Нажмите на клавишу , чтобы переместить файл "ServerConfig.xml" в выбранную папку.



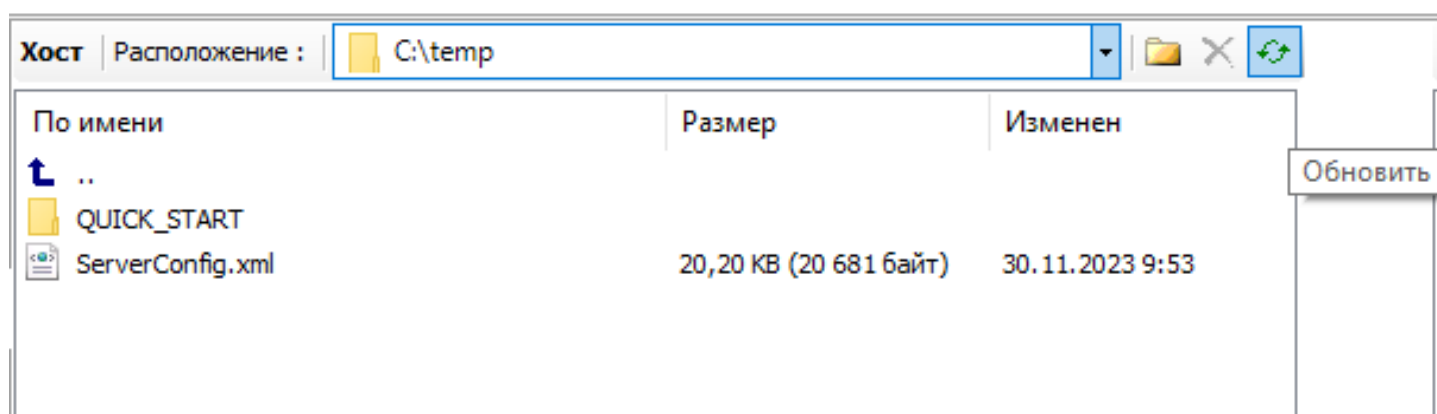
14. Перейдите в папку с файлом и откройте его при помощи редактора Notepad ++.




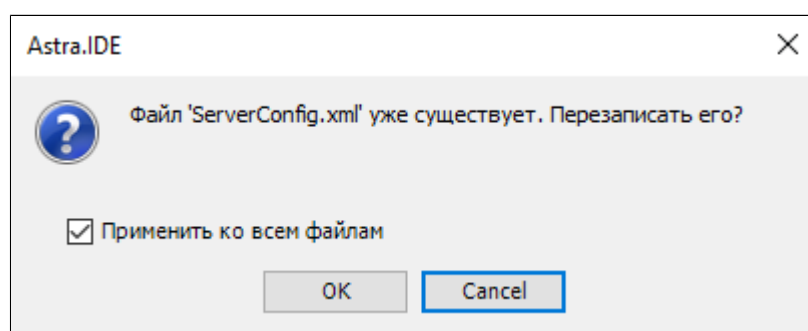
15. Измените значение "true" на "false" в строке <EnableAnonymous> и сохраните измененный файл.



16. Перейдите в Astra.IDE и обновите файловую конфигурацию ОС нажав клавишу "Обновить".

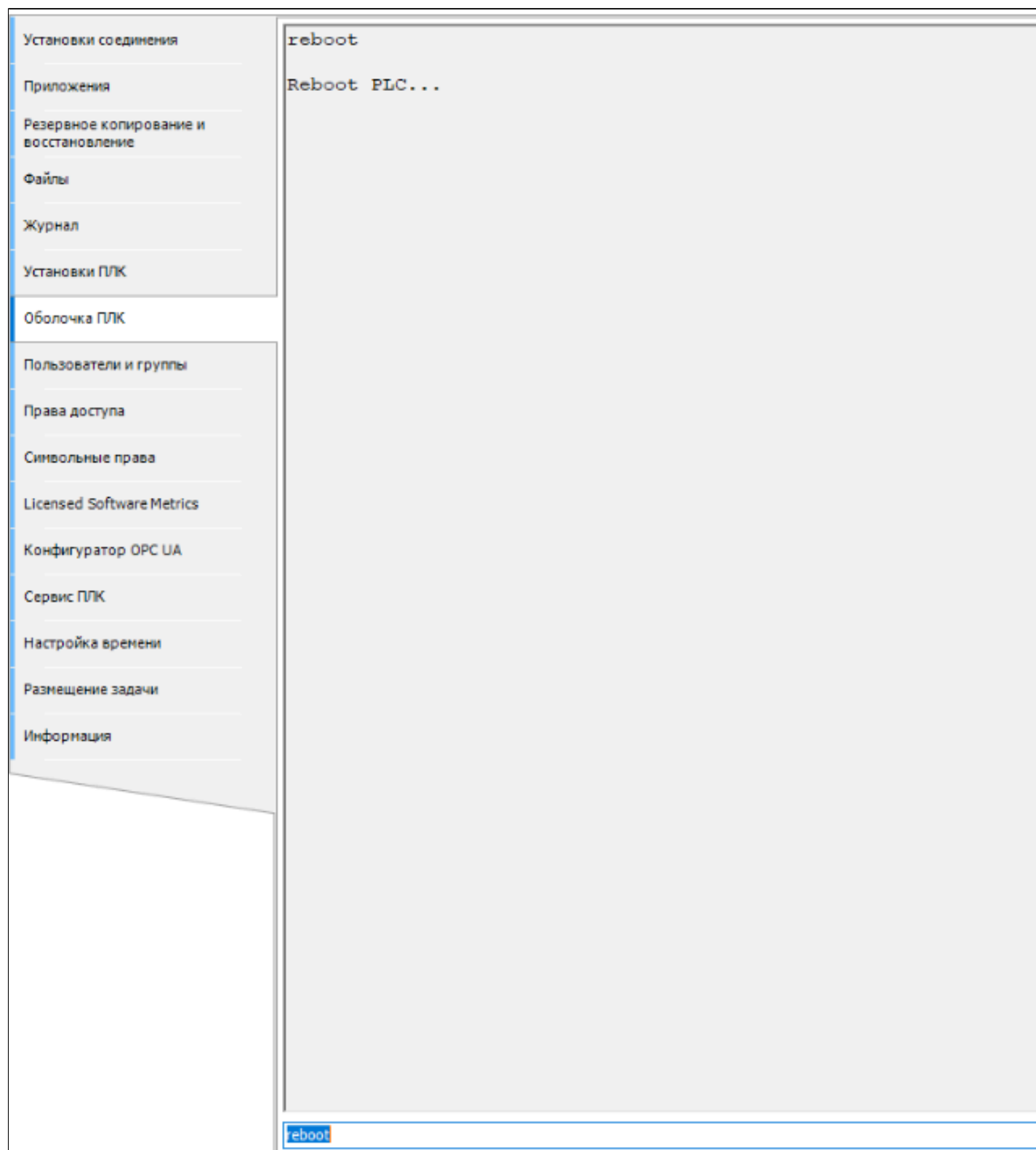


17. Передайте измененный файл в ПЛК, выделите его и нажмите на кнопку . Появится следующее всплывающее окно, нажмите кнопку "OK".



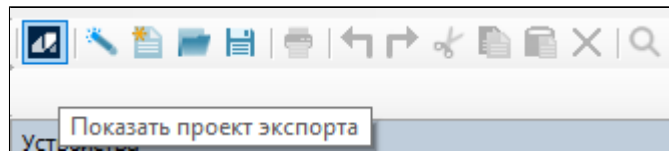
После проделанных действий необходимо перезагрузить ПЛК.

18. Перейдите во вкладку "Оболочка ПЛК". Введите в поле ввода команду "reboot" и нажмите клавишу "Enter". ПЛК будет перезагружен.

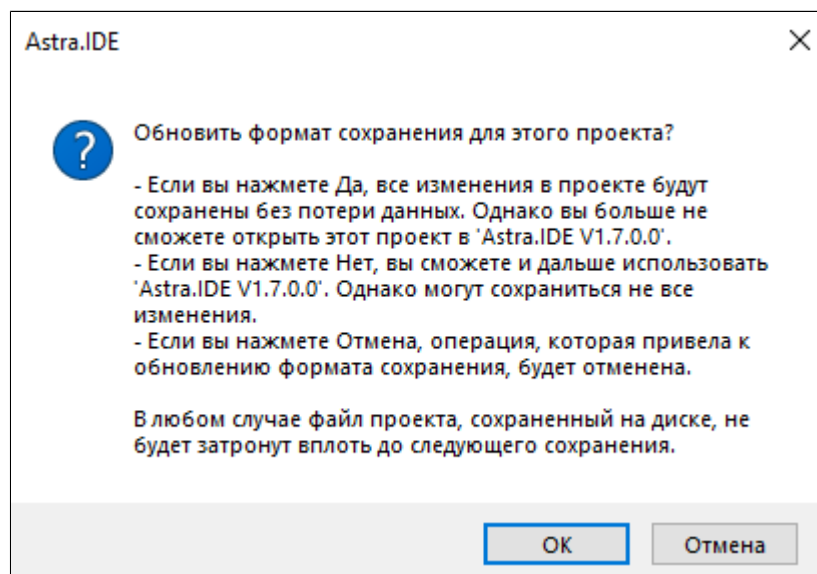


## 1.1.2.2. Создание топологии системы АСУ ТП

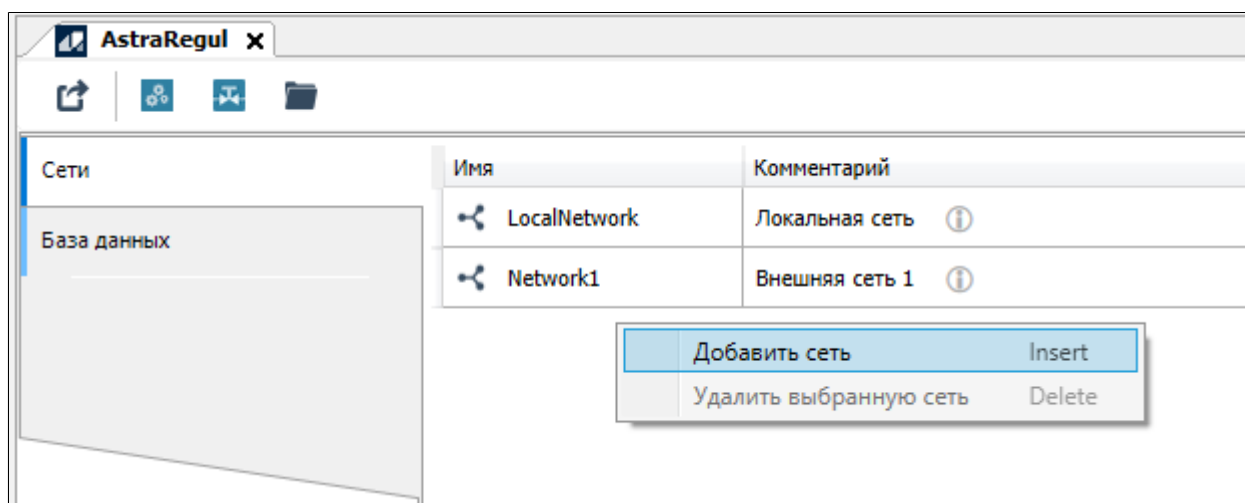
Чтобы приступить к построению топологии системы в Astra.IDE, нажмите на кнопку "Показать проект экспорта" на панели инструментов.



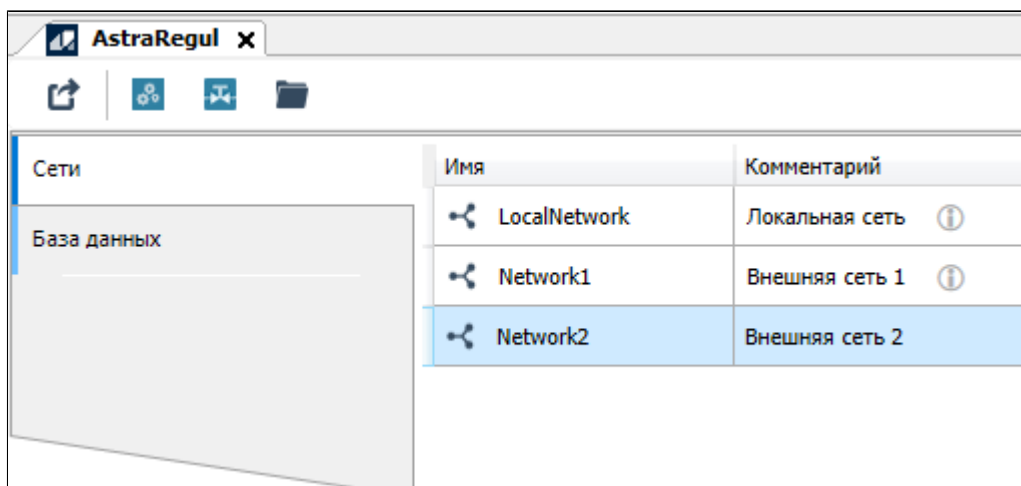
1. Если Astra.IDE предложит обновить формат сохранения для этого проекта, нажмите "OK".



2. В открывшемся окне "AstraRegul" необходимо добавить еще одну сеть, для этого нажмите правой кнопкой мыши по области окна и из контекстного меню выберите "Добавить сеть".



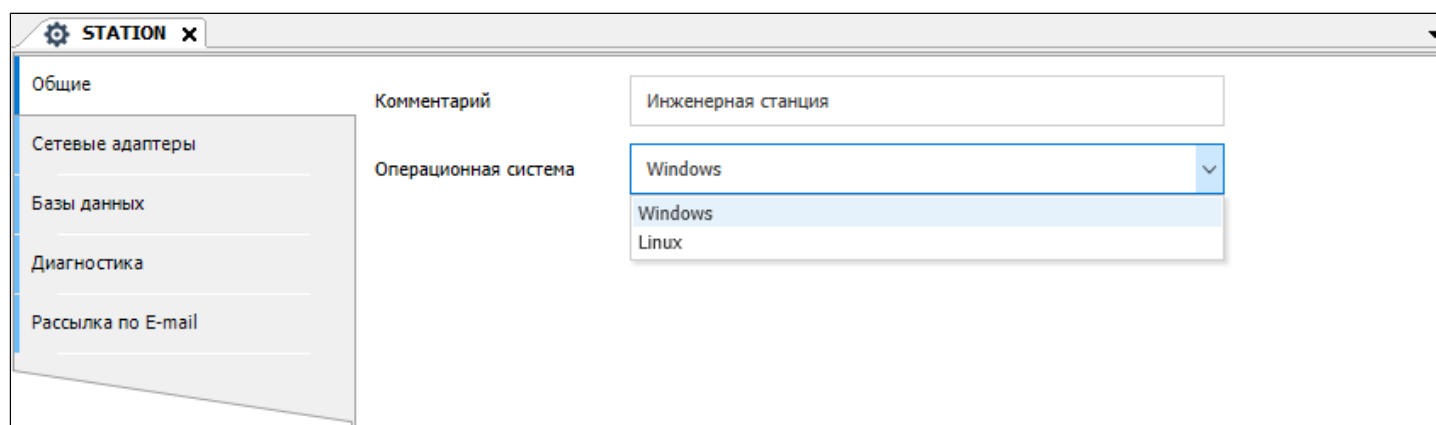
3. Задайте комментарий "Внешняя сеть 2" для появившейся сети "Network2".



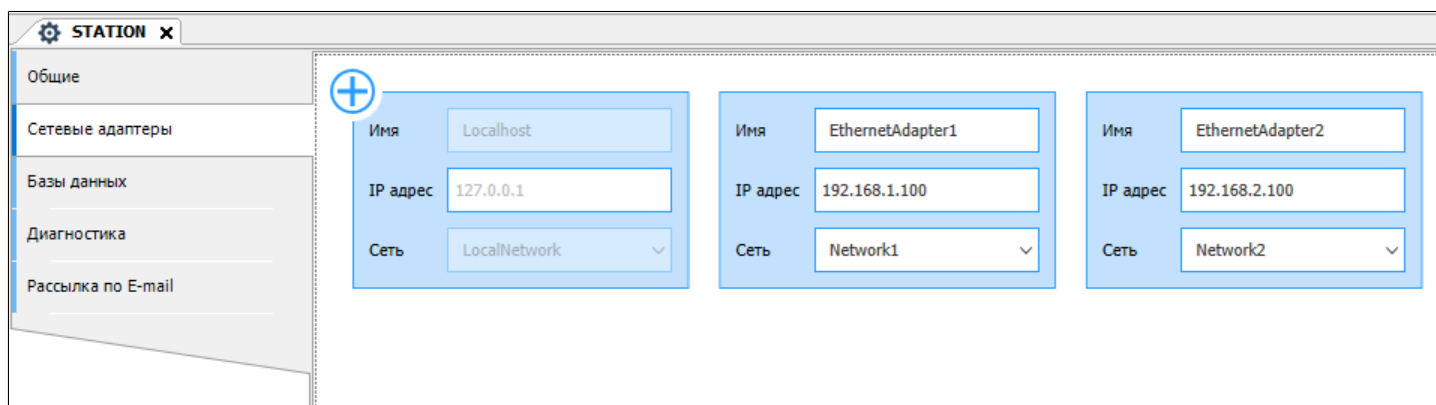


## 1.1.2.2.1. Конфигурирование инженерной станции

1. Перейдите в объект "STATION" двойным щелчком левой кнопкой мыши и во вкладке "Общие" задайте комментарий "Инженерная станция", и выберите операционную систему, на которой будет располагаться станция, в данном случае станция будет располагаться на ОС "Windows".



2. Перейдите во вкладку "Сетевые адаптеры", добавьте еще один Ethernet адаптер, нажав кнопку "Добавить", задайте IP адреса для обеих подсетей инженерной станции. Обратите внимание на название сети.



3. Перейдите во вкладку "Базы данных", проверьте пути расположения баз данных, а также настроечные параметры для хранения данных.

STATION x

Общие

Сетевые адаптеры

Базы данных

Диагностика

Рассылка по E-mail

Базы данных

- Historian
- MS SQL Server
- PostgreSQL Server

Historian

Каталог БД	C:\AstraRegul\Astra.Historian\Databases
Каталог очереди БД	C:\AstraRegul\Astra.Historian\Queues
Объем БД, Мб	2000
Период хранения, дн.	365
Буфер, Мб	512
Кэш, Мб	128
Алгоритм сжатия	Без сжатия
Число блоков в пакете	10
Стратегия пересжатия	Выключено

4. Перейдите во вкладку "Диагностика" и установите флаг "SNMP", настройки оставьте по умолчанию.

The screenshot shows the 'STATION' configuration window with the 'Диагностика' (Diagnosis) tab selected. The left sidebar contains the following menu items: 'Общие' (General), 'Сетевые адаптеры' (Network adapters), 'Базы данных' (Databases), 'Диагностика' (Diagnosis), and 'Рассылка по E-mail' (Email distribution). The main configuration area is titled 'Диагностика' and contains the following settings:

- SNMP
- Syslog

**SNMP**

Порт для опроса	161	Пароль для опроса	public
Порт для уведомлений	162	Пароль для уведомлений	public
Версия SNMP	SNMP v2c		
Уровень безопасности	Без аутентификации и без шифрования		
Тип аутентификации	MD5		
Тип шифрования	AES		

**Параметры опроса**

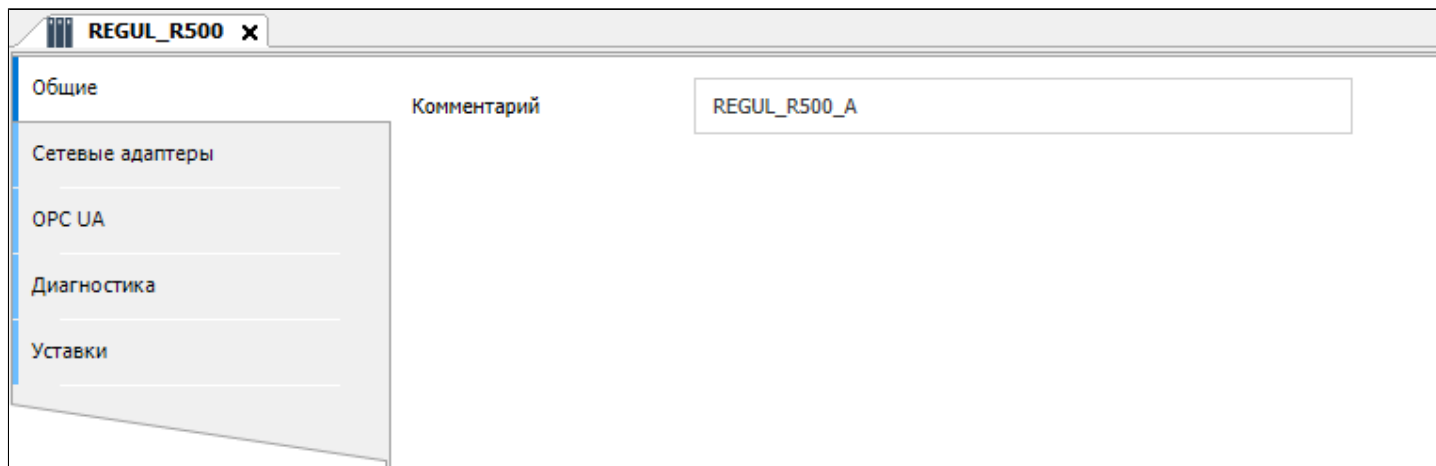
Имя пользователя			
Ключ аутентификации		Ключ шифрования	

**Параметры уведомлений**

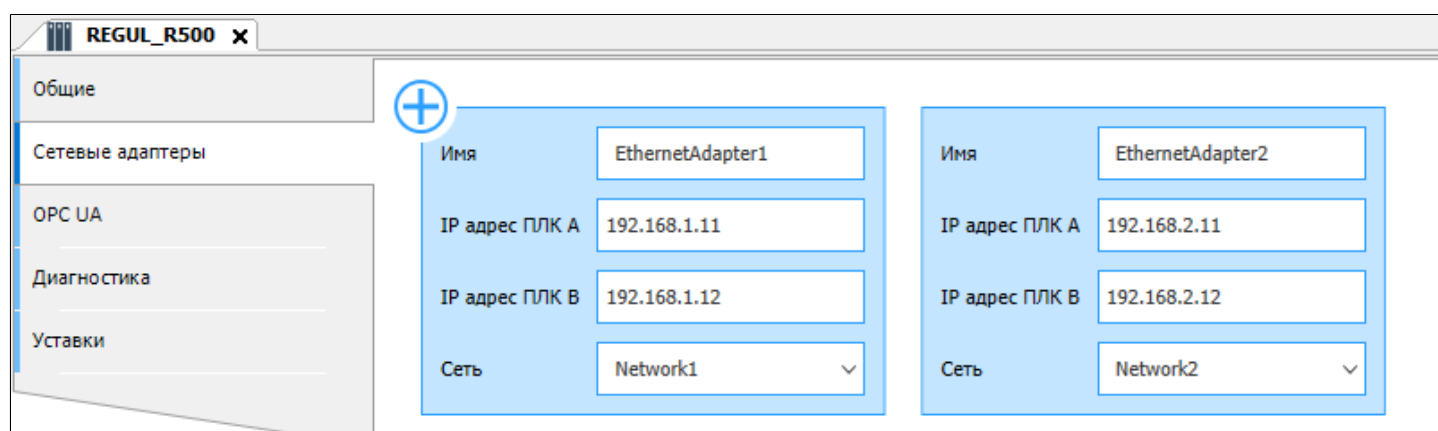
Идентификатор			
Имя пользователя			
Ключ аутентификации		Ключ шифрования	

## 1.1.2.2.2. Конфигурирование ПЛК REGUL R500

1. Перейдите в объект "REGUL\_R500" двойным щелчком левой кнопки мыши, во вкладке "Общие" задайте комментарий "REGUL\_R500\_A".



2. Перейдите во вкладку "Сетевые адаптеры", добавьте еще один Ethernet адаптер, нажав кнопку "Добавить", задайте IP адреса для ПЛК А и ПЛК Б из сканера сети. Обратите внимание на название сети.

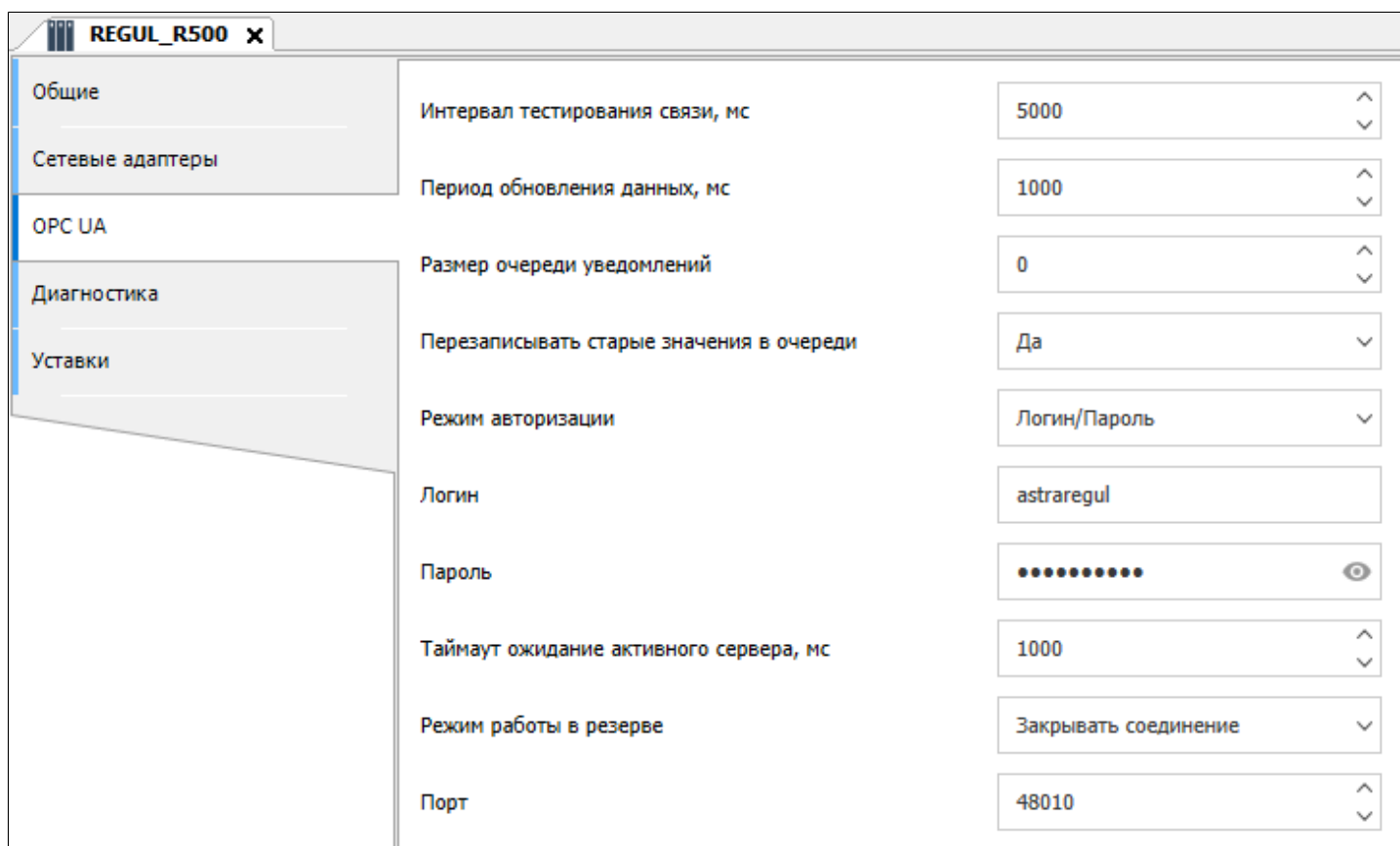


3. Перейдите во вкладку "ОПС UA" и задайте следующие настройки для ОПС UA сервера:

Параметр	Значение
Интервал тестирования связи	5000
Период обновления данных	1000
Размер очереди уведомлений	0
Перезаписывать старые значения в очереди	Да
Режим авторизации	Логин/Пароль
Логин	"astraregul" или тот, который был добавлен при конфигурировании сервера ОПС UA
Пароль	"astraregul" или тот, который был добавлен при конфигурировании сервера ОПС UA
Таймаут ожидания активного сервера	1000
Режим работы в резерве	Закрывать соединение

Порт	48010
------	-------

В результате должны получиться следующие настройки:



The screenshot shows the configuration window for REGUL\_R500. The left sidebar contains the following menu items: Общие, Сетевые адаптеры, OPC UA, Диагностика, and Уставки. The main area displays the following settings:

Интервал тестирования связи, мс	5000
Период обновления данных, мс	1000
Размер очереди уведомлений	0
Перезаписывать старые значения в очереди	Да
Режим авторизации	Логин/Пароль
Логин	astraregul
Пароль	••••••••
Таймаут ожидания активного сервера, мс	1000
Режим работы в резерве	Закрывать соединение
Порт	48010

4. Перейдите на вкладку "Диагностика" и проверьте, что напротив пункта "Диагностика модулей" установлен флаг, настройки оставьте по умолчанию.

REGUL\_R500 x

Общие

Сетевые адаптеры

OPC UA

Диагностика

Уставки

Диагностика

SNMP

Syslog

Диагностика модулей

SNMP

Порт для опроса: 161

Порт для уведомлений: 162

Версия SNMP: SNMP v2c

Уровень безопасности: Без аутентификации и без шифрования

Тип аутентификации: MD5

Тип шифрования: AES

**Параметры опроса**

Имя пользователя:

Ключ аутентификации:

Пароль для опроса: public

Ключ шифрования:

**Параметры уведомлений**

Идентификатор:

Имя пользователя:

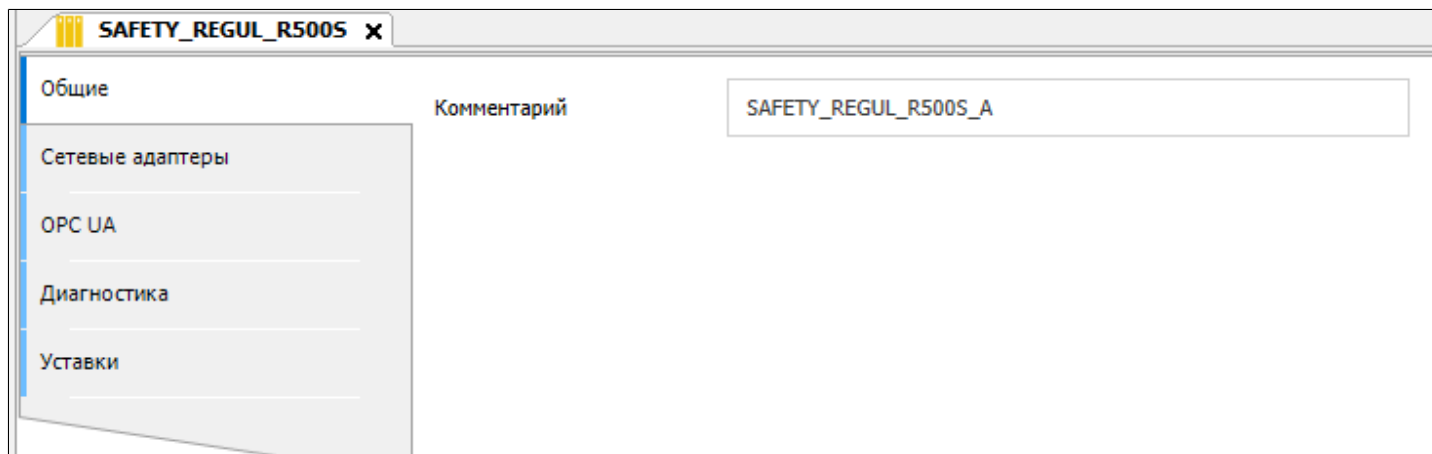
Ключ аутентификации:

Пароль для уведомлений: public

Ключ шифрования:

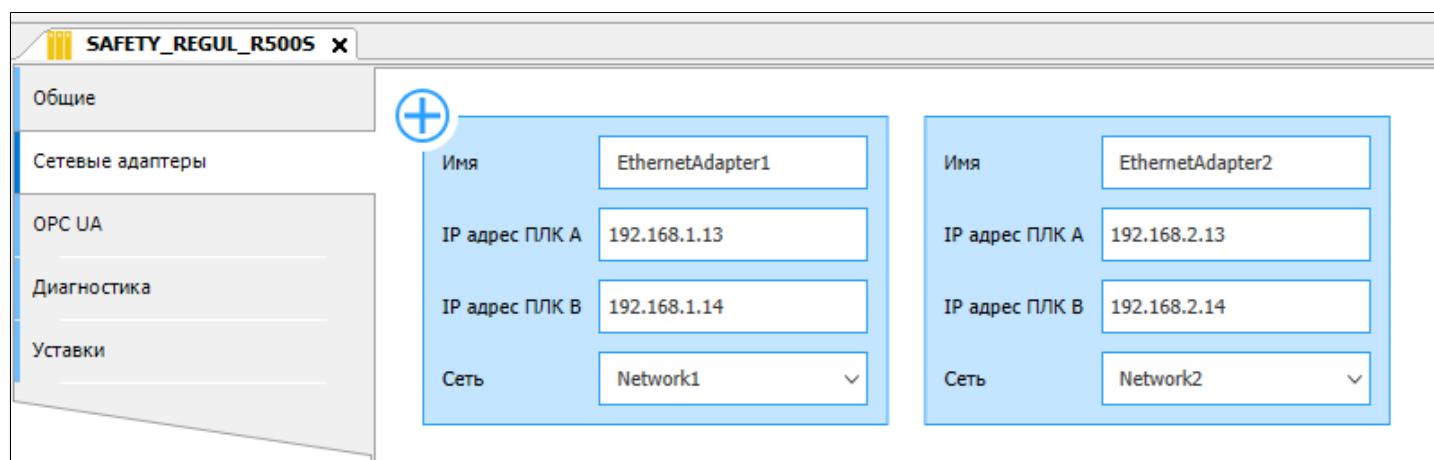
## 1.1.2.2.3. Конфигурирование ПЛК SAFETY REGUL R500S

1. Перейдите в объект "SAFETY\_REGUL\_R500S" двойным щелчком левой кнопки мыши, во вкладке "Общие" задайте комментарий "SAFETY\_REGUL\_R500S\_A".





2. Перейдите во вкладку "Сетевые адаптеры", добавьте еще один Ethernet адаптер, нажав кнопку "Добавить", задайте IP адреса для ПЛК А и ПЛК Б из сканера сети. Обратите внимание на название сети.



3. Перейдите во вкладку "OPC UA" и задайте следующие настройки для OPC UA сервера:

Параметр	Значение
Интервал тестирования связи	5000
Период обновления данных	1000
Размер очереди уведомлений	0
Перезаписывать старые значения в очереди	Да
Режим авторизации	Логин/Пароль
Логин	"astraregul" или тот, который был добавлен при конфигурировании сервера OPC UA
Пароль	"astraregul" или тот, который был добавлен при конфигурировании сервера OPC UA
Таймаут ожидания активного сервера	1000
Режим работы в резерве	Закрывать соединение

Порт	48010
------	-------

В результате должны получиться следующие настройки:

The screenshot shows the configuration window for SAFETY\_REGUL\_R500S. The left sidebar contains the following menu items: Общие, Сетевые адаптеры, OPC UA, Диагностика, and Уставки. The main area displays the following settings:

Интервал тестирования связи, мс	5000
Период обновления данных, мс	1000
Размер очереди уведомлений	0
Перезаписывать старые значения в очереди	Да
Режим авторизации	Логин/Пароль
Логин	astraregul
Пароль	••••••••
Таймаут ожидание активного сервера, мс	1000
Режим работы в резерве	Закрывать соединение
Порт	48010

4. Перейдите на вкладку "Диагностика" и проверьте, что напротив пункта "Диагностика модулей" установлен флаг, настройки оставьте по умолчанию.

**SAFETY\_REGUL\_R500S** X

Общие  
Сетевые адаптеры  
OPC UA  
Диагностика  
Уставки

**Диагностика**

SNMP  
 Syslog  
 Диагностика модулей

**SNMP**

Порт для опроса	161	↕	Пароль для опроса	public
Порт для уведомлений	162	↕	Пароль для уведомлений	public
Версия SNMP	SNMP v2c	↕		
Уровень безопасности	Без аутентификации и без шифрования	↕		
Тип аутентификации	MD5	↕		
Тип шифрования	AES	↕		

**Параметры опроса**

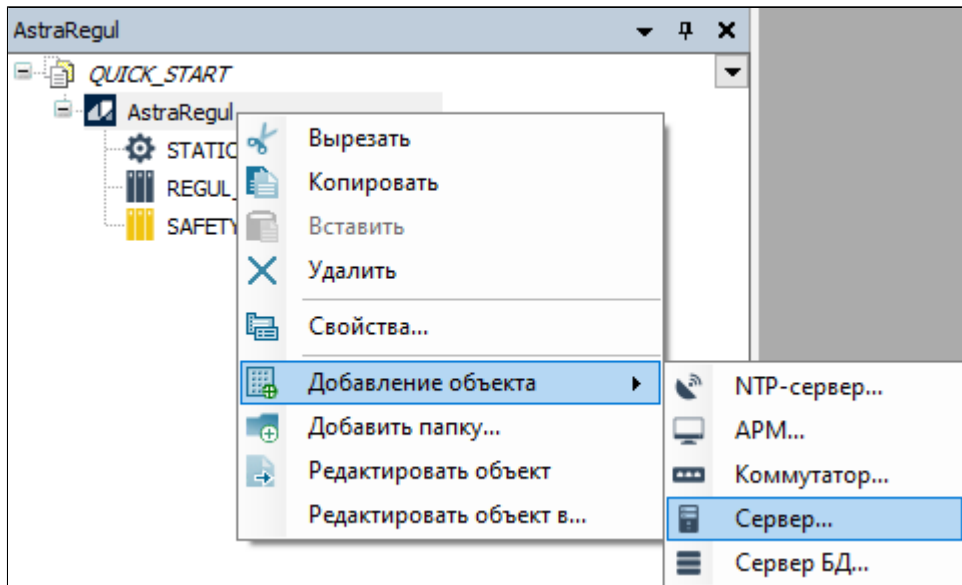
Имя пользователя				
Ключ аутентификации			Ключ шифрования	

**Параметры уведомлений**

Идентификатор				
Имя пользователя				
Ключ аутентификации			Ключ шифрования	


## 1.1.2.2.4. Конфигурирование серверов

1. Добавьте новый объект "Сервер", для этого нажмите правой кнопкой мыши по объекту "AstraRegul" и выберите "Сервер" в меню "Добавление объекта...".



2. Задайте имя "SERVER1" добавляемому объекту и нажмите кнопку "Добавить".

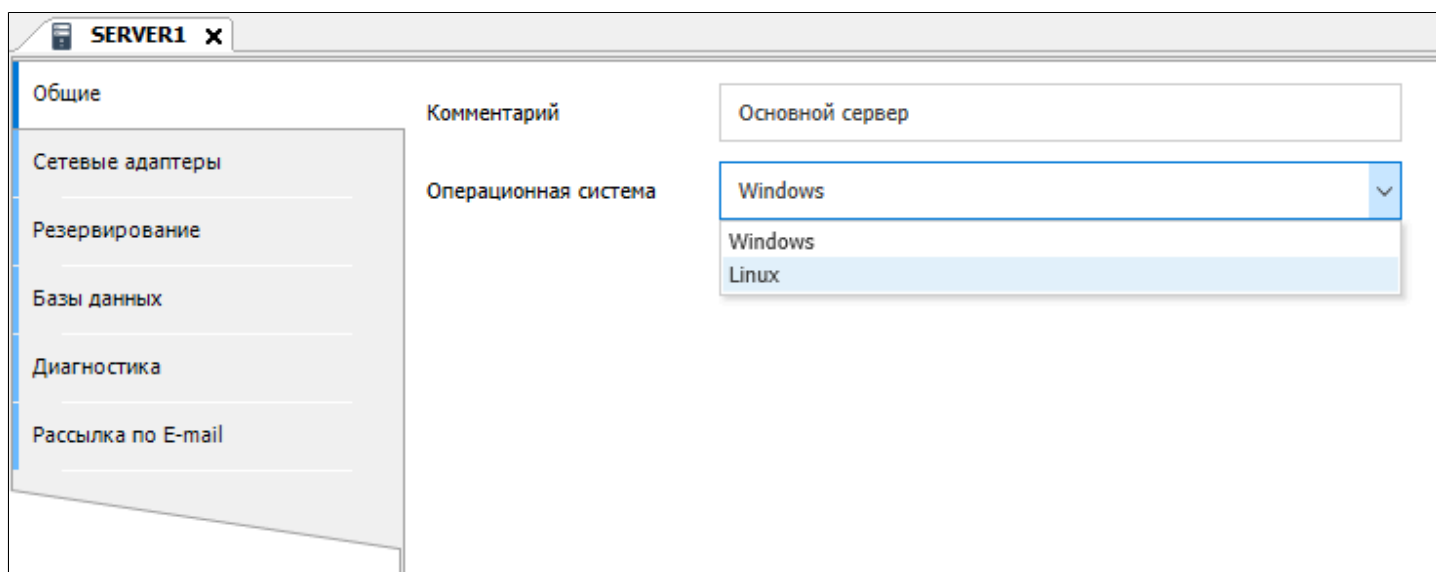
Добавить Сервер ×

 Сервер

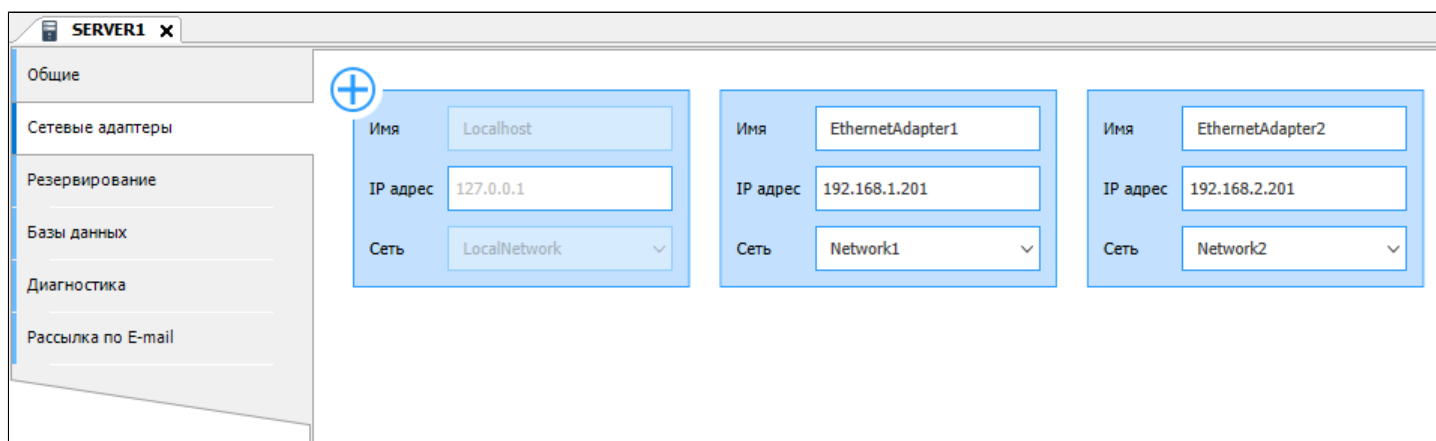
Имя

SERVER1

3. Во вкладке "Общие" объекта "SERVER1" задайте комментарий "Основной сервер" и выберите операционную систему, на которой будет располагаться сервер, в данном случае сервер будет располагаться на ОС "Linux".



4. Перейдите во вкладку "Сетевые адаптеры" и добавьте еще один Ethernet адаптер, нажав кнопку "Добавить", задайте IP адреса для основного сервера. Обратите внимание на название сети.



5. Перейдите во вкладку "Базы данных" и проверьте пути расположения баз данных, а также настроечные параметры для хранения данных.

SERVER1 x

Общие  
Сетевые адаптеры  
Резервирование  
**Базы данных**  
Диагностика  
Рассылка по E-mail

Базы данных

Historian  
 MS SQL Server  
 PostgreSQL Server

Historian

Каталог БД	<input type="text" value="/opt/AstraRegul/Astra.Historian/Databases"/>
Каталог очереди БД	<input type="text" value="/opt/AstraRegul/Astra.Historian/Queues"/>
Объем БД, Мб	<input type="text" value="2000"/>
Период хранения, дн.	<input type="text" value="365"/>
Буфер, Мб	<input type="text" value="512"/>
Кэш, Мб	<input type="text" value="128"/>
Алгоритм сжатия	<input type="text" value="Без сжатия"/>
Число блоков в пакете	<input type="text" value="10"/>
Стратегия пересжатия	<input type="text" value="Выключено"/>

6. Перейдите во вкладку "Диагностика" и установите флаг "SNMP", остальные настройки оставьте по умолчанию.

**SERVER1** x

Общие  
Сетевые адаптеры  
Резервирование  
Базы данных  
**Диагностика**  
Рассылка по E-mail

**Диагностика**

SNMP  
 Syslog

**SNMP**

Порт для опроса	161	Пароль для опроса	public
Порт для уведомлений	162	Пароль для уведомлений	public
Версия SNMP	SNMP v2c		
Уровень безопасности	Без аутентификации и без шифрования		
Тип аутентификации	MD5		
Тип шифрования	AES		

**Параметры опроса**

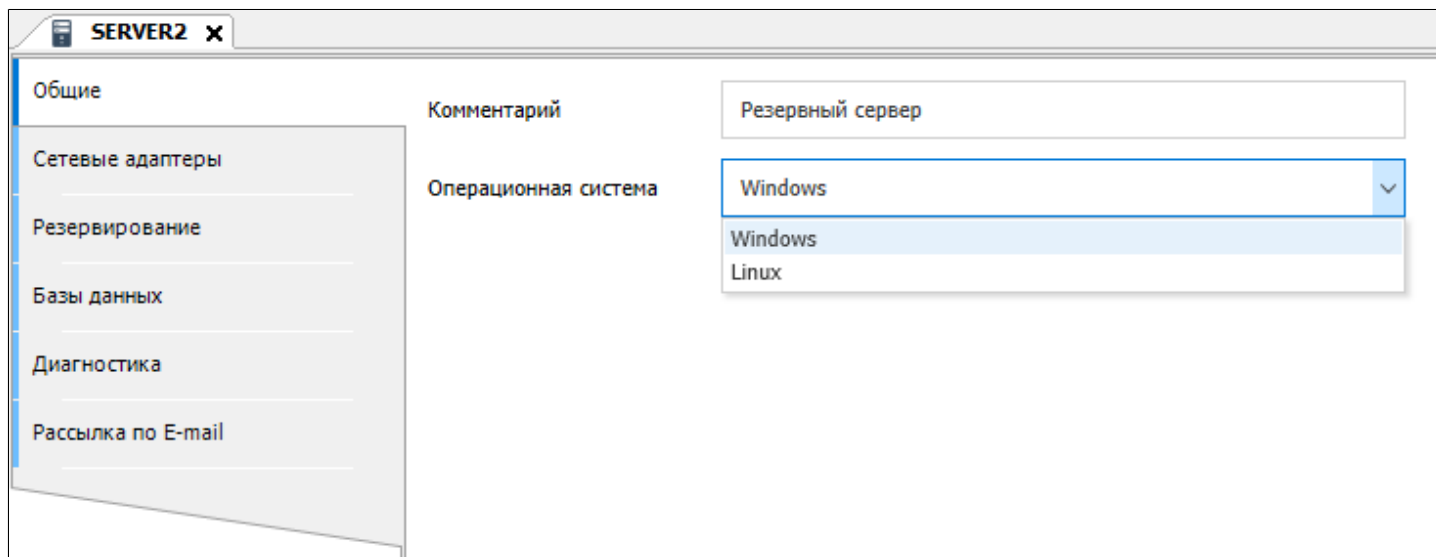
Имя пользователя			
Ключ аутентификации		Ключ шифрования	

**Параметры уведомлений**

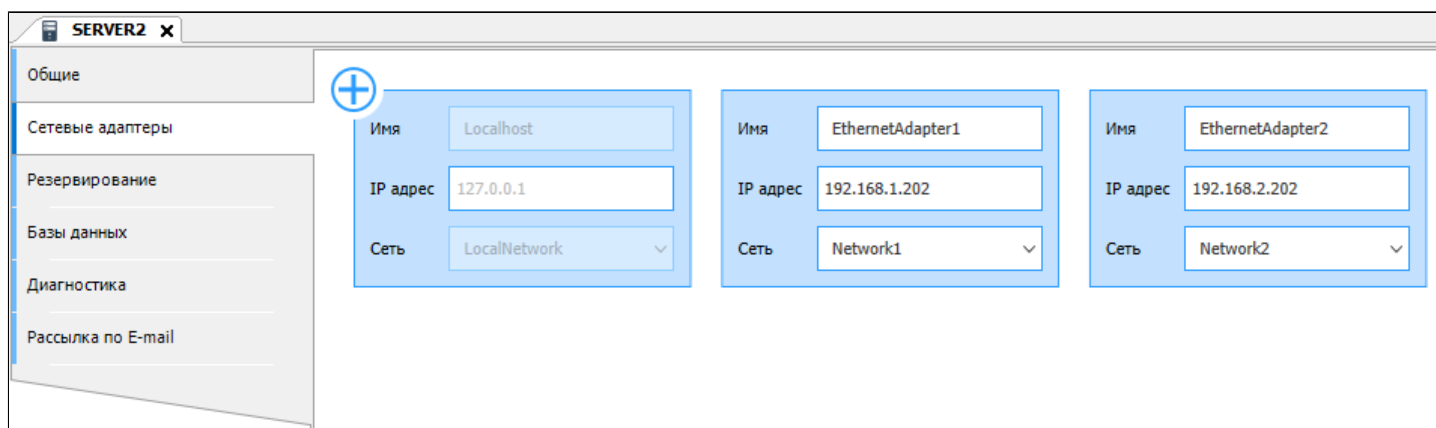
Идентификатор			
Имя пользователя			
Ключ аутентификации		Ключ шифрования	



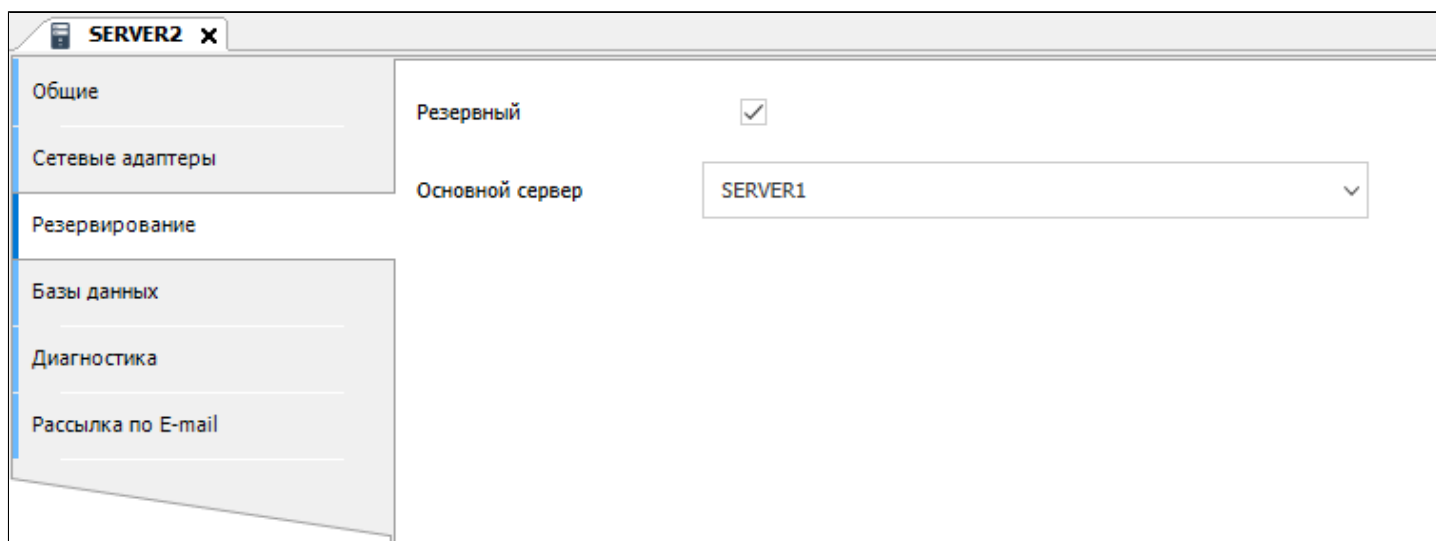
7. Аналогично основному серверу "SERVER1" добавьте резервный сервер "SERVER2". Во вкладке "Общие" задайте комментарий "Резервный сервер" и выберите операционную систему, на которой будет располагаться сервер, в данном случае сервер будет располагаться на ОС "Windows".



8. Перейдите во вкладку "Сетевые адаптеры" и добавьте еще один Ethernet адаптер, нажав кнопку "Добавить", задайте IP адреса для резервного сервера. Обратите внимание на название сети.

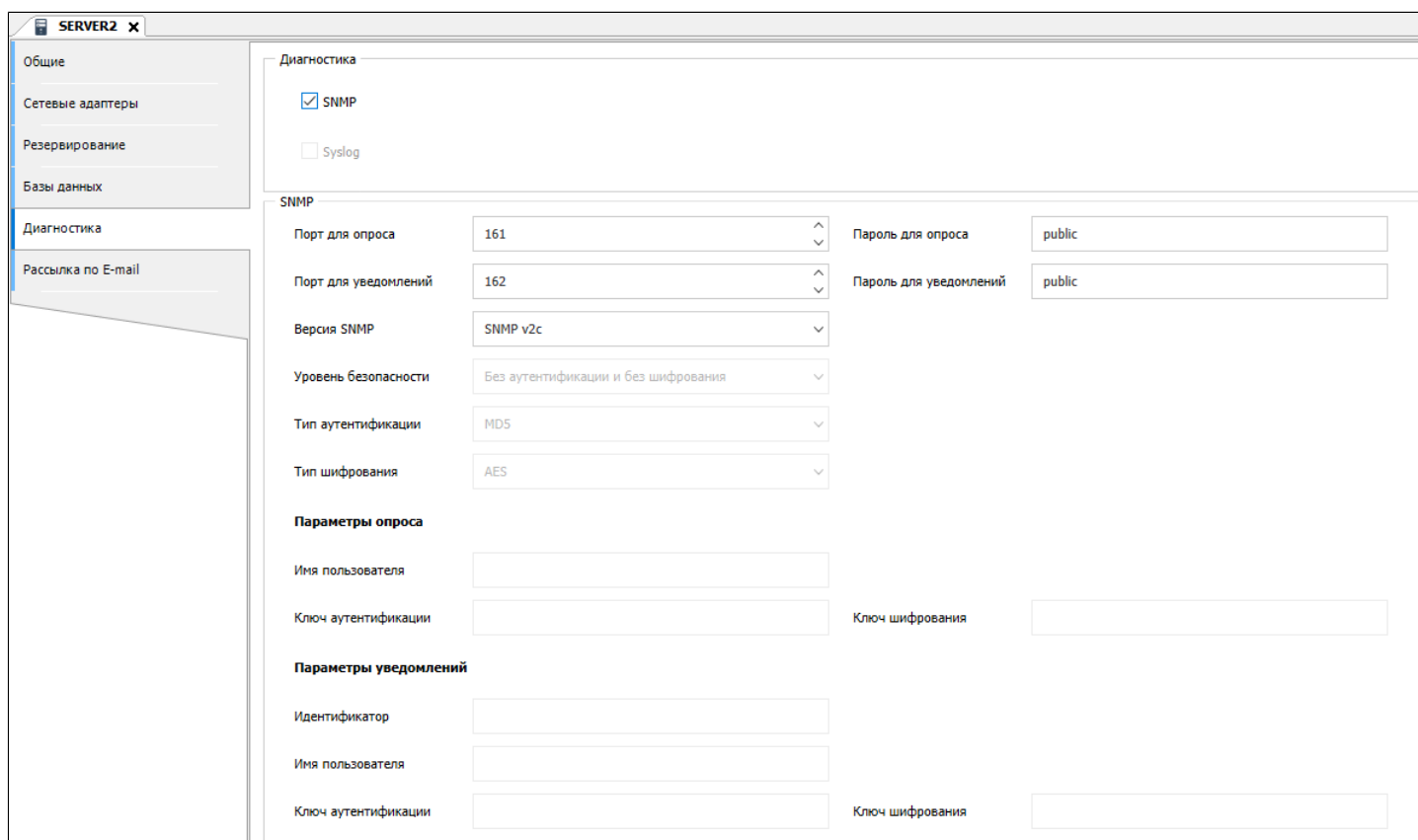


9. Перейдите во вкладку "Резервирование" и установите флажок "Резервный", выберите в качестве основного сервера "SERVER1"



The screenshot shows the configuration window for 'SERVER2'. The left sidebar contains the following menu items: 'Общие', 'Сетевые адаптеры', 'Резервирование', 'Базы данных', 'Диагностика', and 'Рассылка по E-mail'. The 'Резервирование' tab is active. In the main area, the 'Резервный' checkbox is checked, and the 'Основной сервер' dropdown menu is set to 'SERVER1'.

10. Перейдите во вкладку "Диагностика" и установите флаг "SNMP", настройки оставьте по умолчанию.

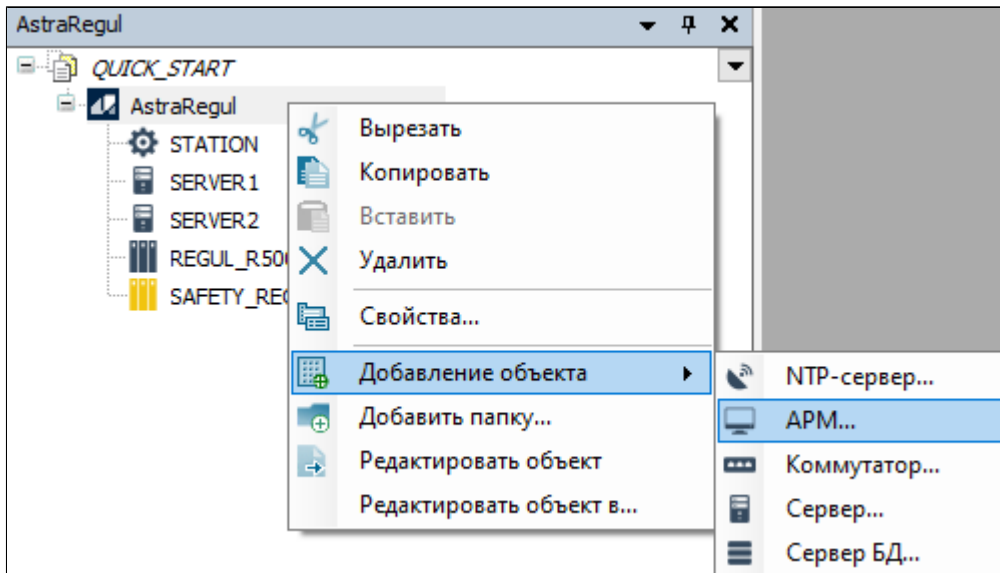


The screenshot shows the configuration window for 'SERVER2' with the 'Диагностика' (Diagnosis) tab selected. The 'SNMP' checkbox is checked, while 'Syslog' is unchecked. The 'SNMP' section contains the following settings:

Parameter	Value	Parameter	Value
Порт для опроса	161	Пароль для опроса	public
Порт для уведомлений	162	Пароль для уведомлений	public
Версия SNMP	SNMP v2c		
Уровень безопасности	Без аутентификации и без шифрования		
Тип аутентификации	MDS		
Тип шифрования	AES		
<b>Параметры опроса</b>			
Имя пользователя			
Ключ аутентификации		Ключ шифрования	
<b>Параметры уведомлений</b>			
Идентификатор			
Имя пользователя			
Ключ аутентификации		Ключ шифрования	


## 1.1.2.2.5. Конфигурирование АРМ

1. Добавьте новый объект "АРМ", для этого нажмите правой кнопкой мыши по объекту "AstraRegul" и выберите "АРМ..." в меню "Добавление объекта...".



2. Задайте объекту "АРМ" название "АРМ1" и нажмите кнопку "Добавить".

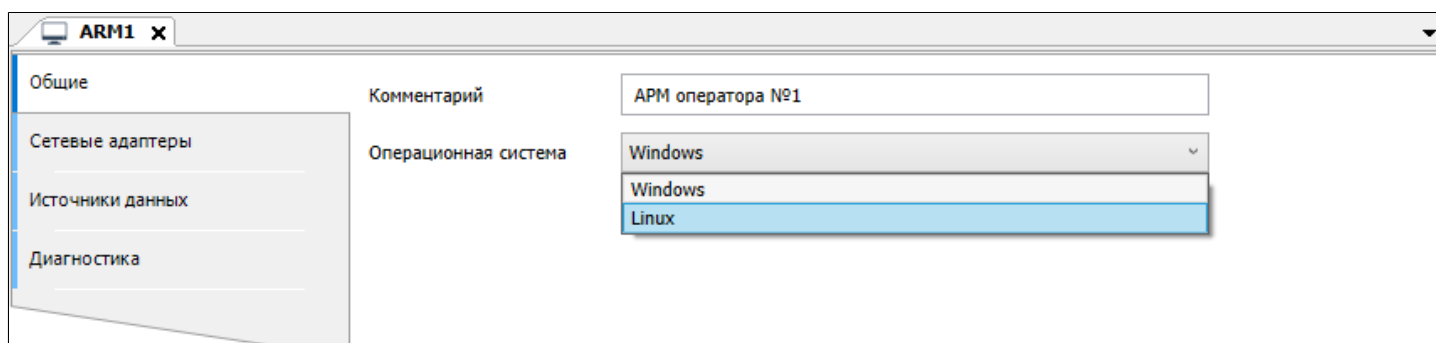
Добавить ARM ×

 ARM

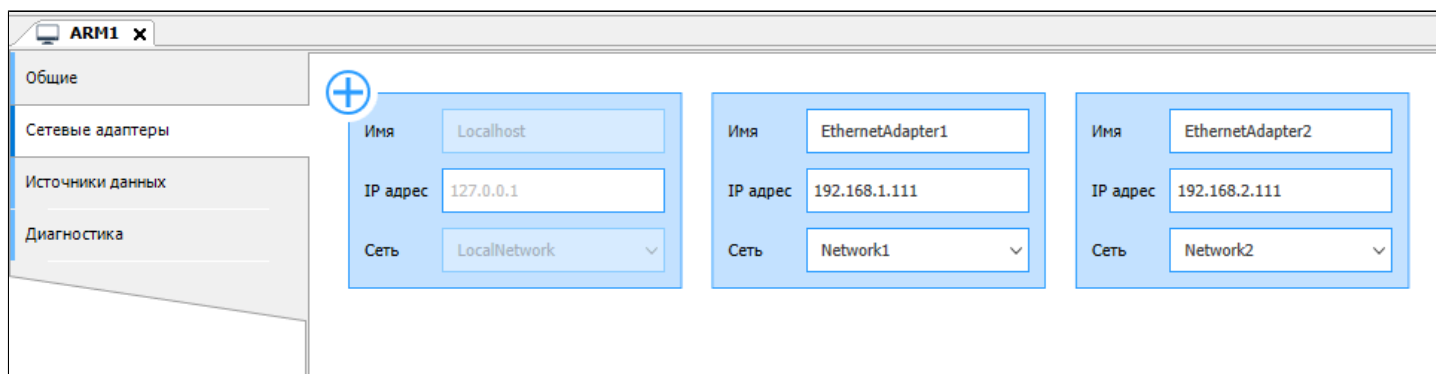
Имя

ARM1

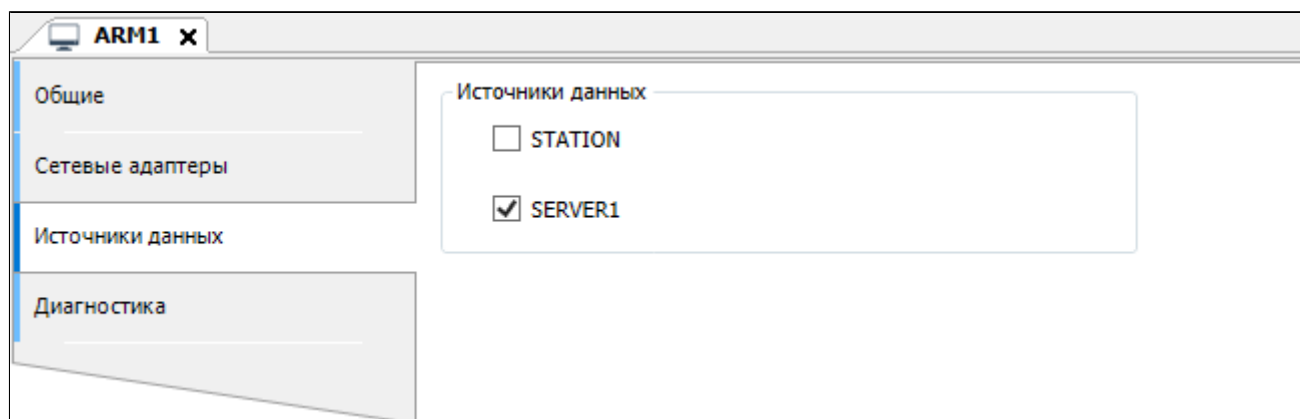
3. Во вкладке "Общие" задайте комментарий "АРМ оператора №1" и выберите операционную систему, на которой будет располагаться АРМ, в данном случае АРМ будет располагаться на ОС "Linux".



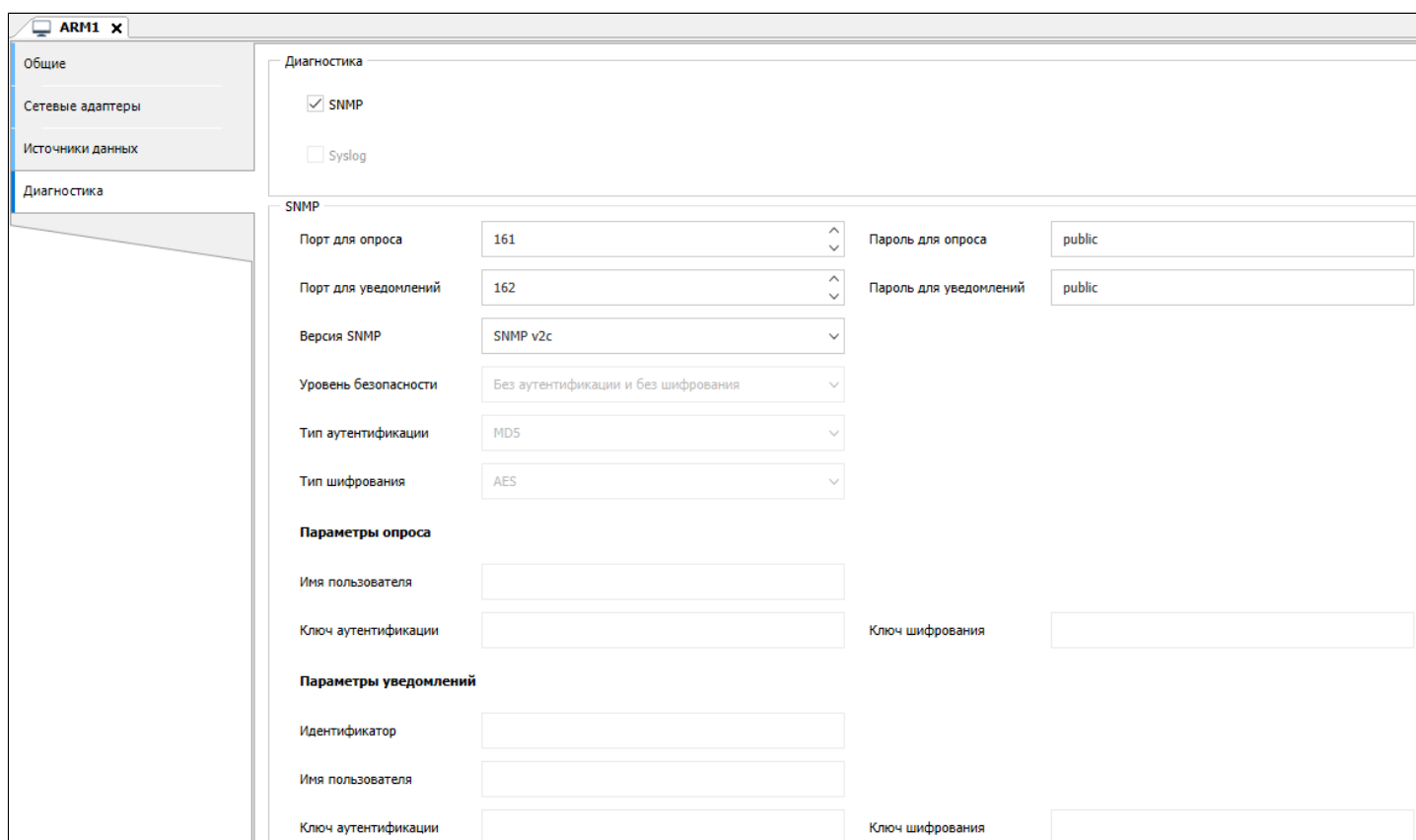
4. Перейдите во вкладку "Сетевые адаптеры" и добавьте еще один Ethernet адаптер, нажав кнопку "Добавить", задайте IP адреса для АРМ оператора. Обратите внимание на название сети.



5. Перейдите во вкладку "Источники данных" и установите флаг "SERVER1". То есть данные для "АРМ оператора №1" будут браться из основного сервера.

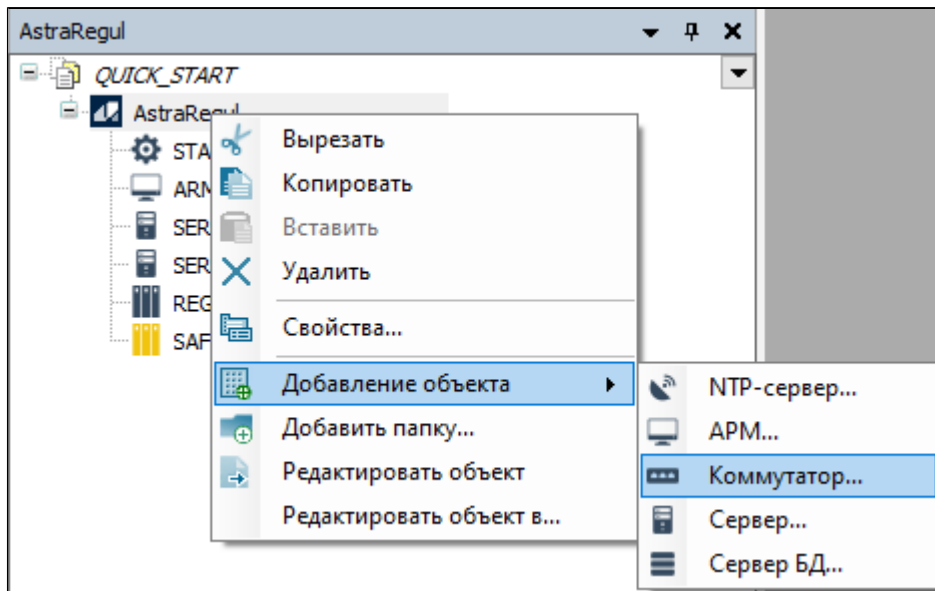


6. Перейдите во вкладку "Диагностика" и установите флаг "SNMP", настройки оставьте по умолчанию.



## 1.1.2.2.6. Конфигурирование коммутаторов

1. Добавьте новый объект "Коммутатор", для этого нажмите правой кнопкой мыши по объекту "AstraRegul" и выберите "Коммутатор..." в меню "Добавление объекта..."



2. Задайте объекту "Коммутатор" имя "SWITCH1" и нажмите "Добавить".

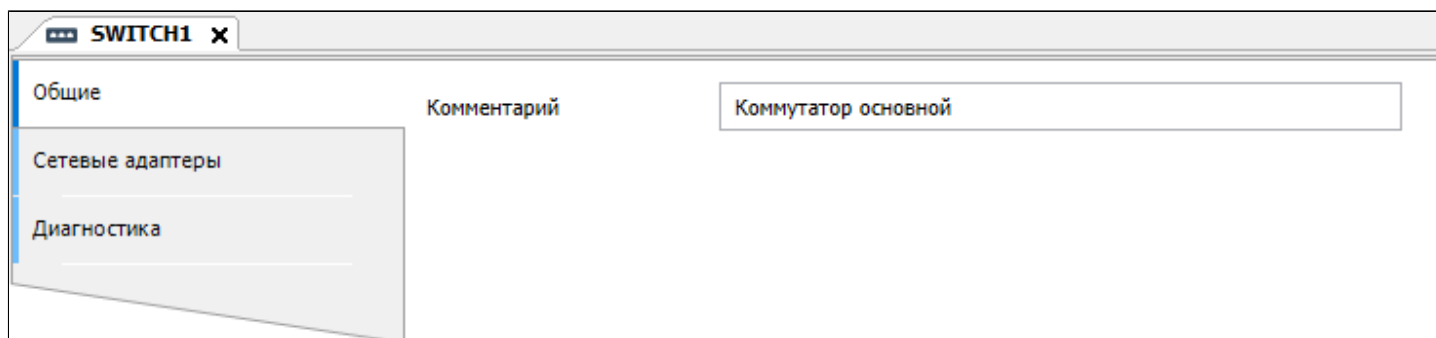
Добавить Коммутатор ×

... Коммутатор

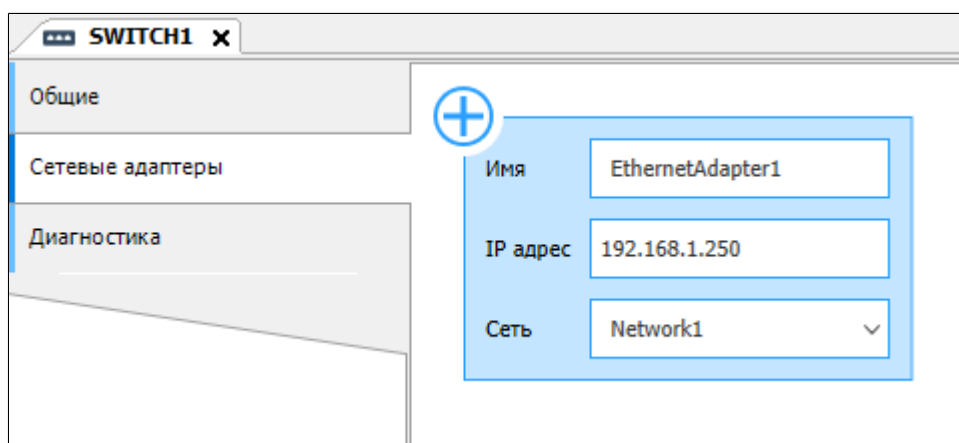
Имя



3. Во вкладке "Общие" задайте комментарий "Коммутатор основной".



4. Перейдите во вкладку "Сетевые адаптеры" и задайте IP адрес для основного коммутатора. Обратите внимание на название сети.

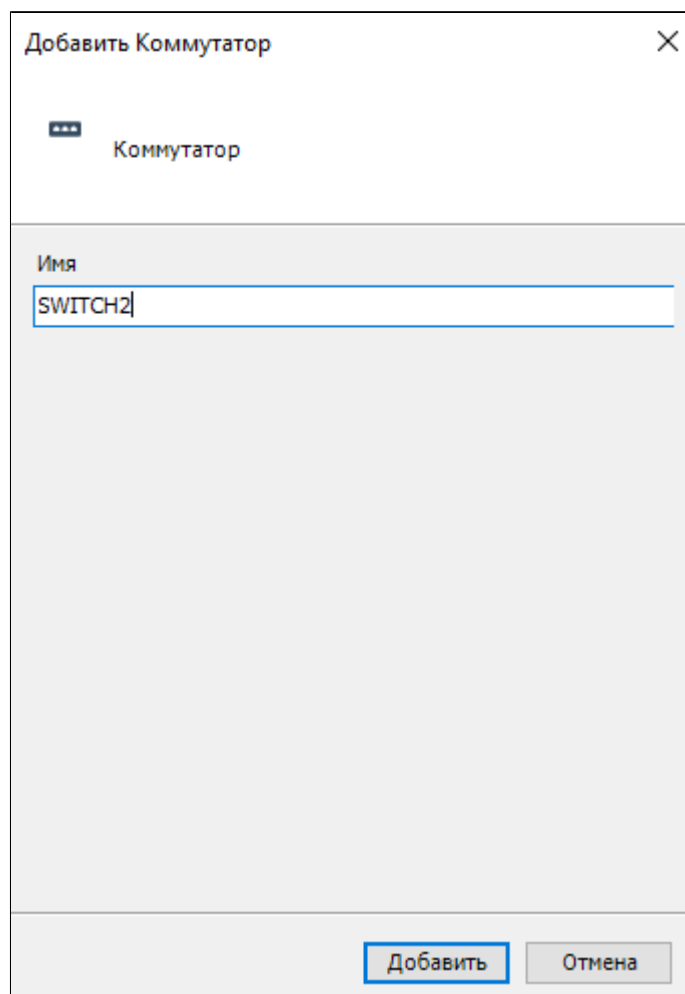


5. Перейдите во вкладку "Диагностика" и установите флаг "SNMP", настройки оставьте по умолчанию.

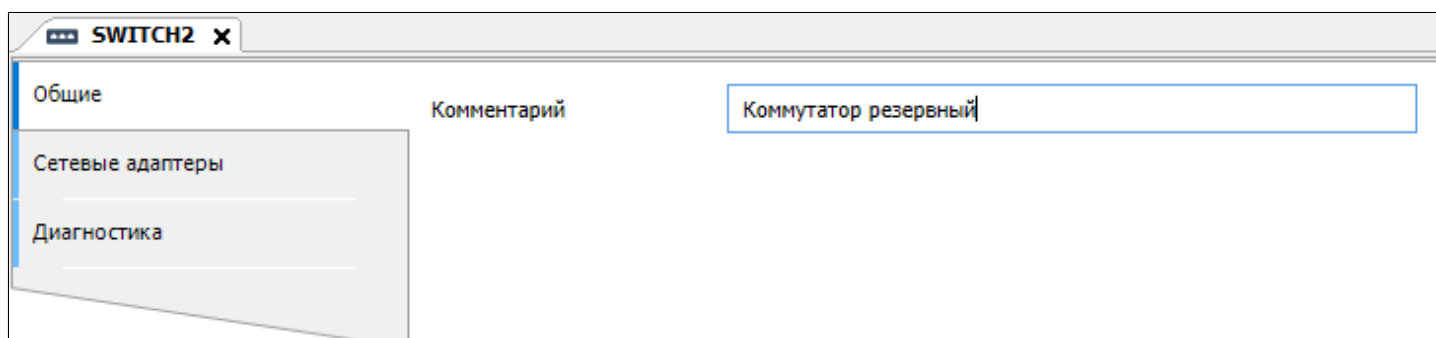
The screenshot shows the configuration interface for a device named SWITCH1. The left sidebar contains three tabs: 'Общие' (General), 'Сетевые адаптеры' (Network adapters), and 'Диагностика' (Diagnosis), with the latter being the active tab. The main content area is titled 'Диагностика' and contains the following settings:

- Диагностика**
  - SNMP
  - Syslog
- SNMP**
  - Порт для опроса: 161
  - Порт для уведомлений: 162
  - Версия SNMP: SNMP v2c
  - Уровень безопасности: Без аутентификации и без шифрования
  - Тип аутентификации: MD5
  - Тип шифрования: AES
- Параметры опроса**
  - Имя пользователя: [empty field]
  - Пароль для опроса: public
  - Ключ аутентификации: [empty field]
  - Ключ шифрования: [empty field]
- Параметры уведомлений**
  - Идентификатор: [empty field]
  - Имя пользователя: [empty field]
  - Ключ аутентификации: [empty field]
  - Ключ шифрования: [empty field]

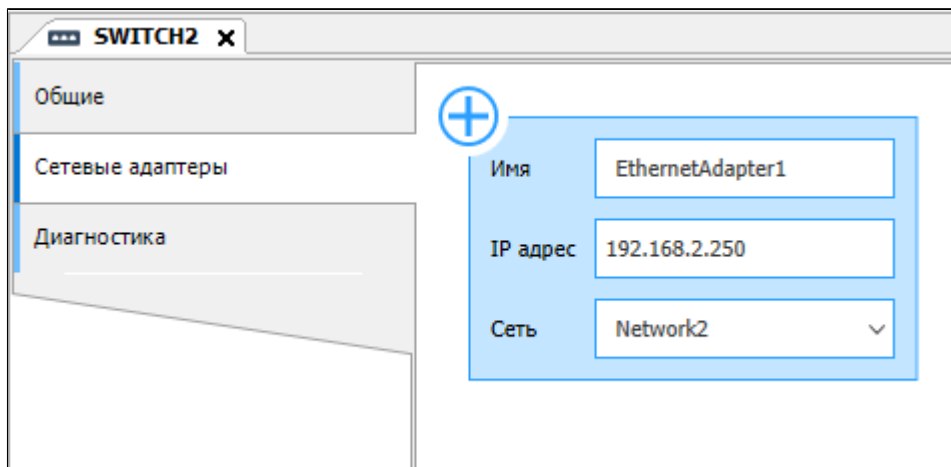
6. Аналогичным образом добавьте еще один коммутатор и задайте ему имя "SWITCH2".



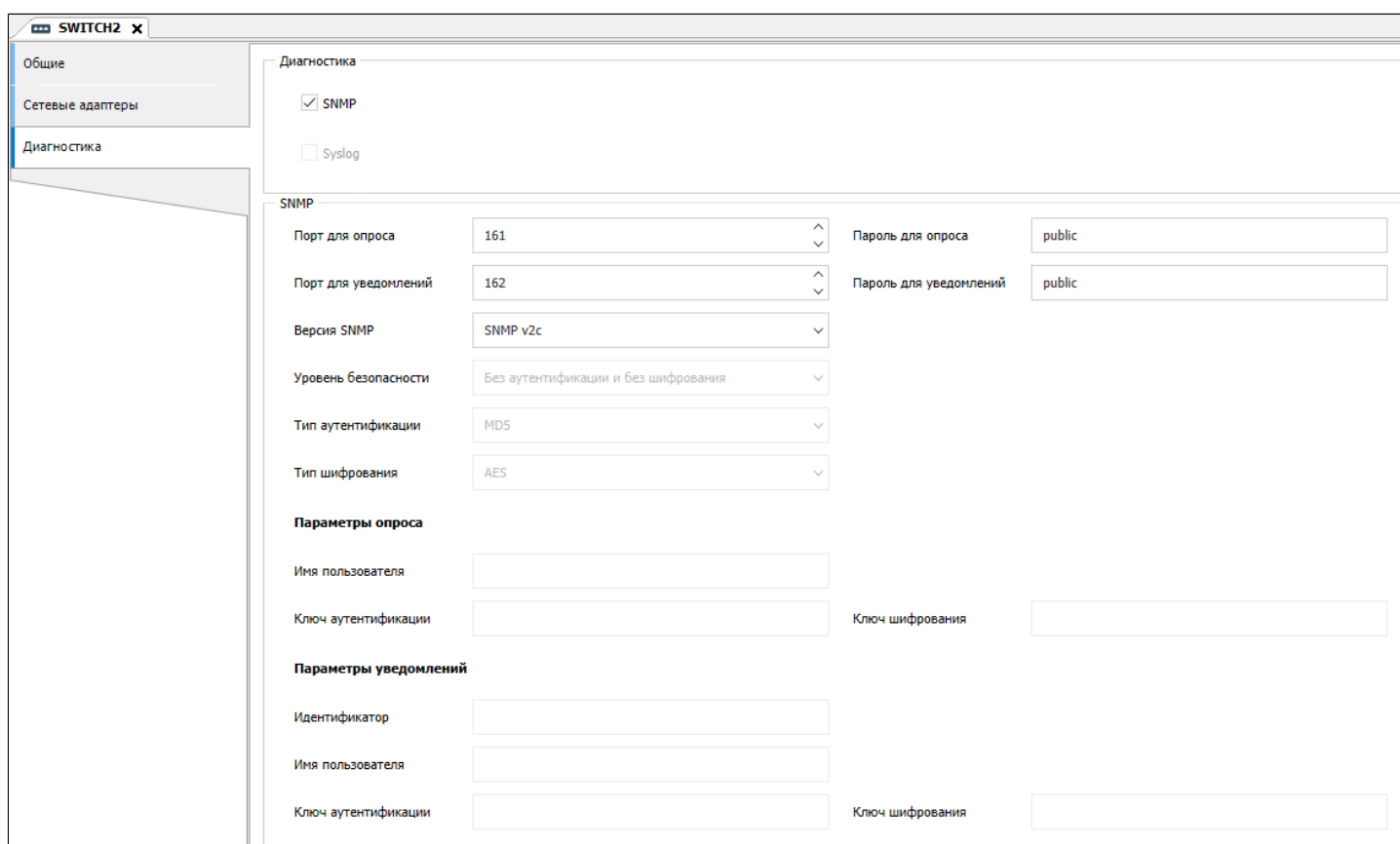
7. Во вкладке "Общие" добавьте комментарий "Коммутатор резервный".



8. Перейдите во вкладку "Сетевые адаптеры" и задайте IP адрес для основного коммутатора. Обратите внимание на название сети.

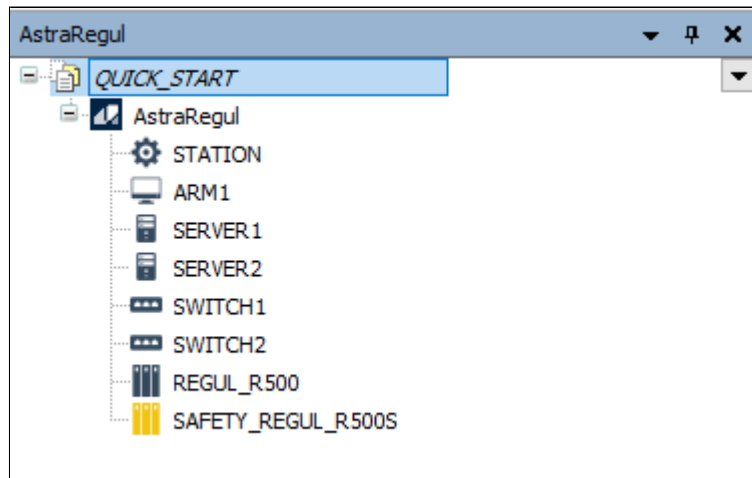


9. Перейдите во вкладку "Диагностика" и установите флаг "SNMP", настройки оставьте по умолчанию.

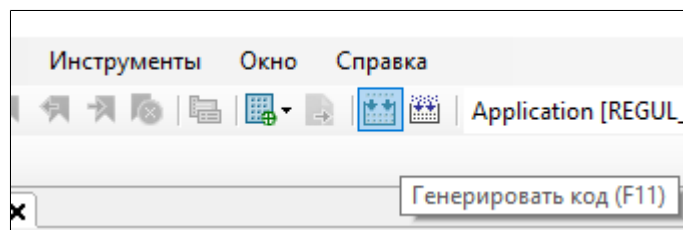


## 1.1.2.2.7. Генерация кода

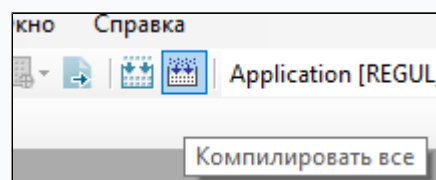
Должна получиться следующая топология сети:



После того как все устройства в топологии были сконфигурированы, нажмите кнопку "Генерировать код(F11)".



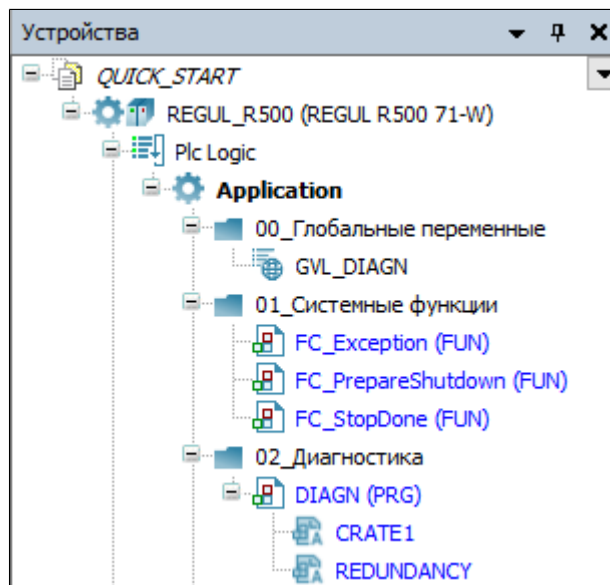
Если в проекте несколько Target-устройств, можно генерировать код с помощью кнопки "Компилировать все".



В проекте автоматически будут сформированы следующие папки:

- › 00\_Глобальные переменные
- › 01\_Системные функции
- › 02\_Диагностика

В каждой папке автоматически будут сформированы объекты, каждый из которых описан далее.



В проект будет добавлен объект "Конфигурация задач", если этот объект не был добавлен ранее вручную.

## Глобальные переменные

В папке "00\_Глобальные переменные" автоматически сформирован список глобальных переменных "GVL\_DIAGN" для диагностики ПЛК в соответствии с конфигурацией крейтов.



Если конфигурация крейта была изменена, необходимо повторно скомпилировать код приложения.



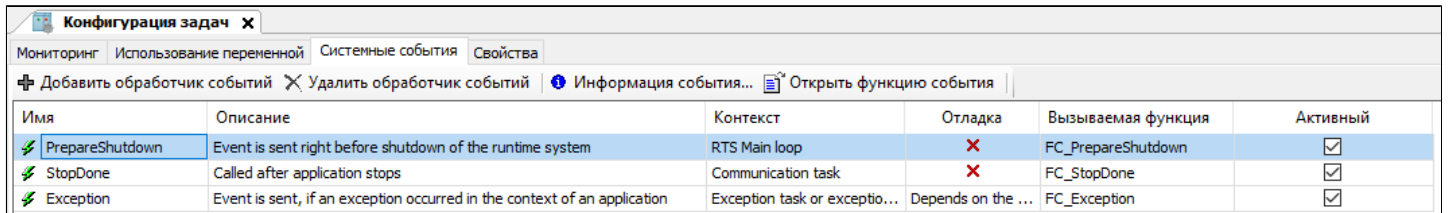
Список глобальных переменных "GVL\_DIAGN" не доступен для редактирования.

Сформированные глобальные переменные для диагностики модулей автоматически привязываются к карте адресов соответствующего модуля.

# Системные функции

В папке "01\_Системные функции" автоматически генерируются функции, которые вызываются при возникновении системных событий:

- › Exception – событие, срабатывающее при исключении;
- › PrepareShutdown – событие, срабатывающее перед выключением ПЛК;
- › StopDone – событие, срабатывающее при переходе в режим "Стоп".



Имя	Описание	Контекст	Отладка	Вызываемая функция	Активный
PrepareShutdown	Event is sent right before shutdown of the runtime system	RTS Main loop	✗	FC_PrepareShutdown	<input checked="" type="checkbox"/>
StopDone	Called after application stops	Communication task	✗	FC_StopDone	<input checked="" type="checkbox"/>
Exception	Event is sent, if an exception occurred in the context of an application	Exception task or exceptio...	Depends on the ...	FC_Exception	<input checked="" type="checkbox"/>

Во всех функциях сбрасываются флаги, которые отвечают за передачу данных на сервер ввода/вывода, тем самым сервер получает информацию, что ПЛК недоступен для опроса.

## Диагностика

В папке "02\_Диагностика" автоматически формируется программа "DIAGN" для диагностики всех модулей ПЛК, а также действие "REDUNDANCY" для определения режима при резервировании ПЛК.



Программа "DIAGN" недоступна для редактирования и автоматически вызывается в задаче "TASK\_DIAGN".

Задача "TASK\_DIAGN" имеет следующие параметры вызова:

- › Приоритет задачи: 10.
- › Интервал исполнения задачи: 1000 мс.
- › Тип исполнения задачи: Циклическое.

### **1.1.2.3.            Разработка            технологических алгоритмов**

После создания топологии системы АСУ ТП и автоматического создания программы для диагностики аппаратных компонентов можно приступить к разработке технологических алгоритмов.



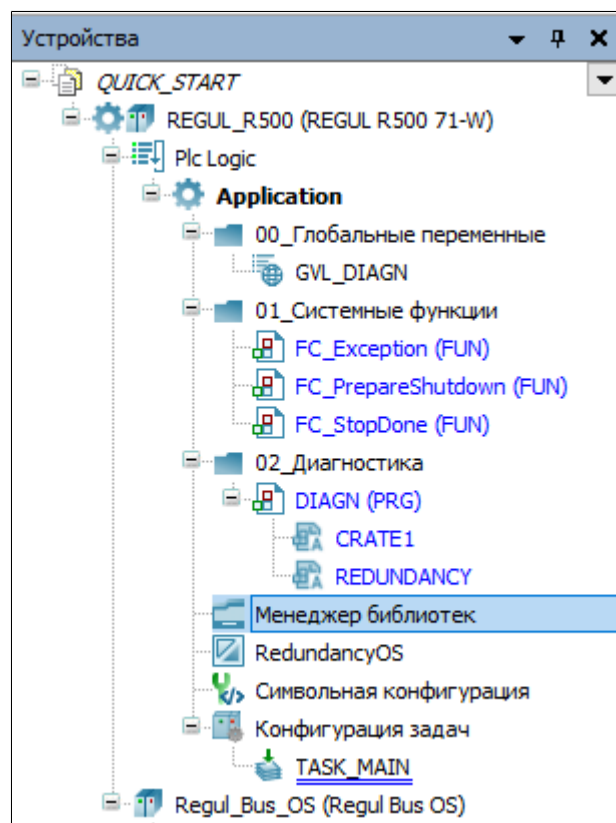
## 1.1.2.3.1. Добавление библиотек

Для добавления стандартных библиотек "PsBase", "PsDiagn", "PsTechEE", "PsSIS" к проекту выполните следующие действия:

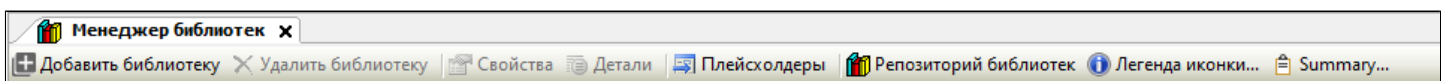


Начиная с версии 1.7.0.2 все библиотеки входят в состав пакета "Astra.IDE".

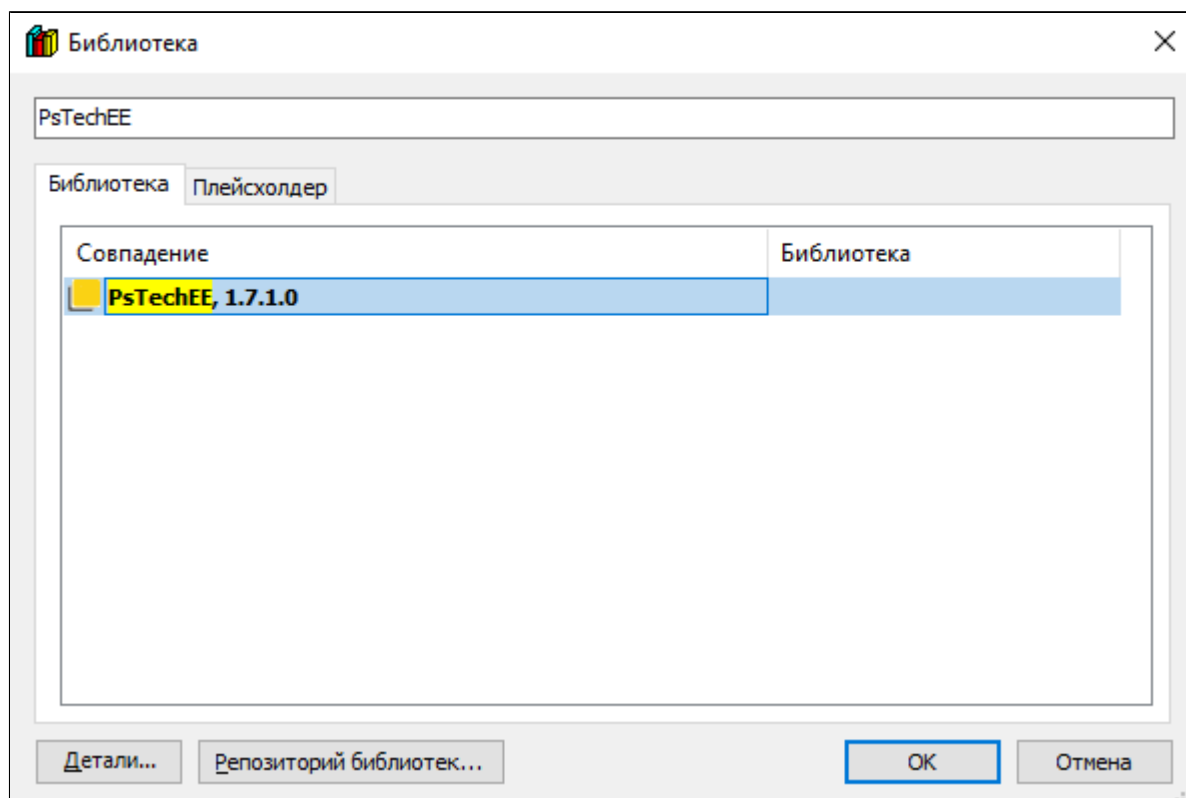
1. Откройте "Менеджер библиотек" двойным кликом левой кнопки мыши по компоненту в дереве устройств.



2. В менеджере библиотек нажмите "Добавить библиотеку" → "Дополнительно...".



3. Введите название библиотеки в строке поиска. Выберите библиотеку и нажмите кнопку "OK".

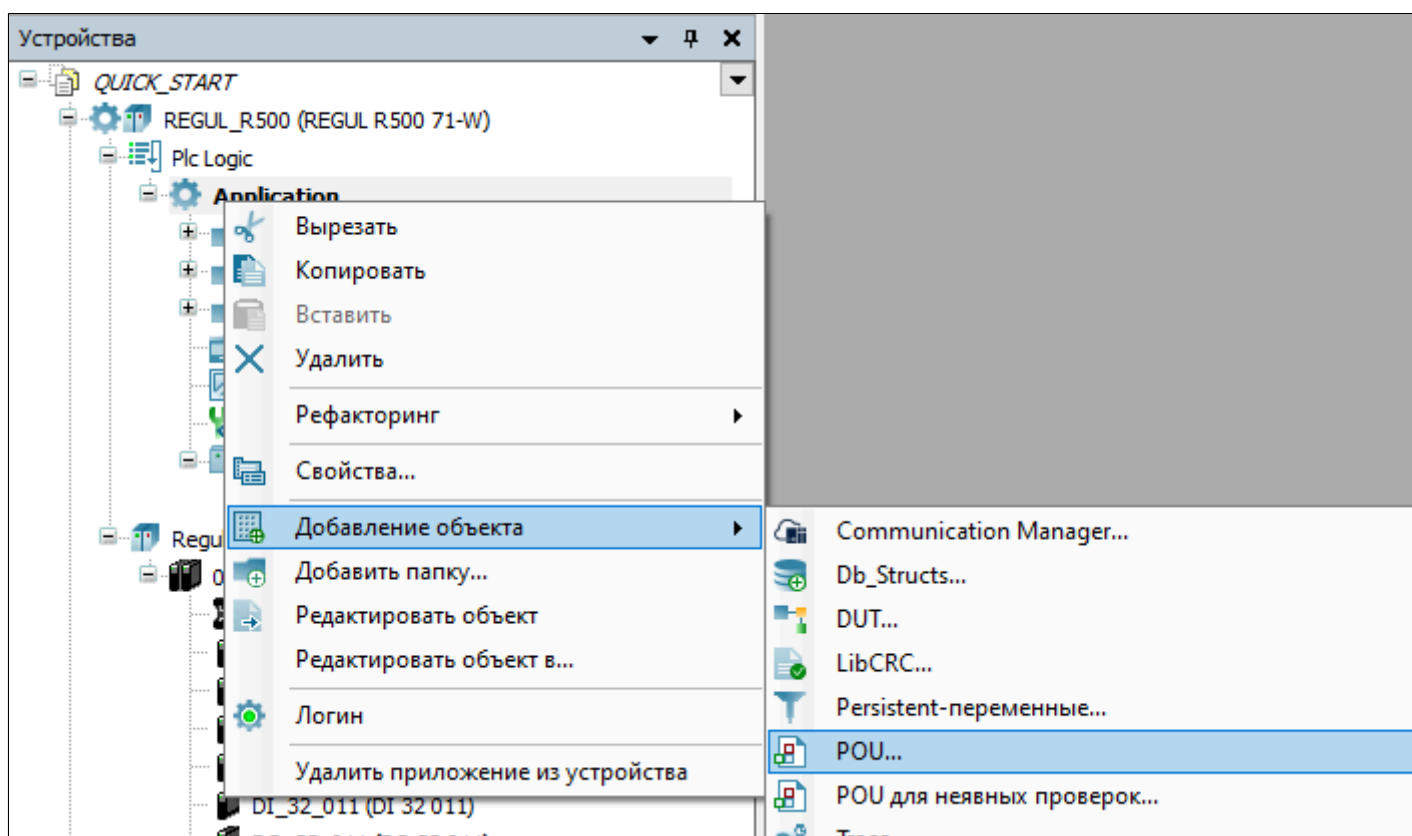


По аналогии с подключением библиотеки "PsTechEE" можно подключать другие библиотеки к проектам.

## 1.1.2.3.2. Создание прикладной программы ПЛК REGUL R500

Рассмотрим процесс разработки прикладного программного обеспечения на примере аналогового датчика.

1. Чтобы создать новую программу (POU) выберите в контекстном меню приложения "Application" пункт "Добавление объекта" → "POU...".



2. Задайте название программы "MAIN", укажите тип "Программа" и язык реализации "CFC постраничный". Нажмите кнопку "Добавить".

Добавить ROU

Создать новый ROU (компонент организации программы)

Имя  
MAIN

Тип

Программа

Функциональный блок

Extends  ...

Implements  ...

Окончательный  Абстрактный

Спецификатор доступа  
...

Язык реализации метода:  
Непрерывные функциональные схемы (CFC) - по

Функция

Тип возвращаемого значения  ...

Язык реализации  
Непрерывные функциональные схемы (CFC) - постраничн

Добавить Отмена

Откроется редактор программы "MAIN":

Устройства

QUICK\_START

REGUL\_R500 (REGUL R500 71-W)

Plc Logic

Application

00\_Глобальные переменные

01\_Системные функции

02\_Диагностика

Менеджер библиотек

MAIN (PRG)

RedundancyOS

Символьная конфигурация

Конфигурация задач

TASK\_MAIN

Regul\_Bus\_OS (Regul Bus OS)

0 - CRATE1 (R500 Крейт)

ST\_02\_012 (ST xx x1x)

PP\_00\_011 (PP 00 xxx)

\* : CU\_00\_071 (CU 00 xxx)

AI\_08\_041 (AI 08 041)

AO\_08\_011 (AO 08 011)

DI\_32\_011 (DI 32 011)

DO\_32\_011 (DO 32 011)

ST\_02\_022 (ST xx x2x)

SAFETY\_REGUL\_R500S (SAFETY REGUL R500S 851)

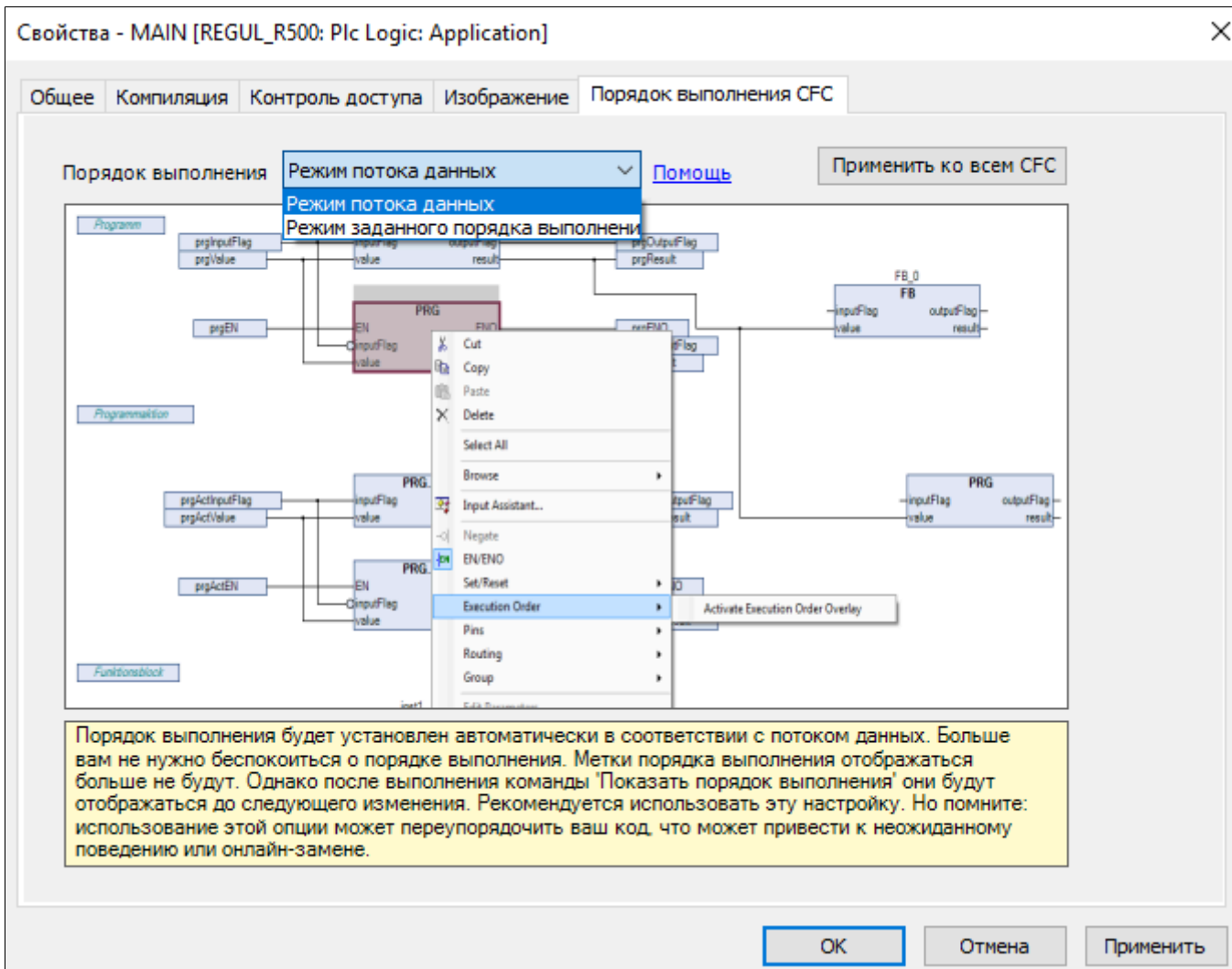
MAIN x

```
1 PROGRAM MAIN
2 VAR
3 END_VAR
4
```

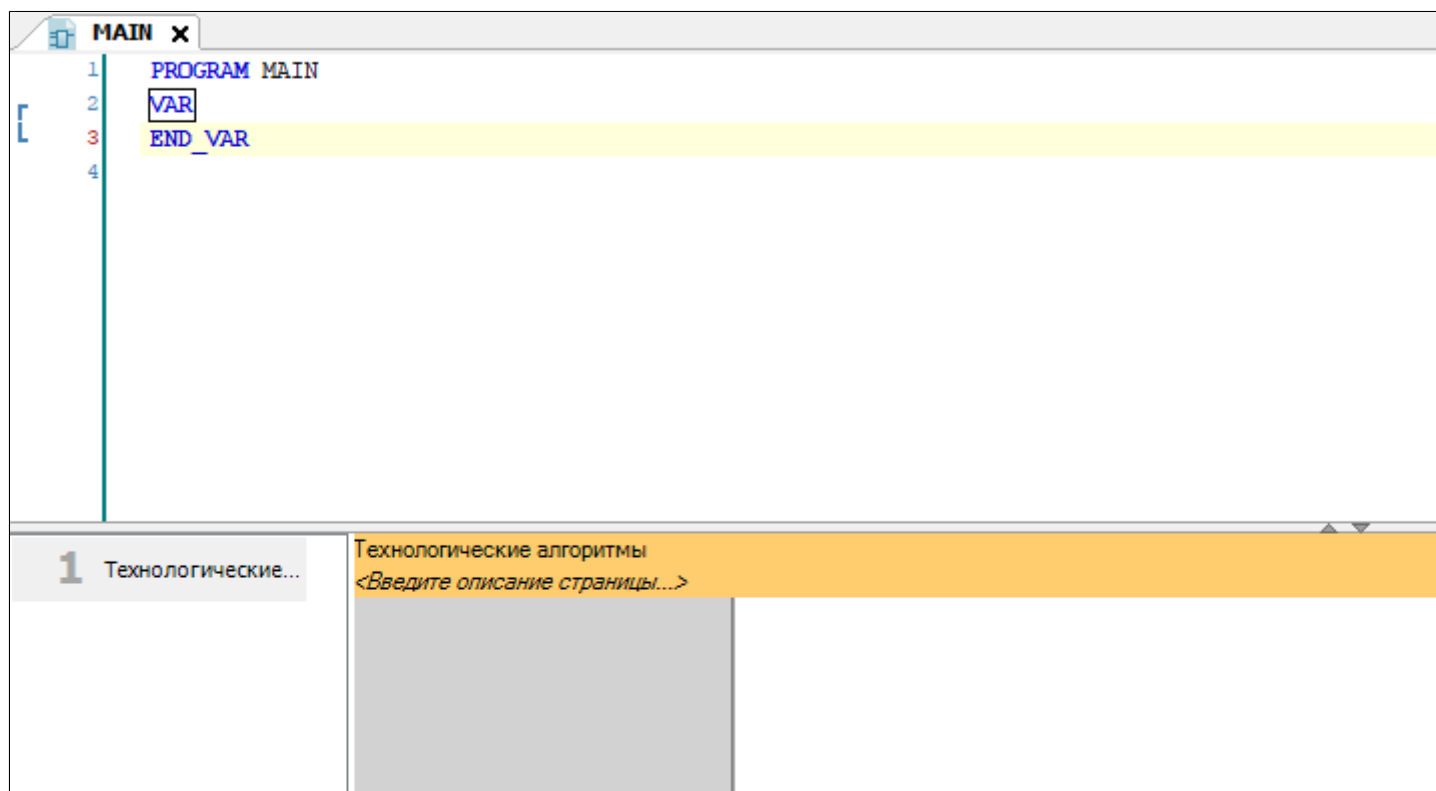
Режим потока данных активирован

Свойства Помощь

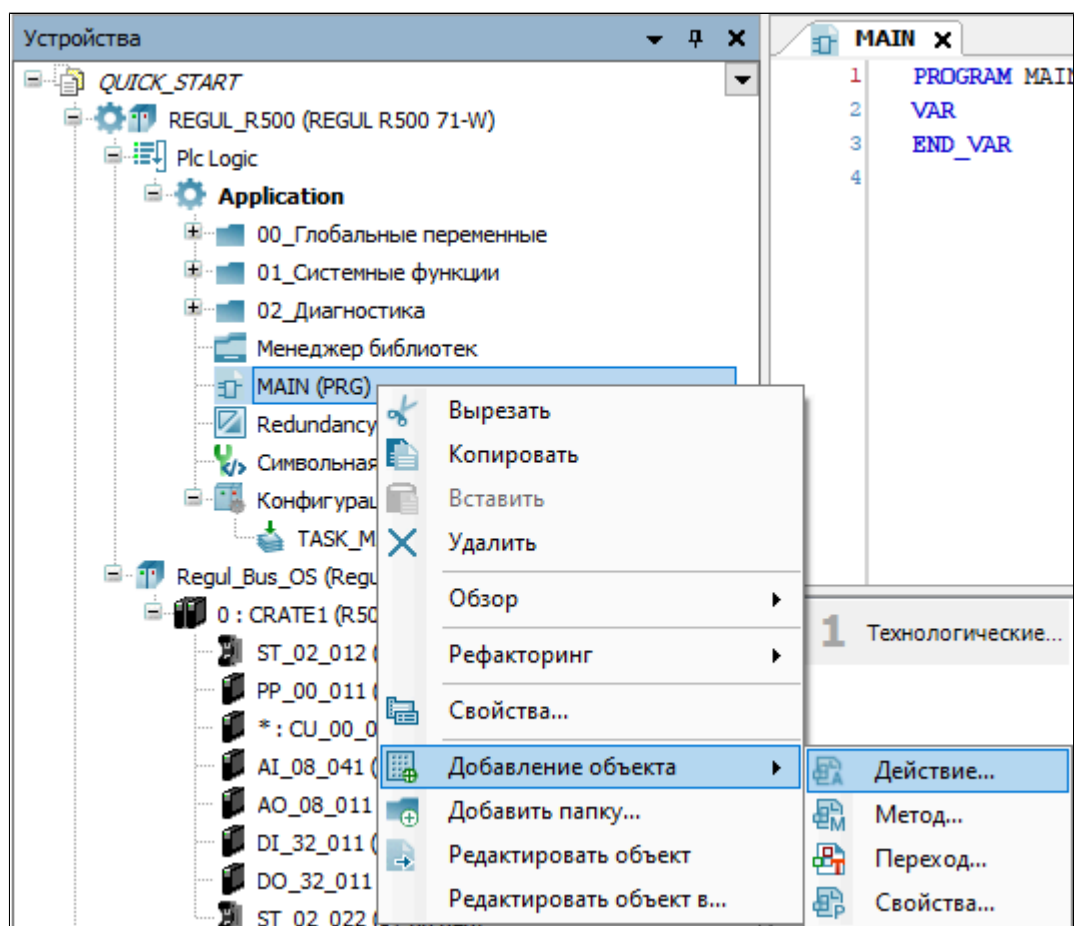
3. Кликните на кнопку "Свойства" и выберите пункт "Режим потока данных" из выпадающего списка, затем нажмите кнопку "Применить".



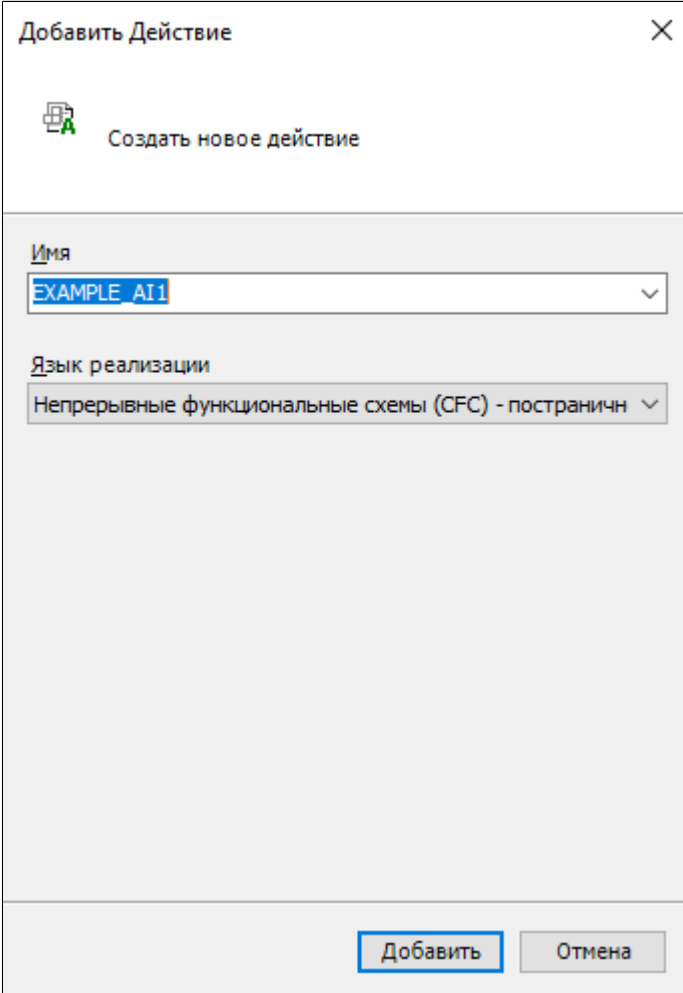
4. В редакторе программы "MAIN" задайте имя страницы – "Технологические алгоритмы".



5. Нажмите правой кнопкой мыши по программе "MAIN" и в контекстном меню выберите "Добавление объекта" → "Действие...".



6. Задайте добавляемому действию имя "EXAMPLE\_AI1" и укажите язык реализации "CFC постраничный".



Добавить Действие

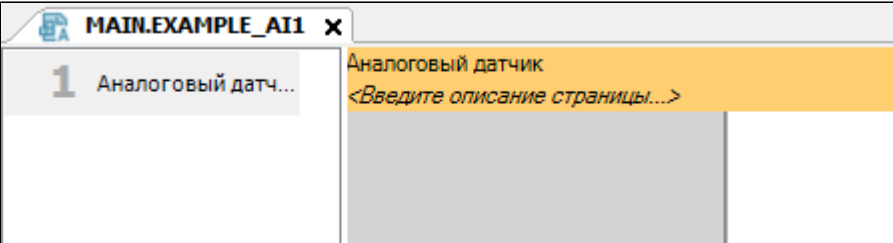
Создать новое действие

Имя  
EXAMPLE\_AI1

Язык реализации  
Непрерывные функциональные схемы (CFC) - постраничн

Добавить Отмена

7. Откроется редактор действия "EXAMPLE\_AI1". Задайте имя страницы "Аналоговый датчик".

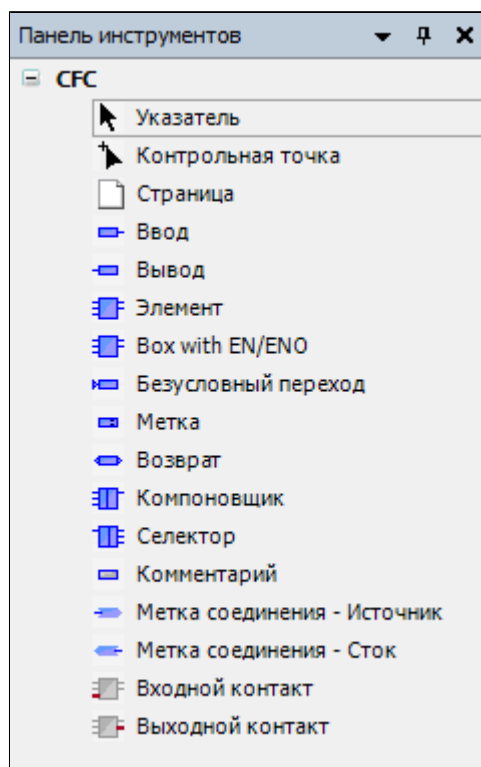


MAIN.EXAMPLE\_AI1 x

1 Аналоговый датч...

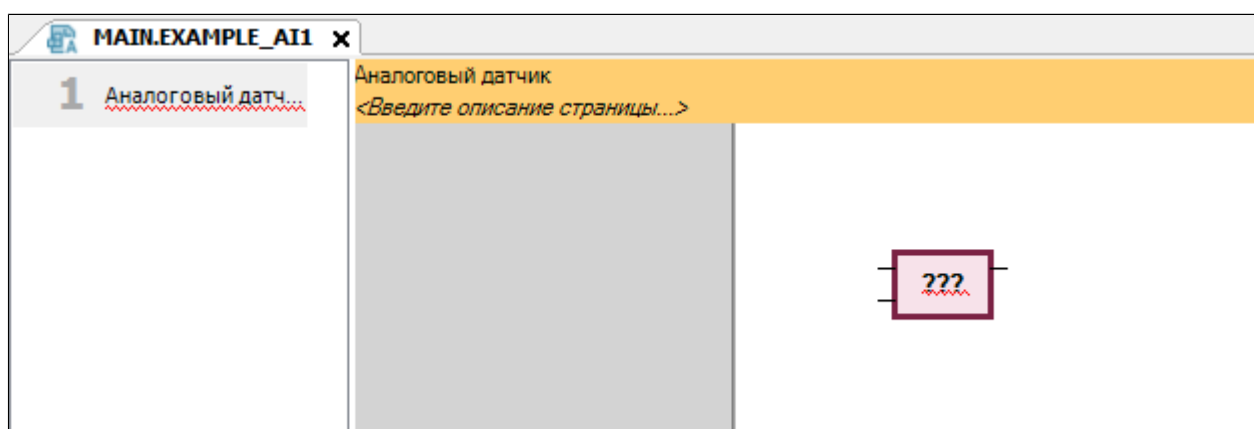
Аналоговый датчик  
<Введите описание страницы...>

8. Добавьте на рабочую область страницы объект "Элемент" из панели инструментов.



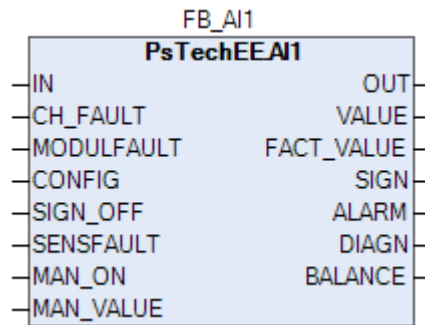
Чтобы добавить объект, перетащите его из панели инструментов в рабочую область.

9. Задайте тип элемента "PsTechEE.AI1" в области "???".





10. Задайте имя экземпляру функционального блока "FB\_AI1" и нажмите клавишу "Enter".



11. В окне "Автообъявление" укажите имя "FB\_AI1" и нажмите кнопку "OK".

Автообъявление

Класс: VAR

Имя: FB\_AI1

Тип: PsTechEE.AI1

Объект: MAIN [Application]

Нач. значение: [ ]

Адрес: [ ]

Флаг:  
 CONSTANT  
 RETAIN  
 PERSISTENT

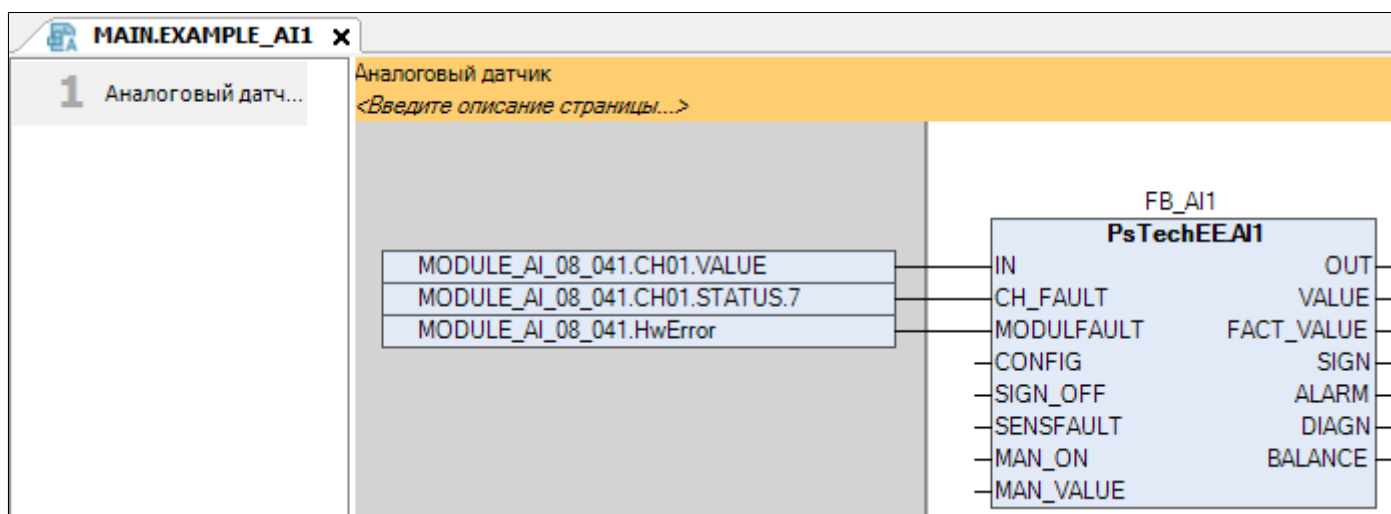
Комментарий: [ ]

OK Отмена

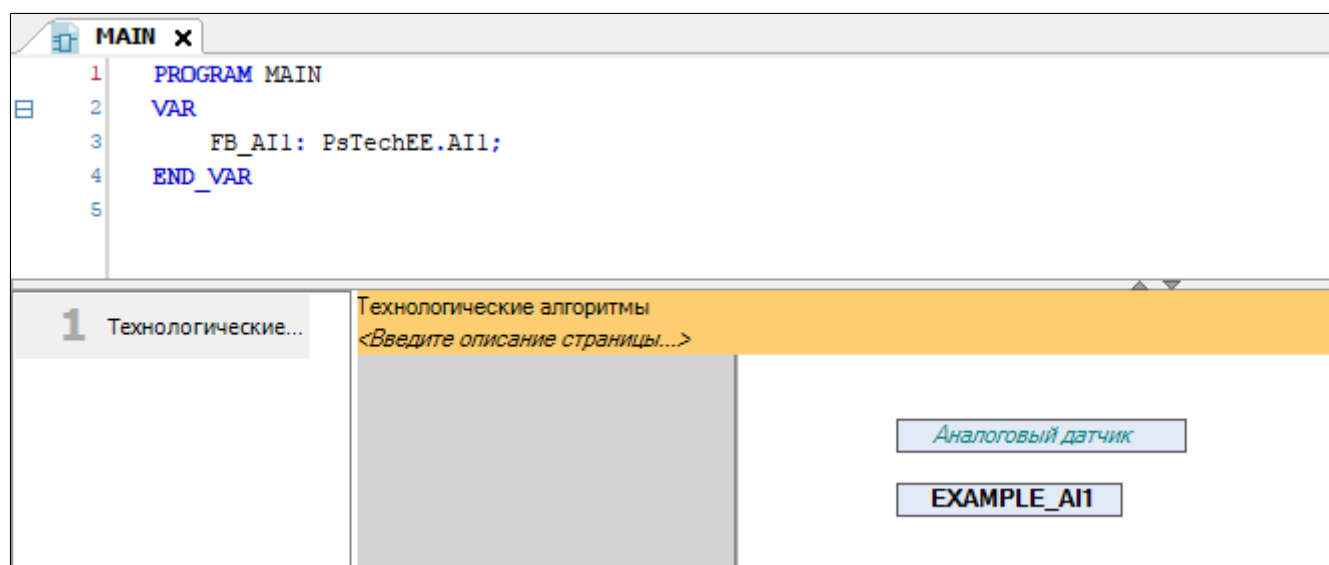
В области определения локальных переменных программы "MAIN" появилось объявление экземпляра функционального блока "FB\_AI1".

```
1 PROGRAM MAIN
2 VAR
3     FB_AI1: PsTechEE.AI1;
4 END_VAR
5
```

12. Привяжите переменные с аналогового модуля ко входам экземпляра функционального блока "PsTechEE.AI1" с помощью элемента "Ввод" из панели инструментов.

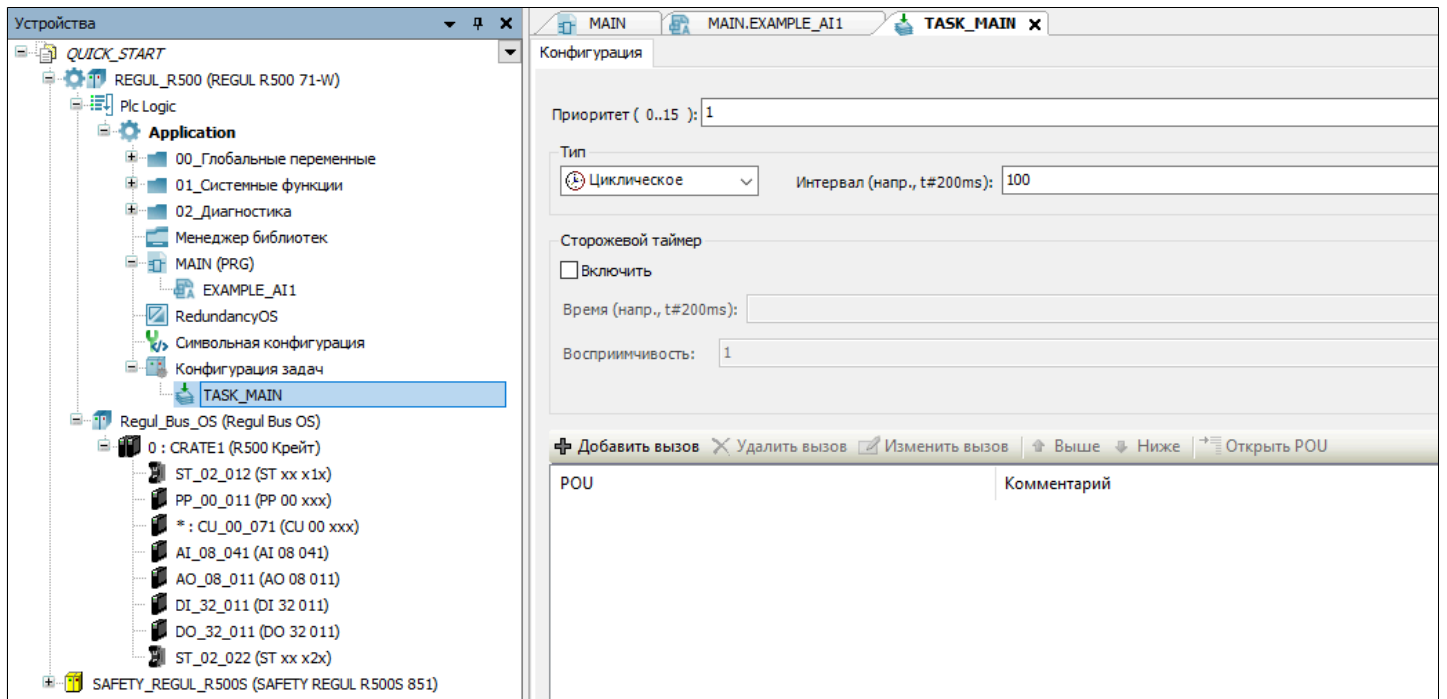


13. Чтобы действие "EXAMPLE\_AI1" вызывалось программой "MAIN" необходимо на странице "Технологические алгоритмы" добавить объект "Элемент" из панели инструментов и задать ему имя действия, которое необходимо вызвать – "EXAMPLE\_AI1". Добавьте объект "Комментарий" из панели инструментов и задайте комментарий "Аналоговый датчик".



14. Чтобы прикладная программа "MAIN" исполнялась в контроллере, необходимо добавить ее вызов в задаче.

Перейдите двойным щелчком мыши в конфигурацию задачи TASK\_MAIN.



15. Нажмите кнопку "Добавить вызов" и в ассистенте ввода выберите программу "MAIN", нажмите кнопку "OK".

Текстовый поиск Категории

Программы

Имя	Тип	Источник
Application	Приложение	
02_Диагностика		
MAIN	PROGRAM	

Структурированный вид

Вставка с аргументами

Вставка с префиксом

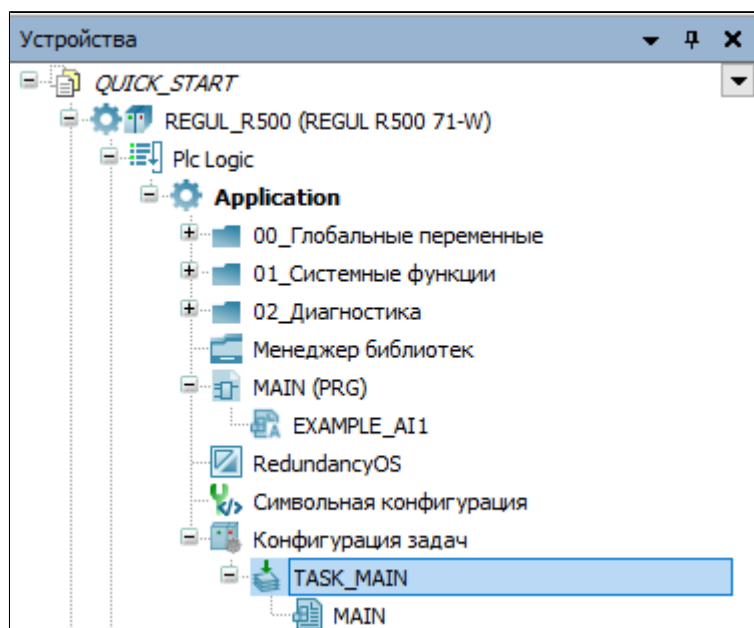
Документация

PROGRAM MAIN

OK

Отмена

При добавлении прикладной программы "MAIN" в задачу, она отобразится в дереве устройств.

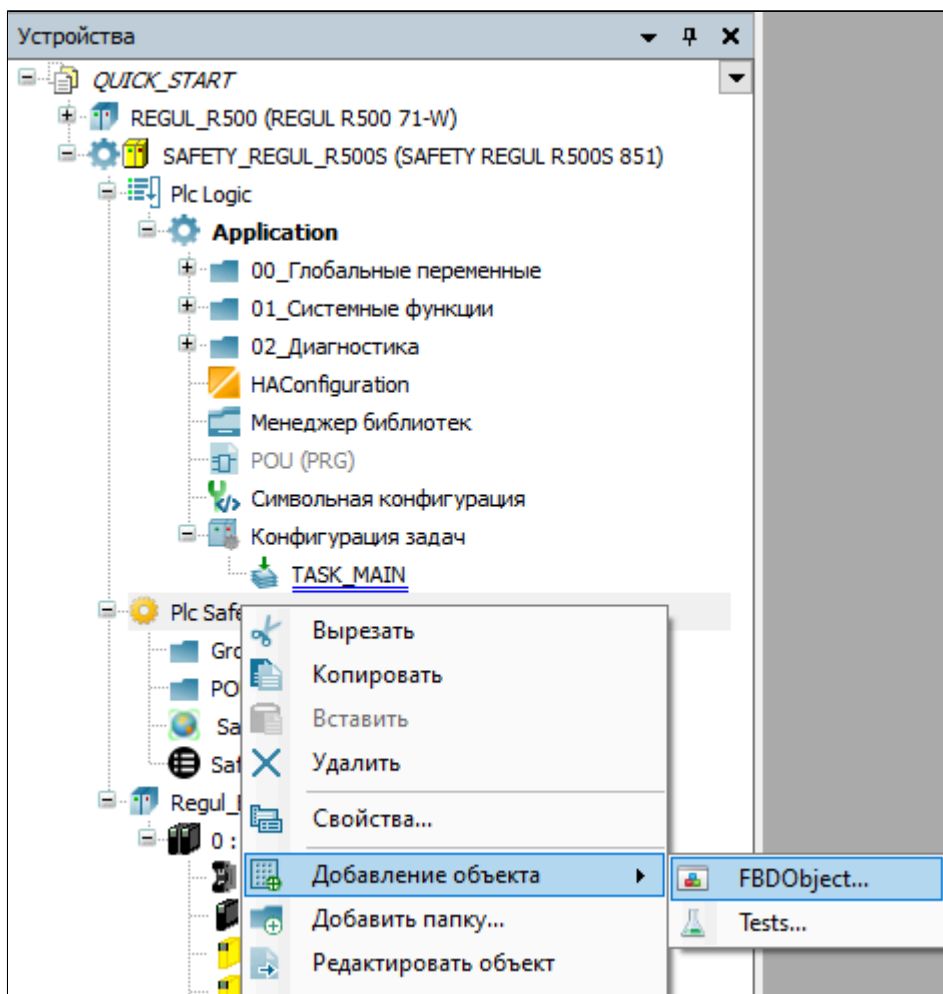


## 1.1.2.3.3. Создание прикладной программы ПЛК SAFETY REGUL R500S

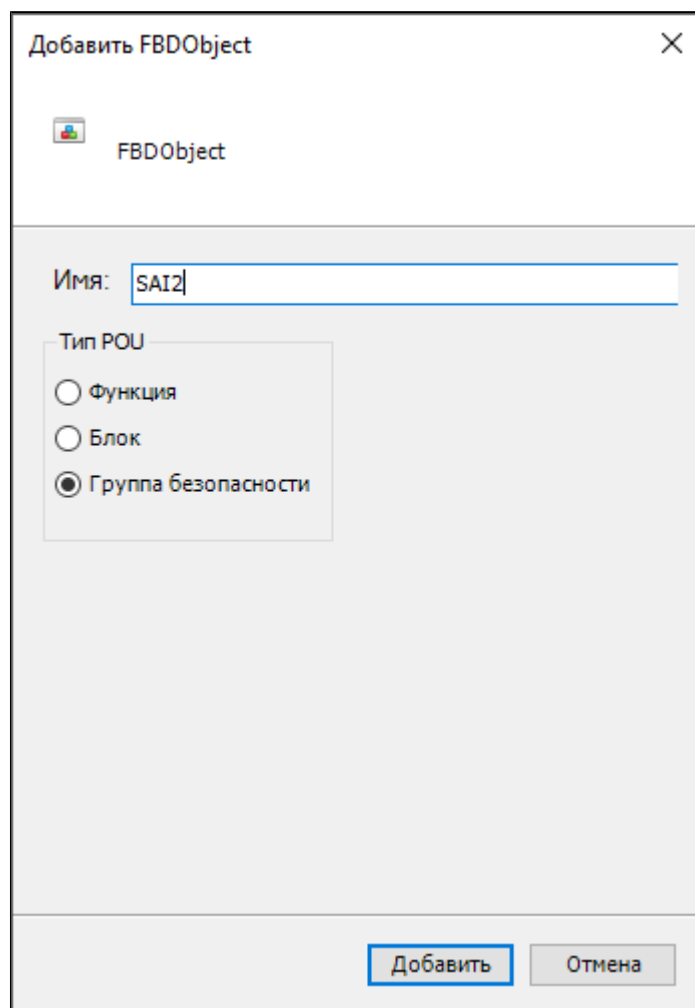
Рассмотрим процесс разработки прикладного программного обеспечения на примере аналогового датчика с двумя порогами сигнализации. Для этого необходимо создать группу безопасности в безопасной части контроллера и программу передачи данных на верхний уровень в небезопасной части.

### Группа безопасности

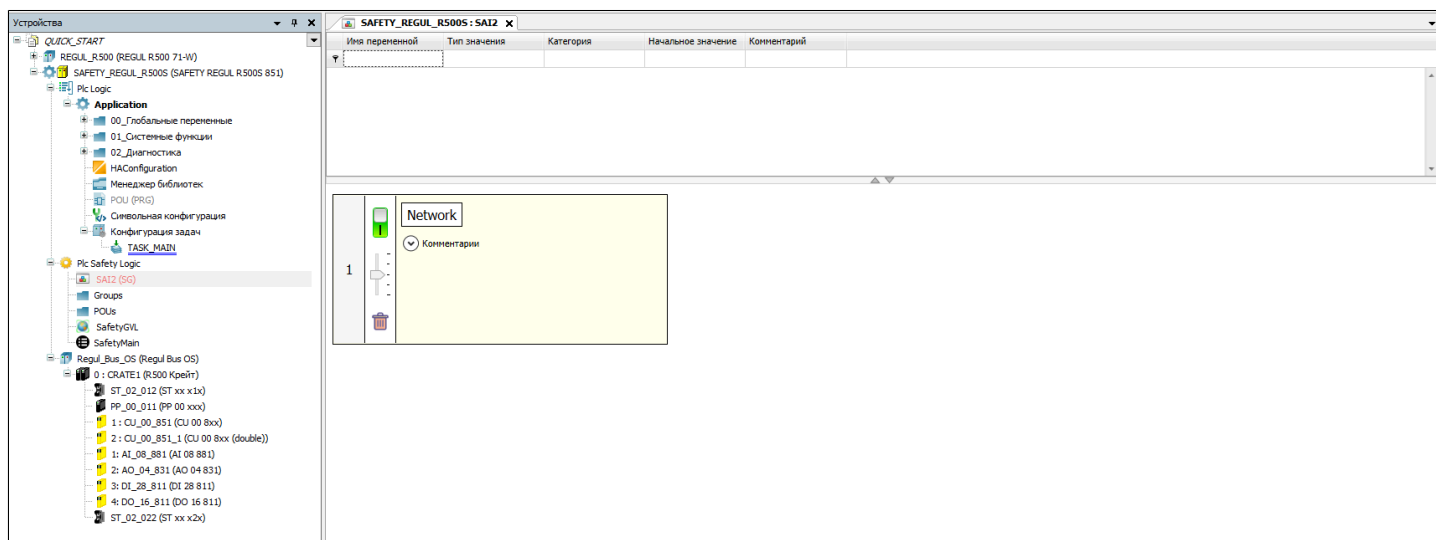
1. Чтобы создать новую группу безопасности выберите в контекстном меню приложения "Plc Safety Logic" пункт "Добавление объекта" → "FBXObject...".



2. Задайте название программы "SAI2", укажите тип "Группа безопасности".  
Нажмите кнопку "Добавить".

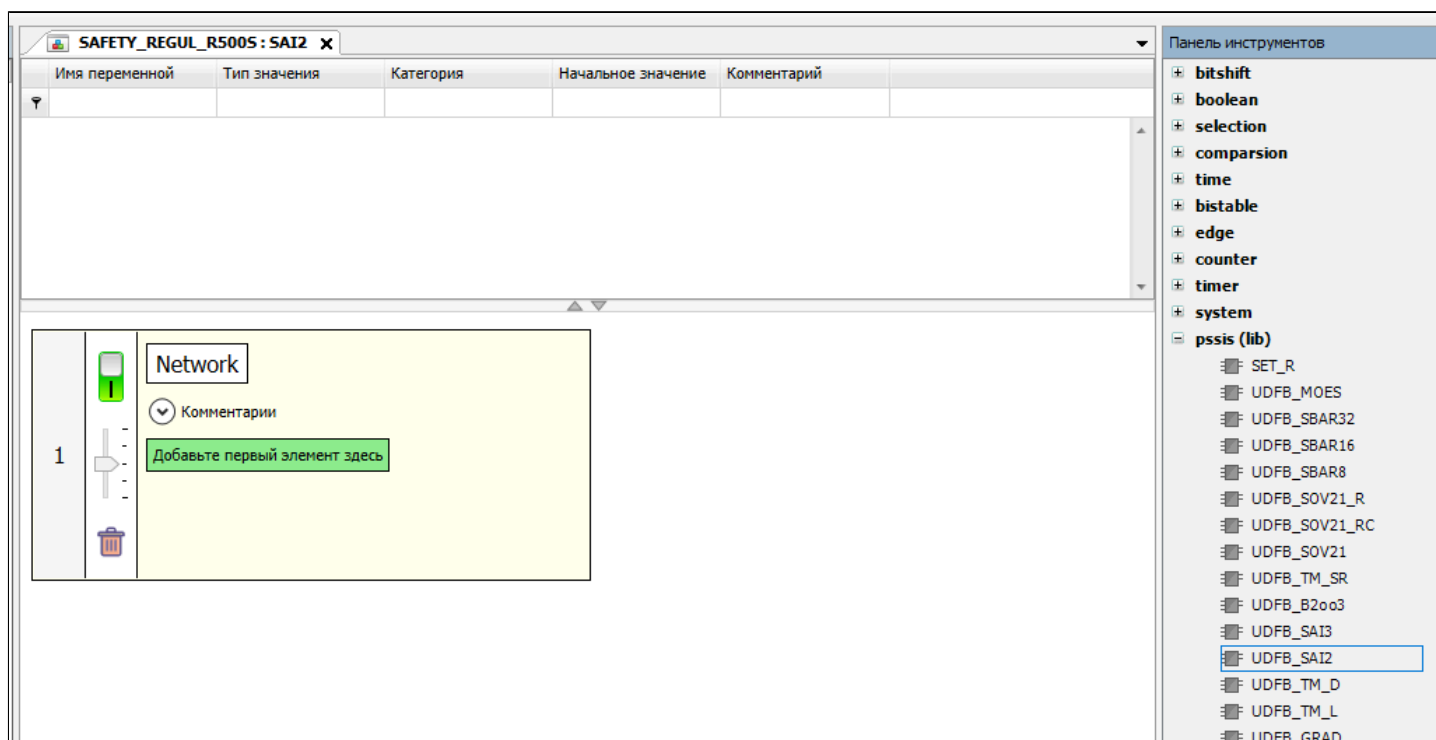


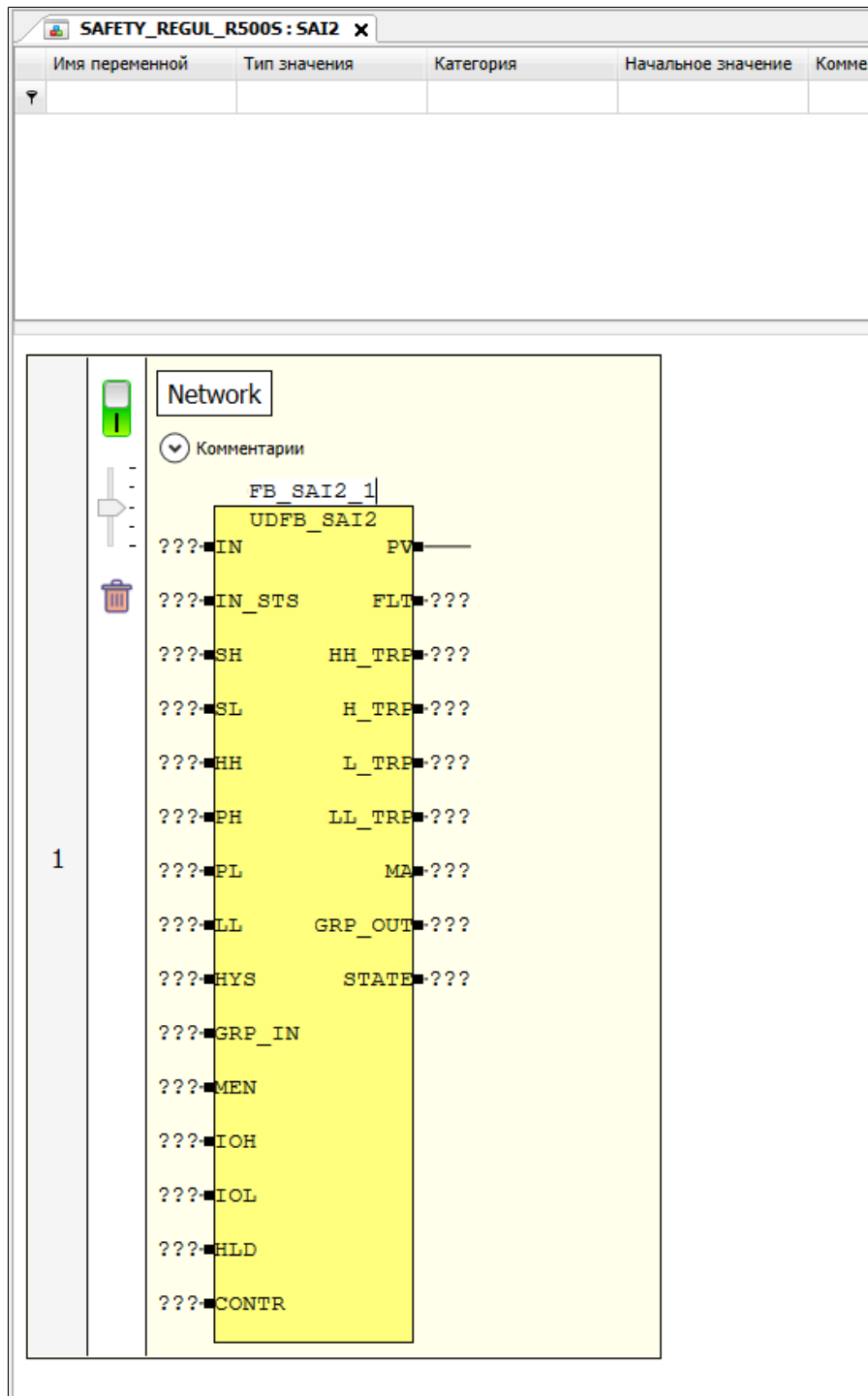
Откроется редактор программы "SAI2":



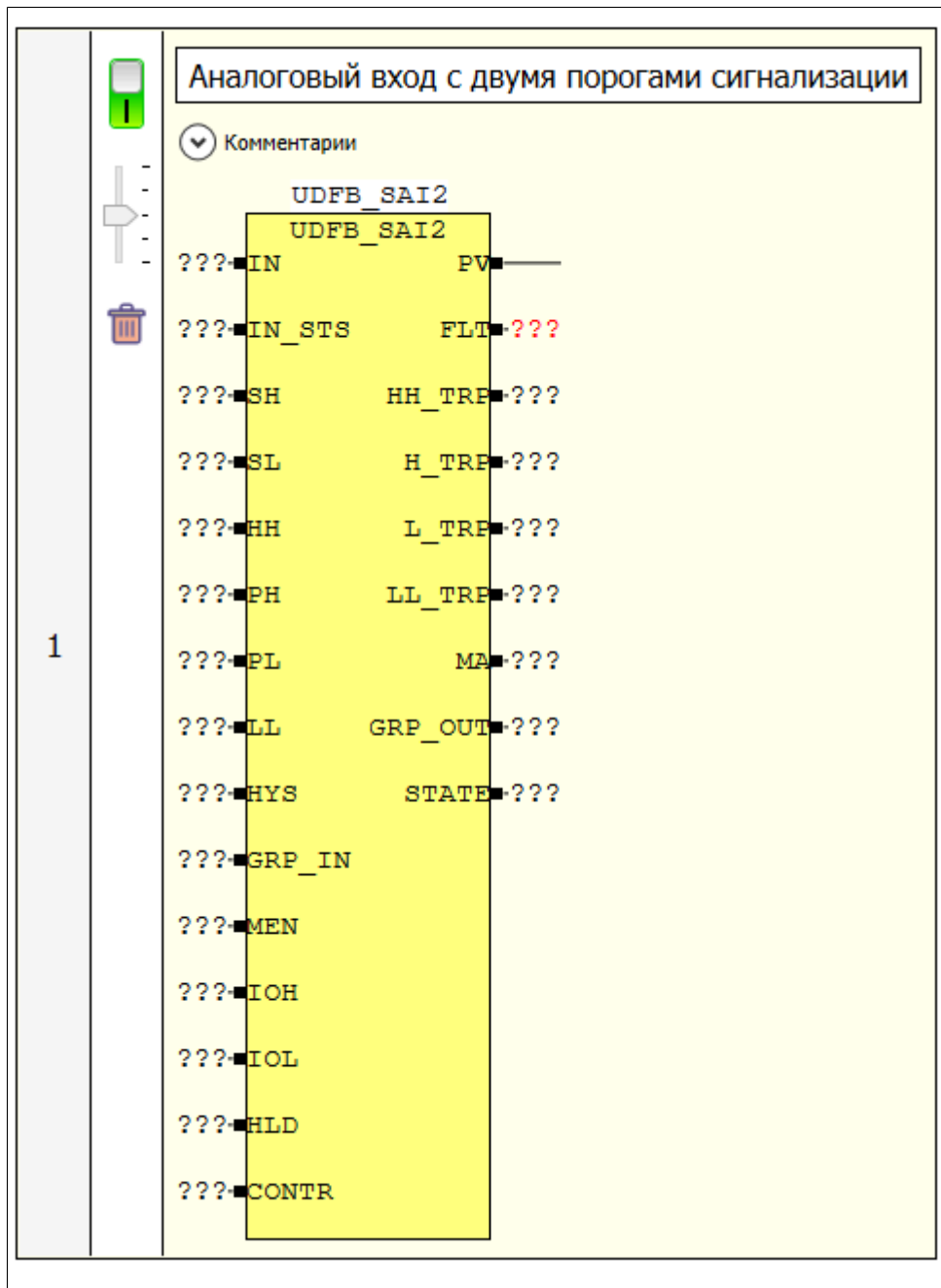


3. На "Панели инструментов" раскройте раздел библиотеки "pssis(lib)" и перетащите в "Network" блок "UDFB\_SAI2".





4. В редакторе функциональной группы безопасности задайте имя цепи – "Аналоговый вход с двумя порогами сигнализации" и переименуйте блок в "UDFB\_SAI2"



5. Двойным кликом откройте редактор глобальных переменных "SafetyGVL" и добавьте следующие переменные:

Имя переменной	Тип значения	Категория	Начальное значение	NSE
FB_UDFB_SAI2_PV	REAL	VAR_GLOBAL	0	R
FB_UDFB_SAI2_STATE	DINT	VAR_GLOBAL	0	R
FB_UDFB_SAI2_PH	REAL	VAR_GLOBAL	0	W
FB_UDFB_SAI2_PL	REAL	VAR_GLOBAL	0	W

FB_UDFB_SAI2_HH	REAL	VAR_GLOBAL	0	W
FB_UDFB_SAI2_SH	REAL	VAR_GLOBAL	0	W
FB_UDFB_SAI2_SL	REAL	VAR_GLOBAL	0	W
FB_UDFB_SAI2_LL	REAL	VAR_GLOBAL	0	W
FB_UDFB_SAI2_HYS	REAL	VAR_GLOBAL	0	W
FB_UDFB_SAI2_CONTR	DINT	VAR_GLOBAL	0	W
GRP_MOS	BOOL	VAR_GLOBAL	FALSE	-

SAFETY\_REGUL\_R5005: SafetyGVL x

Глобальные переменные FSC переменные

Имя переменной	Тип значения	Категория	Начальное значение	Комментарий	FSC	NSE
1 safety_lock	BOOL	VAR_GLOBAL	False		<input type="checkbox"/>	-
2 out_unlock	BOOL	VAR_GLOBAL	True		<input type="checkbox"/>	-
3 Start_SYNC	BOOL	VAR_GLOBAL	True		<input type="checkbox"/>	-
4 FB_UDFB_SAI2_PV	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	R
5 FB_UDFB_SAI2_STATE	DINT	VAR_GLOBAL	0		<input type="checkbox"/>	R
6 FB_UDFB_SAI2_PH	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
7 FB_UDFB_SAI2_PL	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
8 FB_UDFB_SAI2_HH	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
9 FB_UDFB_SAI2_SH	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
10 FB_UDFB_SAI2_SL	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
11 FB_UDFB_SAI2_LL	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
12 FB_UDFB_SAI2_HYS	REAL	VAR_GLOBAL	0		<input type="checkbox"/>	W
13 FB_UDFB_SAI2_CONTR	DINT	VAR_GLOBAL	0		<input type="checkbox"/>	W
14 GRP_MOS	BOOL	VAR_GLOBAL	FALSE		<input type="checkbox"/>	-

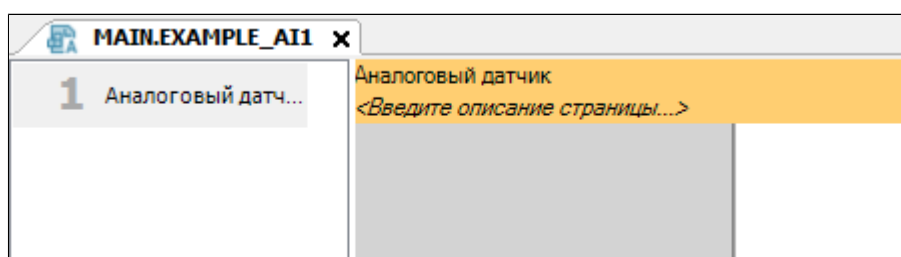
6. В группе безопасности "SAI2" в редакторе переменных добавьте следующие переменные:

Имя переменной	Тип значения	Категория	Начальное значение
UDFB_SAI2_IN	REAL	VAR_INPUT	-
UDFB_SAI2_IN_STS	BOOL	VAR_INPUT	-
UDFB_SAI2_MEN	BOOL	VAR_INPUT	-
UDFB_SAI2_HLD	BOOL	VAR_INPUT	-
UDFB_SAI2_PV	REAL	VAR_OUTPUT	-
UDFB_SAI2_HH_TRP	BOOL	VAR_OUTPUT	-
UDFB_SAI2_H_TRP	BOOL	VAR_OUTPUT	-

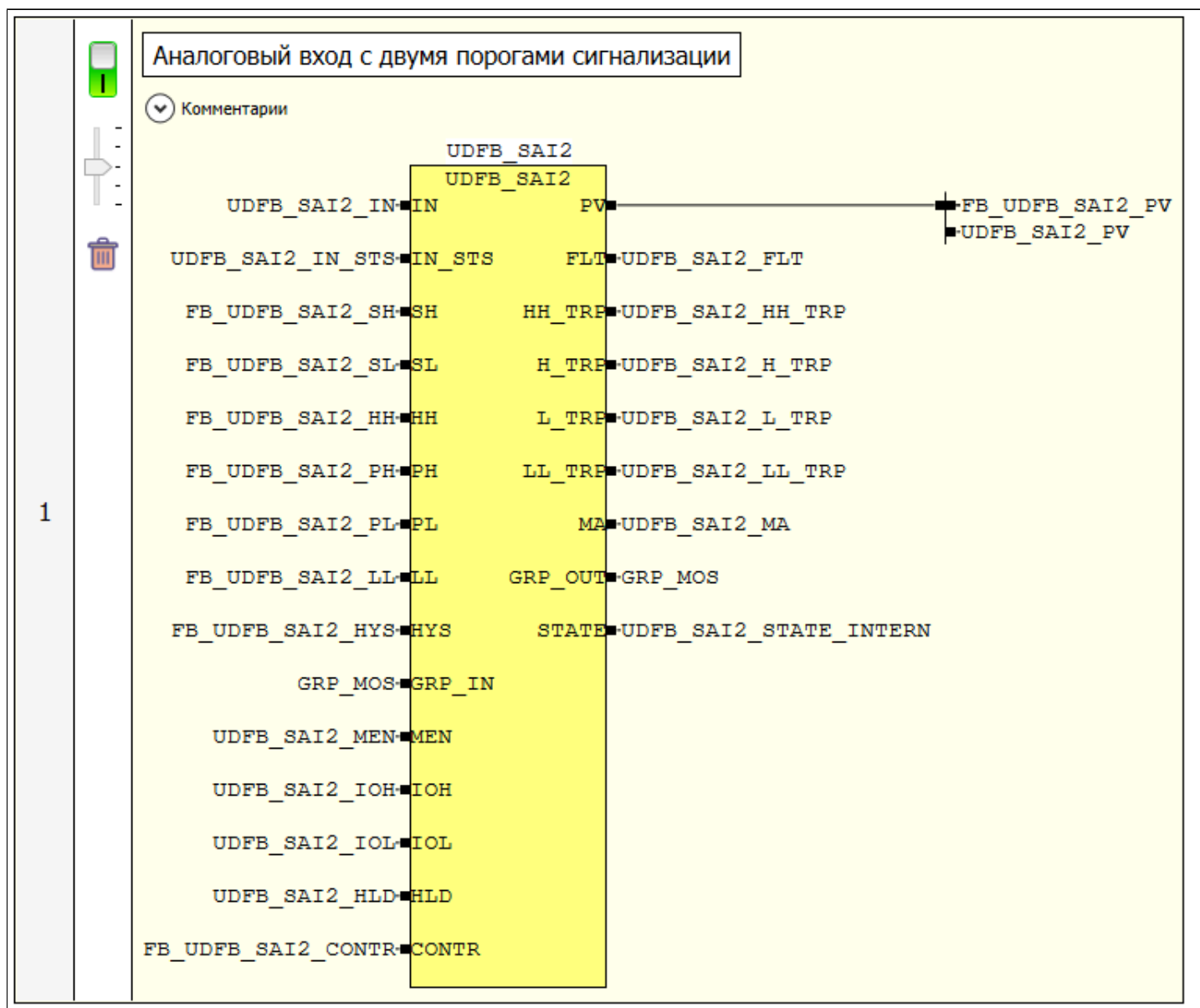
UDFB_SAI2_L_TRP	BOOL	VAR_OUTPUT	-
UDFB_SAI2_LL_TRP	BOOL	VAR_OUTPUT	-
UDFB_SAI2_FLT_TRP	BOOL	VAR_OUTPUT	-
UDFB_SAI2_MA	BOOL	VAR_OUTPUT	-
FB_UDFB_SAI2_PV	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_STATE	DINT	VAR_EXTERNAL	-
FB_UDFB_SAI2_PH	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_PL	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_HH	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_SH	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_SL	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_LL	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_HYS	REAL	VAR_EXTERNAL	-
FB_UDFB_SAI2_CONTR	DINT	VAR_EXTERNAL	-
GRP_MOS	BOOL	VAR_EXTERNAL	-
UDFB_SAI2_IOH	REAL	VAR_CONST	20.0
UDFB_SAI2_IOL	REAL	VAR_CONST	4.0
UDFB_SAI2_STATE_ITERN	DINT	VAR	0

SAFETY_REGUL_R5005 : SAI2 x					
	Имя переменной	Тип значения	Категория	Начальное значение	Комментарий
1	UDFB_SAI2_IN	REAL	VAR_INPUT	0	
2	UDFB_SAI2_IN_STS	BOOL	VAR_INPUT	False	
3	UDFB_SAI2_MEN	BOOL	VAR_INPUT	False	
4	UDFB_SAI2_HLD	BOOL	VAR_INPUT	False	
5	UDFB_SAI2_PV	REAL	VAR_OUTPUT	0	
6	UDFB_SAI2_HH_TRP	BOOL	VAR_OUTPUT	False	
7	UDFB_SAI2_H_TRP	BOOL	VAR_OUTPUT	False	
8	UDFB_SAI2_L_TRP	BOOL	VAR_OUTPUT	False	
9	UDFB_SAI2_LL_TRP	BOOL	VAR_OUTPUT	False	
10	UDFB_SAI2_FLT	BOOL	VAR_OUTPUT	False	
11	UDFB_SAI2_MA	BOOL	VAR_OUTPUT	False	
12	FB_UDFB_SAI2_CONTR	DINT	VAR_EXTERNAL	0	
13	FB_UDFB_SAI2_SH	REAL	VAR_EXTERNAL	0	
14	FB_UDFB_SAI2_SL	REAL	VAR_EXTERNAL	0	
15	FB_UDFB_SAI2_HH	REAL	VAR_EXTERNAL	0	
16	FB_UDFB_SAI2_PH	REAL	VAR_EXTERNAL	0	
17	FB_UDFB_SAI2_PL	REAL	VAR_EXTERNAL	0	
18	FB_UDFB_SAI2_LL	REAL	VAR_EXTERNAL	0	
19	FB_UDFB_SAI2_HYS	REAL	VAR_EXTERNAL	0	
20	FB_UDFB_SAI2_PV	REAL	VAR_EXTERNAL	0	
21	FB_UDFB_SAI2_STATE	DINT	VAR_EXTERNAL	0	
22	GRP_MOS	BOOL	VAR_EXTERNAL	False	
23	UDFB_SAI2_IOH	REAL	VAR_CONST	20,0	
24	UDFB_SAI2_IOL	REAL	VAR_CONST	4,0	
25	UDFB_SAI2_STATE_INTERN	DINT	VAR	0	

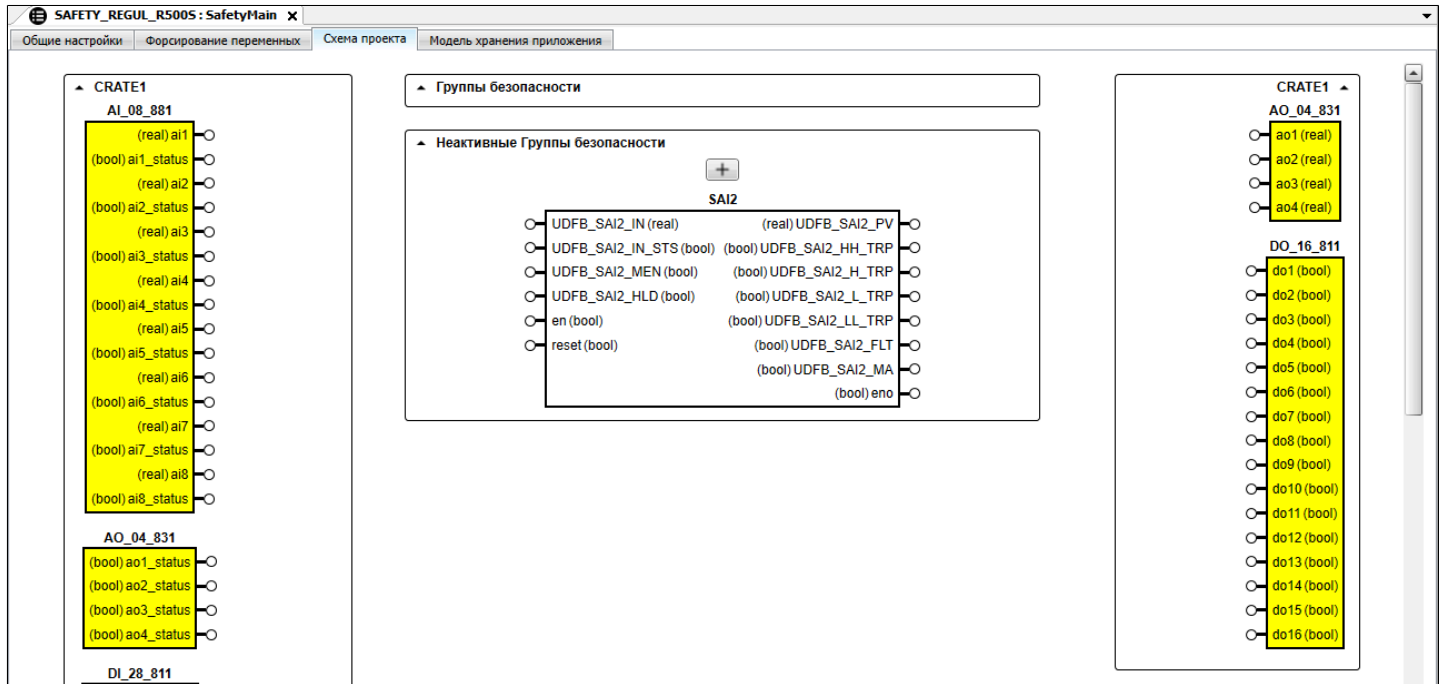
7. Откроется редактор действия "EXAMPLE\_AI1". Задайте имя страницы "Аналоговый датчик".




8. Задайте переменные в областях "???".

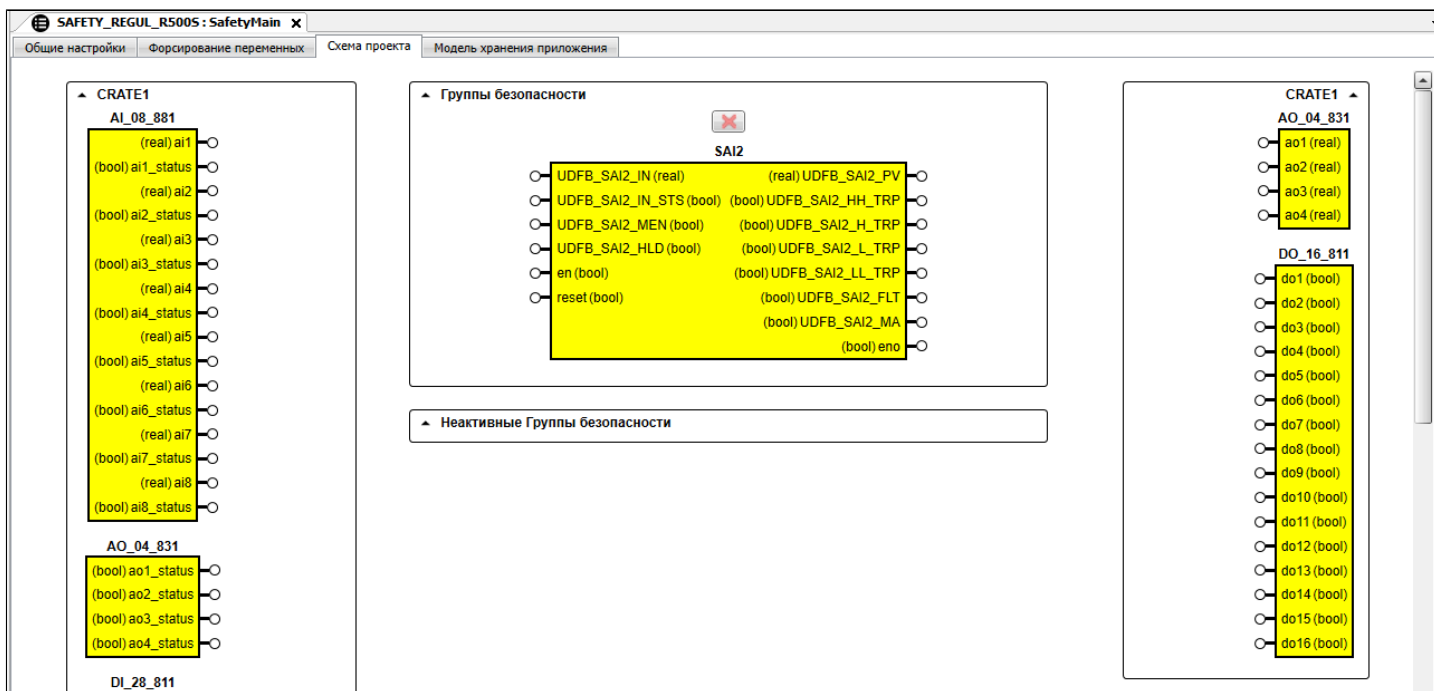




9. Откройте "SafetyMain" и перейдите во вкладку "Схема проекта".



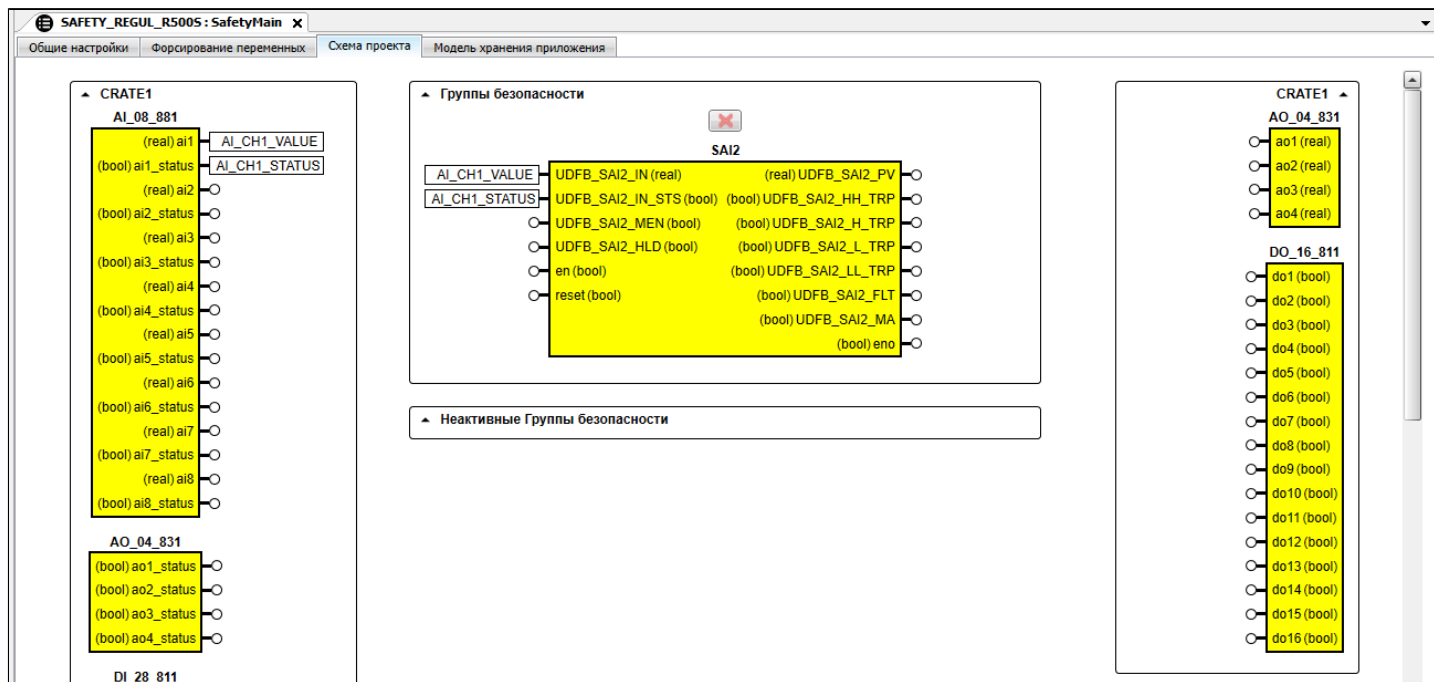


10. Нажмите на  , чтобы сделать созданную группу безопасности активной.



 Для того, чтобы сделать данную группу безопасности неактивной нажмите кнопку  .

11. Создайте информационные связи между модулями ввода и группой безопасности.



Создание группы безопасности в безопасной части контроллера завершена.

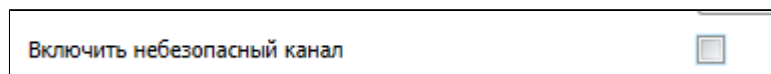
## Передача данных на верхний уровень

Для передачи данных на верхний уровень необходимо активировать небезопасный канал связи.

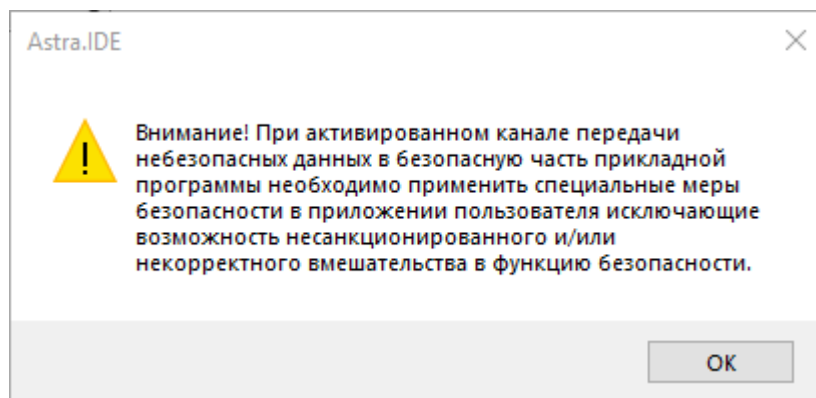
1. Откройте "SafetyGVL" и создайте переменную активации небезопасного канала "SafetyChannel" типа "BOOL" и значение "True".

14	GRP_MOS	BOOL	VAR_GLOBAL	FALSE	
15	SafetyChannel	BOOL	VAR_GLOBAL	True	

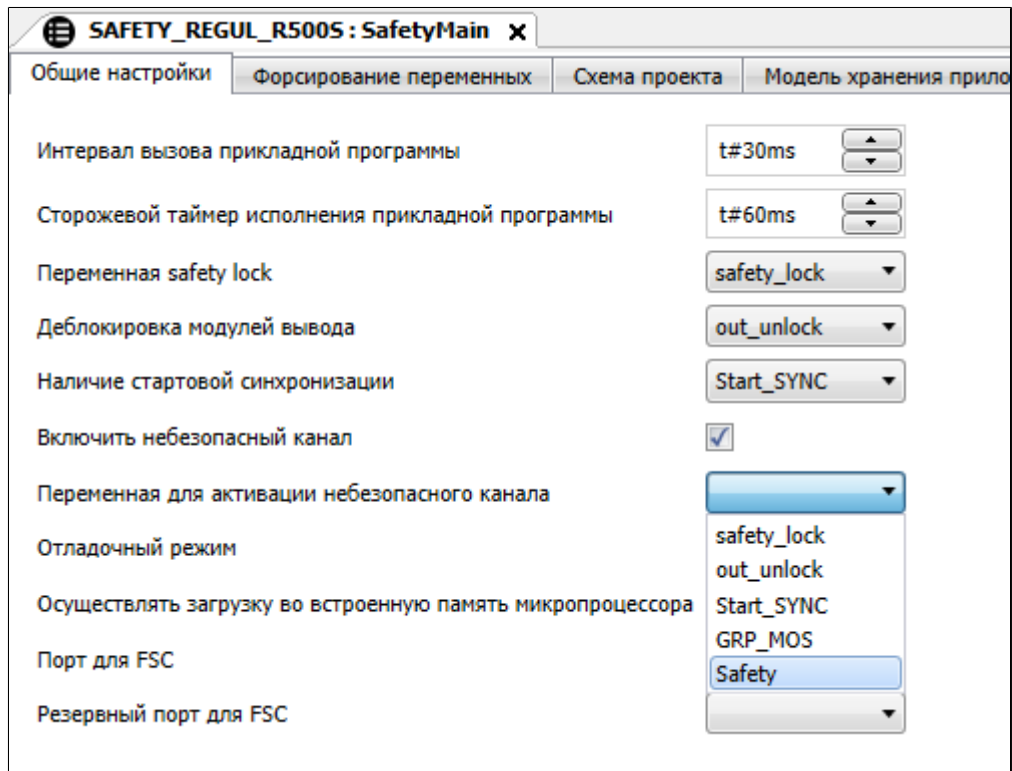
2. Перейдите во вкладку "Общие настройки" в "SafetyMain" и выберите пункт "Включить небезопасный канал".



3. Откроется окно с предупреждением, нажмите "OK".



4. В выпадающем списке пункта "Переменная для активации небезопасного канала" выберите созданную переменную "SafetyChannel".



5. Перейдите в группу безопасности "SAI2" и добавьте цепь, перетащив её из "Панели инструментов".

Имя переменной	Тип значения	Категория	Начальное значение	Комментарий	
3	UDFB_SAI2_MEN	BOOL	VAR_INPUT	False	
4	UDFB_SAI2_HLD	BOOL	VAR_INPUT	False	
5	UDFB_SAI2_PV	REAL	VAR_OUTPUT	0	
6	UDFB_SAI2_HH_TRP	BOOL	VAR_OUTPUT	False	
7	UDFB_SAI2_H_TRP	BOOL	VAR_OUTPUT	False	
8	UDFB_SAI2_L_TRP	BOOL	VAR_OUTPUT	False	

1

UDFB\_SAI2\_IN\_STS:IN\_STS FLT:UDFB\_SAI2\_FLT

FB\_UDFB\_SAI2\_SH:SH HH\_TRP:UDFB\_SAI2\_HH\_TRP

FB\_UDFB\_SAI2\_SL:SL H\_TRP:UDFB\_SAI2\_H\_TRP

FB\_UDFB\_SAI2\_HH:HH L\_TRP:UDFB\_SAI2\_L\_TRP

FB\_UDFB\_SAI2\_PH:PH LL\_TRP:UDFB\_SAI2\_LL\_TRP

FB\_UDFB\_SAI2\_PL:PL MA:UDFB\_SAI2\_MA

FB\_UDFB\_SAI2\_LL:LL GRP\_OUT:GRP\_MOS

FB\_UDFB\_SAI2\_HYS:HYS STATE:UDFB\_SAI2\_STATE\_INTERN

GRP\_MOS:GRP\_IN

UDFB\_SAI2\_MEN:MEN

UDFB\_SAI2\_IOH:IOH

UDFB\_SAI2\_IOL:IOL

UDFB\_SAI2\_HLD:HLD

FB\_UDFB\_SAI2\_CONTR:CONTR

2

Network

Комментарии

6. Задайте имя цепи - "Состояние для верхнего уровня", и создайте следующее присвоение:

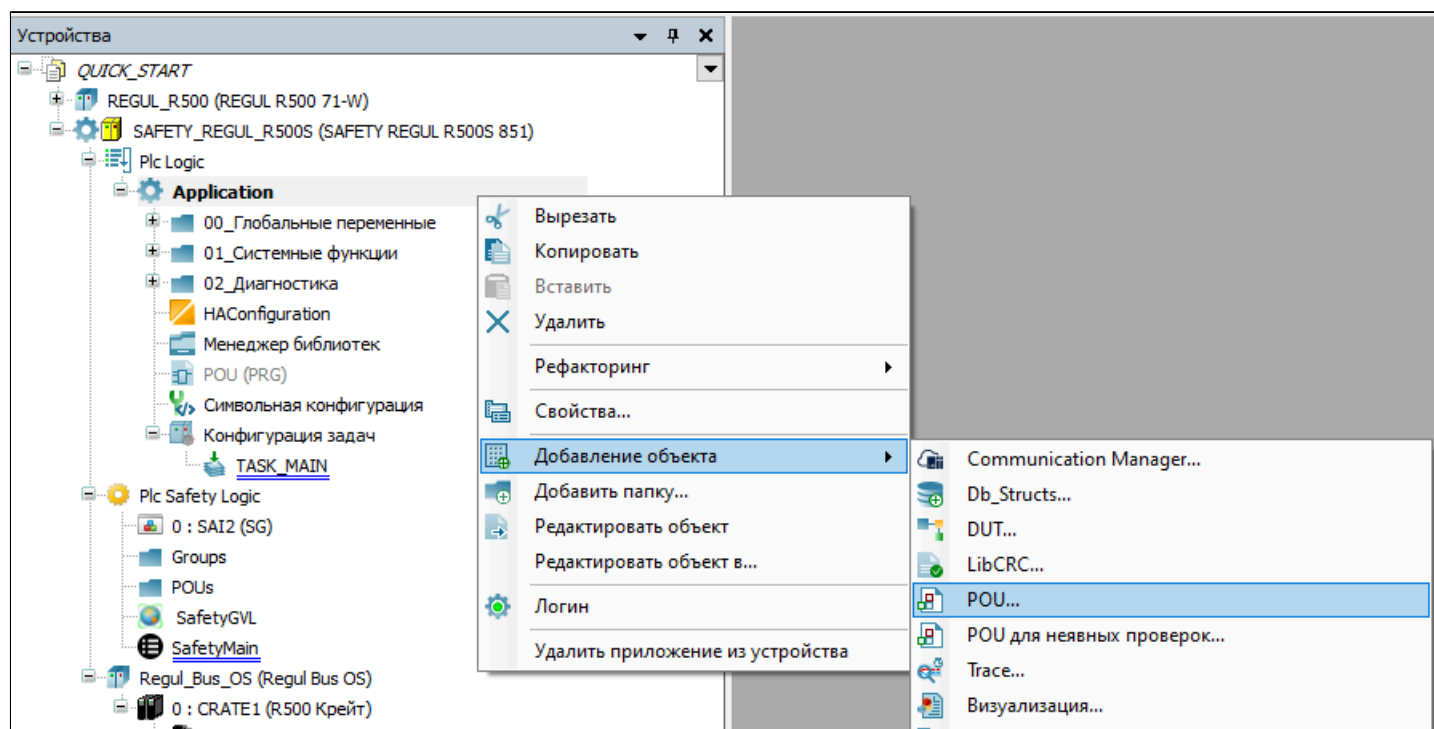
2

Состояние для верхнего уровня

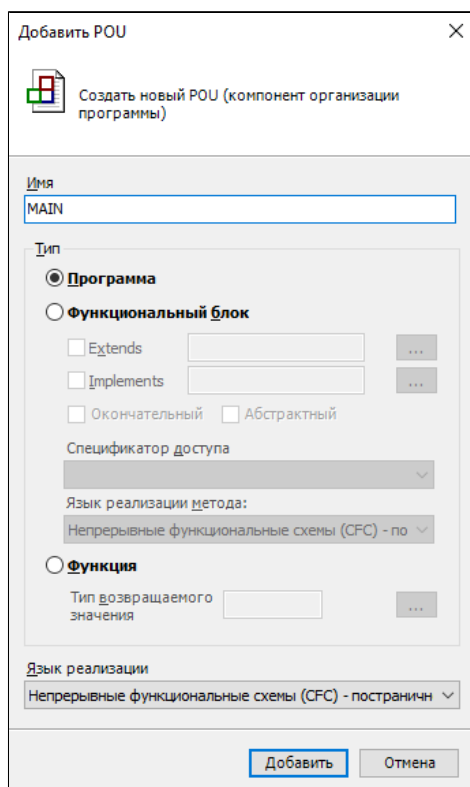
Комментарии

UDFB\_SAI2\_STATE\_INTERN:FB\_UDFB\_SAI2\_STATE

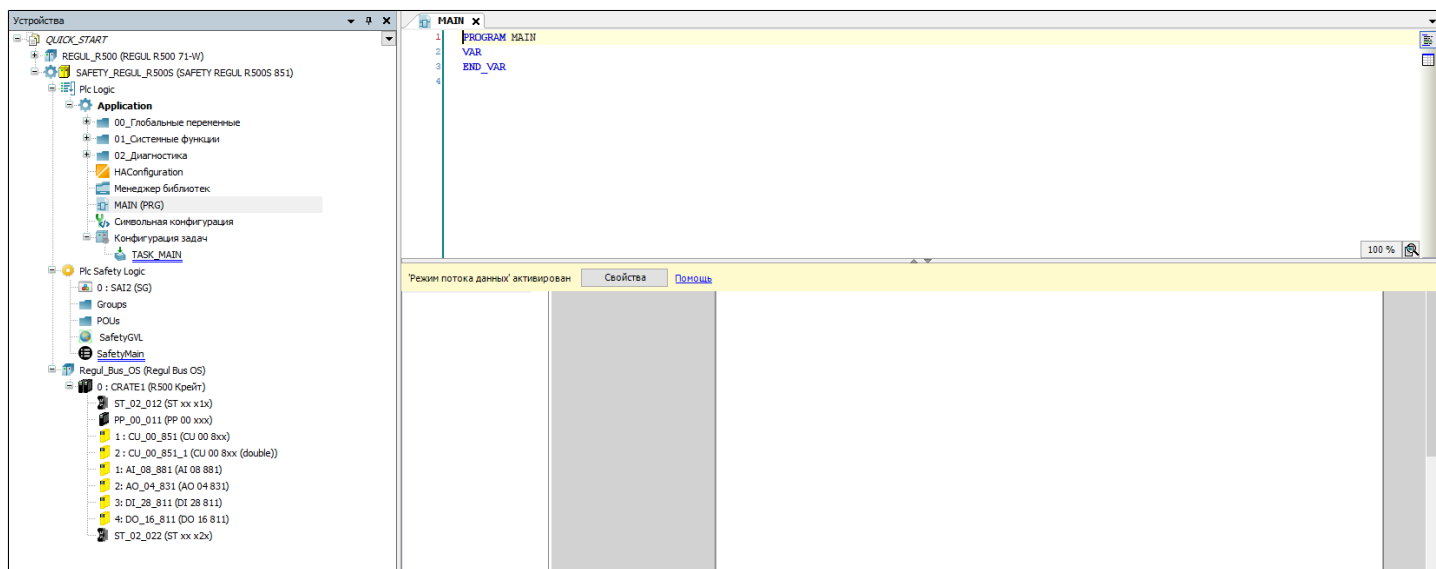
7. Создайте новую программу (POU), выберите в контекстном меню приложения "Application" пункт "Добавление объекта" → "POU...".



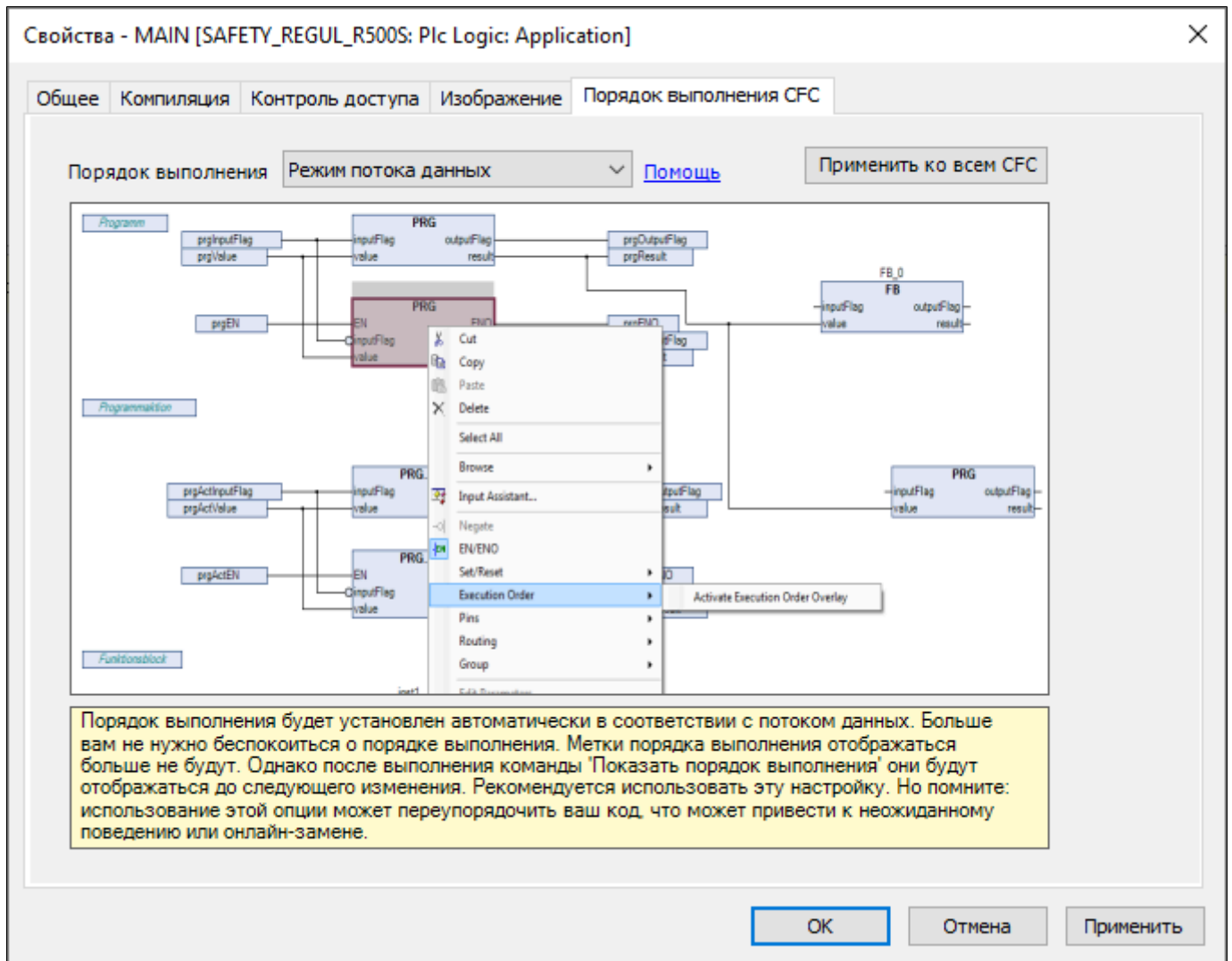
8. Задайте название программы "MAIN", укажите тип "Программа" и язык реализации "CFC постраничный". Нажмите кнопку "Добавить".



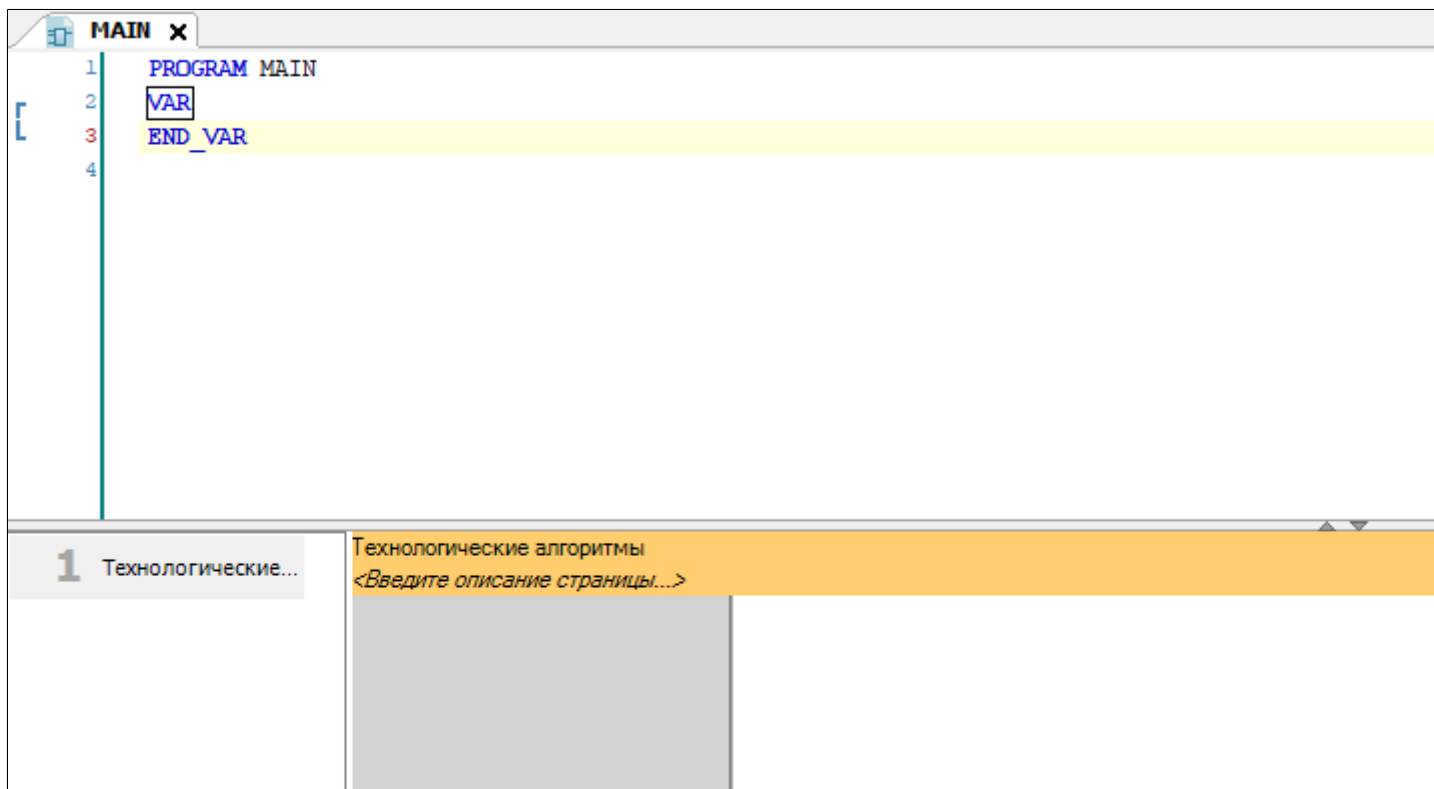
Откроется редактор программы "MAIN":



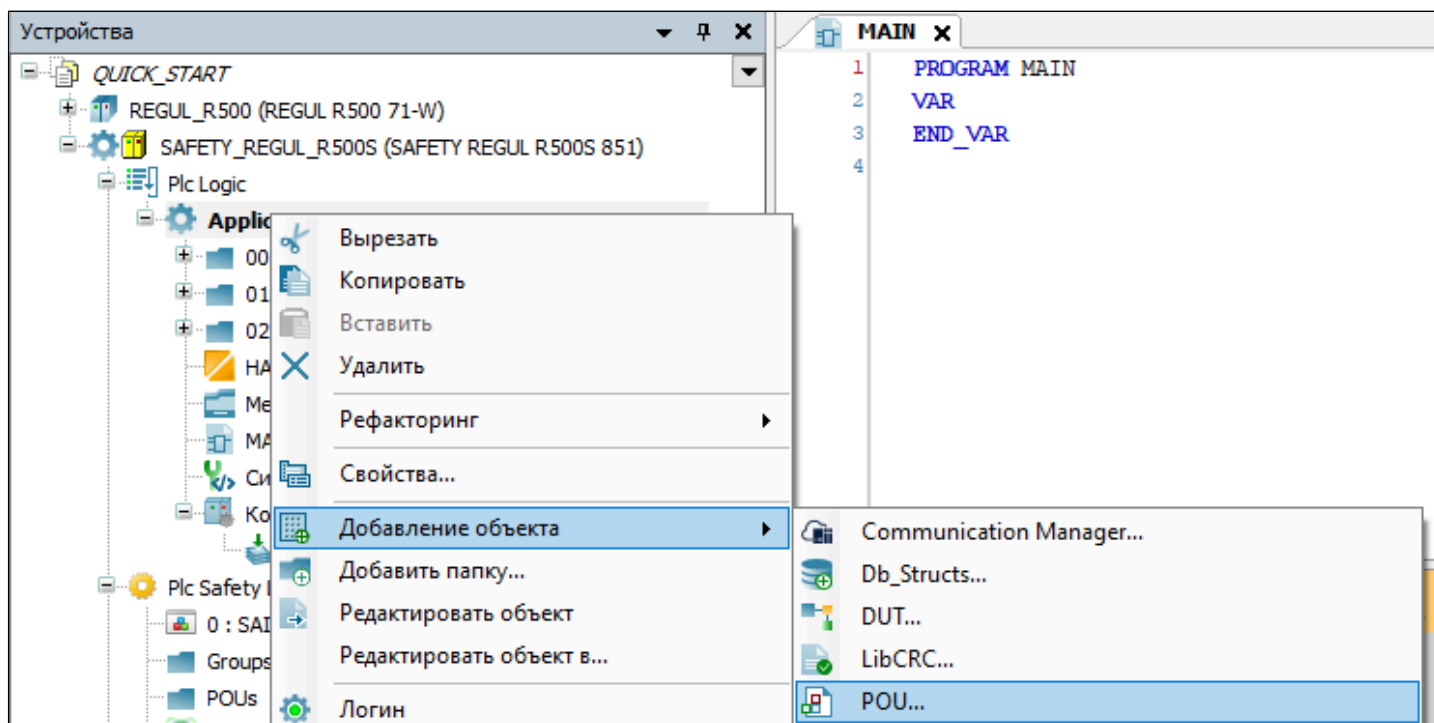
9. Кликните на кнопку "Свойства" и выберите пункт "Режим потока данных" из выпадающего списка, затем нажмите кнопку "Применить".



10. В редакторе программы "MAIN" задайте имя страницы – "Технологические алгоритмы".



11. Создайте вторую программу (POU), выберите в контекстном меню приложения "Application" пункт "Добавление объекта" → "POU...".



12. Задайте добавляемой задаче имя "PROXY\_SAI2" и укажите язык реализации "CFC постраничный".



Добавить POU

Создать новый POU (компонент организации программы)

Имя

Тип

**Программа**

**Функциональный блок**

Extends  ...

Implements  ...

Окончательный  Абстрактный

Спецификатор доступа

Язык реализации метода:

**Функция**

Тип возвращаемого значения  ...

Язык реализации

13. Откроется редактор программы "PROXY\_SAI2". Задайте имя страницы "Аналоговый вход с двумя порогами сигнализации" выберите "Режим потока данных".

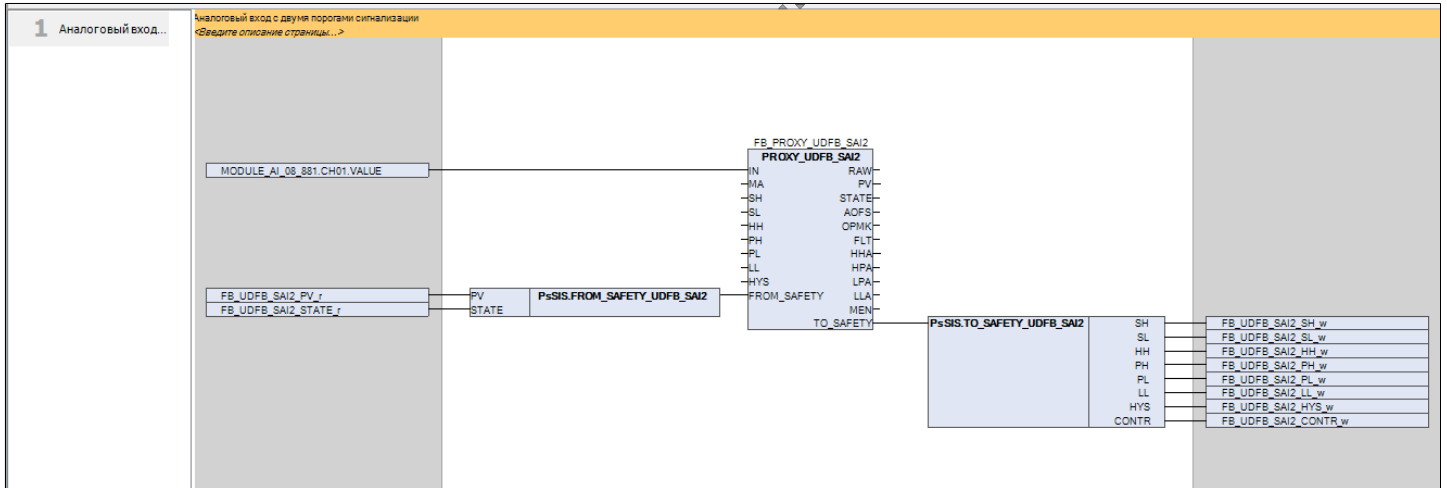
```

1  PROGRAM PROXY_SAI2
2  VAR
3
4  END_VAR
5

```

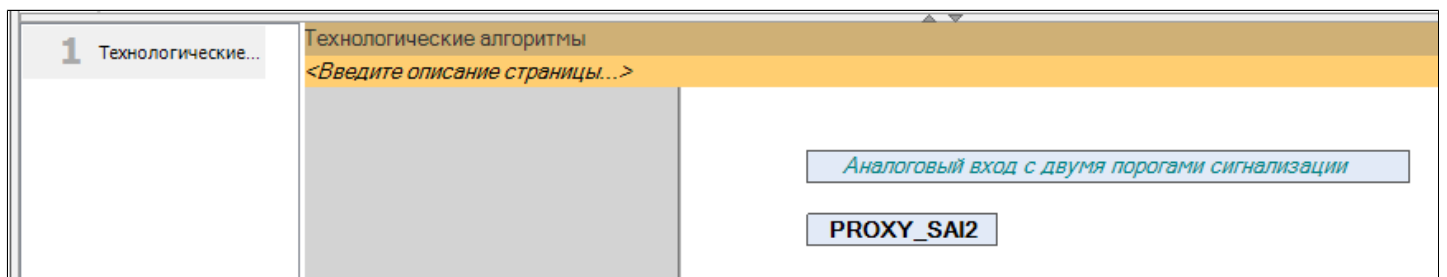
1 Аналоговый вх... Аналоговый вход с двумя порогами сигнализации  
 <Введите описание страницы...>

14. Создайте в рабочей области следующую структуру, состоящую из элемента "PROXY\_UDFB\_SAI2", компоновщика "PsSIS.FROM\_SAFETY\_UDFB\_SAI2" и селектора "PsSIS.TO\_SAFETY\_UDFB\_SAI2". Привяжите переменные из безопасной части ко входам/выходам блоков.

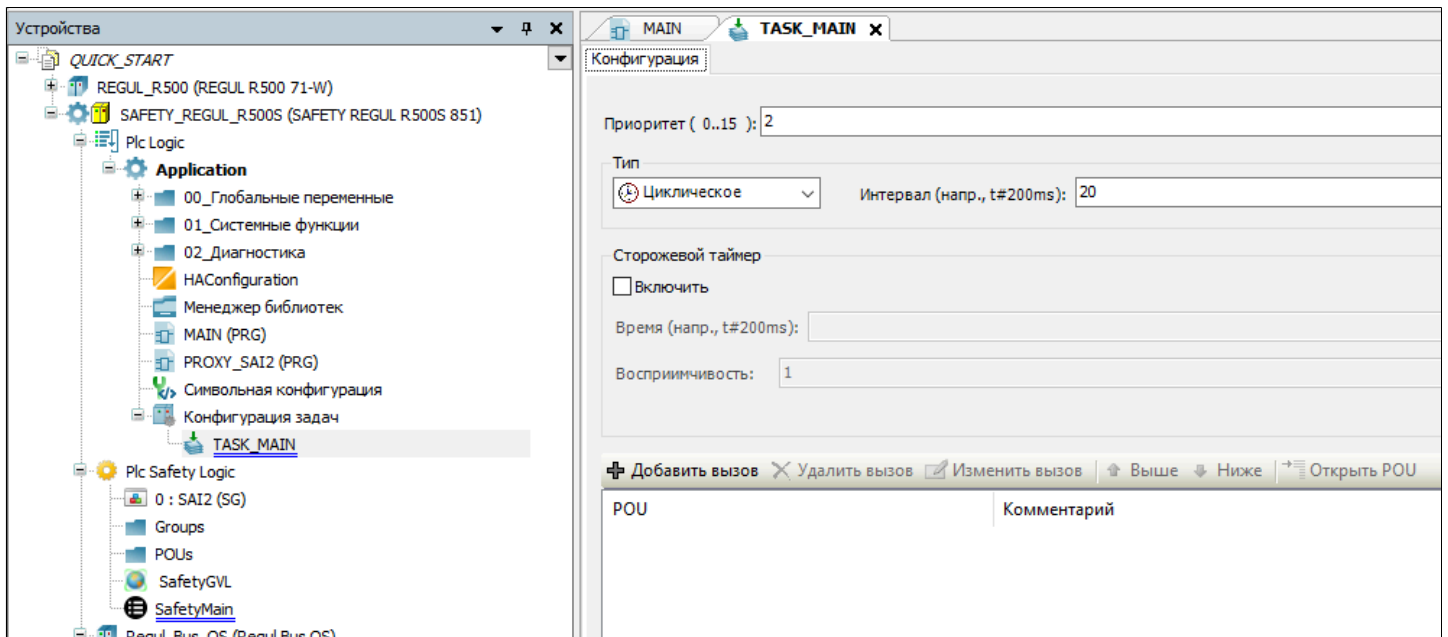


Чтобы добавить объект, перетащите его из панели инструментов в рабочую область.

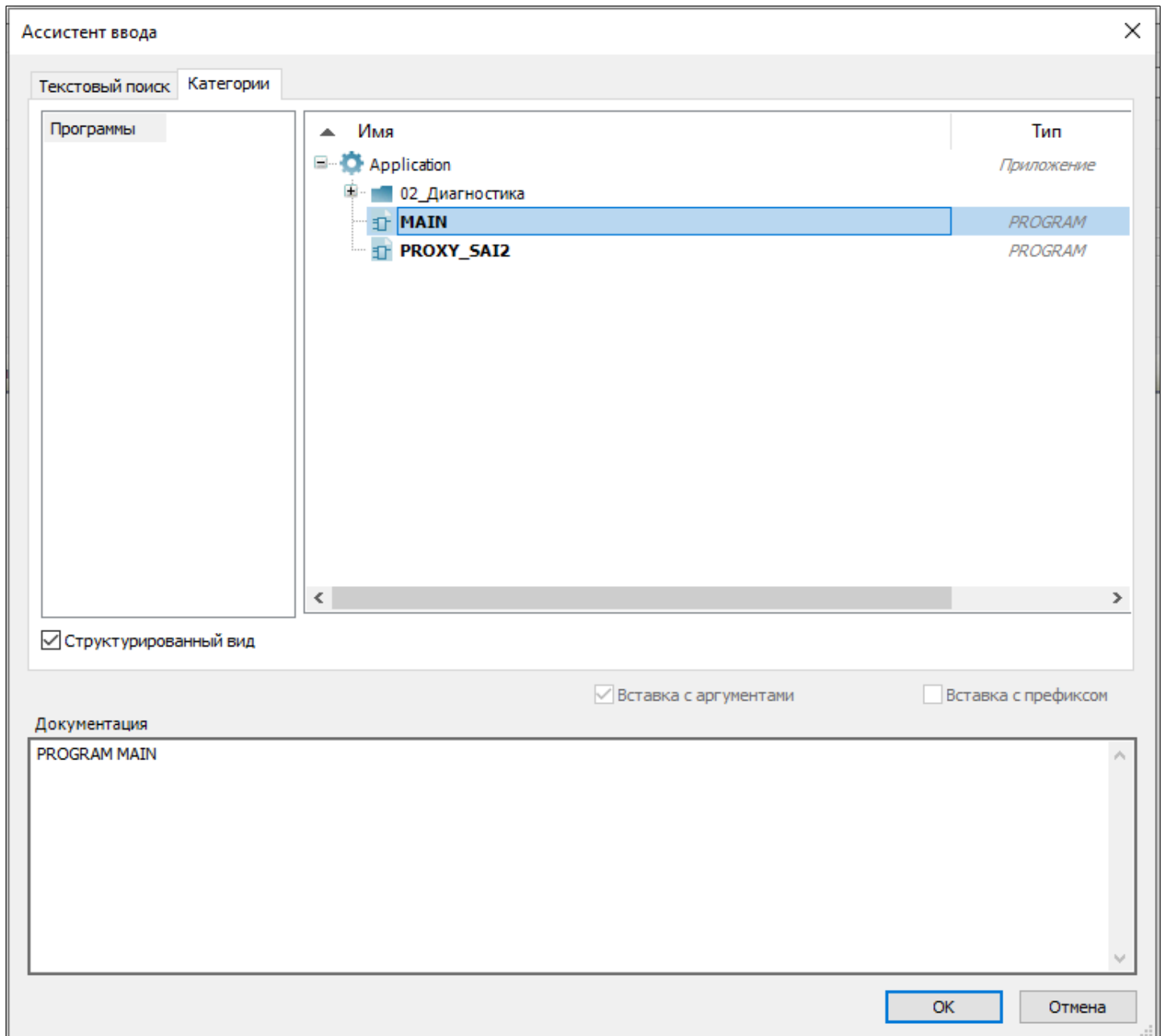
15. Чтобы программа "PROXY\_SAI2" вызывалась программой "MAIN" необходимо на странице "Технологические алгоритмы" добавить объект "Элемент" из панели инструментов и задать ему имя действия, которое необходимо вызвать – "PROXY\_SAI2". Добавьте объект "Комментарий" из панели инструментов и задайте комментарий "Аналоговый вход с двумя порогам сигнализации".



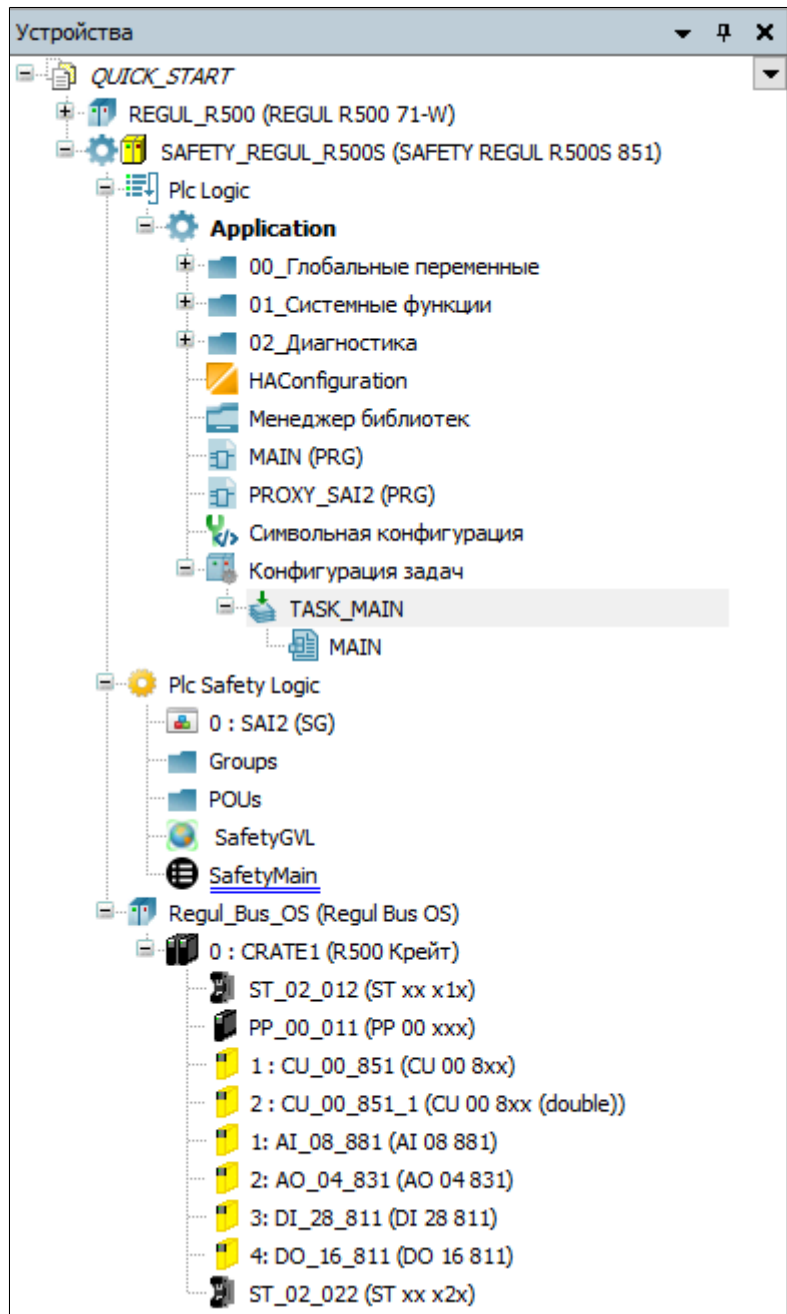
16. Чтобы прикладная программа "MAIN" исполнялась в контроллере, необходимо добавить ее вызов в задаче.  
Перейдите двойным щелчком мыши в конфигурацию задачи TASK\_MAIN.



17. Нажмите кнопку "Добавить вызов" и в ассистенте ввода выберите программу "MAIN", нажмите кнопку "OK".



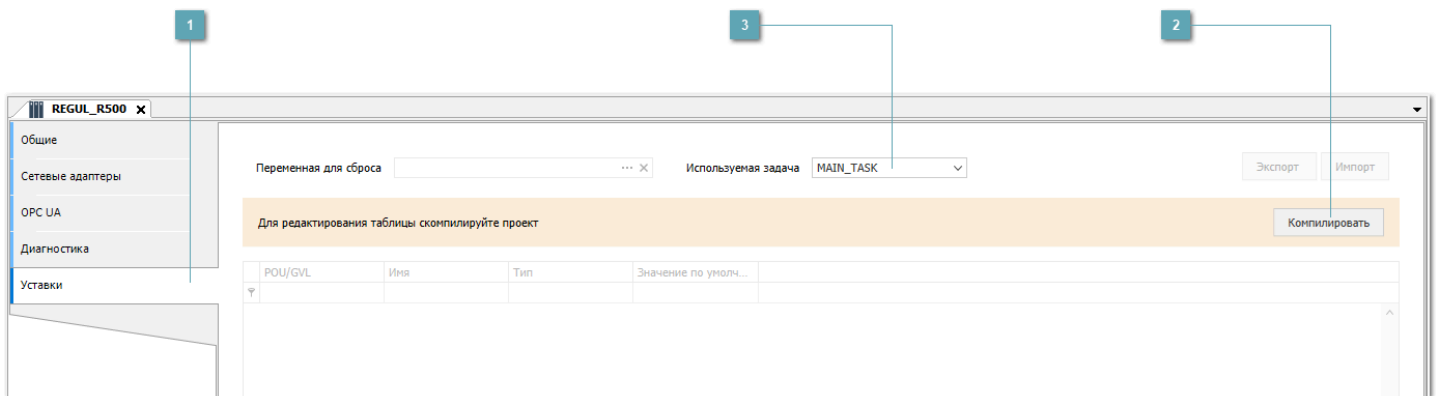
При добавлении прикладной программы "MAIN" в задачу, она отобразится в дереве устройств.



## 1.1.2.3.4. Редактор "Уставки"

В данном разделе описан подробный интерфейс компонента "Уставки" плагина "AstraRegul".

После создания конфигурации системы АСУ ТП с помощью плагина "AstraRegul" и создания технологических алгоритмов, необходимо задать значения по умолчанию для настроечных параметров. Для этого необходимо открыть объект контроллера в созданной конфигурации системы и выбрать вкладку "Уставки".



### 1 Вкладка "Уставки"

Вкладка для задания значений по умолчанию в объекте используемого контроллера.

### 2 Кнопка "Компилировать"

Для редактирования таблицы с уставками необходимо выполнить компиляцию проекта, нажав данную кнопку.

### 3 Выпадающий список "Используемая задача"

Выпадающий список для выбора задачи из созданного проекта для формирования таблицы переменных.



При выборе используемой задачи запрещено выбирать задачу диагностики "TASK\_DIAGN". При выборе неверной задачи, могут возникнуть серьезные ошибки.

## 1.1.2.3.4.1. Интерфейс пользователя

После компиляции проекта и выбора используемой задачи формируется таблица с установочными значениями.

1

5 2 6 7 8 3 4

REGUL\_R500 x

Общие  
Сетевые адаптеры  
ОРС UA  
Диагностика  
Уставки

Переменная для сброса ... X

Используемая задача TASK\_MAIN

Экспорт Импорт

POU/GVL	Имя	Тип	Значение по умолч...
MAIN	FB_AI1.CONFIG.SCA...	BOOL	true
MAIN	FB_AI1.CONFIG.XMIN	REAL	4.0
MAIN	FB_AI1.CONFIG.XMAX	REAL	20.0
MAIN	FB_AI1.CONFIG.YMIN	REAL	0.0
MAIN	FB_AI1.CONFIG.YMAX	REAL	100.0
MAIN	FB_AI1.CONFIG.BRE...	REAL	3.5
MAIN	FB_AI1.CONFIG.OVE...	REAL	21.0
MAIN	FB_AI1.CONFIG.TF	REAL	0.0
MAIN	FB_AI1.CONFIG.AH2	REAL	95.0
MAIN	FB_AI1.CONFIG.AH	REAL	90.0
MAIN	FB_AI1.CONFIG.WH	REAL	80.0
MAIN	FB_AI1.CONFIG.WL	REAL	20.0
MAIN	FB_AI1.CONFIG.AL	REAL	10.0
MAIN	FB_AI1.CONFIG.AL2	REAL	5.0
MAIN	FB_AI1.CONFIG.HYST	REAL	0.0
MAIN	FB_AI1.CONFIG.TEMP	REAL	1E6
MAIN	FB_AI1.CONFIG.SIG...	BYTE	0
MAIN	FB_AI1.MAN_ON	BOOL	false
MAIN	FB_AI1.MAN_VALUE	REAL	0.0

### 1 Таблица с уставками


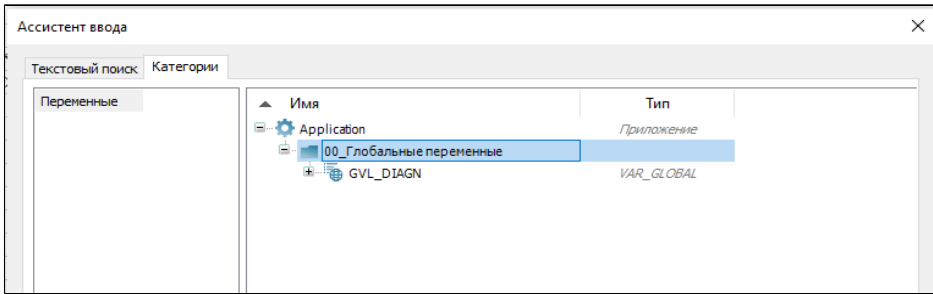

Таблица со значениями по умолчанию.



## 2 Поле "Переменная для сброса"

В данном поле указывается логическая переменная для сброса уставок в изначальное значение.

### Область выбора переменной для сброса

	<p>Кнопка "Многоточие" открывает окно "Ассистент ввода" для выбора переменной для сброса из программы.</p> 
	<p>Кнопка "Крестик" удаляет значение переменной для сброса и очищает поле ввода.</p>

## 3 Кнопка "Экспорт"

Экспортирование таблицы с уставками в файл формата \*.xlsx

## 4 Кнопка "Импорт"

Импортирование таблицы с уставками из файла формата \*.xlsx

## 5 Столбец "POU/GVL"

Источник переменной (программа или список глобальных переменных).

## 6 Столбец "Имя"

Наименование переменной.



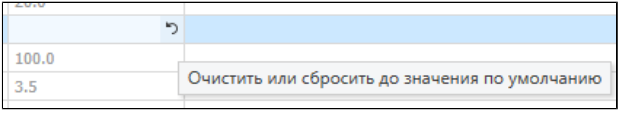
## 7 Столбец "Тип"

Тип данных переменной.

## 8 Столбец "Значения по умолчанию"

Значения по умолчанию для уставок. Отображаются значения, указанные в атрибуте при инициализации переменной.

### Область редактирования переменных

	Для логических переменных в области редактирования есть возможность выбора значения через выпадающий список.
	Для численных переменных (INT, REAL и др.) используется поле ввода.
 <p>Очистить или сбросить до значения по умолчанию</p>	Кнопка "Очистить или сбросить до значения по умолчанию" очищает или сбрасывает данное поле, и заполняет значением, указанным в атрибуте переменной.



В области редактирования невозможно задать неверный формат записи для переменной.

## 1.1.2.3.4.2. Инициализация уставок

Для формирования установочных значений в компоненте "Уставки" необходимо их проинициализировать. Для этого используется атрибут, добавляемый перед вызовом переменной в программе.

Все настроечные параметры необходимо хранить в энергонезависимой памяти. Переменные, которые необходимо хранить в энергонезависимой памяти (уставки) помечаются атрибутом {attribute 'ps.setpoint':='xxxxx'}, где xxxxx - начальное значение параметра (значения по умолчанию). Пример использования атрибута приведен ниже:

```
{attribute 'ps.setpoint':='0.0'}  
YMIN          : REAL:=0.0;      //Минимум выходной величины
```

Данная переменная автоматически сформируется в компоненте "Уставки" после компиляции проекта со значением "0.0".

В библиотечных блоках также используется данный атрибут для передачи переменных в энергонезависимую память, поэтому достаточно будет скомпилировать проект для формирования таблицы с уставками. Дополнительных действий не требуется.

## 1.1.2.3.4.3. Экспорт и импорт уставок

Для более удобного редактирования значения уставочных параметров в компоненте "Уставки" реализован экспорт и импорт файла с таблицей значений в файл формата \*.xlsx для редактирования переменных в любом табличном редакторе.

[Экспорт уставок](#)

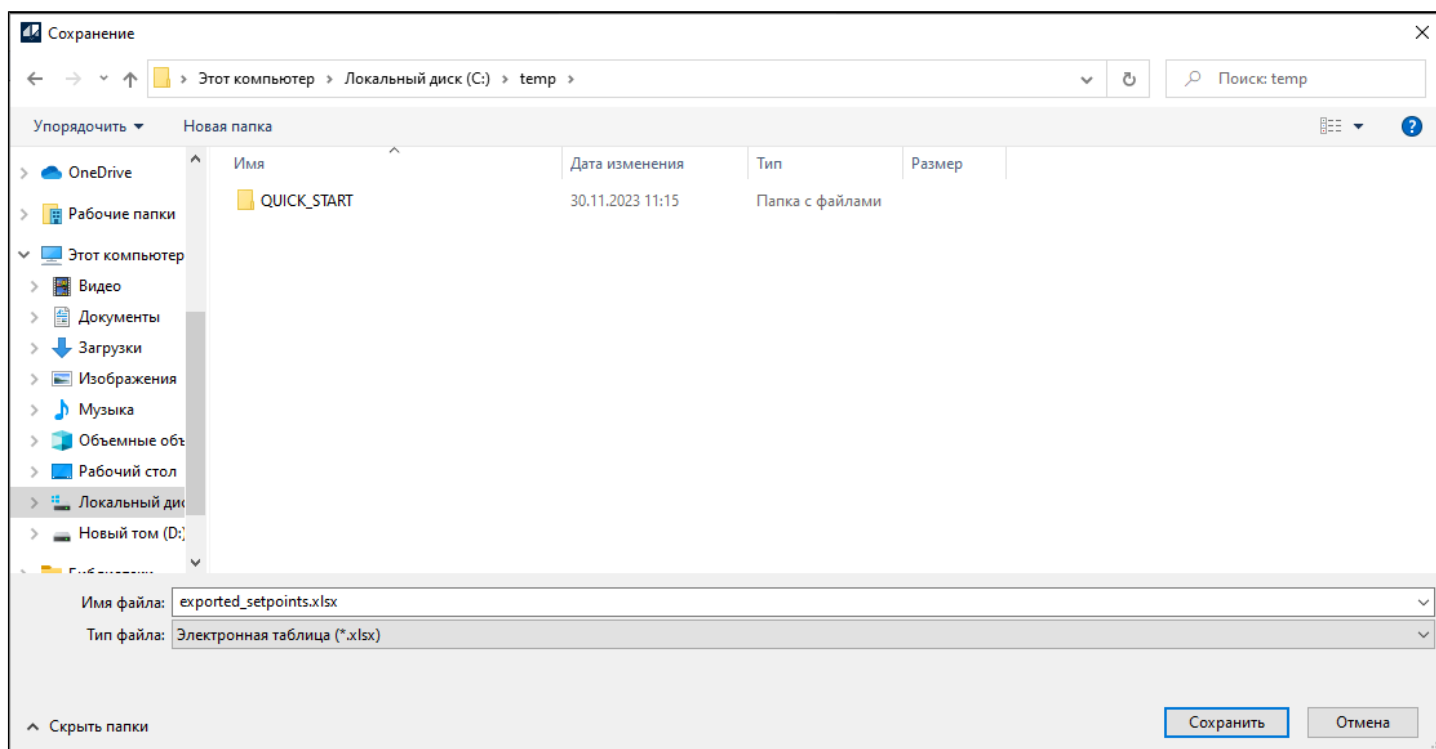
[Импорт уставок](#)

## 1.1.2.3.4.3.1. Экспорт уставок

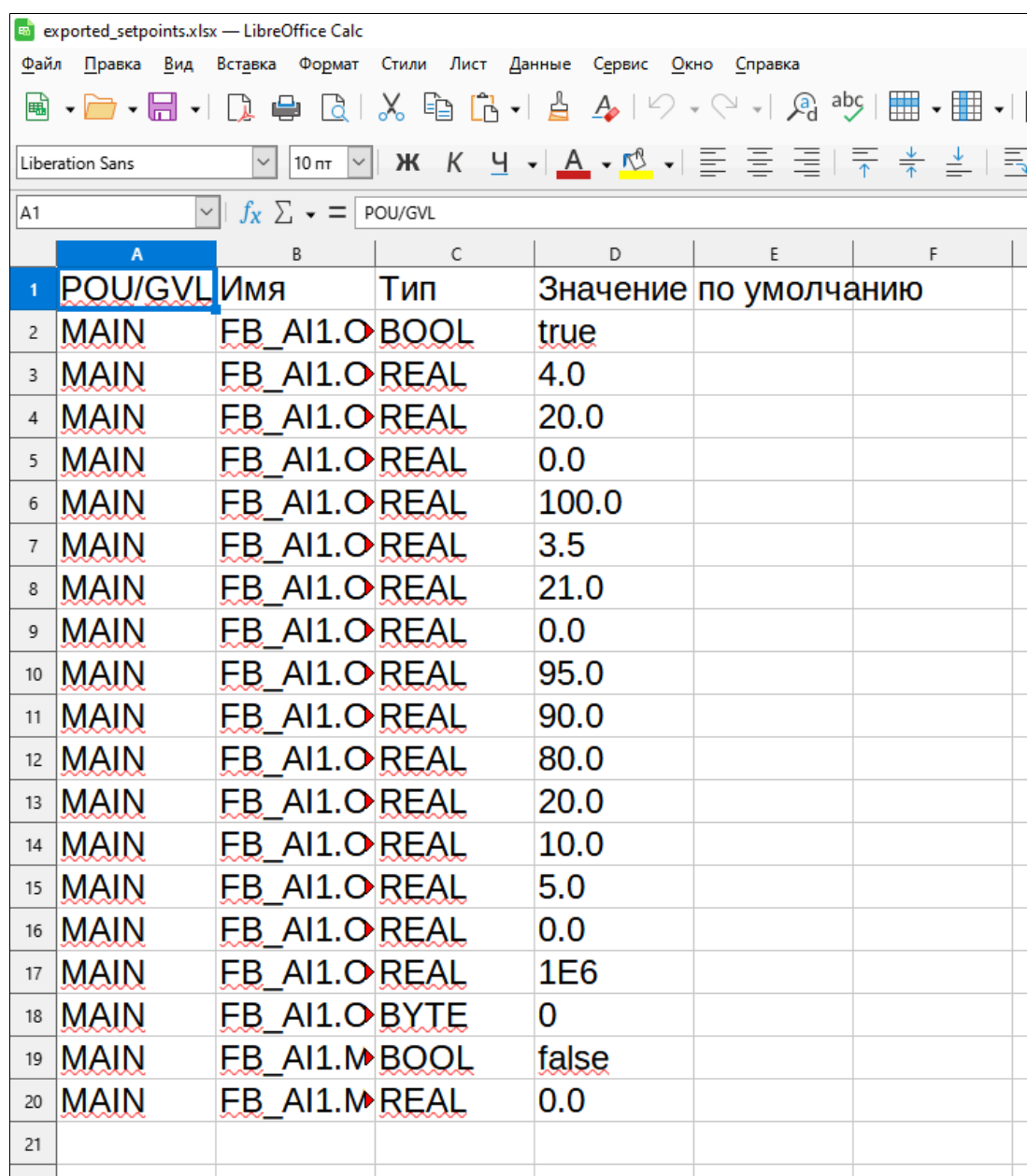
1. Для экспорта таблицы нажмите кнопку "Экспорт" в правом верхнем углу в редакторе уставок.

POU/GVL	Имя	Тип	Значение по умолч...
> MAIN	FB_AI1.CONFIG.SCA...	BOOL	true
MAIN	FB_AI1.CONFIG.XMIN	REAL	4.0
MAIN	FB_AI1.CONFIG.XMAX	REAL	20.0
MAIN	FB_AI1.CONFIG.YMIN	REAL	0.0
MAIN	FB_AI1.CONFIG.YMAX	REAL	100.0
MAIN	FB_AI1.CONFIG.BRE...	REAL	3.5
MAIN	FB_AI1.CONFIG.OVE...	REAL	21.0
MAIN	FB_AI1.CONFIG.TF	REAL	0.0
MAIN	FB_AI1.CONFIG.AH2	REAL	95.0
MAIN	FB_AI1.CONFIG.AH	REAL	90.0
MAIN	FB_AI1.CONFIG.WH	REAL	80.0
MAIN	FB_AI1.CONFIG.WL	REAL	20.0
MAIN	FB_AI1.CONFIG.AL	REAL	10.0
MAIN	FB_AI1.CONFIG.AL2	REAL	5.0
MAIN	FB_AI1.CONFIG.HYST	REAL	0.0
MAIN	FB_AI1.CONFIG.TEMP	REAL	1E6
MAIN	FB_AI1.CONFIG.SIG...	BYTE	0
MAIN	FB_AI1.MAN_ON	BOOL	false
MAIN	FB_AI1.MAN_VALUE	REAL	0.0

2. Откроется окно проводника с возможностью изменения имени файла и выбором каталога сохранения файла.



3. Откройте файл с помощью табличного редактора. Таблица из Astra.IDE экспортировалась в файл и доступна для редактирования. После изменения значений переменных, сохраните файл и закройте его.



	A	B	C	D	E	F
1	POU/GVL	Имя	Тип	Значение по умолчанию		
2	MAIN	FB_AI1.0	BOOL	true		
3	MAIN	FB_AI1.0	REAL	4.0		
4	MAIN	FB_AI1.0	REAL	20.0		
5	MAIN	FB_AI1.0	REAL	0.0		
6	MAIN	FB_AI1.0	REAL	100.0		
7	MAIN	FB_AI1.0	REAL	3.5		
8	MAIN	FB_AI1.0	REAL	21.0		
9	MAIN	FB_AI1.0	REAL	0.0		
10	MAIN	FB_AI1.0	REAL	95.0		
11	MAIN	FB_AI1.0	REAL	90.0		
12	MAIN	FB_AI1.0	REAL	80.0		
13	MAIN	FB_AI1.0	REAL	20.0		
14	MAIN	FB_AI1.0	REAL	10.0		
15	MAIN	FB_AI1.0	REAL	5.0		
16	MAIN	FB_AI1.0	REAL	0.0		
17	MAIN	FB_AI1.0	REAL	1E6		
18	MAIN	FB_AI1.0	BYTE	0		
19	MAIN	FB_AI1.M	BOOL	false		
20	MAIN	FB_AI1.M	REAL	0.0		
21						



Не рекомендуется изменять другие параметры переменных. Это может привести к ошибкам импорта файла в Astra.IDE. Подробнее в разделе ["Основные ошибки при импорте файла с уставками"](#)

## 1.1.2.3.4.3.2. Импорт уставок

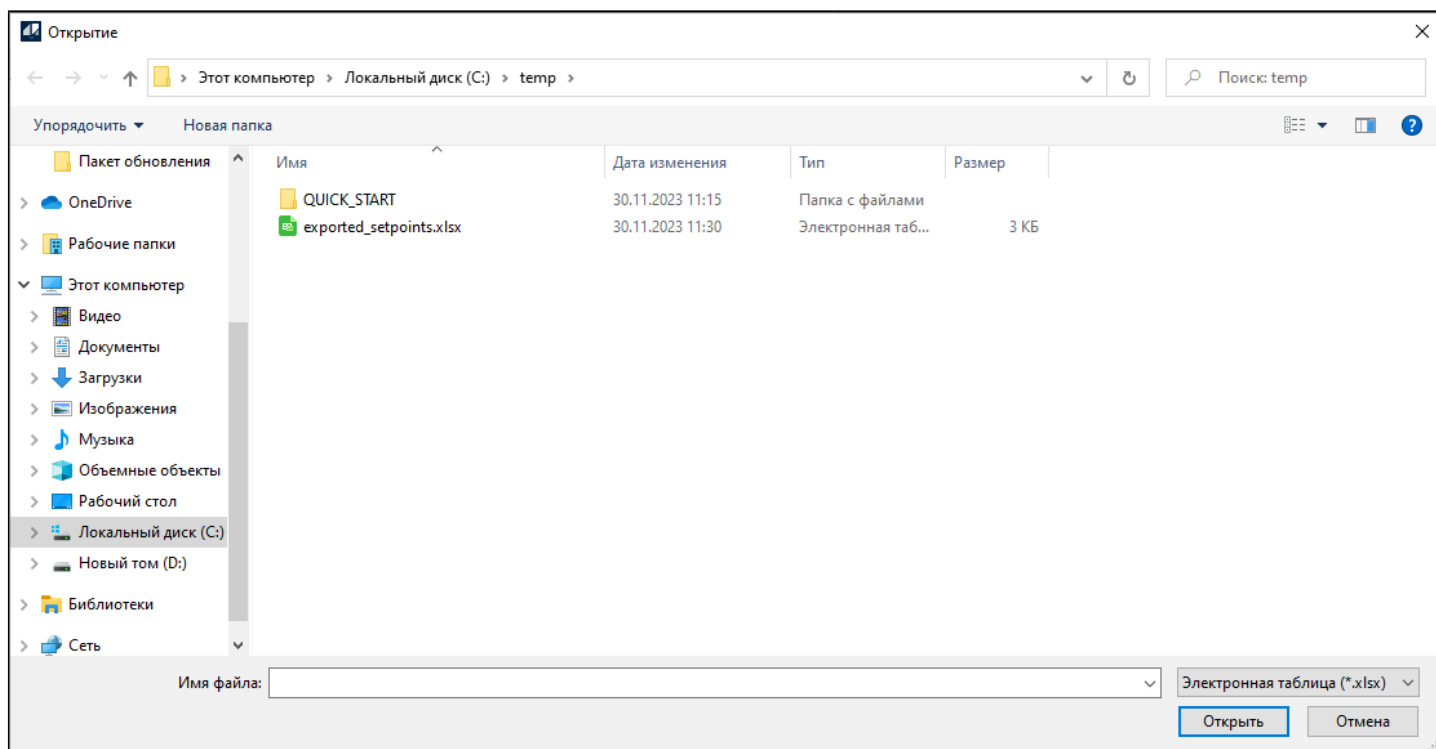
1. Для импорта файла нажмите кнопку "Импорт" в редакторе уставок.

Переменная для сброса  ... X      Используемая задача  ▾     

POU/GVL	Имя	Тип	Значение по умолч...	
> MAIN	FB_AI1.CONFIG.SCA...	BOOL	true	
MAIN	FB_AI1.CONFIG.XMIN	REAL	4.0	
MAIN	FB_AI1.CONFIG.XMAX	REAL	20.0	
MAIN	FB_AI1.CONFIG.YMIN	REAL	0.0	
MAIN	FB_AI1.CONFIG.YMAX	REAL	100.0	
MAIN	FB_AI1.CONFIG.BRE...	REAL	3.5	
MAIN	FB_AI1.CONFIG.OVE...	REAL	21.0	
MAIN	FB_AI1.CONFIG.TF	REAL	0.0	
MAIN	FB_AI1.CONFIG.AH2	REAL	95.0	
MAIN	FB_AI1.CONFIG.AH	REAL	90.0	
MAIN	FB_AI1.CONFIG.WH	REAL	80.0	
MAIN	FB_AI1.CONFIG.WL	REAL	20.0	
MAIN	FB_AI1.CONFIG.AL	REAL	10.0	
MAIN	FB_AI1.CONFIG.AL2	REAL	5.0	
MAIN	FB_AI1.CONFIG.HYST	REAL	0.0	
MAIN	FB_AI1.CONFIG.TEMP	REAL	1E6	
MAIN	FB_AI1.CONFIG.SIG...	BYTE	0	
MAIN	FB_AI1.MAN_ON	BOOL	false	
MAIN	FB_AI1.MAN_VALUE	REAL	0.0	



## 2. Выберите необходимый файл с таблицей в окне проводника.



## 3. После выбора файла в таблицу импортируются данные из файла.

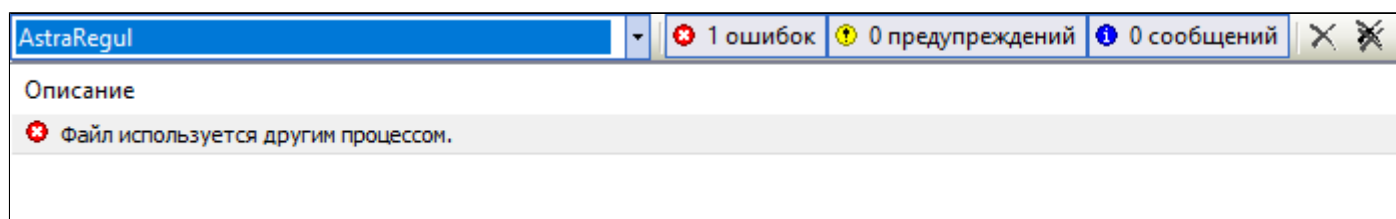
Переменная для сброса  ... X Используемая задача TASK\_MAIN Экспорт Импорт

POU/GVL	Имя	Тип	Значение по умолч...	
> MAIN	FB_AI1.CONFIG.SCA...	BOOL	false	
MAIN	FB_AI1.CONFIG.XMIN	REAL	5.0	
MAIN	FB_AI1.CONFIG.XMAX	REAL	25.0	
MAIN	FB_AI1.CONFIG.YMIN	REAL	0.0	
MAIN	FB_AI1.CONFIG.YMAX	REAL	100.0	
MAIN	FB_AI1.CONFIG.BRE...	REAL	3.5	
MAIN	FB_AI1.CONFIG.OVE...	REAL	21.0	
MAIN	FB_AI1.CONFIG.TF	REAL	0.0	
MAIN	FB_AI1.CONFIG.AH2	REAL	95.0	
MAIN	FB_AI1.CONFIG.AH1	REAL	95.0	

## 1.1.2.3.4.3.2.1. Основные ошибки при импорте файла

### Ошибка "Файл используется другим процессом"

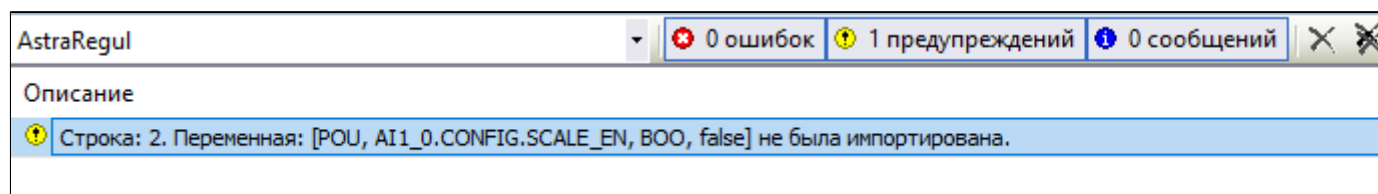
Если файл открыт в табличном редакторе во время импорта будет выведено сообщение об ошибке:



Для устранения необходимо закрыть табличный редактор и повторно произвести импорт файла.

### Предупреждение "Строка: X. Переменная: [X, X, X, X] не была импортирована"

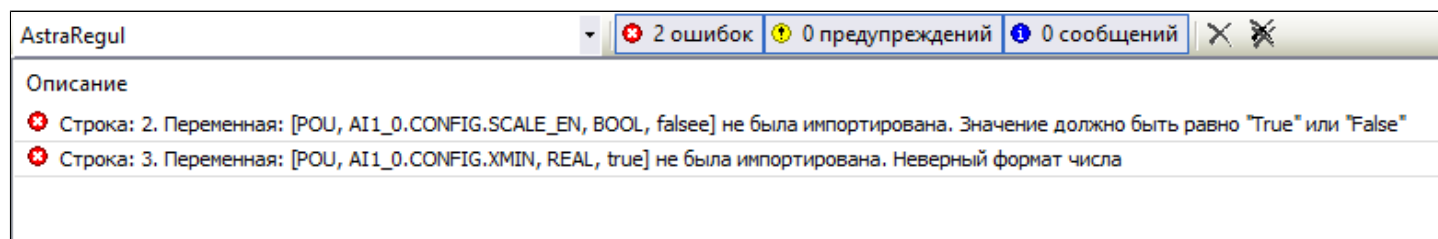
При некорректном вводе параметров "POU/GVL", "Имя" и "Тип" в табличном редакторе будет выведено сообщение с предупреждением:



В предупреждении указывается строка таблицы, в которой допущена ошибка. Для устранения необходимо исправить данные переменной (источник переменной, имя или тип) в табличном редакторе и повторно произвести импорт файла.

## Ошибка "Строка: X. Переменная [X, X, X, X] не была импортирована"

При некорректном вводе параметра "Значение по умолчанию" будет выведено следующее сообщение об ошибке:



В данном сообщении указывается номер строки таблицы, в которой допущена ошибка, а также вариант исправления. Для устранения необходимо исправить значение переменной в табличном редакторе и повторно произвести импорт файла.

## 1.1.2.3.4.4. Фильтр в таблице уставок

Первая строка таблицы уставок необходима для настройки фильтрации данных.

Для фильтрации укажите в первой ячейке выбранного столбца значение, по которому надо отфильтровать переменные.

Переменная для сброса  ... X      Используемая задача          

POU/GVL	Имя	Тип	Значение по умолч...	
&				
> MAIN	FB_AI1.CONFIG.SCA...	BOOL	false	
MAIN	FB_AI1.CONFIG.XMIN	REAL	5.0	
MAIN	FB_AI1.CONFIG.XMAX	REAL	25.0	
MAIN	FB_AI1.CONFIG.YMIN	REAL	0.0	
MAIN	FB_AI1.CONFIG.YMAX	REAL	100.0	

## 1.1.2.3.4.4.1. Пример фильтрации

1. Пример настройки фильтрации на выбор данных типа "BOOL". Для этого в первой ячейке столбца "Тип" укажите параметр "BOOL".

Переменная для сброса  ... X      Используемая задача **TASK\_MAIN** ▾      Экспорт      Импорт

POU/GVL	Имя	Тип	Значение по умолч...
☿		BOOL	
> MAIN	FB_AI1.CONFIG.SCA...	BOOL	false
MAIN	FB_AI1.MAN_ON	BOOL	True

2. Пример комбинации фильтрации для выбора данных типа "BOOL" со значением "True". Для этого в первой ячейке столбца "Тип" укажите параметр "BOOL" и в первой ячейке столбца "Значение по умолчанию" укажите значение "True".

POU/GVL	Имя	Тип	Значение по умолч...
☿		BOOL	True
MAIN	FB_AI1.MAN_ON	BOOL	True



Фильтрация по "Значение по умолчанию" возможна только после импорта таблицы уставок, либо после ручного заполнения значений переменных.

## 1.1.2.3.4.5. ENUM-переменные

Переменные ENUM (перечисления) представляют типы данных, которые содержат набор констант. Каждой константе сопоставлено определенное числовое значение. Данный тип переменной также может храниться в энергонезависимой памяти и редактироваться во время работы контроллера в онлайн режиме.

Для изменения значения константы данной переменной во вкладке уставки необходимо указывать числовое значение, которое ей присвоено. При попытке изменить значение с помощью констант, будет выведена ошибка и значение переменной невозможно будет сохранить.


## 1.1.2.3.4.5.1. Пример работы с ENUM-переменными

Пример изменения значения ENUM переменной в онлайн режиме с помощью числового значения:

ENUM_HHAO.NONE	ENUM_HHAO.NONE	1
ENUM_HAO.NONE	ENUM_HAO.NONE	
enum_test.two	enum_test.two	

ENUM_HHAO.NONE	ENUM_HHAO.NONE	ENUM_HHAO.LL
ENUM_HAO.NONE	ENUM_HAO.NONE	
enum_test.two	enum_test.two	

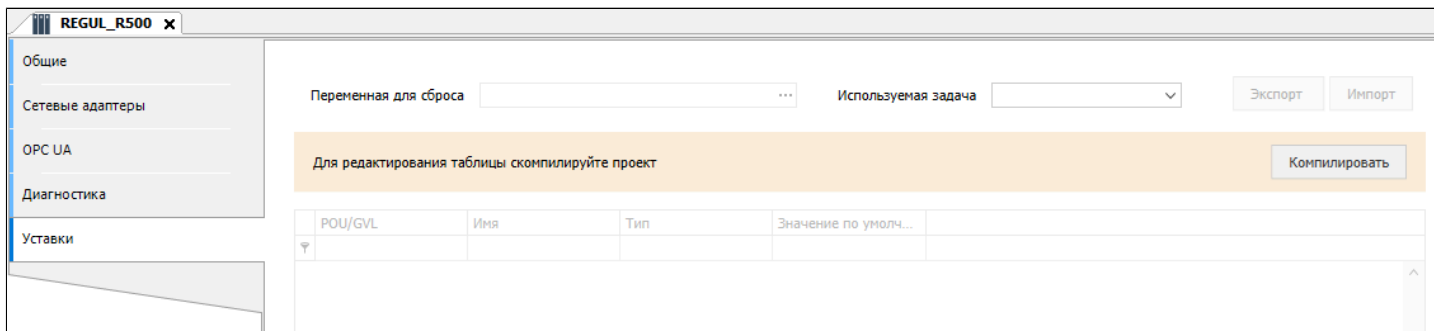
При попытке ввода названия константы происходит ошибка ввода подготовленного значения:

ENUM_HHAO.NONE	ENUM_HHAO.NONE	 ENUM_HHAO.LL
ENUM_HAO.NONE	ENUM_HAO.NONE	
enum_test.two	enum_test.two	

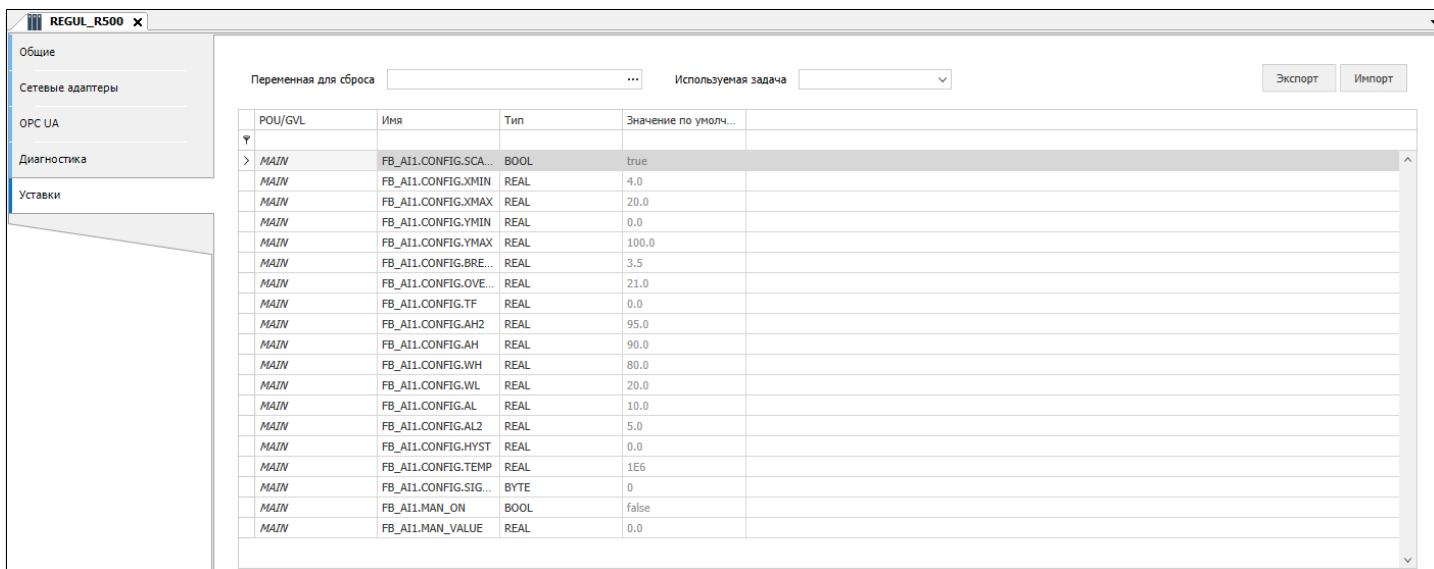
## 1.1.2.3.4.6. Формирования уставок

Чтобы задать значения по умолчанию для настроечных параметров, выполните следующие действия:

1. Перейдите на вкладку "AstraRegul" и откройте вкладку "Уставки" объекта "REGUL\_R500".



2. Для формирования и редактирования таблицы с уставками выполните компиляцию проекта, нажав кнопку "Компилировать".



3. В выпадающем списке поля "Используемая задача" выберите задачу "TASK\_MAIN" для инициализации переменных. Для параметров указаны значения по умолчанию, при необходимости измените значения этих переменных вручную или произведите импорт файла с подготовленными значениями уставок.

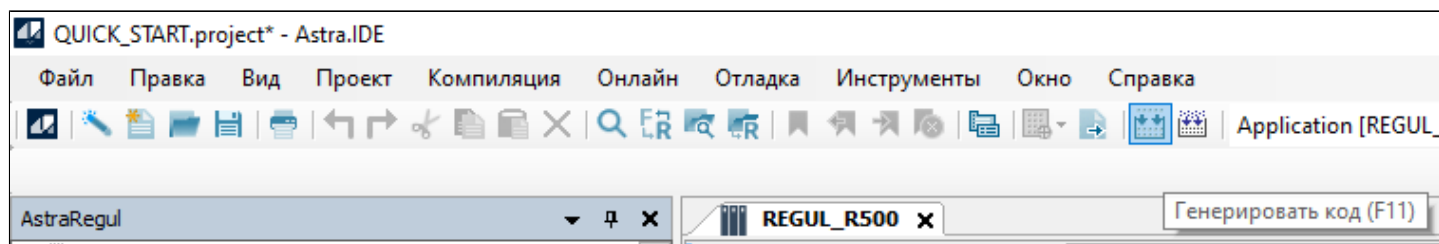


Переменная для сброса

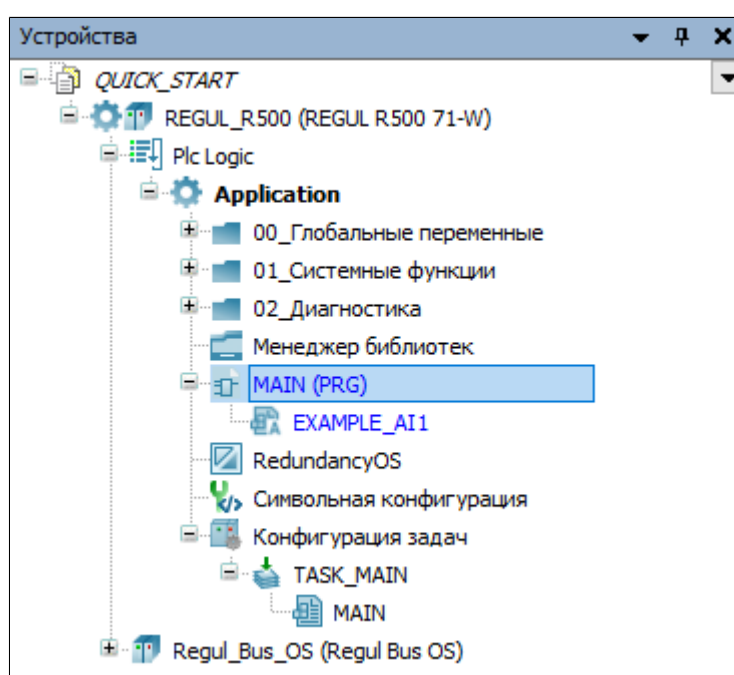
Используемая задача

	POU/GVL	Имя	Тип	Значение по умолчанию	
▼					
>	<i>MAIN</i>	FB_AI1.CONFIG.SCA...	BOOL	true	
	<i>MAIN</i>	FB_AI1.CONFIG.XMIN	REAL	4.0	
	<i>MAIN</i>	FB_AI1.CONFIG.XMAX	REAL	20.0	
	<i>MAIN</i>	FB_AI1.CONFIG.YMIN	REAL	0.0	
	<i>MAIN</i>	FB_AI1.CONFIG.YMAX	REAL	100.0	
	<i>MAIN</i>	FB_AI1.CONFIG.BRE...	REAL	3.5	
	<i>MAIN</i>	FB_AI1.CONFIG.OVE...	REAL	21.0	
	<i>MAIN</i>	FB_AI1.CONFIG.TF	REAL	0.0	
	<i>MAIN</i>	FB_AI1.CONFIG.AH2	REAL	95.0	
	<i>MAIN</i>	FB_AI1.CONFIG.AH	REAL	90.0	
	<i>MAIN</i>	FB_AI1.CONFIG.WH	REAL	80.0	
	<i>MAIN</i>	FB_AI1.CONFIG.WL	REAL	20.0	
	<i>MAIN</i>	FB_AI1.CONFIG.AL	REAL	10.0	
	<i>MAIN</i>	FB_AI1.CONFIG.AL2	REAL	5.0	
	<i>MAIN</i>	FB_AI1.CONFIG.HYST	REAL	0.0	
	<i>MAIN</i>	FB_AI1.CONFIG.TEMP	REAL	1E6	
	<i>MAIN</i>	FB_AI1.CONFIG.SIG...	BYTE	0	
	<i>MAIN</i>	FB_AI1.MAN_ON	BOOL	false	
	<i>MAIN</i>	FB_AI1.MAN_VALUE	REAL	0.0	

4. На панели инструментов нажмите кнопку "Генерировать код (F11)".



После генерации кода прикладная программа выделится синим цветом – программа успешно вызвана задачей "TASK\_MAIN".



## 1.1.2.3.4.7. Ограничения по работе с энергонезависимой памятью

Переменные, хранящиеся в энергонезависимой памяти (уставки), синхронизируются при любом изменении, даже если была произведена запись того же значения. Для уменьшения нагрузки на ЦП необходимо уменьшить частоту записи в эти переменные.

Период записи переменных определяется файловой системой, и не зависит от времени выполнения цикла задачи.



Максимальный период записи данных на диск составляет 10 с.

Другими словами, как бы часто вы не записывали значения, реальная запись на диск будет определяться файловой системой. Все значения отфильтруются и запишется только последнее.

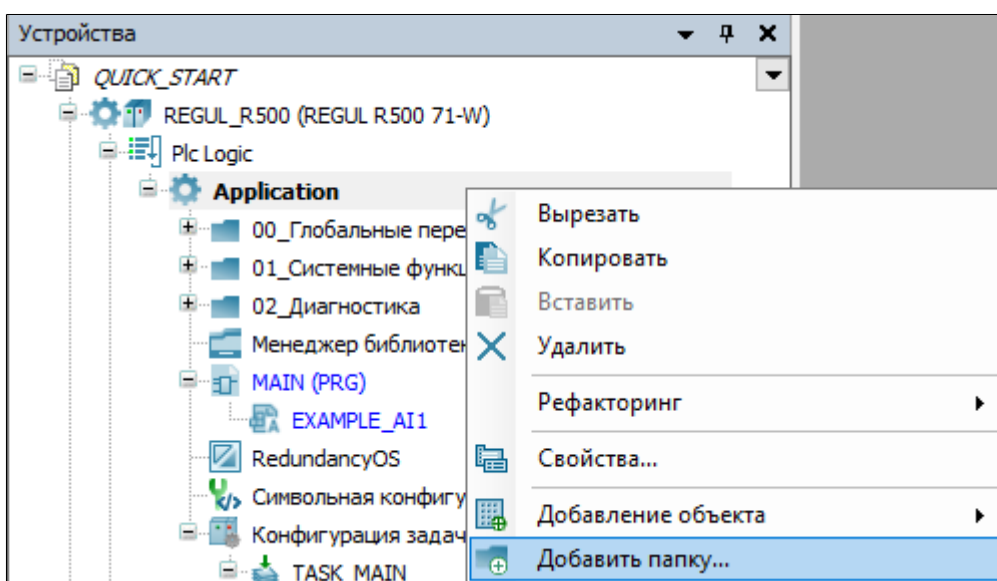
Поэтому рекомендуется в редакторе уставок указывать задачу с низким приоритетом (например, 12) и с циклом 5000 - 10000 мс.

## 1.1.2.3.5. Структурирование проекта

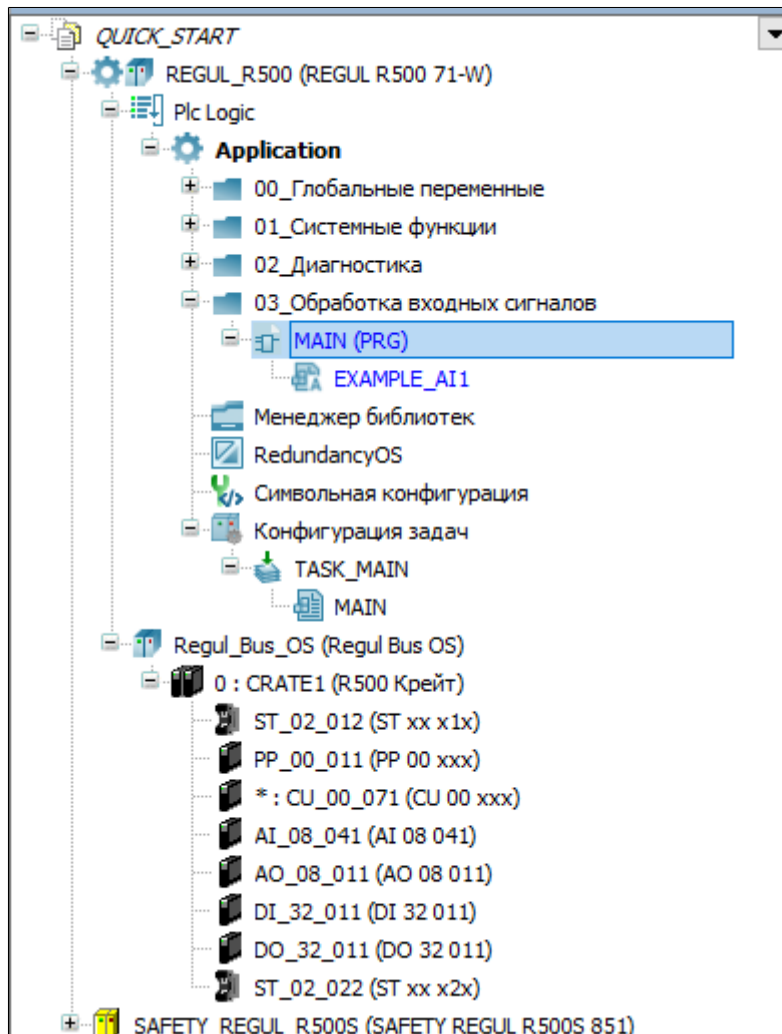
### REGUL R500

Чтобы в проекте было проще ориентироваться, рекомендуется структурировать данные:

1. В контекстном меню приложения "Application" выберите пункт "Добавить папку..."



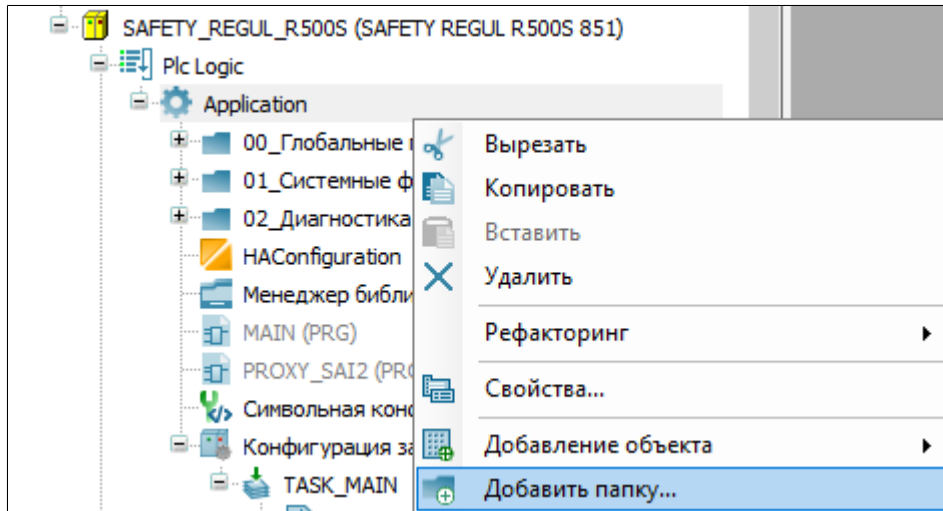
2. Создайте папку "03\_Обработка входных сигналов" и переместите в нее программу "MAIN". Должна получиться следующая структура проекта:



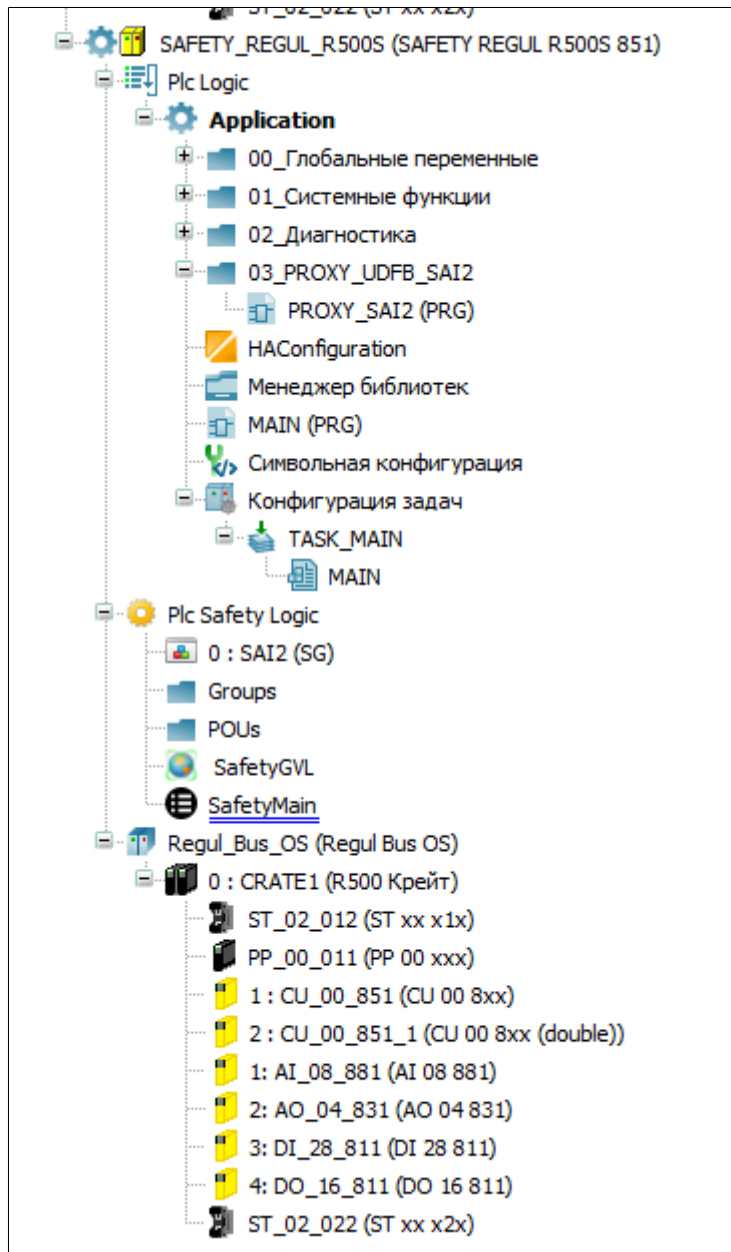
## SAFETY REGUL R500S

Чтобы в проекте было проще ориентироваться, рекомендуется структурировать данные:

1. В контекстном меню приложения "Application" выберите пункт "Добавить папку...".



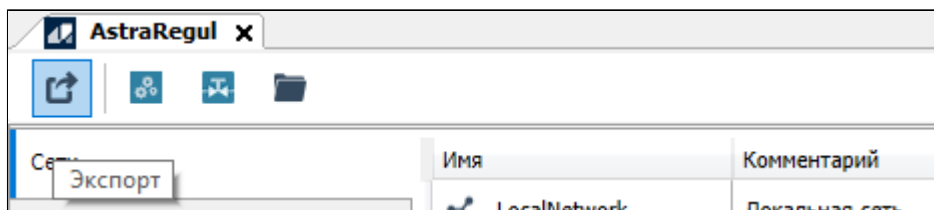
2. Создайте папку "03\_PROXY\_UDFB\_SAI2" и переместите в нее программу "PROXY\_SAI2". Также переместите группу безопасности "SAI2" в папку "Groups". Должна получиться следующая структура проекта:



## 1.1.2.3.6. Экспорт конфигурационных файлов

Необходимо добавить приложения для сервера ввода/вывода в созданные нами объекты сети, для этого необходимо экспортировать нашу топологию сети в конфигуратор Astra.Astudio.

1. Перейдите в объект "AstraRegul" и нажмите кнопку "Компиляция" для проверки правильности конфигурации сети, далее на панели инструментов "AstraRegul" нажмите кнопку "Экспорт".

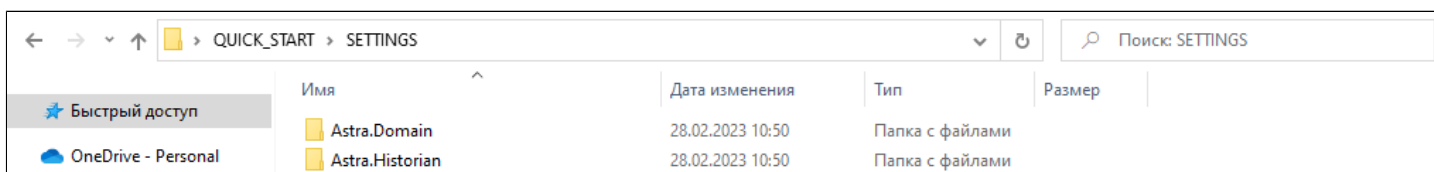


2. После экспорта в папке с проектом Astra.IDE добавились 3 папки:

- AS – папка с проектом для Astra.AStudio.
- HMI – папка для проекта Astra.HMI.
- SETTINGS – папка с конфигурационными файлами "Historian" и "Domain" для каждого объекта сети.

Имя	Дата изменения	Тип	Размер
AS	30.11.2023 10:49	Папка с файлами	
HMI	30.11.2023 10:49	Папка с файлами	
SETTINGS	30.11.2023 10:49	Папка с файлами	
QUICK_START.project	30.11.2023 10:49	Astra.IDE project	323 КБ

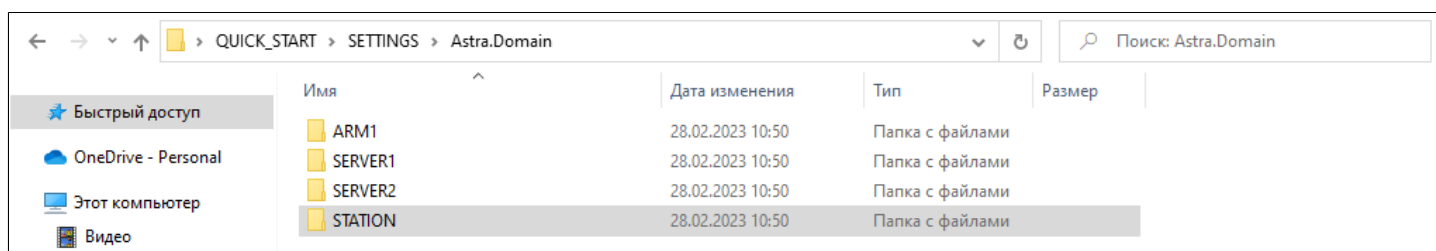
3. Перейдите в папку "SETTINGS". В ней располагаются 2 папки "Astra.Domain" и "Astra.Historian", которые содержат в себе конфигурационные файлы для каждого компонента системы (сервер, станция, АРМ и т.д.).

The screenshot shows a Windows File Explorer window. The address bar shows the path "QUICK\_START > SETTINGS". The search bar contains "Поиск: SETTINGS". The main area displays a table of files and folders.

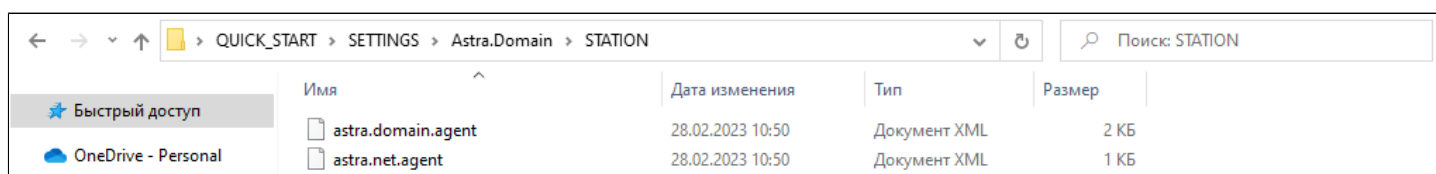
Имя	Дата изменения	Тип	Размер
Astra.Domain	28.02.2023 10:50	Папка с файлами	
Astra.Historian	28.02.2023 10:50	Папка с файлами	



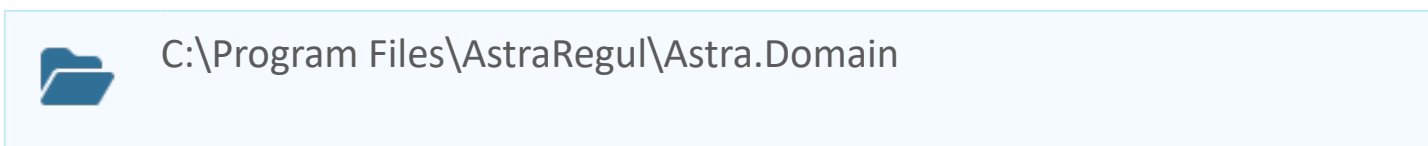
4. Перейдите в папку "Astra.Domain" и выберите для настройки устройства папку с конфигурационными файлами, например, необходимо произвести настройку Astra.Domain для инженерной станции.



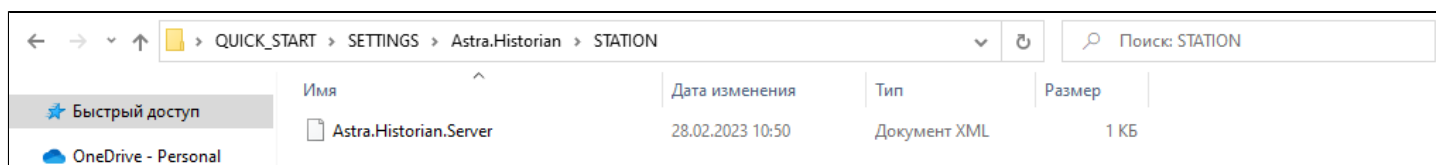
5. Внутри папки "STATION" располагаются 2 файла "astra.domain.agent.xml" и "astra.net.agent.xml".



Скопируйте их и подложите по следующему пути с заменой:



6. Перейдите в папку "Astra.Historian" и выберите для настройки устройства папку с конфигурационными файлами, например, необходимо произвести настройку Astra.Historian для инженерной станции. Внутри папки будет располагаться файл "Astra.Historian.Server.xml".

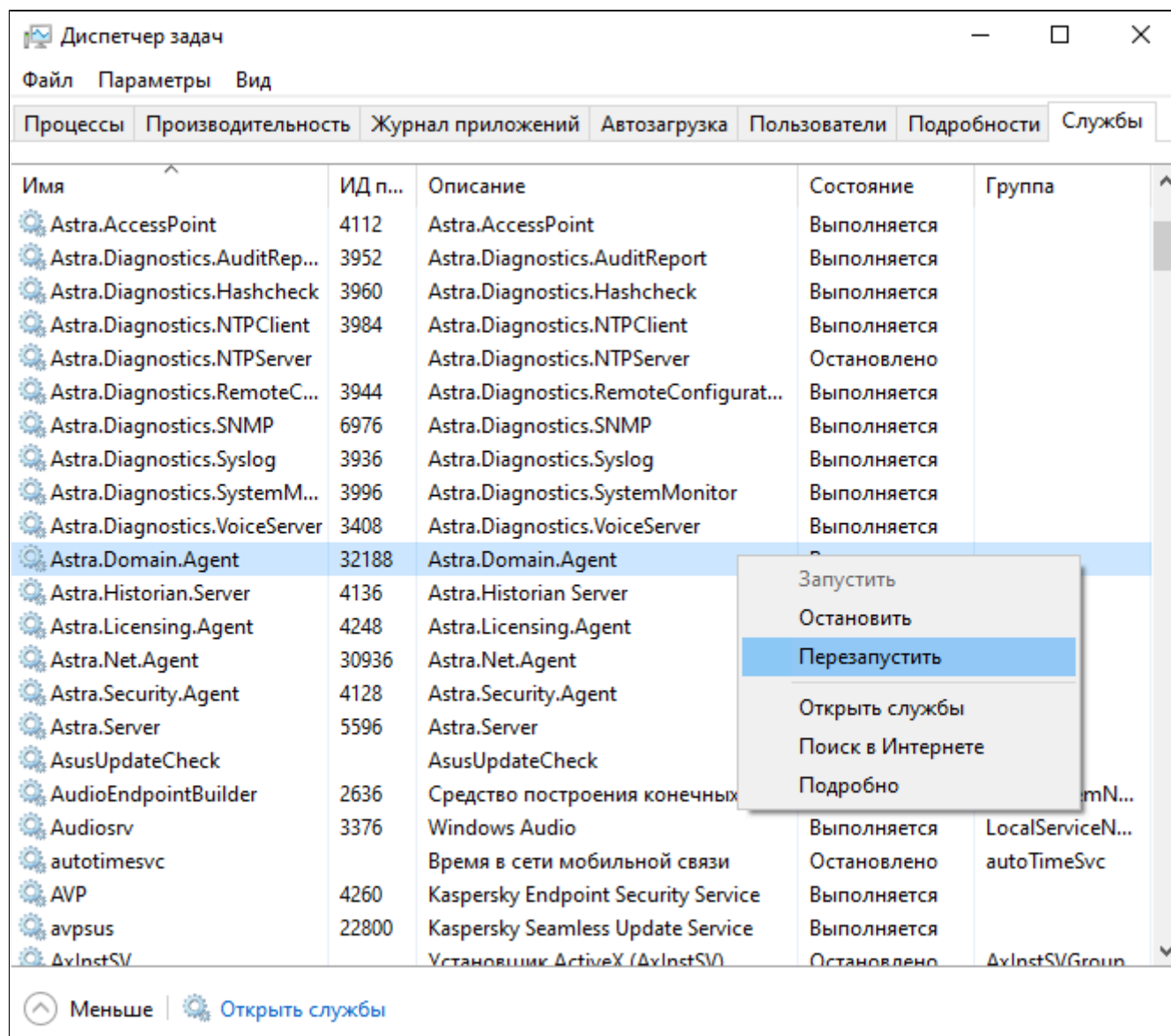


Скопируйте его и подложите по следующему пути с заменой:



C:\Program Files\AstraRegul\Astra.Historian

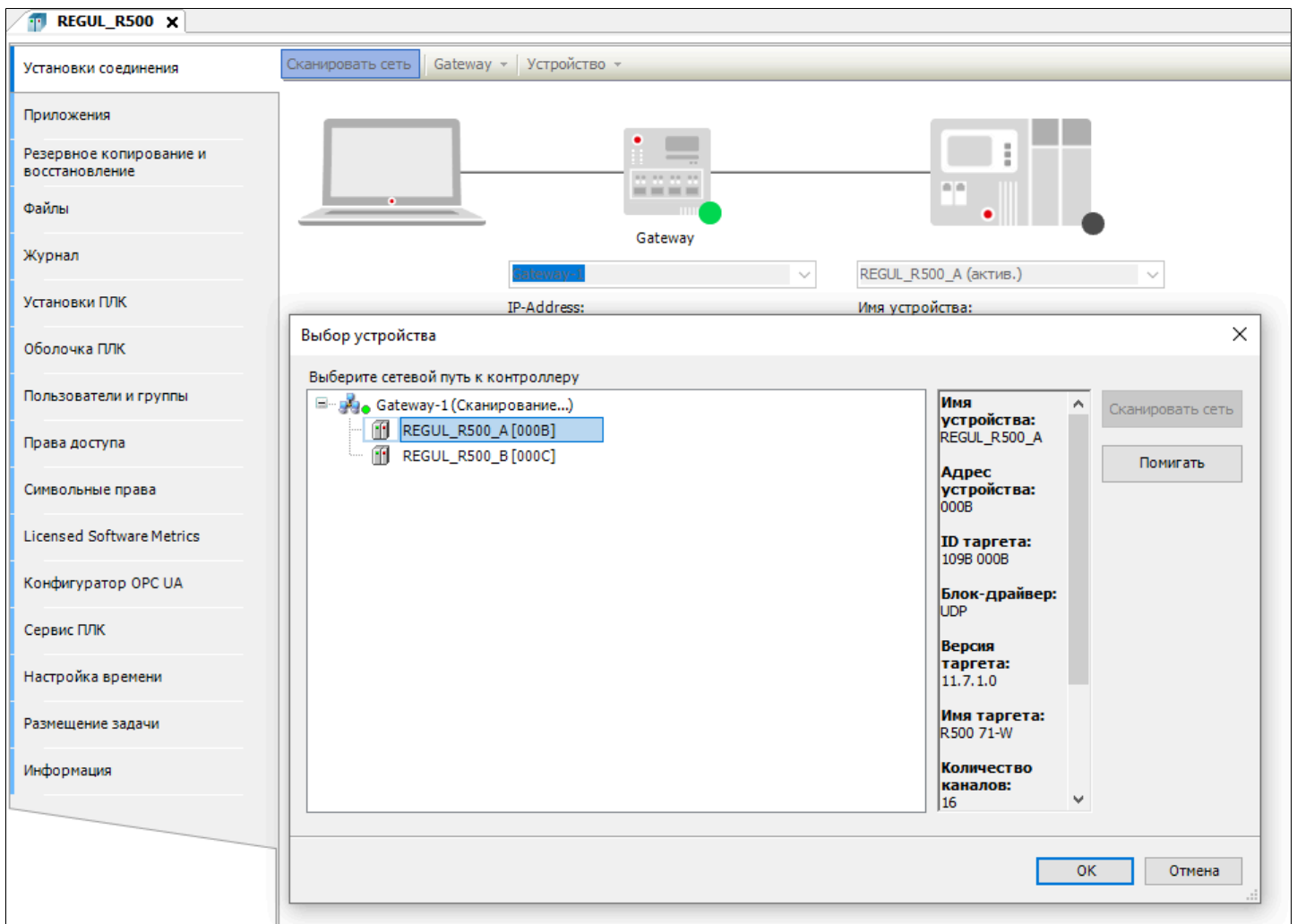
7. Перезапустите службы "Astra.Domain.Agent", "Astra.Net.Agent" и "Astra.Historian.Server" в диспетчере задач для применения настроек.



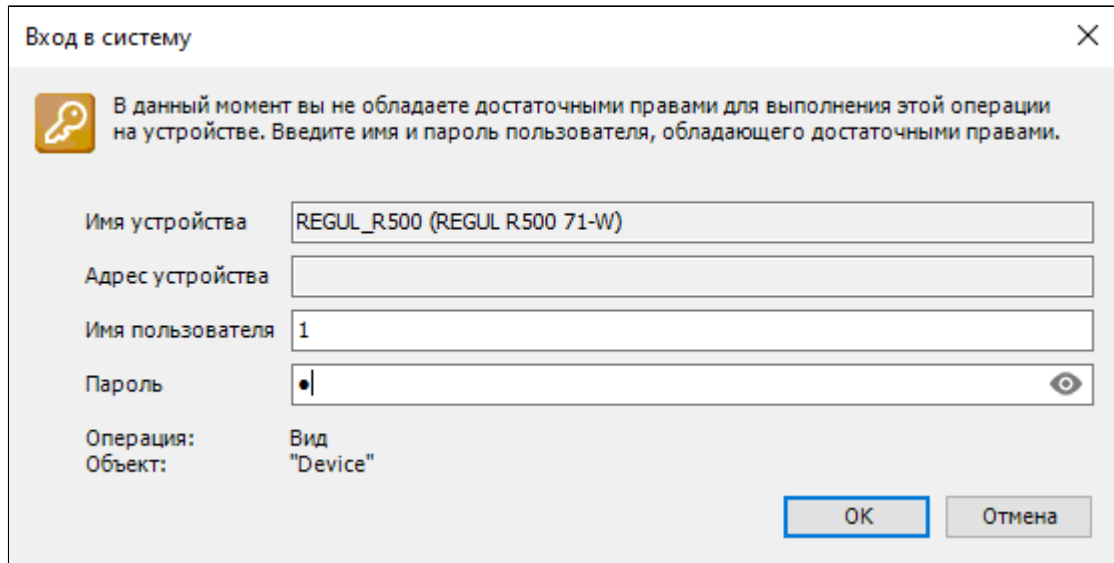
## 1.1.2.3.7. Загрузка проекта в ПЛК

Чтобы загрузить проект в ПЛК, необходимо выполнить следующие действия:

1. Скомпилируйте проект, нажав кнопку "Генерировать код(F11)".
2. Двойным кликом левой кнопкой мыши по ПЛК откройте вкладку установки соединения, нажмите кнопку "Сканировать сеть" и выберите ПЛК "REGUL\_R500\_A".

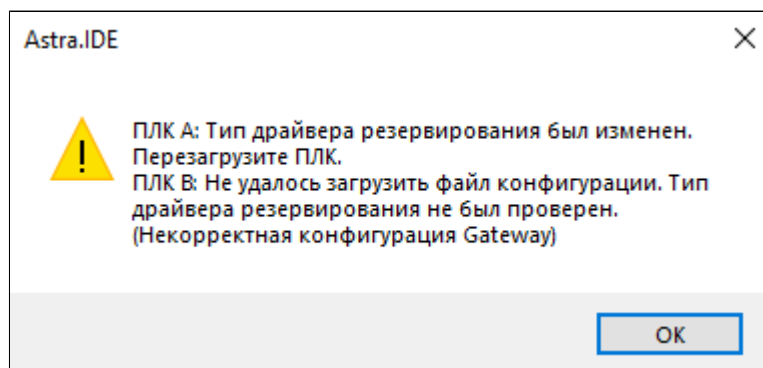


### 3. Пройдите авторизацию пользователя.

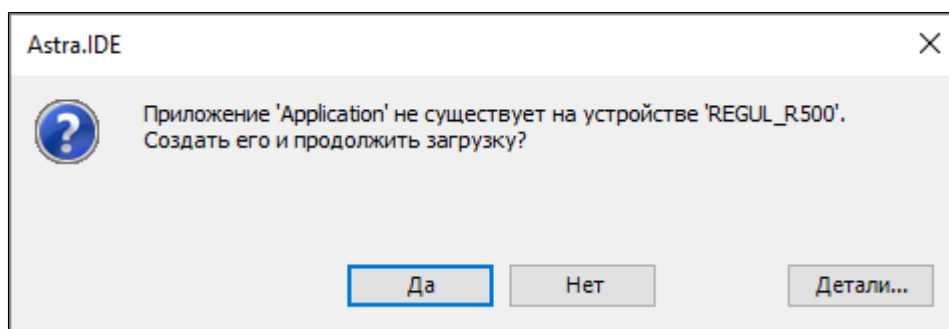


4. После успешного подключения нажмите кнопку "Логин" на панели инструментов, либо комбинацию клавиш "Alt+F8".

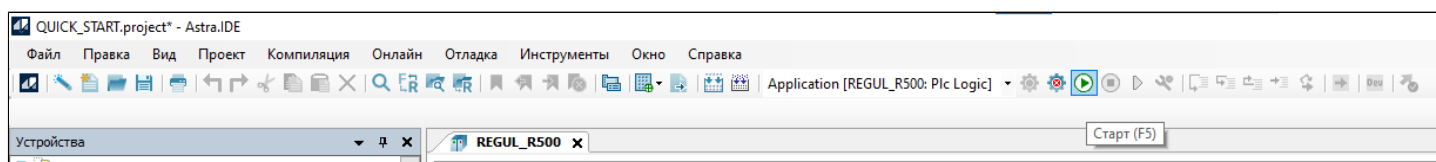
Откроется диалоговое окно, которое предупреждает о том, что тип драйвера резервирования был изменен и необходимо перезагрузить ПЛК. Нажмите кнопку "OK".



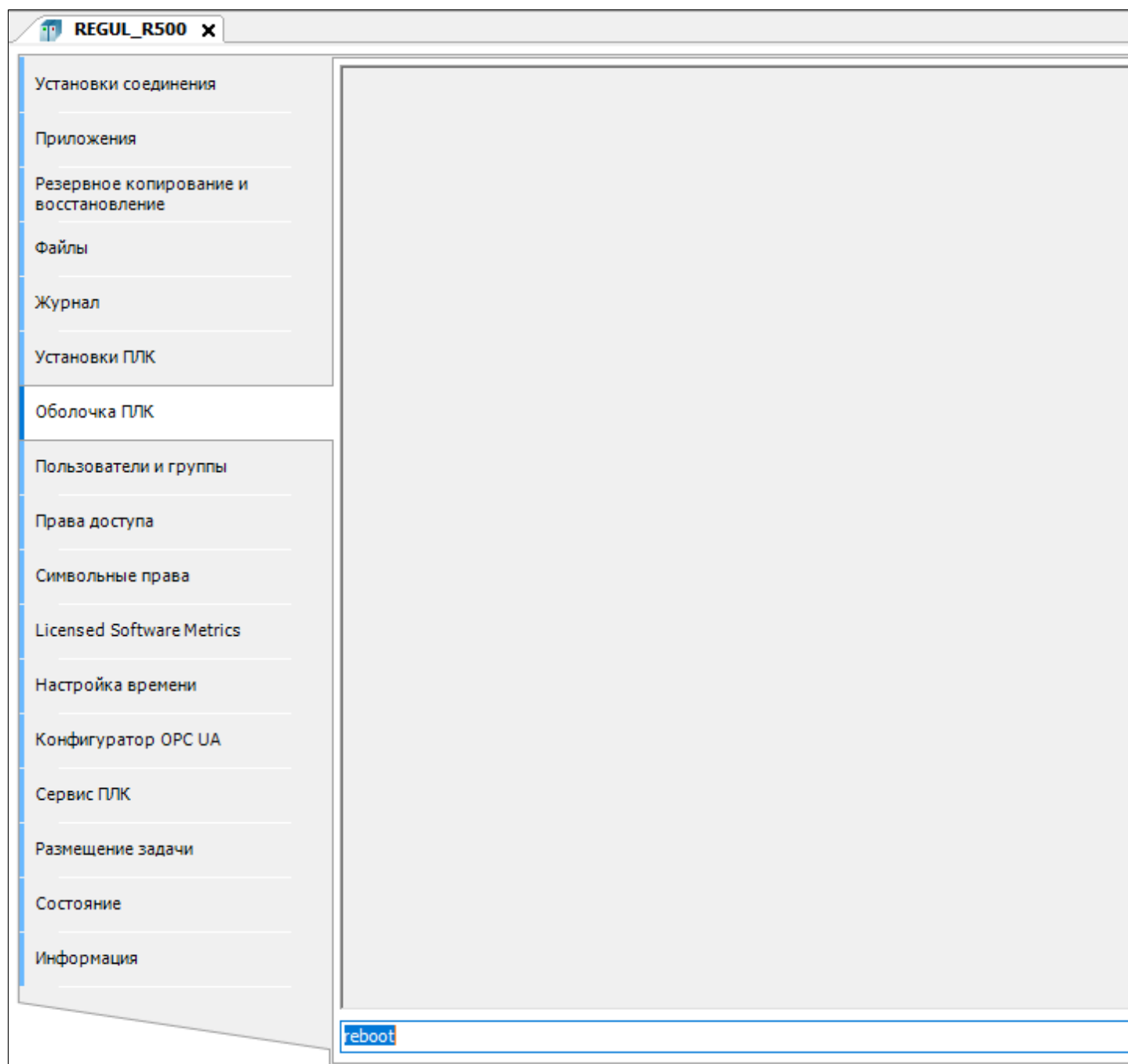
5. Далее откроется диалоговое окно, которое информирует о том, что на контроллере не существует приложения. Будет предложено создать приложение и продолжить загрузку. Нажмите кнопку "Да".



6. После завершения загрузки запустите приложение, нажав кнопку "Старт" на панели инструментов или нажмите клавишу "F5".



7. После успешной загрузки ПЛК и старта программы необходимо перезагрузить ПЛК для того, чтобы был применен актуальный тип драйвера резервирования. Для этого перейдите во вкладку ПЛК, выберите пункт "Оболочка ПЛК" и введите команду "reboot".



8. Подключитесь к ПЛК и запустите его. Перейдите во вкладку "GVL\_DIAGN" в папке "00\_Глобальные переменные" и присвойте переменным "IS\_ACTIVE" и "IS\_CPU\_A" фиксированное значение "TRUE". Для этого в поле "Подготовленное значение" введите значение "TRUE" и нажмите клавишу "F7".

Выражение	Тип	Значение	Подготовленное ...	Адрес	Комментарий
IS_ACTIVE	BOOL	TRUE			Флаг активности ПЛК
IS_CPU_A	BOOL	TRUE			Активность ПЛК А
IS_CPU_B	BOOL	TRUE			Активность ПЛК Б
RED_MODE	REDMODE	DISABLED			Режим резервирования ПЛК
SYNC_STATE	PsRedundancy_OS...				Состояние синхронизации проектов
MODULE_ST_02_012	PsDiagn.MODULE_S...				
MODULE_PP_00_011	PsDiagn.MODULE_P...				
MODULE_CU_00_071	PsDiagn.MODULE_C...				
MODULE_AI_08_041	PsDiagn.MODULE_A...				
MODULE_AO_08_011	PsDiagn.MODULE_A...				
MODULE_DI_32_011	PsDiagn.MODULE_D...				
MODULE_DO_32_011	PsDiagn.MODULE_D...				
MODULE_ST_02_022	PsDiagn.MODULE_S...				

Дождитесь полной перезагрузки ПЛК и проделайте все те же действия с ПЛК "REGUL\_R500\_B" и ПЛК серии Safety.



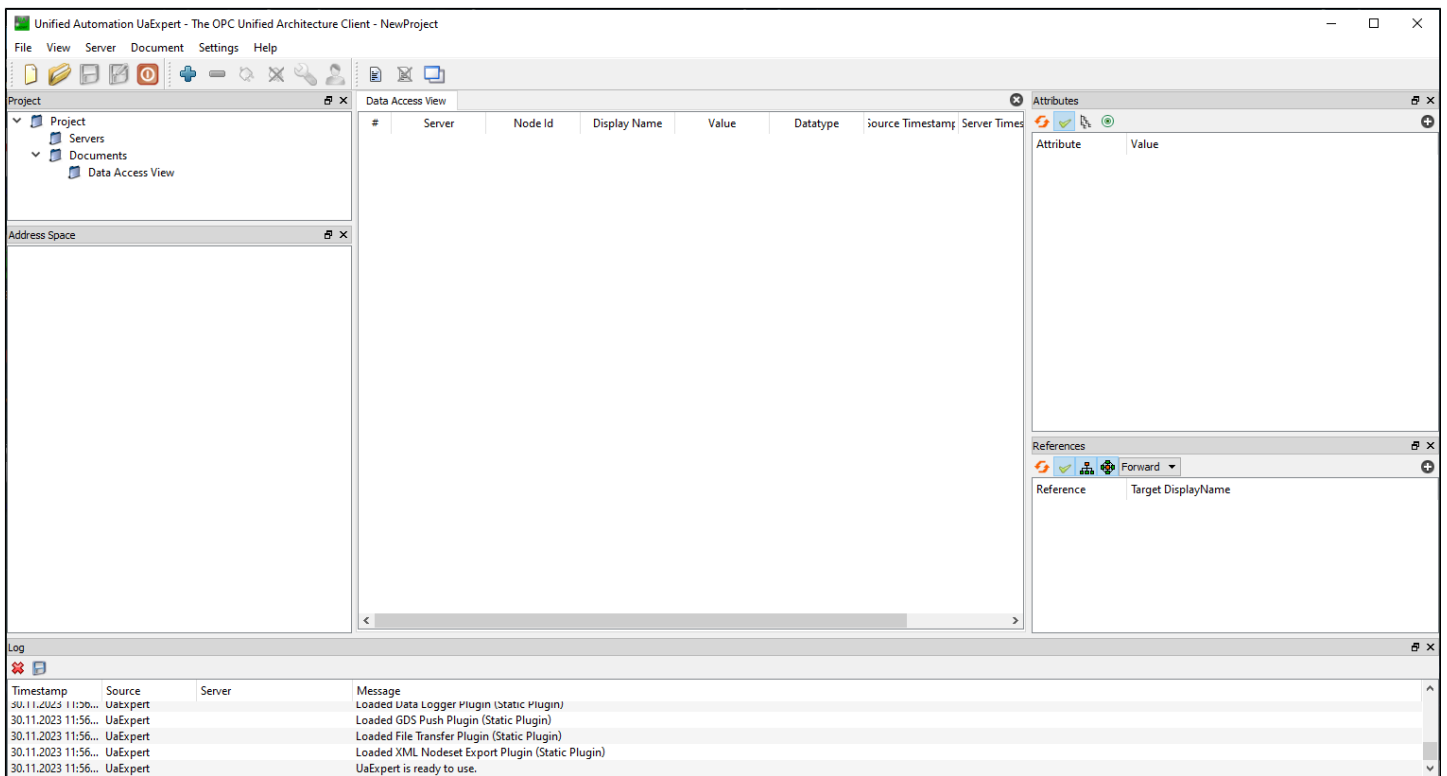
# 1.1.2.3.8. Проверка с помощью утилиты UA Expert

Чтобы проверить передачу данных по протоколу OPC UA, можно использовать утилиту UA Expert.

Установите утилиту и запустите ярлык "UaExpert" на рабочем столе.

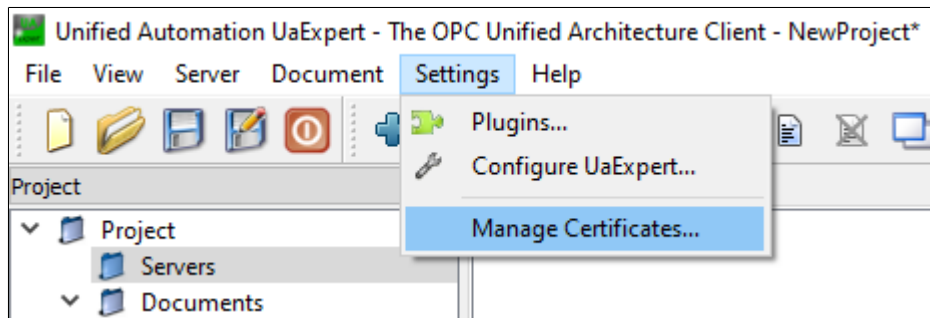


Откроется окно программы:

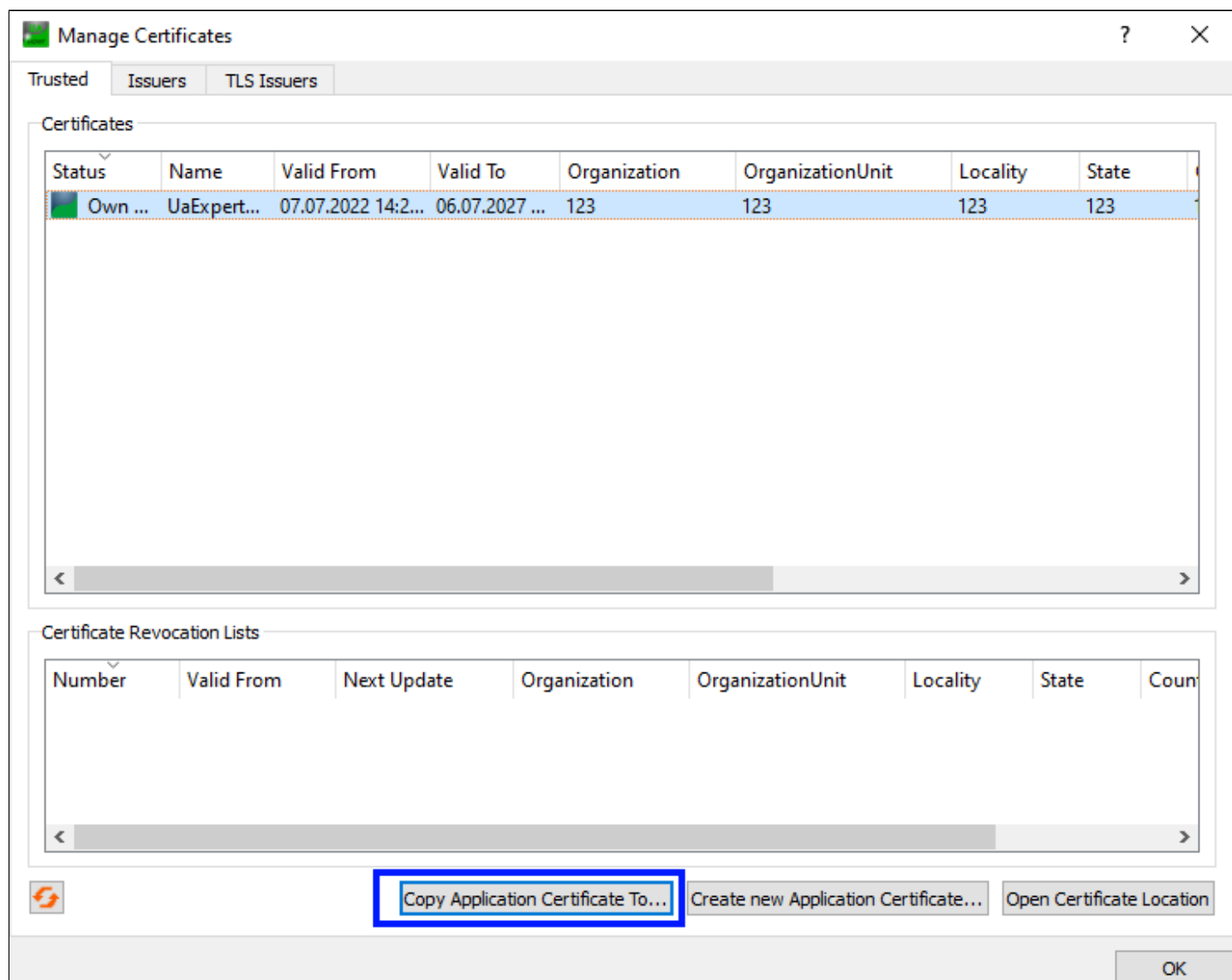


Теперь необходимо добавить сертификат утилиты в контроллер, для этого:

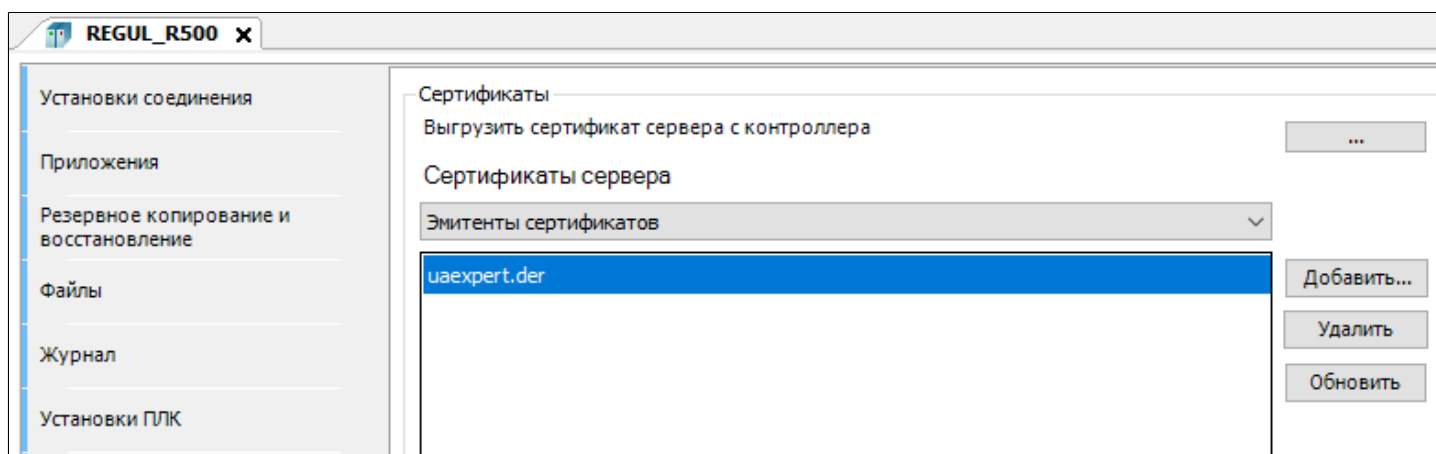
1. Откройте меню "Settings" и перейдите во вкладку "Manage Certificates".



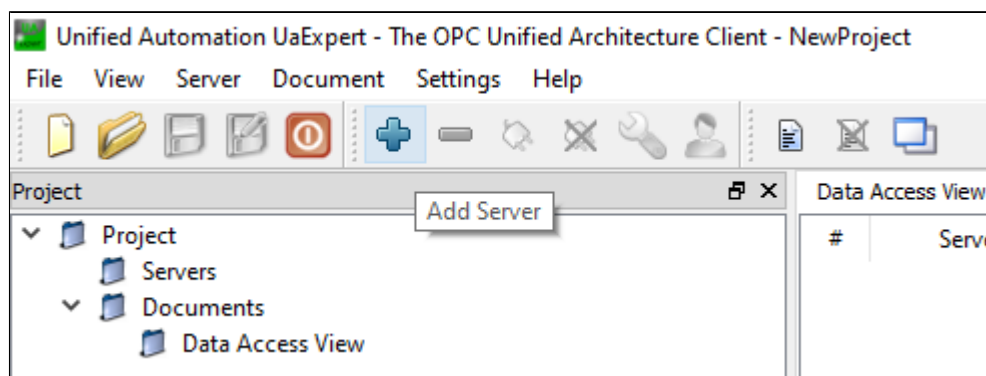
2. В открывшемся окне нажмите кнопку "Copy Application Certificate To.." и выберите папку для экспорта.



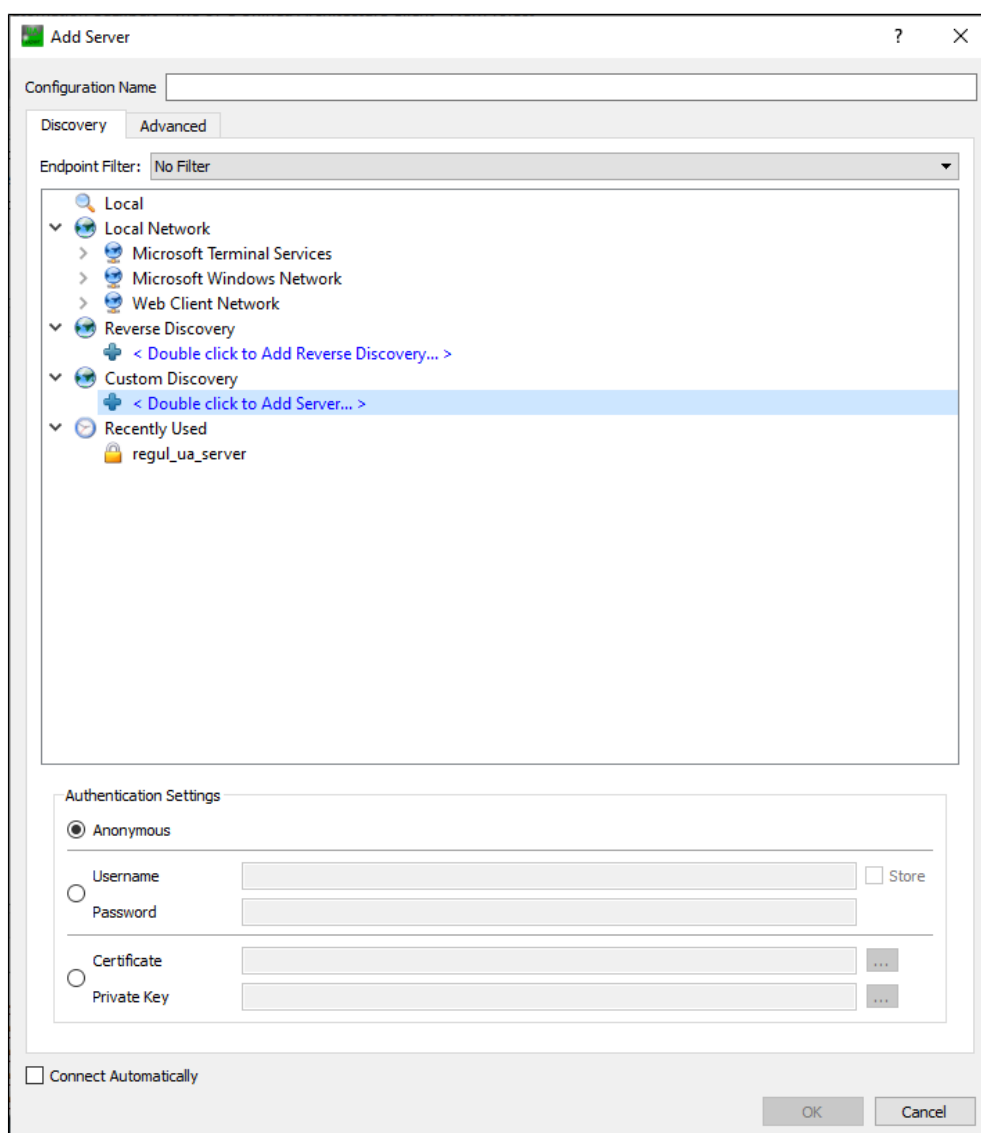
3. Откройте проект Astra.IDE и подключитесь к контроллеру. Перейдите во вкладку "Конфигуратор OPC UA" и добавьте сертификат утилиты "UaExpert" в сертификаты сервера.



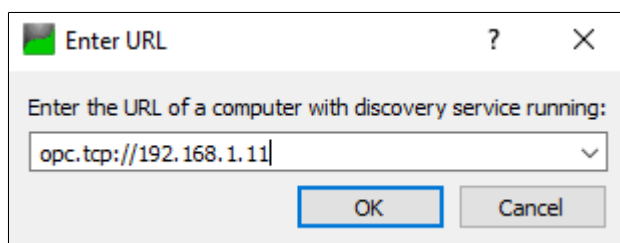
4. Перейдите в утилиту "UaExpert" и добавьте OPC UA сервер контроллера REGUL. Для этого нажмите кнопку "Add Server" на панели инструментов.



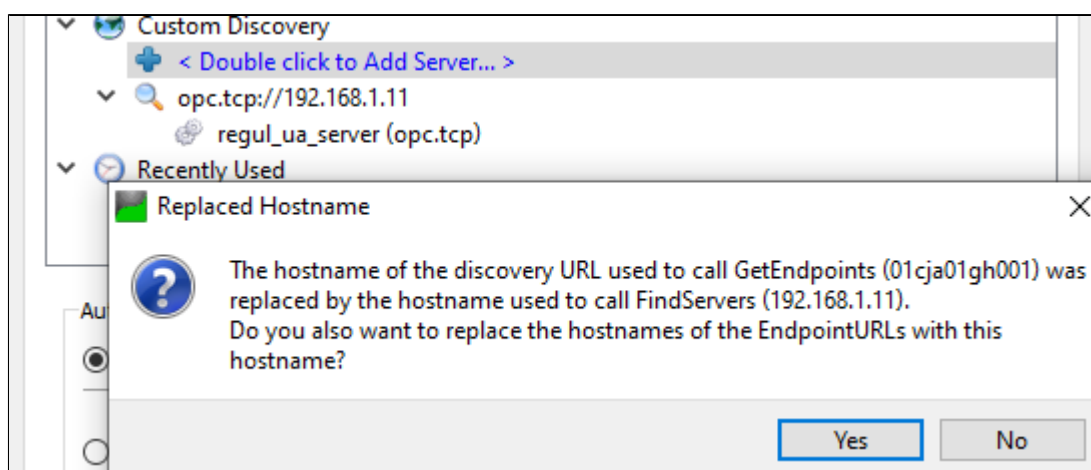
5. В открывшемся окне оставьте все настройки по умолчанию и выберите пункт "<Double click to Add Server..>".



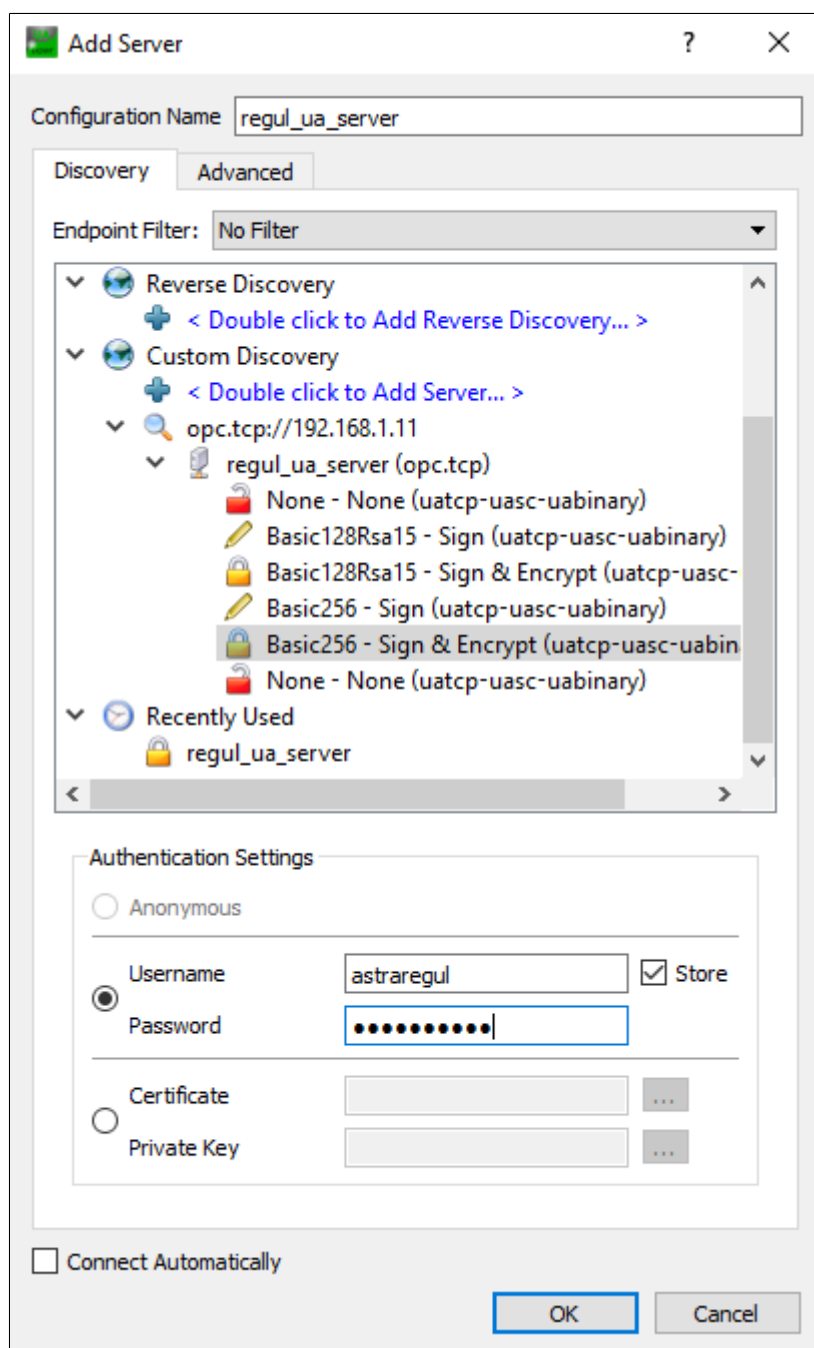
6. Задайте IP-адрес контроллера и нажмите кнопку "OK".



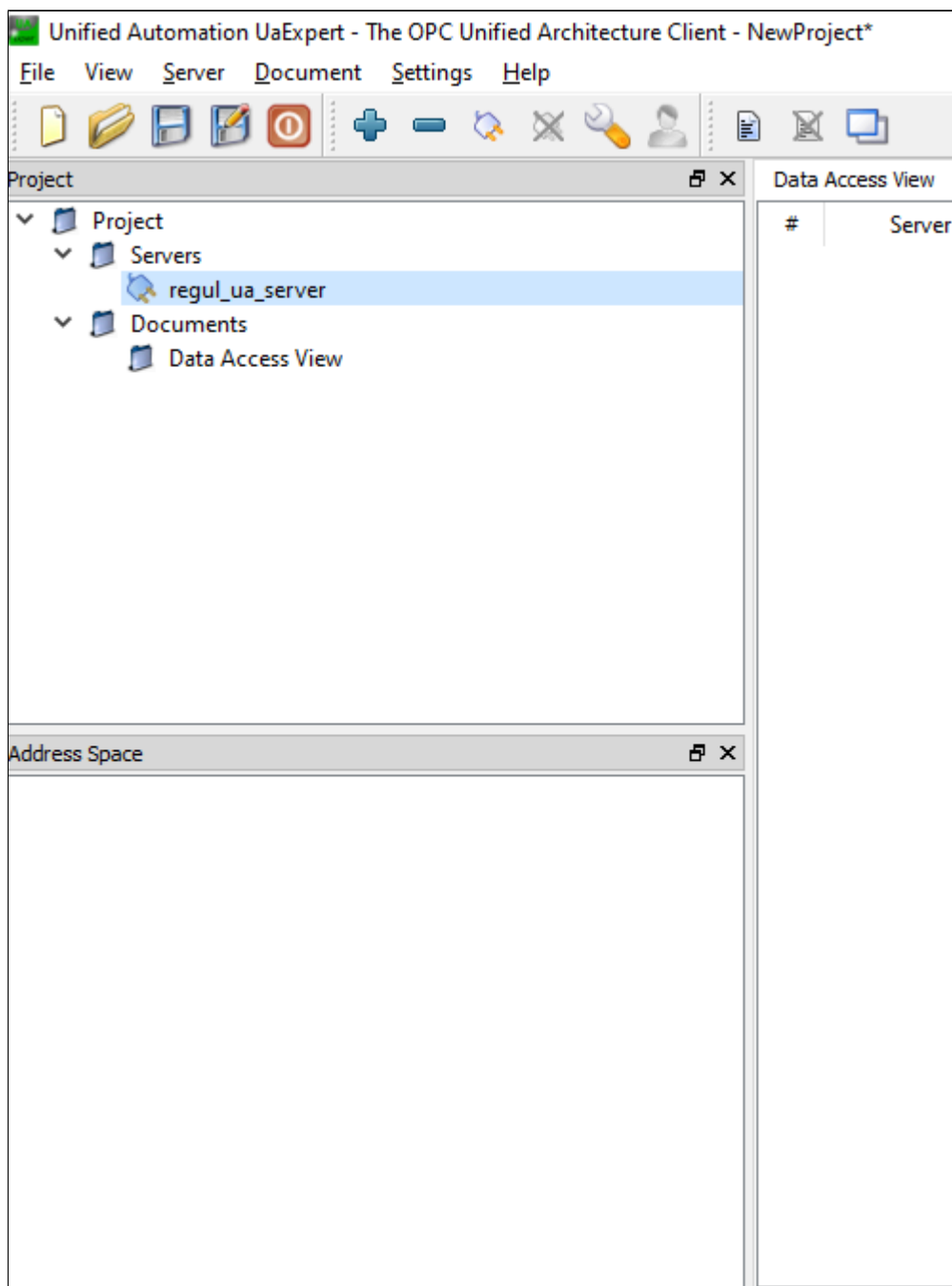
7. Сервер отобразится в списке. Раскройте содержимое сервера "regul\_ua\_server (opc.tcp)". Появится следующее всплывающее окно, нажмите кнопку "Yes".



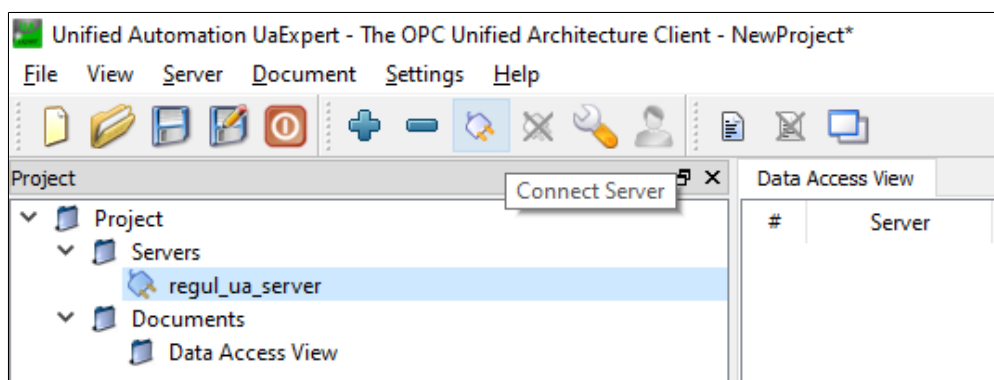
8. Выберите режим безопасности "Basic 256 - Sign & Encrypt (uatcp-uasc-uabinary)". При подключении необходимо использовать логин и пароль из раздела "Конфигурирование сервера OPC UA". Чтобы ввести пароль, необходимо установить флаг "Store". Введите логин "astraregul" и пароль "astraregul", нажмите "OK".



OPC UA сервер будет добавлен в дерево проекта UA Expert.

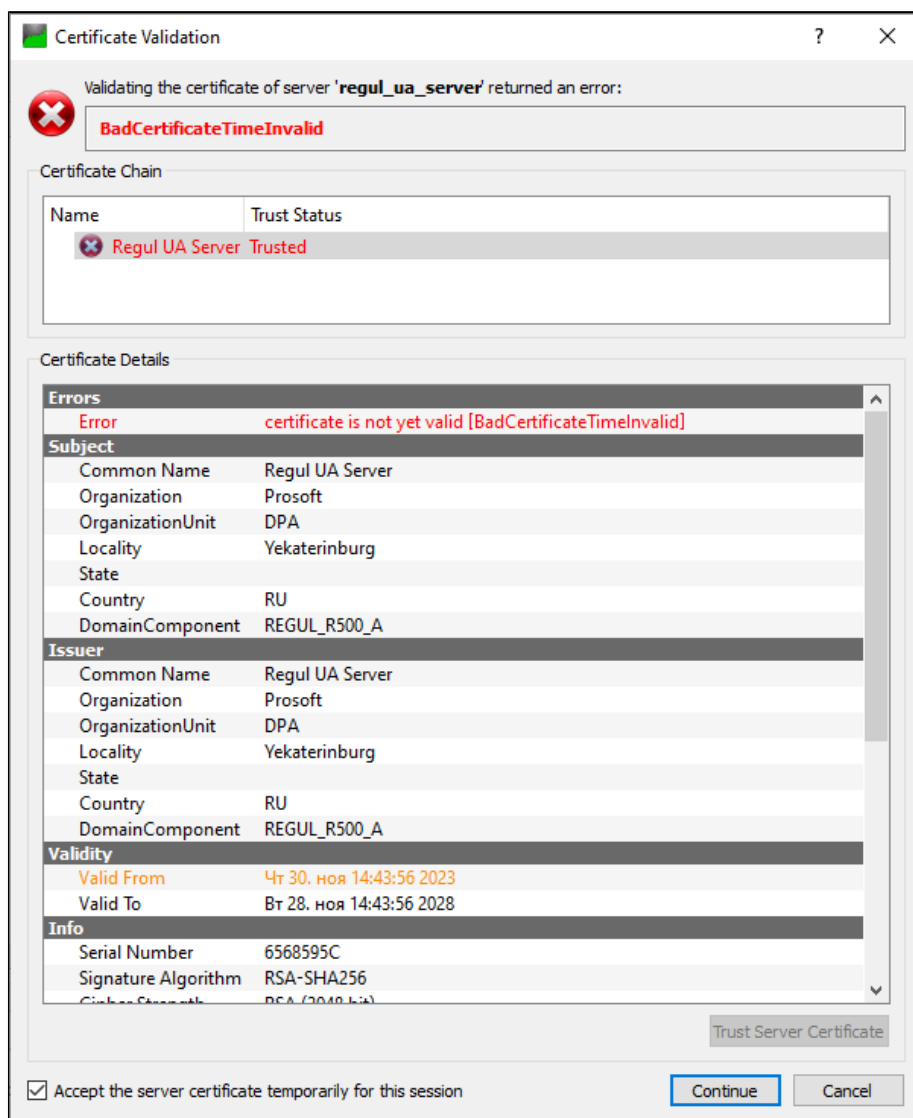


9. Выберите сервер "regul\_ua\_server" и нажмите кнопку "Connect Server".

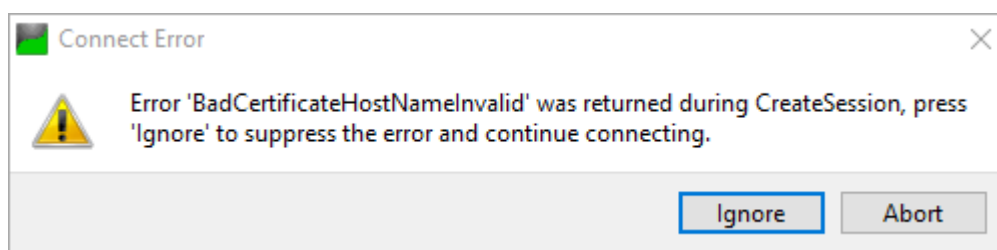




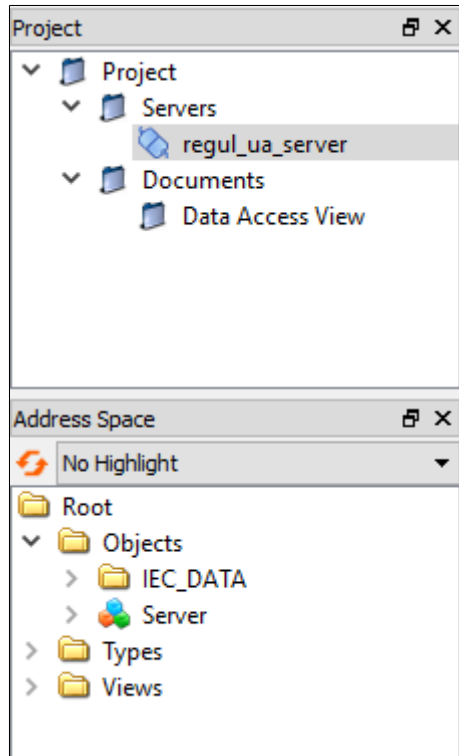
10. Появится следующее окно, установите флаг "Accept the server certificate temporarily for this session" и нажмите кнопку "Continue":



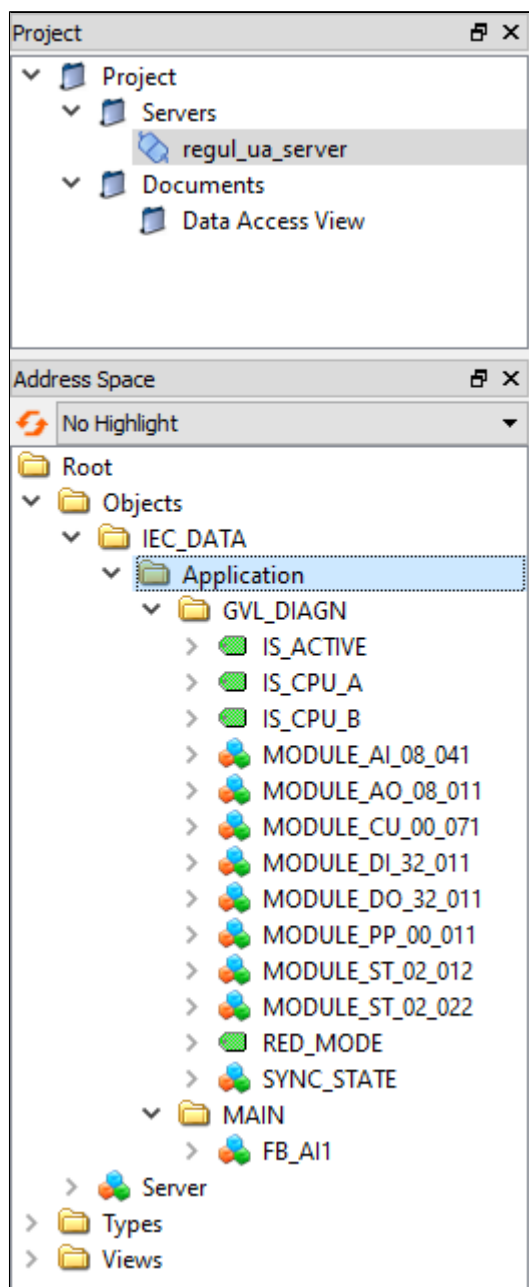
Появится следующее всплывающее окно, нажмите кнопку "Ignore":



Отобразится адресное пространство OPC UA сервера.



10. Раскройте папку "IEC\_DATA" → "Application". В адресном пространстве сервера отображаются сигналы, передаваемые на верхний уровень.



11. Выберите сигнал, чтобы посмотреть его атрибуты в окне "Attributes".

Attribute	Value
NodeId	ns=2;s=Application.MAIN.FB_AI1.DIAGN
NamespaceIndex	2
IdentifierType	String
Identifier	Application.MAIN.FB_AI1.DIAGN
NodeClass	Variable
BrowseName	2, "DIAGN"
DisplayName	"" , "DIAGN"
Description	"" , ""
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeValueInvalid (0x80350000)
UserRolePermissions	BadAttributeValueInvalid (0x80350000)
AccessRestrictions	BadAttributeValueInvalid (0x80350000)
Value	
SourceTimestamp	30.11.2023 17:05:27.356
SourcePicoseconds	0
ServerTimestamp	30.11.2023 17:05:27.356
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	0 (FAULT)
Data Type	PsTechEE.ENUM_DIAGN_CH
NamespaceIndex	2

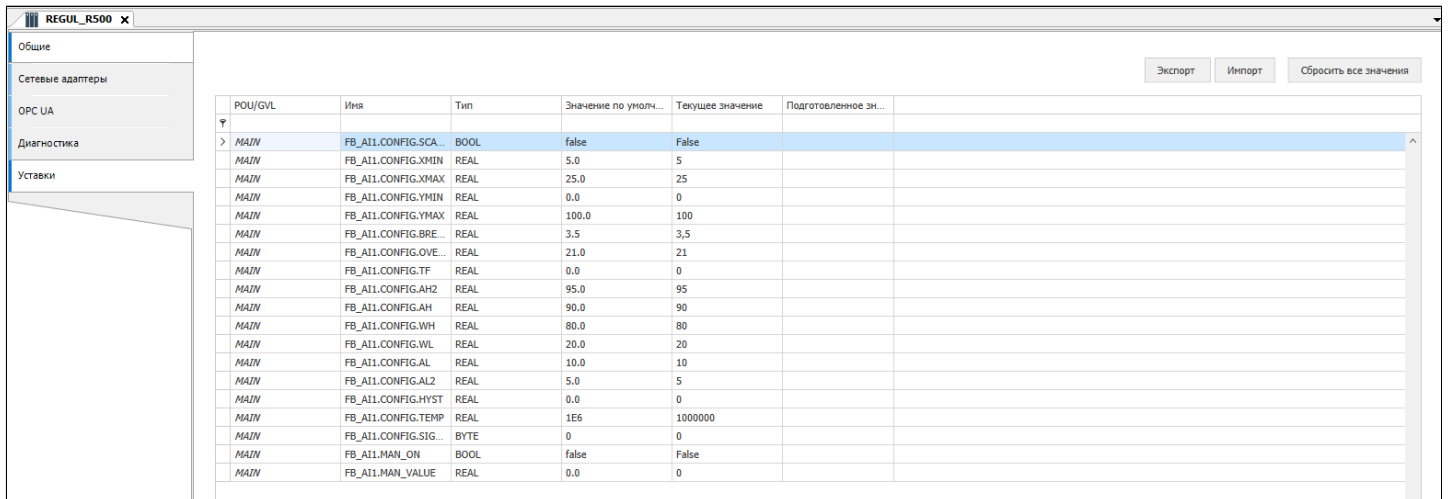
12. Чтобы узнать параметры переменной, перетащите ее в окно "Data Access View".

#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	regul_ua_server	NS2 String Appl...	DIAGN	0 (FAULT)	Int32	17:06:25.308	17:06:25.308	Good

## 1.1.2.3.9. Изменение переменных

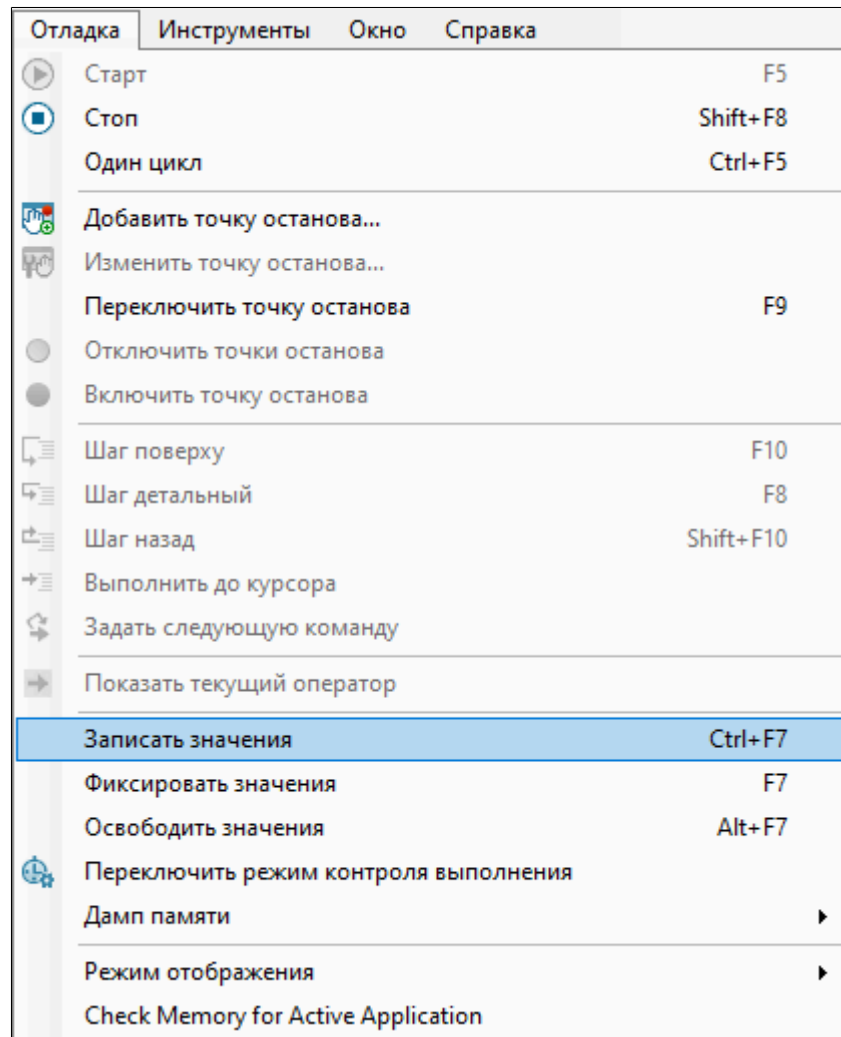
Чтобы изменить значения настроечных параметров в онлайн-режиме выполните следующие действия:

1. В онлайн-режиме перейдите на вкладку "AstraRegul" и откройте вкладку "Уставки" объекта "REGUL\_R500".



POU/GVL	Имя	Тип	Значение по умолч...	Текущее значение	Подготовленное зн...
MAIN	FB_A11.CONFIG.SCA...	BOOL	false	False	
MAIN	FB_A11.CONFIG.XMIN	REAL	5.0	5	
MAIN	FB_A11.CONFIG.XMAX	REAL	25.0	25	
MAIN	FB_A11.CONFIG.YMIN	REAL	0.0	0	
MAIN	FB_A11.CONFIG.YMAX	REAL	100.0	100	
MAIN	FB_A11.CONFIG.BRE...	REAL	3.5	3,5	
MAIN	FB_A11.CONFIG.OVE...	REAL	21.0	21	
MAIN	FB_A11.CONFIG.TF	REAL	0.0	0	
MAIN	FB_A11.CONFIG.AH2	REAL	95.0	95	
MAIN	FB_A11.CONFIG.AH	REAL	90.0	90	
MAIN	FB_A11.CONFIG.WH	REAL	80.0	80	
MAIN	FB_A11.CONFIG.WL	REAL	20.0	20	
MAIN	FB_A11.CONFIG.AL	REAL	10.0	10	
MAIN	FB_A11.CONFIG.AL2	REAL	5.0	5	
MAIN	FB_A11.CONFIG.HYST	REAL	0.0	0	
MAIN	FB_A11.CONFIG.TEMP	REAL	1E6	1000000	
MAIN	FB_A11.CONFIG.SIG...	BYTE	0	0	
MAIN	FB_A11.MAN_ON	BOOL	false	False	
MAIN	FB_A11.MAN_VALUE	REAL	0.0	0	

2. Задайте новое значение настроечных параметров в области "Подготовленное значение" и нажмите "Записать значение" ("Панель инструментов" → "Отладка" → "Записать значения") или сочетание клавиш "Ctrl +F7".



Новые значения будут записаны в соответствующие переменные.

	POU/GVL	Имя	Тип	Значение по умолч...	Текущее значение	Подготовленное зн...	
▼							
	MAIN	FB_AI1.CONFIG.SCA...	BOOL	false	True		
	MAIN	FB_AI1.CONFIG.XMIN	REAL	5.0	6		
	MAIN	FB_AI1.CONFIG.XMAX	REAL	25.0	26		
	MAIN	FB_AI1.CONFIG.YMIN	REAL	0.0	0		
	MAIN	FB_AI1.CONFIG.YMAX	REAL	100.0	100		
>	MAIN	FB_AI1.CONFIG.BRE...	REAL	3.5	3,5		
	MAIN	FB_AI1.CONFIG.OVE...	REAL	21.0	21		
	MAIN	FB_AI1.CONFIG.TF	REAL	0.0	0		
	MAIN	FB_AI1.CONFIG.AH2	REAL	95.0	95		
	MAIN	FB_AI1.CONFIG.AH	REAL	90.0	90		
	MAIN	FB_AI1.CONFIG.WH	REAL	80.0	80		

Чтобы после внесения изменений вернуть переменным значение по умолчанию, нажмите кнопку "Сбросить все значения".

[Экспорт](#)[Импорт](#)[Сбросить все значения](#)

POU/GVL	Имя	Тип	Значение по умолч...	Текущее значение	Подготовленное зн...
<i>MAIN</i>	FB_A11.CONFIG.SCA...	BOOL	false	False	
<i>MAIN</i>	FB_A11.CONFIG.XMIN	REAL	5.0	5	
<i>MAIN</i>	FB_A11.CONFIG.XMAX	REAL	25.0	25	
<i>MAIN</i>	FB_A11.CONFIG.YMIN	REAL	0.0	0	
<i>MAIN</i>	FB_A11.CONFIG.YMAX	REAL	100.0	100	
> <i>MAIN</i>	FB_A11.CONFIG.BRE...	REAL	3.5	3,5	
<i>MAIN</i>	FB_A11.CONFIG.OVE...	REAL	21.0	21	
<i>MAIN</i>	FB_A11.CONFIG.TF	REAL	0.0	0	
<i>MAIN</i>	FB_A11.CONFIG.AH2	REAL	95.0	95	
<i>MAIN</i>	FB_A11.CONFIG.AH	REAL	90.0	90	
<i>MAIN</i>	FB_A11.CONFIG.WH	REAL	80.0	80	
<i>MAIN</i>	FB_A11.CONFIG.WL	REAL	20.0	20	
<i>MAIN</i>	FB_A11.CONFIG.AL	REAL	10.0	10	
<i>MAIN</i>	FB_A11.CONFIG.AL2	REAL	5.0	5	
<i>MAIN</i>	FB_A11.CONFIG.HYST	REAL	0.0	0	
<i>MAIN</i>	FB_A11.CONFIG.TEMP	REAL	1E6	1000000	
<i>MAIN</i>	FB_A11.CONFIG.SIG...	BYTE	0	0	
<i>MAIN</i>	FB_A11.MAN_ON	BOOL	false	False	
<i>MAIN</i>	FB_A11.MAN_VALUE	REAL	0.0	0	

## 1.1.3. Astra.AStudio

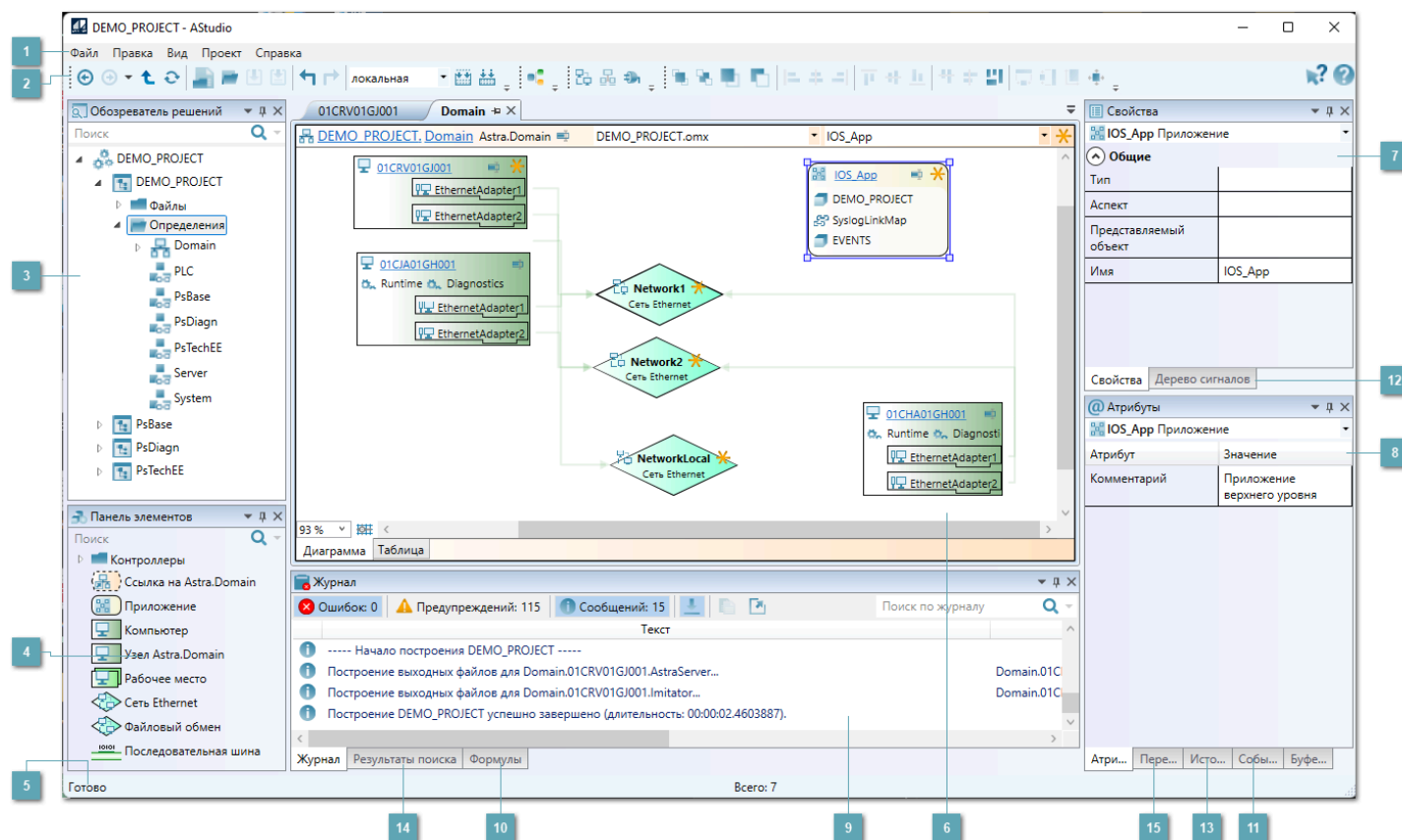
**Astra.AStudio** – программный компонент, предназначенный для конфигурирования серверов ввода/вывода Astra.Server и точек доступа Astra.AccessPoint.



Для автоматизации разработки проекта на уровне Astra.Astudio используйте плагин AstraRegul.



# 1.1.3.1. Пользовательский интерфейс



## 1 Меню

Содержит:

- Файл – действия с решением, проектами, открытие параметров приложения.
- Правка – отмена последних действий, копирование/вставка/удаление элементов, упорядочивание и выравнивание элементов.
- Вид – включение окон инструментов и панелей инструментов, переход вверх/назад/вперёд по иерархии элементов, сброс макета окон.
- Проект – компиляция, построение, развёртывание, подключение к исполняющим компонентам, поиск элементов.
- Справка – открытие встроенной справки, информации о программе.

## 2 Панель инструментов

Панель инструментов содержит группы кнопок быстрых действий:

- › Стандартная.
- › Мастеры.
- › Диаграмма.
- › Контекстно-зависимые кнопки. Состав этой группы зависит от элемента, открытого в активной вкладке.

Панели инструментов Стандартная, Мастеры и Диаграмма можно включать/отключать в меню Вид → Панели инструментов.

## 3 Окно Обзоратель решений

Окно предназначено для просмотра и редактирования структуры проекта. Чтобы открыть окно инструмента или переключиться на него, выберите инструмент в меню Вид → Обзоратель решений.

Структура проекта имеет вид дерева и содержит несколько уровней вложенности объектов.

В верхней части окна расположена строка поиска, позволяющая фильтровать содержимое дерева проекта в соответствии с искомым текстом.

## 4 Окно Панель элементов

Панель элементов предназначена для добавления новых элементов в топологию.

На панели элементов отображается список элементов, которые можно добавить в решение, в том числе типы из добавленных в проект библиотек.



В списке отображаются только те элементы, которые можно добавить в элемент, открытый в рабочей области в настоящее время.

Чтобы добавить элемент в решение, перетащите его из панели элементов в рабочую область.

Чтобы найти необходимый элемент, введите его название целиком или частично в поле поиска и нажмите клавишу "Enter".

## 5 Строка состояния

В строке состояния отображаются иконки связи с исполняющими компонентами.

## 6 Рабочая область

В рабочей области в виде вкладок открываются редакторы.

Вкладки можно разместить в отдельных областях внутри рабочей области или вынести за пределы окна приложения в виде отдельного окна.

Вкладки можно закреплять. Чтобы закрепить вкладку, нажмите иконку на изображении вкладки.

## 7 Окно Свойства

Окно Свойства предназначено для настройки свойств выделенного элемента проекта.

Каждый элемент имеет индивидуальные свойства. Для настройки свойств элемента следует выделить нужный объект и заполнить необходимые поля в окне. Элемент с заданными свойствами включается в состав проекта и автоматически связывается с контактирующими элементами. В случае ошибки задания поле соответствующего значения и элемент выделяются красной рамкой.



Чтобы открыть окно инструмента или переключиться на него, выберите инструмент в меню Вид → Свойства.

В качестве защиты от ошибок Astra.AStudio позволяет:

- › Не указывать наименование элемента. При пустом значении в качестве имени элемента будет отображаться его наименование по умолчанию.
- › Выбирать значение свойства из выпадающего списка.
- › Выбирать элемент в диалоговом окне – доступно для свойств, ссылающихся на элементы проекта.

В диалоговом окне отображаются только те элементы, которые можно указать в свойстве.

## 8 Окно Атрибуты

Окно предназначено для добавления и редактирования атрибутов. Некоторые атрибуты добавляются элементу автоматически при выполнении действий в интерфейсе и работе в других окнах.



Атрибуты, добавленные типу, при построении применяются к объектам этого типа.

## 9 Окно Журнал

Окно Журнал предназначено для отображения информации:

- › Описание – основной текст уведомлений.
- › Элемент – путь к элементу в рамках проекта.
- › Файл – полный путь к файлу, в котором определен элемент.
- › Проект – наименование проекта.

Виды уведомлений:

- › Сообщения – оповещения информационного характера.
- › Предупреждения – сообщения некритического характера, оповещают о возникновении незначительных ошибок при формировании и компиляции проекта, влекущих за собой некорректную работу отдельных его частей;
- › Ошибки – критические неполадки, возникшие в процессе сборки проекта.

Кнопки Ошибок, Предупреждений, Сообщений отражают количество каждого вида сообщений. В случае значительного количества сообщений есть возможность включить отображение только одного типа уведомлений. Для этого необходимо нажать соответствующую кнопку. Повторное нажатие кнопки восстанавливает отображение содержимого таблицы полностью.

Кнопка Автоматическое прокручивание списка исключает необходимость ручного прокручивания списка сообщений. При нажатии кнопки автоматически выполняется переход к концу списка. При добавлении в журнал новых сообщений последнее сообщение, добавленное в список, всегда остается видимым.

Комбинации клавиш:

- › Ctrl + A – выделить все сообщения в списке.
- › Ctrl + C – скопировать выделенные сообщения.

## 10 Окно Формулы

Окно Формулы предназначено для добавления в решение вычислений на языке Astra.Оm.

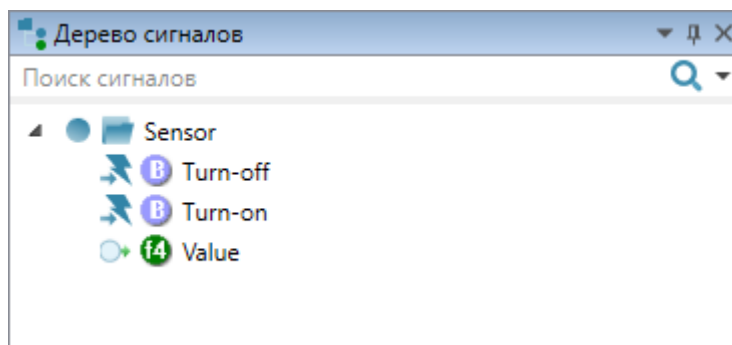
## 11 Окно События

Окно События предназначено для настройки генерации и агрегации событий.

## 12 Окно Дерево сигналов

Окно предназначено для отображения дерева сигналов, которое строится в соответствии со значениями атрибута для объектов Имя узла в OPC. При построении учитывается вложенность объектов.

Содержимое окна меняется в зависимости от выделенного объекта. При выделении узла все его дочерние объекты отображаются в виде вложенных узлов.



## 13 Окно История

Окно предназначено для настройки сохранения значений сигналов. Параметры, указанные в окне История, автоматически переносятся в атрибут сигнала "Ведение истории".

## 14 Окно Результаты поиска

Чтобы открыть окно или перейти к нему, в меню выберите Вид → Результаты поиска. Окно автоматически открывается при выполнении поиска или поиска с заменой.

В окне в виде списка отображаются результаты поиска: свойства и атрибуты, в значении которых найдена искомая строка.

Отображение отдельных параметров можно включить/отключить с помощью контекстного окна в строке заголовков списка.

Параметр	Описание
Название	Название свойства или атрибута.
Текст	Значение свойства или атрибута. Если выполнялся поиск с заменой, в поле будет указано значение до выполнения замены.
Элемент	Элемент, которому принадлежит свойство или атрибут.
Проект	Проект, в котором находится элемент.
Файл	Файл проекта, в котором находится элемент.

Доступные действия с элементами списка:

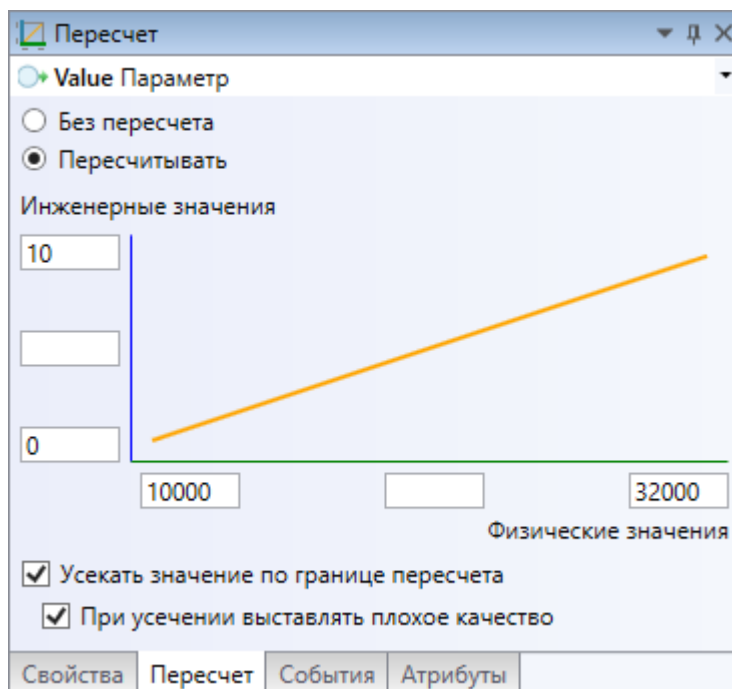
- › Переход по двойному клику к элементу списка.
- › Поиск/фильтрация элементов в списке.
- › Копирование выбранных элементов списка (Ctrl + C).
- › Вставка скопированных элементов списка в таблицу Excel.
- › Группировка результатов поиска по параметру.
- › Экспорт в файл.



В файл сохраняются все элементы списка, которые отображаются в окне в момент выполнения экспорта.

## 15 Окно Пересчет

Окно позволяет задавать свойства пересчета физических значений сигналов в инженерные и в обратную сторону.



Свойства пересчета можно задать только для элементов типа Параметр и Событие.

В зависимости от типа сигналов возможны следующие варианты:

- › Без пересчета.
- › Инвертировать.
- › Пересчитывать.

Согласно указанным значениям строится график, отражающий зависимость физических значений сигнала от инженерных и наоборот. В зависимости от присутствующих свойств сигнала выполняется линейный пересчет либо линейный пересчет с изломом.

В случае установки флага "Усекать значение по границе пересчета" результирующее значение сигнала при выходе за диапазон пересчета приравнивается к ближайшей границе пересчета.



При выходе значения за границы пересчета для сигнала может устанавливаться плохое качество.

# Редактор элемента

Под редактированием содержимого элемента следует понимать редактирование состава вложенных элементов и настройка их свойств. Отдельный элемент редактируется в собственном редакторе. Редактором называется отдельная закладка с содержимым элемента. Можно открыть одновременно несколько редакторов для разных элементов.

Переход к редактору элемента может выполняться с помощью:

- › нажатия на ссылку с наименованием элемента, расположенную в верхней части образа элемента;
- › двойного нажатия на элемент в Обзорщике решений;
- › выбора команды Перейти к определению в контекстном меню (F12).

В верхней части редактора элемента отображается информация:

- › составное имя элемента с учетом его принадлежности к верхним уровням иерархии проекта. Для разделения уровней расположения элемента в дереве проекта используются точки;
- › кнопка переименования элемента;
- › поле для навигации по файлам проекта приложения;
- › поле для навигации по элементам выбранного файла.

Содержимое элемента можно представить в двух видах:

- › [диаграмма](#) (вид по умолчанию);
- › [таблица](#).



Переключение между видами проводится в нижней части области просмотра и редактирования.

## 1.1.3.1.1.1. Диаграмма

Диаграмма элемента отражает весь состав вложенных элементов и связи между ними. Каждый отдельный элемент отображается в виде образа элемента.

Если содержимое диаграммы не помещается на экран и появляются полосы прокрутки. В этом случае перемещаться по ней можно:

- › С помощью полос прокрутки.
- › С помощью горячих клавиш:

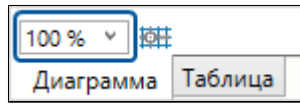
Клавиша или комбинация клавиш	Результат
Стрелки	Переместиться в направлении стрелки
PageUp	Переместиться на страницу выше
PageDown	Переместиться на страницу ниже
Home	Переместиться до конца влево
End	Переместиться до конца вправо
Ctrl + Home	Переместиться до конца вверх
Ctrl + End	Переместиться до конца вниз

- › С помощью мыши:

Действие	Результат
Прокрутка колёсика мыши	Переместиться выше/ниже
Alt + прокрутка колёсика мыши	Переместиться влево/вправо
Ctrl + зажатая левая клавиша мыши	Переместиться, захватив и переместив точку под курсором

Изменять масштаб можно:

- › С помощью поля задания масштаба внизу диаграммы.



- С помощью прокрутки колёсика мыши с зажатой клавишей Ctrl.

## 1.1.3.1.1.2. Таблица

Табличный вид редактора элемента отражает весь набор вложенных элементов в виде списка. В отличие от диаграммы, таблица позволяет одновременно увидеть:

- › все дочерние элементы, вплоть до самого нижнего уровня иерархии;
- › свойства и атрибуты всех дочерних элементов.

В таблице доступны следующие действия:

- › редактирование значений. Редактировать можно значения в ячейках, выделенных белым цветом. При редактировании можно использовать буфер обмена;
- › фильтрация. Фильтрация доступна в каждом столбце и проводится по значениям в строках. При фильтрации можно пользоваться выпадающим списком либо вводить собственное условие фильтра (в т.ч. и регулярные выражения);
- › навигация по элементам. Для перехода к таблице отдельного элемента следует щелкнуть по заголовку строки или нередактируемой ячейке;
- › добавление/удаление элементов (действия - аналогично диаграмме). Любой элемент добавляется в верхний уровень иерархии отображаемых элементов.

Контекстное меню элементов в табличном виде полностью совпадает с контекстным меню элементов в виде диаграмм.

# Редактор исходного кода

**Редактор исходного кода** – текстовый редактор, предназначенный для работы с исходным кодом, написанном на языке Astra.Om.

Редактор открывается в виде вкладки в рабочей области. Чтобы открыть редактор:

- › кликните по заголовку обработчика события.
- › дважды кликните по обработчику события в обозревателе решений.

Возможности:

- › подсветка синтаксиса.
- › автодополнение – при наборе имени редактор предлагает подходящие ключевые слова и имена сигналов.

# Редактор карт адресов

Редактор карт адресов предназначен для работы с картами адресов, добавленными в решение. Окно редактора карт адресов открывается в отдельной вкладке в основной рабочей области при открытии карты адресов, добавленной в решение. В окне в табличном виде отображается список сигналов того экземпляра приложения, в котором находится данная карта адресов.

Для поиска сигналов в редакторе используется фильтрация: фильтрация доступна в каждом столбце и проводится по значениям в строках. При фильтрации можно пользоваться выпадающим списком либо вводить собственное условие фильтра, в том числе и регулярные выражения.



Столбцы Сигнал, Тип и Привязка описывают сигнал и являются общими для всех карт связывания.

- › Сигнал – OPC-тег элемента.
- › Тип – тип сигнала.
- › Привязка – способ задания адреса элемента в карте адресов:
  - › не привязан – адрес не указан в карте адресов, редактировать параметры адреса нельзя.
  - › непосредственно – адрес указан в данной карте адресов.
  - › только чтение – адрес указан в карте адресов, но изменять его параметры в редакторе нельзя. Указывается, чтобы предотвратить непреднамеренное изменение адреса.
  - › унаследован – адрес задан в наследуемой карте адресов, редактировать параметры адреса нельзя.
  - › переопределен – адрес задан в наследуемой карте адресов и может быть переопределён в данной карте адресов.

Остальные столбцы представляют собой отдельные параметры адреса сигнала. Список параметров адреса зависит от карты адресов.

# Редактор карт атрибутов

Редактор карт атрибутов предназначен для работы с картами атрибутов, добавленными в решение.

Окно редактора карт атрибутов открывается в отдельной вкладке в основной рабочей области при открытии карты значений атрибутов.

Для поиска элементов в редакторе используется фильтрация: фильтрация доступна в каждом столбце и проводится по значениям в строках. При фильтрации можно пользоваться выпадающим списком либо вводить собственное условие фильтра, в том числе и регулярные выражения.



# Схема представлений

Схема представлений предназначена для отображения объекта/типа со всеми его представлениями и связями между ними.

Чтобы открыть схему представлений объекта/типа, выберите объект/тип или любое его представление и нажмите на кнопку Схема представлений на панели инструментов.



Схема представлений открывается в отдельной вкладке рабочей области. Название вкладки – имя исходного объекта/типа, для которого открыта схема представлений.

На схеме представлений отображаются:

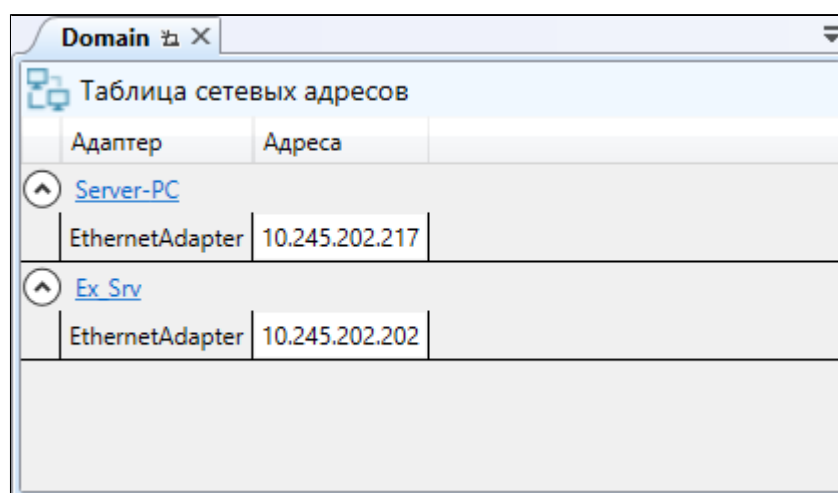
- Аспекты, в которых представлен объект/тип – отображаются в виде вертикальных регионов;
- Исходный объект/тип. Имя исходного объекта/типа отображается в заголовке вкладки;
- Представления исходного объекта/типа – объекты/типы, у которых в свойстве Представляемый объект/Представляемый тип указан исходный объект или его представление. Каждое представление отображается в регионе своего аспекта (представления без аспекта отображаются в левой части в неаспектном регионе);
- Связи – направлены от объекта/типа, содержащего ссылку, к объекту/типу, на который эта ссылка указывает.

Цвет связи:

- голубой – между ссылкой и типом;
- золотой – между ссылкой и объектом;
- оранжевый – между ссылкой и объектом, которым она инициализирована.

# Таблица сетевых адресов

Редактор предназначен для просмотра и редактирования IP-адресов устройств в домене. В редакторе в виде списка отображаются все Ethernet-адаптеры домена.



Адаптер	Адреса
Server-PC	
EthernetAdapter	10.245.202.217
Ex_Srv	
EthernetAdapter	10.245.202.202

Чтобы открыть таблицу сетевых адресов, нажмите кнопку Таблица сетевых адресов на панели инструментов.



Таблица сетевых адресов содержит следующие столбцы:

- Адаптер – наименование сетевого адаптера;
- Адреса – сетевой адрес (адреса) устройства.

Каждому адаптеру можно задать один или несколько сетевых адресов:

- IP-адреса разделяются с помощью прямой черты |
- у источника нескольких каналов передачи данных их адреса следует разделять запятой

Сетевые адреса можно вставить из таблиц Excel. Для этого необходимо скопировать строки со значениями адресов в буфер обмена, выделить одну из ячеек, содержащую адрес в таблице адресов и нажать Ctrl + V.

## 1.1.3.1.7. Горячие клавиши

Команда	Сочетание клавиш
<b>Приложение</b>	
Создать решение	Ctrl+N
Открыть решение	Ctrl+O
Сохранить	Ctrl+S
Сохранить все	Ctrl+Shift+S
Вызвать контекстную справку	F1
<b>Решение</b>	
Компилировать решение	F7
Построить решение	F6
Прервать компиляцию/построение	Ctrl+Break
<b>Навигация в проекте</b>	
Вернуться назад	Ctrl+-
Перейти вперед	Ctrl+Shift+-
Подняться на уровень выше	Ctrl++
<b>Правка</b>	
Отменить действие	Ctrl+Z
Повторить отмененное действие	Ctrl+Y
Копировать	Ctrl+C
Вырезать	Ctrl+X
Вставить	Ctrl+V
Удалить	Delete
Найти...	Ctrl+F
Заменить...	Ctrl+H

<b>Элемент</b>	
Открыть редактор элемента	F12
Переименовать	F2
Добавить комментарий	Ctrl+*
Удалить комментарий	Ctrl+Alt+*
<b>Атрибуты</b>	
Добавить атрибут	Insert
Удалить атрибут	Delete
Перейти к определению (для типов атрибутов, созданных пользователем)	F12
<b>События</b>	
Добавить подусловие	Insert

# Настройка безопасности

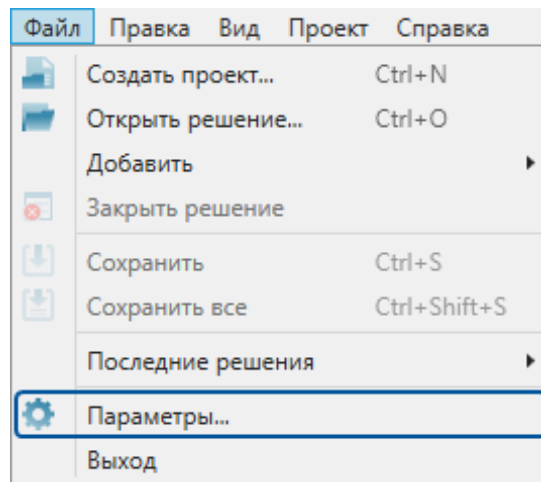
Настройка безопасности позволяет ограничить права пользователей при работе в Astra.AStudio:

- › разрешить доступ к среде разработки только авторизованным пользователям;
- › разрешить развёртывание только определённым пользователям.

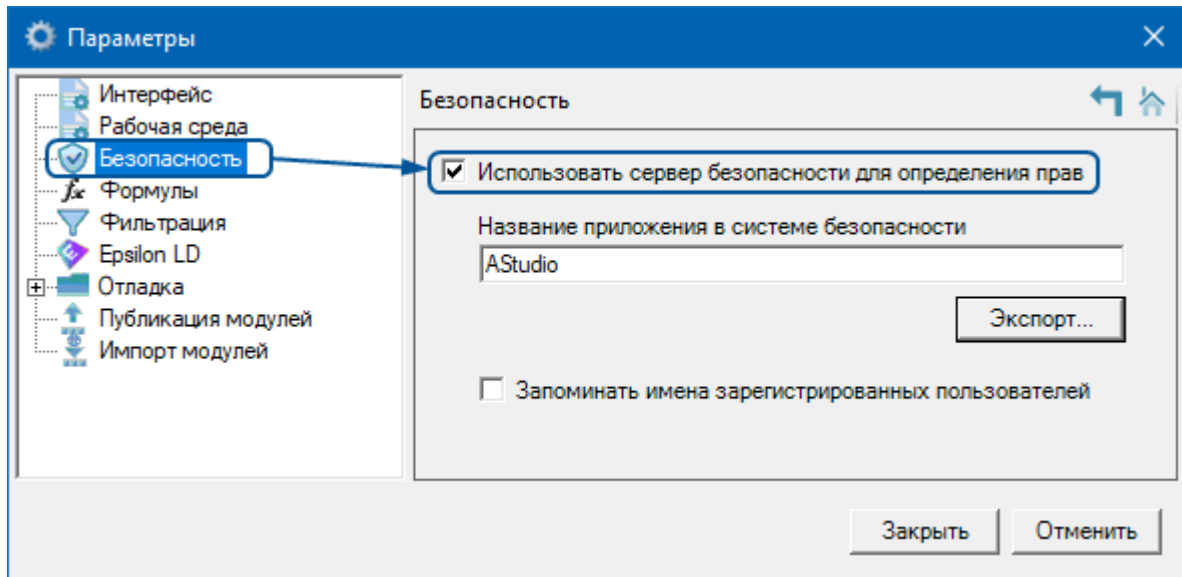
Для определения прав пользователя используется Astra.Security.

Чтобы использовать настройки безопасности:

1. Откройте параметры приложения: меню "Файл" → "Параметры".



2. На вкладке "Безопасность" установите флаг "Использовать сервер безопасности для определения прав".



3. Нажмите кнопку "Закреть" – изменения будут сохранены.

## 1.1.3.1.9. Мастер развертывания

Функции Мастера развертывания:

- › применение разработанной конфигурации;
- › подтверждение стабильности текущей конфигурации;
- › возврат к последней стабильной версии конфигурации;
- › индикация различия построенной и текущей конфигураций.

Для открытия Мастера развертывания следует нажать кнопку "Мастер развертывания" на Панели инструментов.



Основная область мастера развертывания представляет собой таблицу, в строках которой содержится перечень конфигурируемых исполняющих компонентов, описанных в решении: серверов ввода вывода и точек доступа.

Исполняющий компонент	Построенная версия	Активная версия	Стабильная версия
Domain			
ENGINEERING STATION			
<input checked="" type="checkbox"/> AstraServer	1.0.0 ↻	1.0.0.d1 от 20.06.2022 18:26:17 ✓	1.0.0.1.d3 от ↻
SERVER1			
<input checked="" type="checkbox"/> AstraServer	1.0.0 ↻	1.0.0.d1 от 20.06.2022 18:26:17 ✓	1.0.0.1.d3 от ↻
SERVER2			
<input checked="" type="checkbox"/> AstraServer	1.0.0 ↻	1.0.0.d1 от 20.06.2022 18:26:17 ✓	1.0.0.1.d3 от ↻
THIN CLIENT			
<input type="checkbox"/> AccessPoint	1.0.0 ↻	от ✓	↻
APM оператора			
<input type="checkbox"/> AccessPoint	1.0.0 ↻	от ✓ !	↻ !

Callouts: 1 points to the 'Built version' column; 2 points to the 'Active version' column; 3 points to the 'Stable version' column; 4 points to the refresh icon in the 'Stable version' column.

## Применить конфигурацию

В результате нажатия кнопки номер активной конфигурации на вычислителе будет заменен на номер версии, указанной в проекте. Кнопка недоступна, если версия построенной конфигурации и версия активной конфигурации совпадают.

## Зафиксировать стабильную конфигурацию

В результате подтверждения версия стабильной конфигурации на сервере будет заменена на номер отмеченной конфигурации. Кнопка недоступна, если конфигурация уже отмечена как стабильная.

## Ошибка связи

Нет доступа до узла домена. При наведении будет показан комментарий.

## Откатить до стабильной конфигурации

Необходимо нажать, чтобы откатиться к последней стабильной конфигурации. В результате возврата версия активной конфигурации на сервере будет заменена на последнюю конфигурацию, зафиксированную в качестве стабильной. Кнопка недоступна, если указана верная стабильная конфигурация.

Объединяющая строка, содержащая имя домена, содержит также адрес домена в сети Astra.Net. Отображаемый адрес указывается в свойствах элемента Домен. Наименования вычислителей и доменов, отображаемые в мастере развертывания, имеют вид ссылок, с помощью которых можно перейти к дигаграммам соответствующих элементов.

Описание столбцов:

- Исполняющий компонент – имя исполняющего компонента в описании домена;
- Построенная версия – номер версии разработанной конфигурации. Значение задается в свойствах проекта и обновляется после построения решения;



- Активная версия – номер текущей конфигураций, работающей на Astra.Server. Версию активной конфигурации можно выбрать в выпадающем списке, доступном при нажатии на поле значения;
- Стабильная версия – номер конфигурации, отмеченной как стабильная. Стабильной считается конфигурация, не содержащая ошибок.

В зависимости от вида конфигурации для каждого вычислителя возможно выполнить следующее:

- использовать построенную конфигурацию в качестве рабочей. Для применения следует нажать кнопку (Применить конфигурацию). В результате нажатия кнопки номер активной конфигурации на вычислителе будет заменен на номер версии, указанной в проекте. Кнопка недоступна, если версия построенной конфигурации и версия активной конфигурации совпадают;
- зафиксировать стабильность активной конфигурации на сервере. Для подтверждения следует нажать кнопку (Подтвердить, что активная конфигурация стабильна). В результате подтверждения версия стабильной конфигурации на сервере будет заменена на номер отмеченной конфигурации. Кнопка недоступна, если конфигурация уже отмечена как стабильная;
- возвратиться к последней стабильной конфигурации. Для возврата следует нажать кнопку (Вернуть стабильную конфигурацию). В результате возврата версия активной конфигурации на сервере будет заменена на последнюю конфигурацию, зафиксированную в качестве стабильной. Кнопка недоступна, если указана верная стабильная конфигурация.

Операции с конфигурациями можно производить также с помощью кнопок, расположенных на панели инструментов.

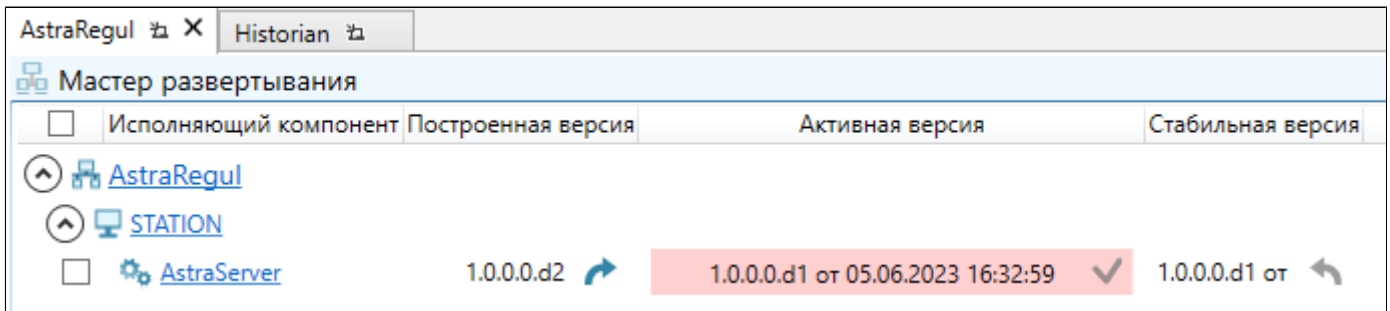


Команды кнопок аналогичны командам кнопок, расположенных непосредственно внутри таблицы мастера развертывания. Основным предназначением кнопок на панели инструментов является возможность одновременного выполнения команды для всех отмеченных вычислителей. Указание выполняется с помощью установки флага рядом с каждым

вычислителем. Для того чтобы одновременно отметить все вычислители, следует установить флаг в левом верхнем углу мастера.

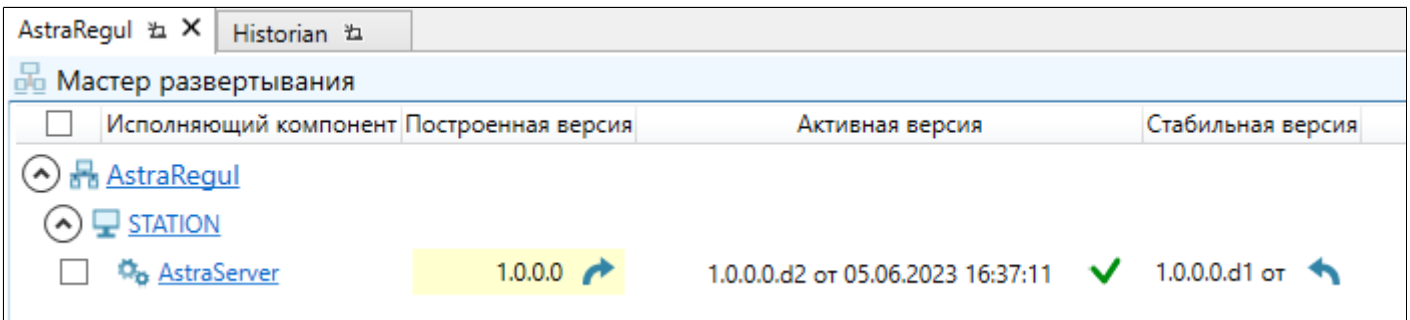
Для наглядного отображения различий рабочей конфигурации и конфигурации, сформированной в проекте, используется следующая цветовая индикация:

- › в случае если рабочая конфигурация сервера младше построенной конфигурации, номер версии активной конфигурации окрашивается в красный цвет;



<input type="checkbox"/>	Исполняющий компонент	Построенная версия	Активная версия	Стабильная версия
<input checked="" type="checkbox"/>	AstraRegul			
<input checked="" type="checkbox"/>	STATION			
<input type="checkbox"/>	AstraServer	1.0.0.d2 ↻	1.0.0.d1 от 05.06.2023 16:32:59 ✓	1.0.0.d1 от ↻

- › в случае если рабочая конфигурация сервера старше построенной конфигурации, номер версии построенной конфигурации окрашивается в желтый цвет;



<input type="checkbox"/>	Исполняющий компонент	Построенная версия	Активная версия	Стабильная версия
<input checked="" type="checkbox"/>	AstraRegul			
<input checked="" type="checkbox"/>	STATION			
<input type="checkbox"/>	AstraServer	1.0.0.0 ↻	1.0.0.d2 от 05.06.2023 16:37:11 ✓	1.0.0.d1 от ↻

- › в случае если номера версий рабочей и построенной конфигурации совпадают, цветовая индикация отсутствует.

<input type="checkbox"/>	Исполняющий компонент	Построенная версия	Активная версия	Стабильная версия
<input type="checkbox"/>	AstraRegul			
<input type="checkbox"/>	STATION			
<input type="checkbox"/>	AstraServer	1.0.0.0.d1 ↻	1.0.0.0.d1 от 05.06.2023 16:32:59 ✓	1.0.0.0.d1 от ↻



Перед развертыванием необходимо построить решение.

При развертывании для серверов ввода/вывода Astra.Server и точек доступа Astra.AccessPoint будут применены построенные для них конфигурации.

Чтобы развернуть решение, выполните следующие действия:

1. Запустите Мастер развёртывания на панели инструментов.



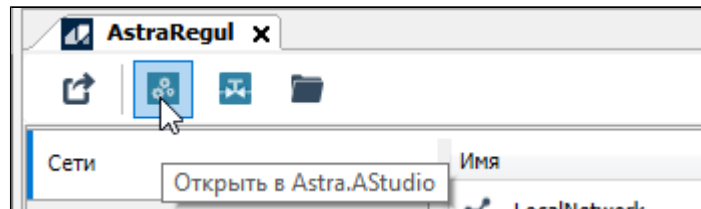
2. Выберите все компоненты ПТК, к которым вы хотите применить построенные для них конфигурации.

3. На панели инструментов нажмите кнопку "Применить все".

После выполнения развёртывания, выбранные компоненты будут сконфигурированы.

## 1.1.3.2. Разработка проекта

Чтобы открыть конфигуратор Astra.AStudio, нажмите кнопку "Открыть в Astra.AStudio" на панели инструментов AstraRegul в среде разработки Astra.IDE.



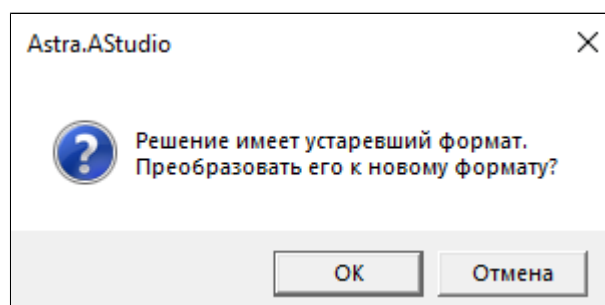
Также, чтобы запустить проект в Astra.AStudio, вы можете перейти в папку с проектом "QUICK\_START" → "AS" и дважды нажать левой кнопки мыши по файлу "QUICK\_START.solution".

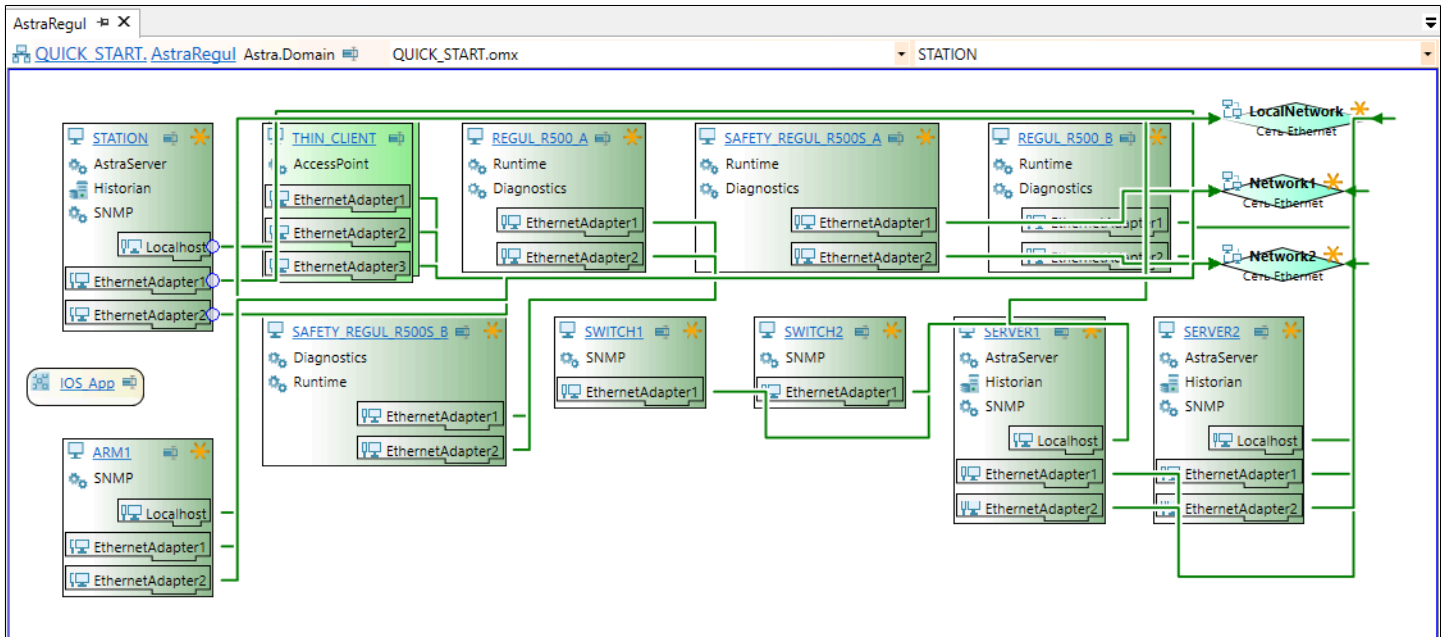
Откроется решение в Astra.AStudio с уже созданной топологией проекта автоматизации.

**Решение** – это контейнер для одного или нескольких связанных между собой проектов.

Само по себе решение не содержит данных, работа с решением – это работа с входящими в него проектами. Проект всегда открывается внутри решения, в которое он входит.

Если Astra.AStudio предложит преобразовать устаревший формат для этого проекта, нажмите "ОК".

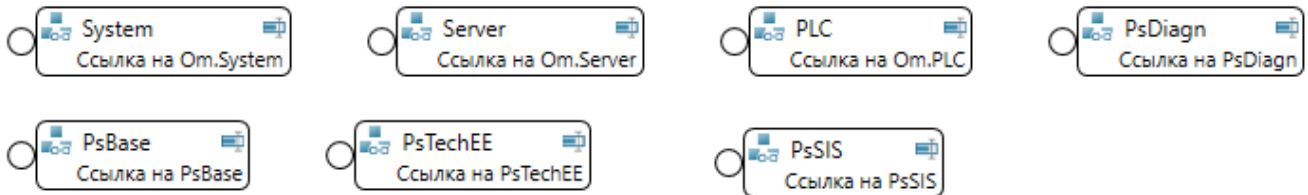




## 1.1.3.2.1. Добавление библиотек



Стандартные библиотеки ПТК добавляются к проекту автоматически при выполнении [экспорта проекта в Astra.IDE](#).



### Ручное добавление библиотек

Чтобы подключить стандартные библиотеки вручную, необходимо выполнить следующие действия:

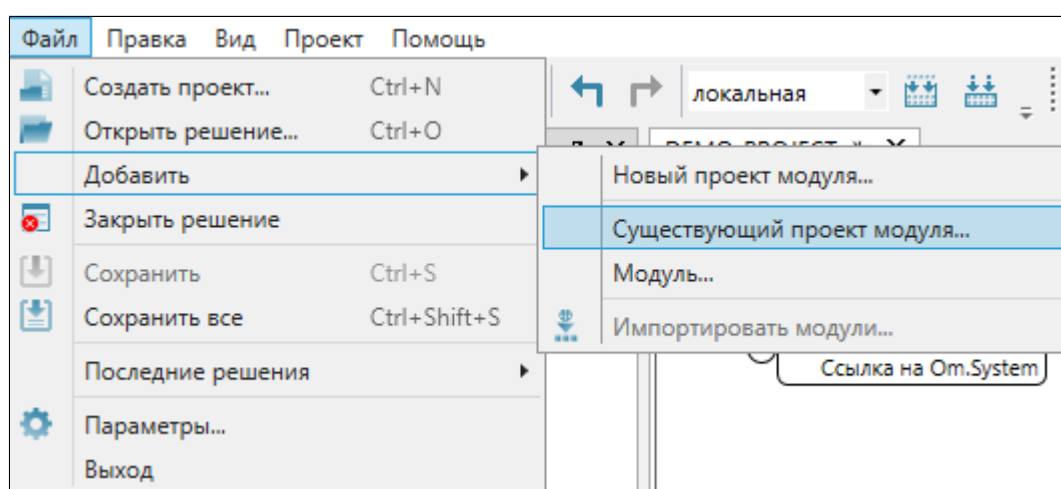
1. Создайте в каталоге с проектом папку "LIBRARIES".

Имя	Дата изменения	Тип	Размер
LIBRARIES	29.01.2024 15:23	Папка с файлами	
QUICK_START	29.01.2024 15:25	Папка с файлами	
QUICK_START.solution	29.01.2024 15:23	AStudio Solution	1 КБ

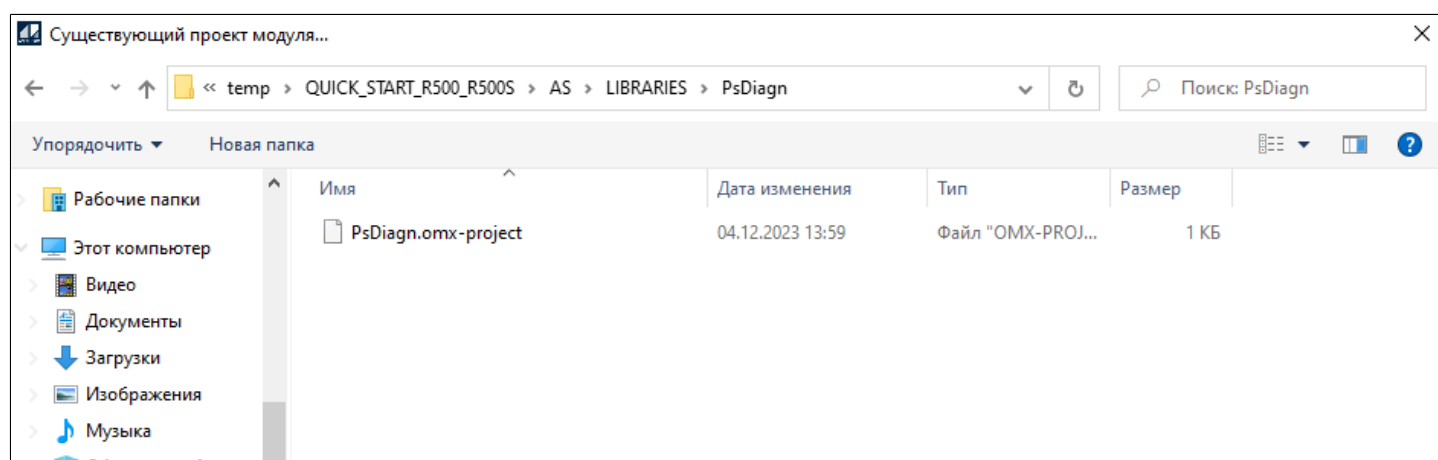
2. Поместите папки с необходимыми библиотеками в папку "LIBRARIES".

PsBase	29.01.2024 15:23	Папка с файлами
PsDiagn	29.01.2024 15:23	Папка с файлами
PsSIS	29.01.2024 15:22	Папка с файлами
PsTechEE	29.01.2024 15:23	Папка с файлами

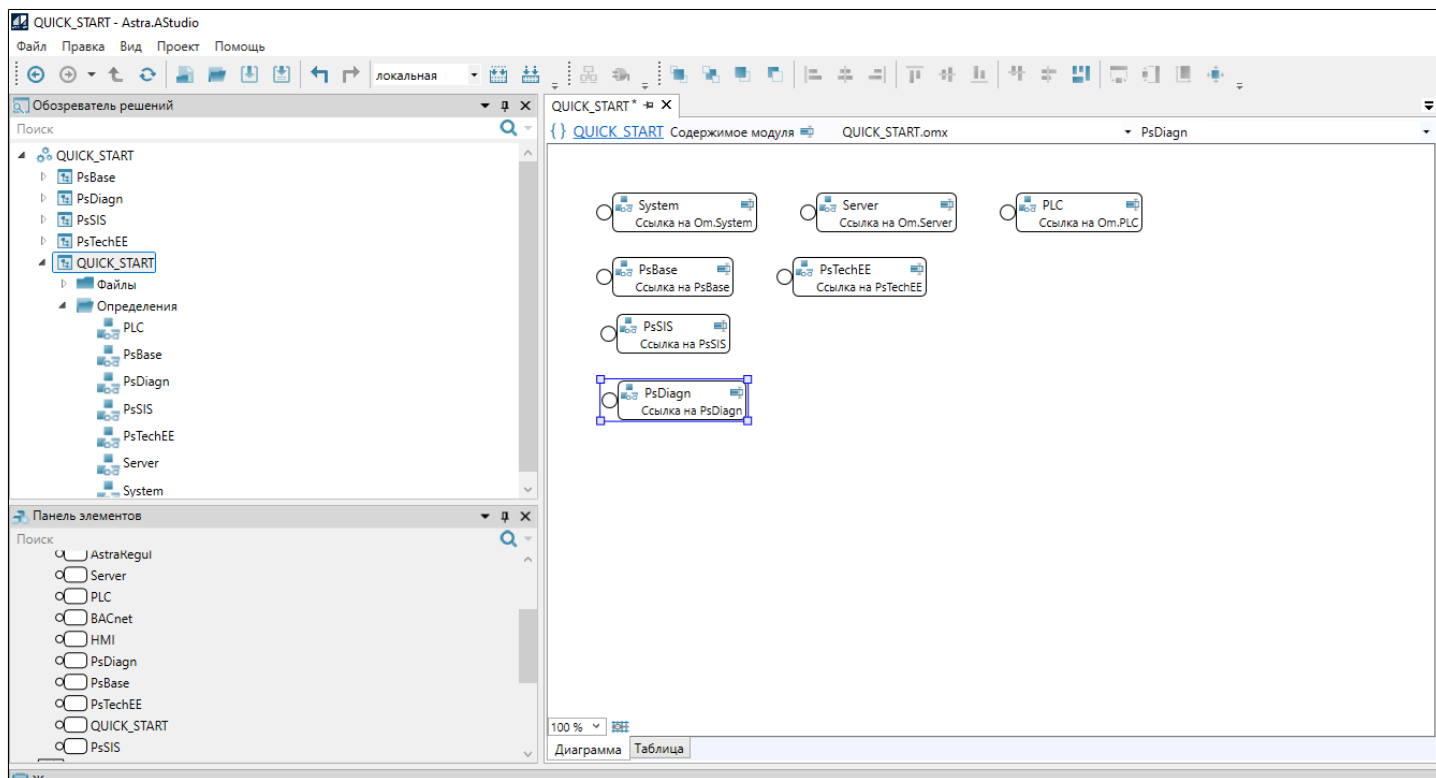
3. В меню "Файл" выберите пункт "Добавить" → "Добавить существующий проект модуля...".



4. Выберите файл с разрешением .omx-project необходимой нам библиотеки из каталога и нажмите кнопку "Открыть".



5. Перетащите модуль библиотеки "PsDiagn" из Панели элементов в рабочую область проекта.



Аналогичным образом можно подключить другие библиотеки.



## 1.1.3.2.2. Топология системы

Описание топологии системы содержит информацию о том, какие компоненты есть в домене, где они расположены и как связываются друг с другом. В описании указываются данные реальной среды исполнения, включая сетевые имена и IP-адреса компьютеров, а также интерфейсы и протоколы, по которым компоненты передают данные.

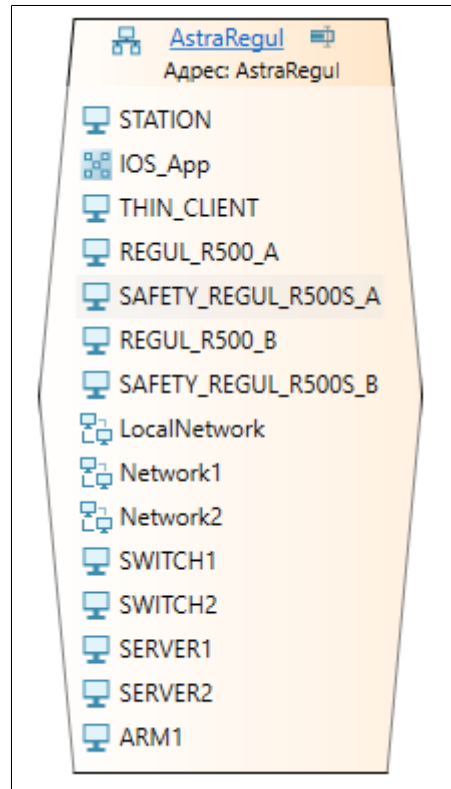
Создание топологии системы начинается с добавления в проект домена AstraRegul.

В домен добавляются компоненты, для которых создаётся конфигурация (АРМ, серверы ввода/вывода), а также компоненты и устройства, с которыми они взаимодействуют:

- › серверы истории;
- › контроллеры;
- › исполняющие компоненты.



При выполнении [экспорта проекта в Astra.IDE](#) автоматически происходит создание топологии системы в домене "AstraRegul".



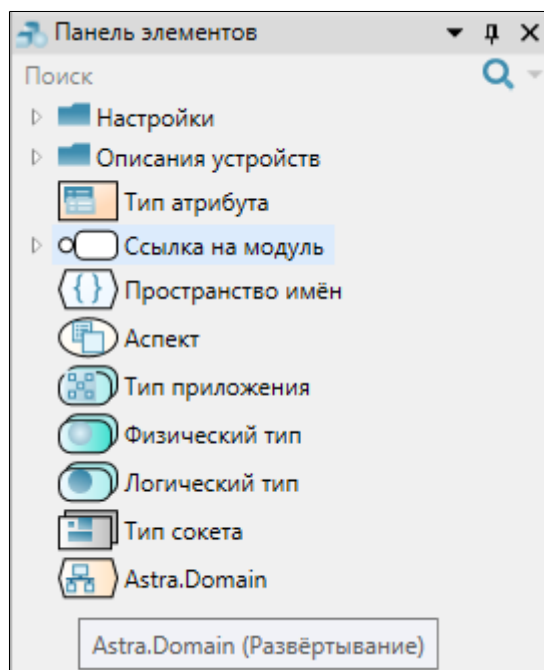
## Ручное добавление топологии системы

Чтобы добавить домен "AstraRegul" в проект вручную, выполните следующие действия:

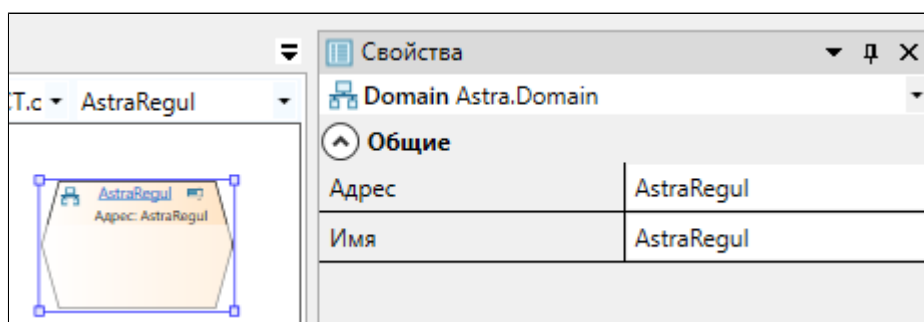
1. Добавьте элемент "Astra.Domain" из Панели элементов в проект.



Чтобы добавить элемент, перетащите его из Панели элементов в рабочую область.



2. Выберите добавленный элемент "Astra.Domain" и в свойстве "Адрес" укажите значение "AstraRegul".



3. Перейдите в элемент "Astra.Domain" для дальнейшей настройки топологии.



Чтобы перейти в элемент, кликните по его имени или дважды кликните по нему в Обзорере решений.

## 1.1.3.2.2.1. Сети

Чтобы компоненты системы могли осуществлять обмен данными, необходимо объединить их в сеть.

Есть три типа сетей:

- › [Ethernet](#)
- › [Последовательная шина](#)
- › [Файловый обмен](#)



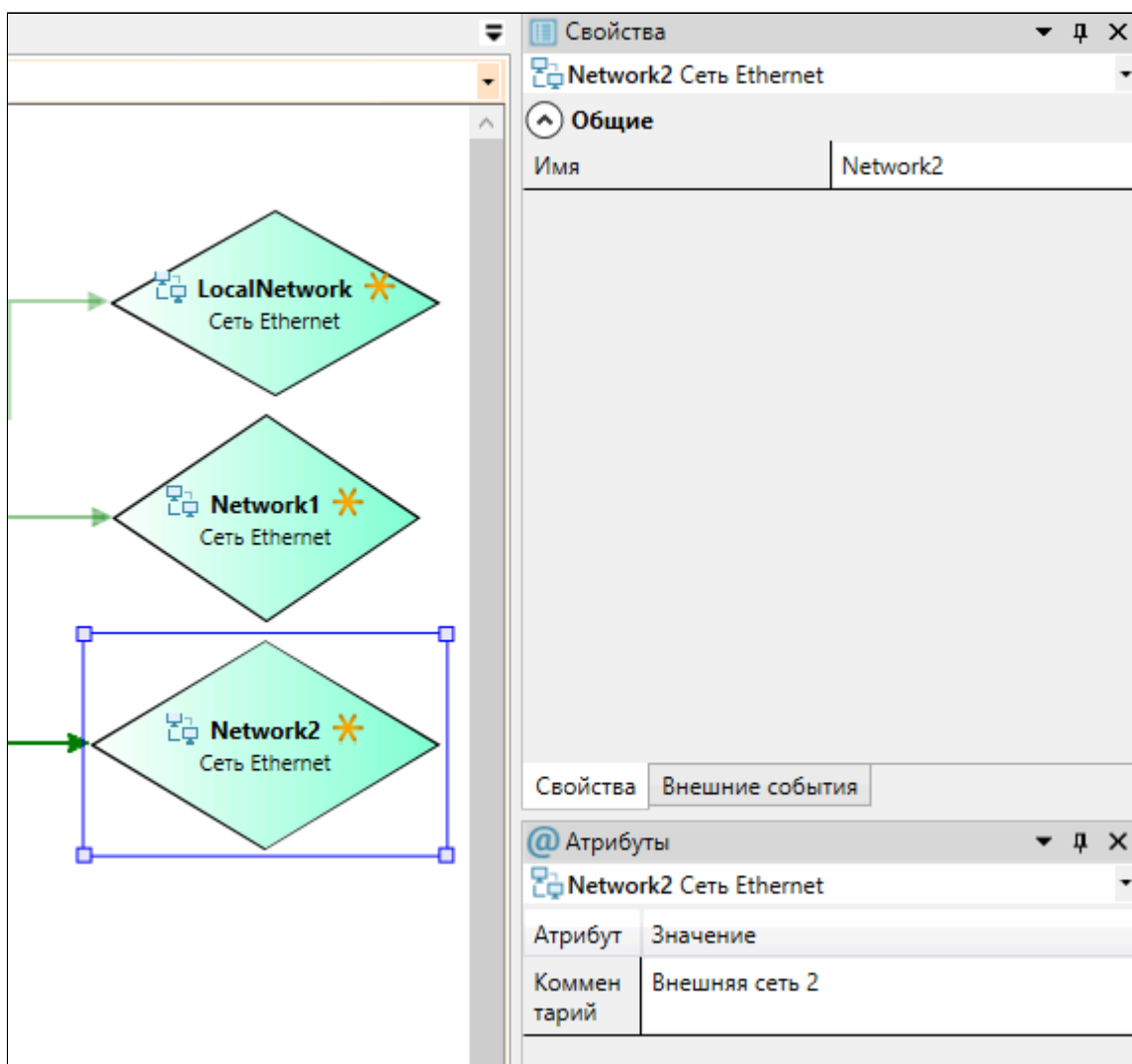
Сеть необходимо добавлять даже если все компоненты расположены на одном компьютере.

## 1.1.3.2.2.1.1. Ethernet

**Сеть Ethernet** – стандартное сетевое соединение между компьютерами, используется для передачи данных по большинству протоколов. В одну сеть включаются все компьютеры и устройства, которые имеют прямое сетевое соединение друг с другом.



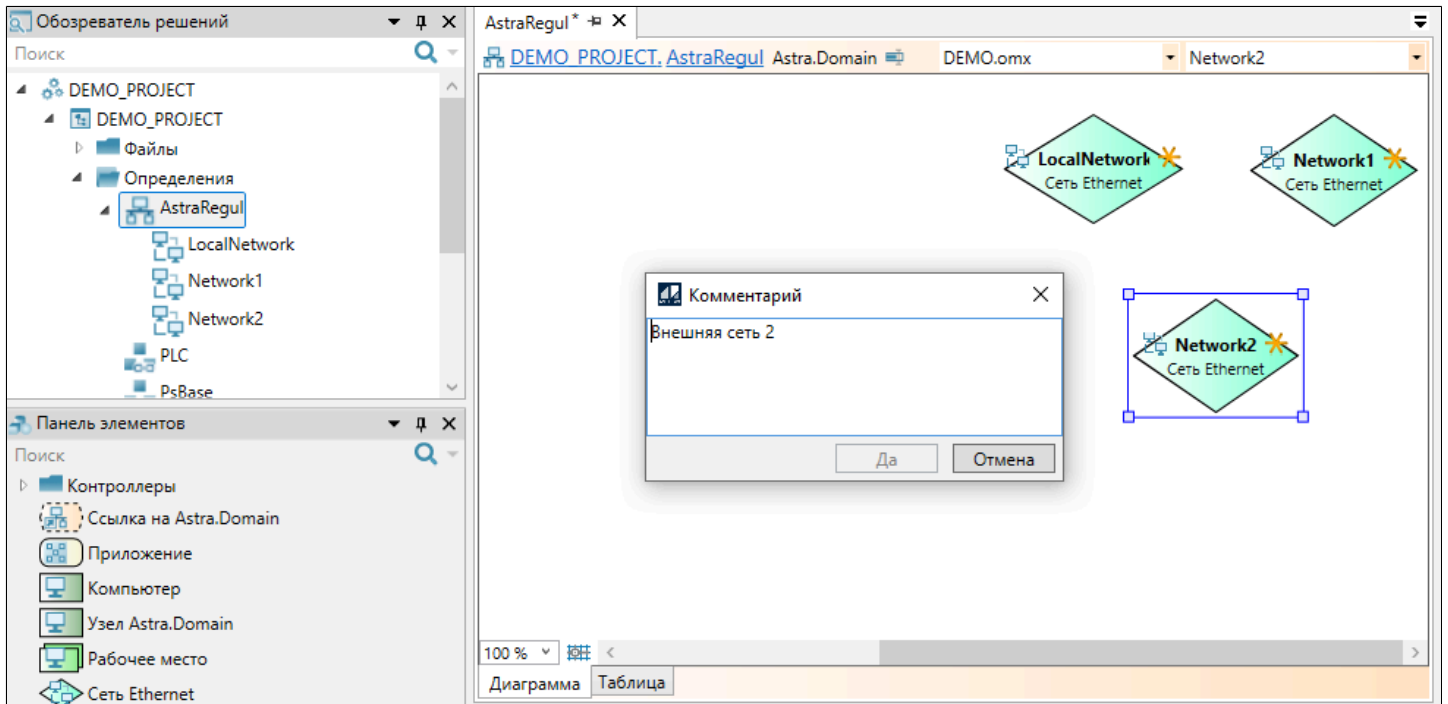
Сети Ethernet добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).



## Ручное добавление сетей Ethernet

Чтобы добавить сеть Ethernet вручную, выполните следующие действия:

1. Перейдите в домен и перетащите в рабочую область элемент "Сеть Ethernet" из Панели элементов.
2. В свойствах элемента "Сеть Ethernet" задайте название сети.
3. При необходимости добавьте комментарий у элемента "Сеть Ethernet" с помощью контекстного меню.

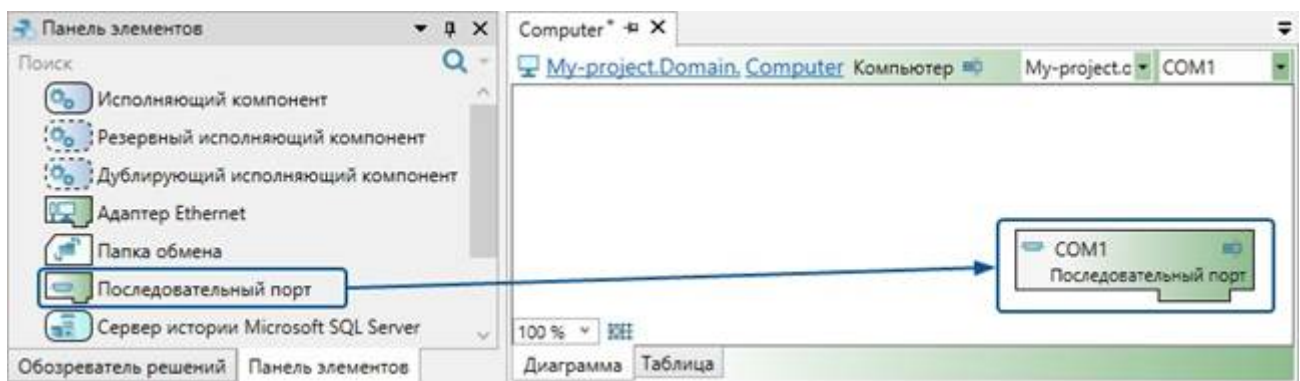


## 1.1.3.2.2.1.2. Последовательная шина

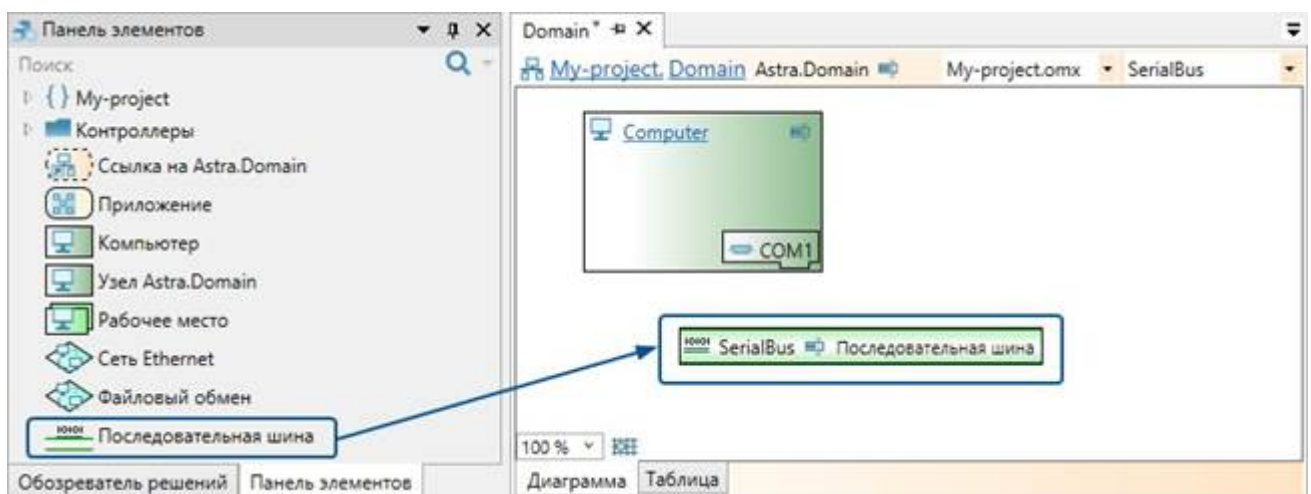
**Последовательная шина** – физический интерфейс, по которому данные передаются последовательно бит за битом, используется для передачи данных по протоколам Modbus RTU и МЭК 870-5-101. Одна последовательная шина соединяет один опросчик и одну или несколько станций, которые он опрашивает.

Чтобы добавить последовательную шину вручную, выполните следующие действия:

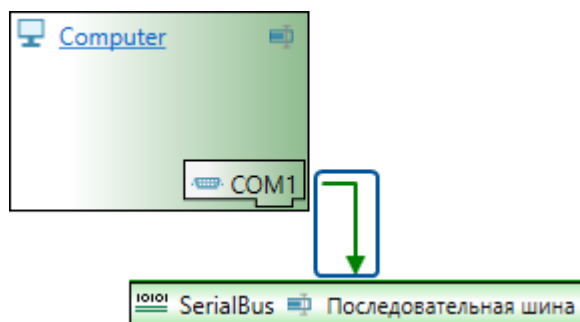
1. В компьютеры, которые должны находиться на шине, добавьте Последовательный порт.



2. В домен добавьте элемент "Последовательная шина" из панели элементов.



3. В свойствах укажите параметры шины: скорость, чётность, количество стоповых битов и способ управления потоком.



4. Соедините компьютеры с добавленной шиной.

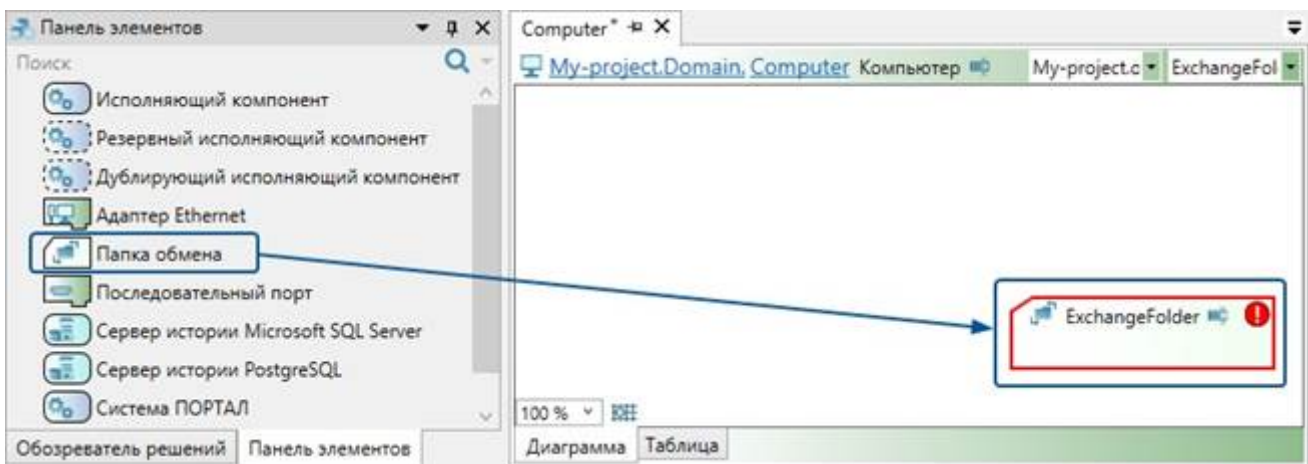


## 1.1.3.2.2.1.3. Файловый обмен

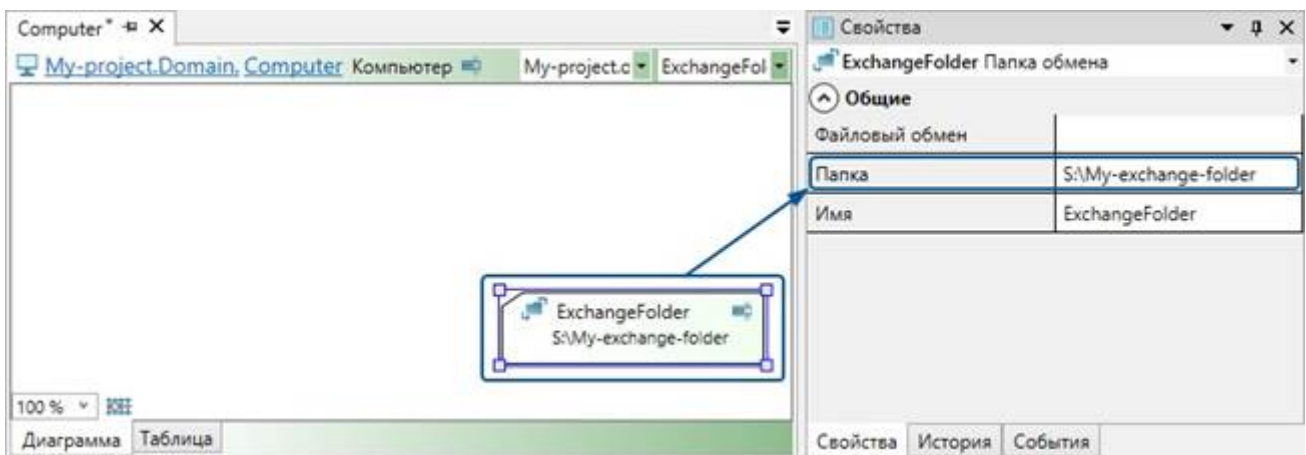
Файловый обмен соединяет пару компонентов (Astra.Server – Astra.AccessPoint или Astra.Server – Astra.Server), в которых модули TCP Server и HUB-модуль используют файловый интерфейс для обмена данными: модуль TCP Server записывает данные в файлы в сетевой папке, а модуль HUB считывает эти файлы.

Чтобы настроить файловый обмен, выполните следующие действия:

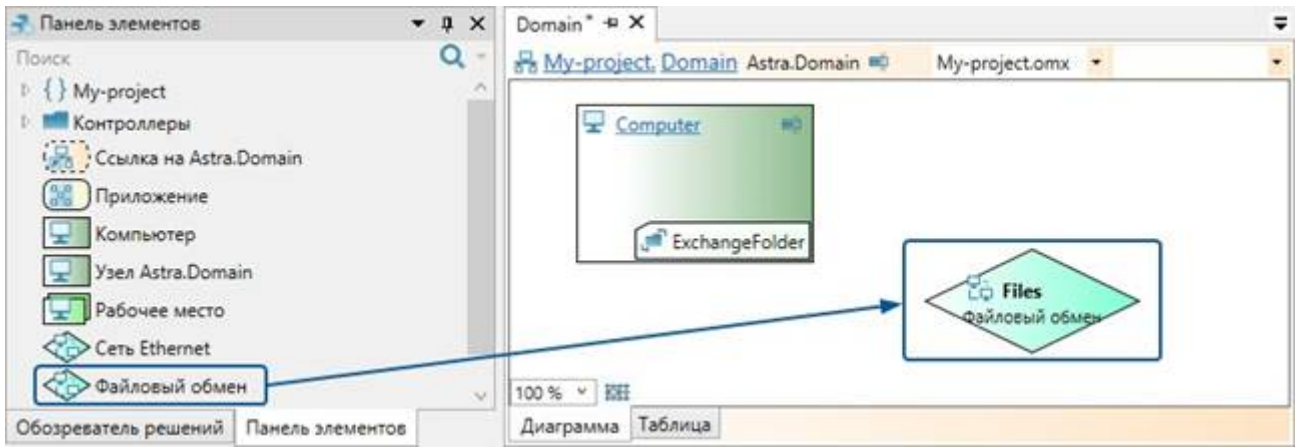
1. В компьютеры, между которыми передаются данные, добавьте элемент "Папка обмена".



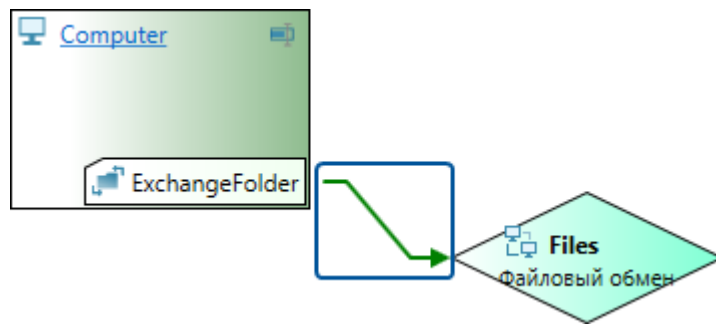
2. В свойстве "Папка" укажите путь к сетевой папке.



3. В домен добавьте элемент "Файловый обмен".



4. Соедините компьютеры с добавленным элементом.

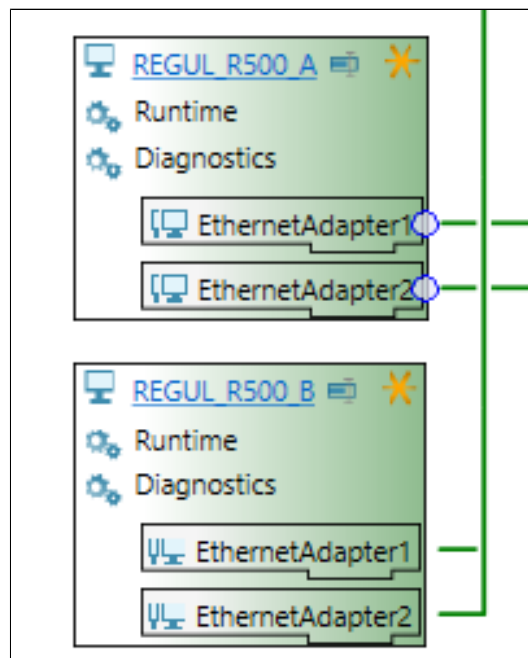


## 1.1.3.2.2. ПЛК REGUL R500

В качестве источника данных будет использоваться ПЛК REGUL R500.



ПЛК REGUL R500 вместе с адаптерами Ethernet и исполняющими компонентами (Runtime и Diagnostics) добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).



### Ручное добавление ПЛК

Чтобы описать ПЛК REGUL в топологии вручную, необходимо выполнить следующие действия:

1. Перейдите в компонент "Astra.Domain" и перетащите в рабочую область из Панели элементов компонент "Компьютер".



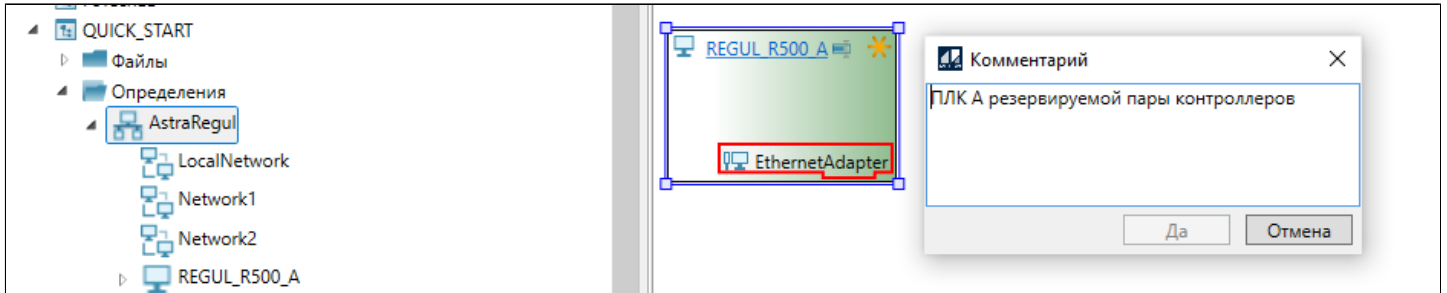
Компонент "Компьютер" - это компьютер или иное сетевое устройство (например, ПЛК), на котором нет компонентов AstraRegul, но с которыми они взаимодействуют.

2. Выделите элемент "Компьютер" и в окне "Свойства" задайте ему свойство "Имя".



Имя элемента "Компьютер" должно соответствовать структурной схеме проекта (например, KKS).

3. Добавьте комментарий у объекта "Компьютер".

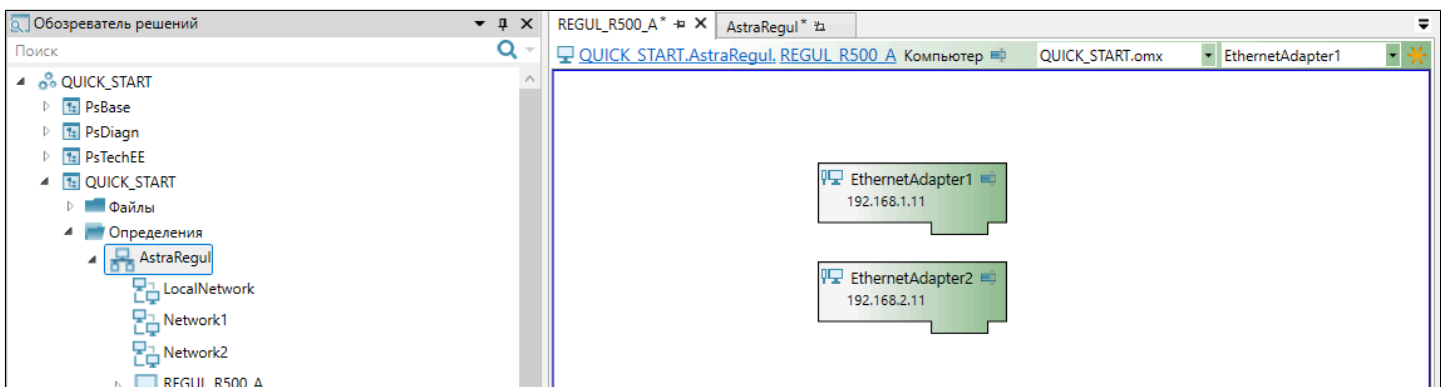


4. Перейдите в компонент "Компьютер" и перетащите в рабочую область из "Панели элементов" необходимое количество адаптеров "Ethernet".



Рекомендуется использовать резервируемую линию связи (два адаптера Ethernet).

У каждого адаптера необходимо настроить свойства "Имя", "Адрес" и "Сеть". В свойстве "Адрес" укажите IP-адрес порта ПЛК. Сеть выбирается из выпадающего списка: для каждого адаптера - своя сеть.



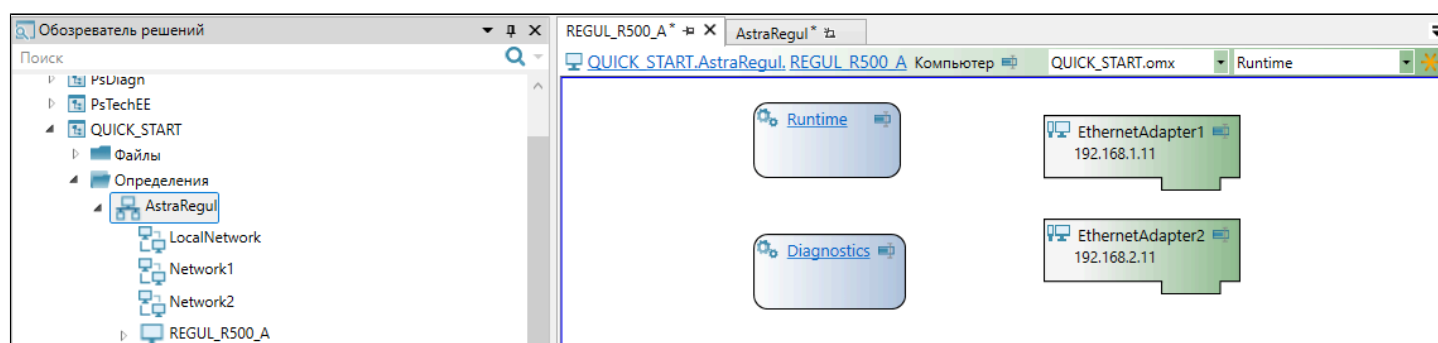
5. Перетащите в рабочую область два элемента "Исполняющий компонент" из "Панели элементов".

**Исполняющий компонент** – это любая сторонняя информационная система. Может иметь как программную реализацию (например, служба на компьютере), так и физическую (датчик или другое устройство).



Astra.AStudio не конфигурирует сторонние системы, но использует их описание при создании конфигураций для экземпляров Astra.Server, которые с ними взаимодействуют.

Первому исполняющему компоненту в окне Свойства задайте имя "Runtime" (для технологических данных), второму - "Diagnostics" (для диагностики ПЛК).



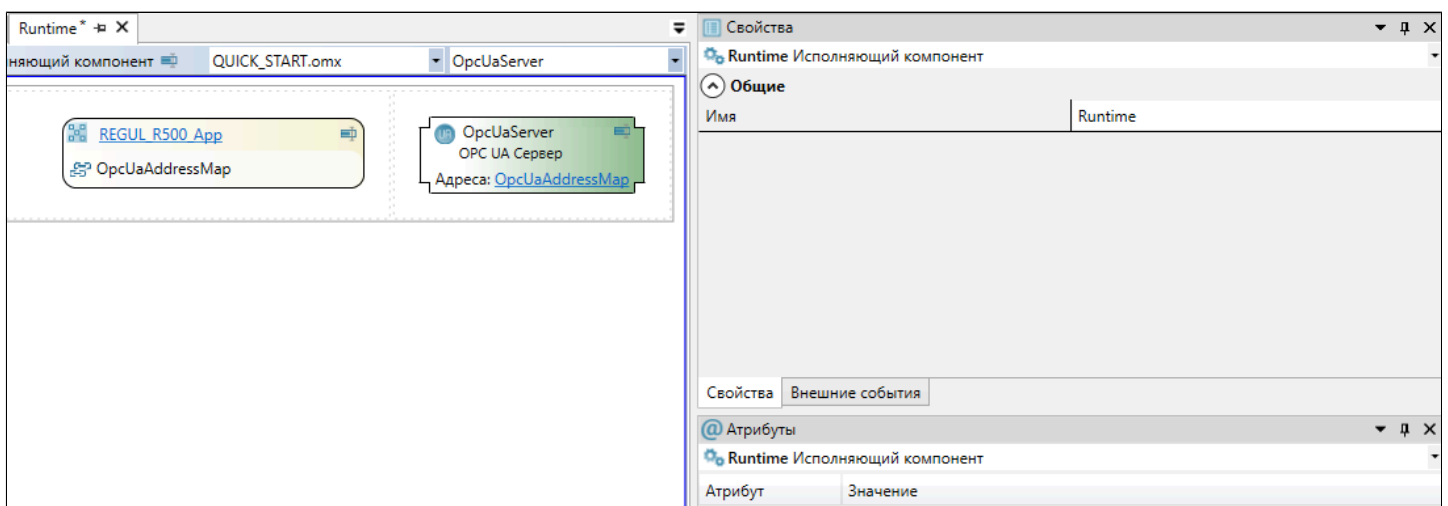
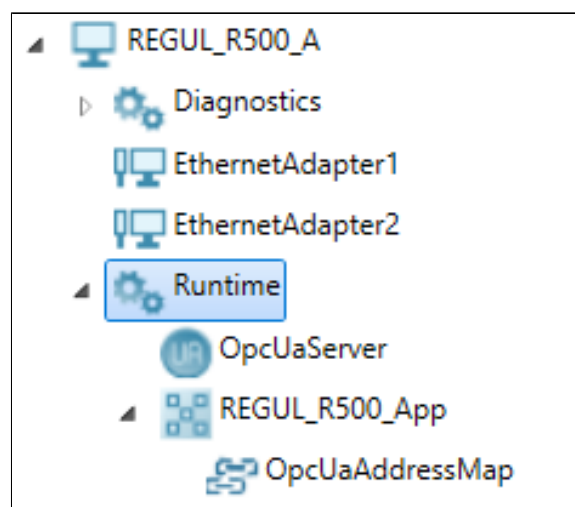
Далее необходимо создать приложения для каждого исполняющего компонента ПЛК.

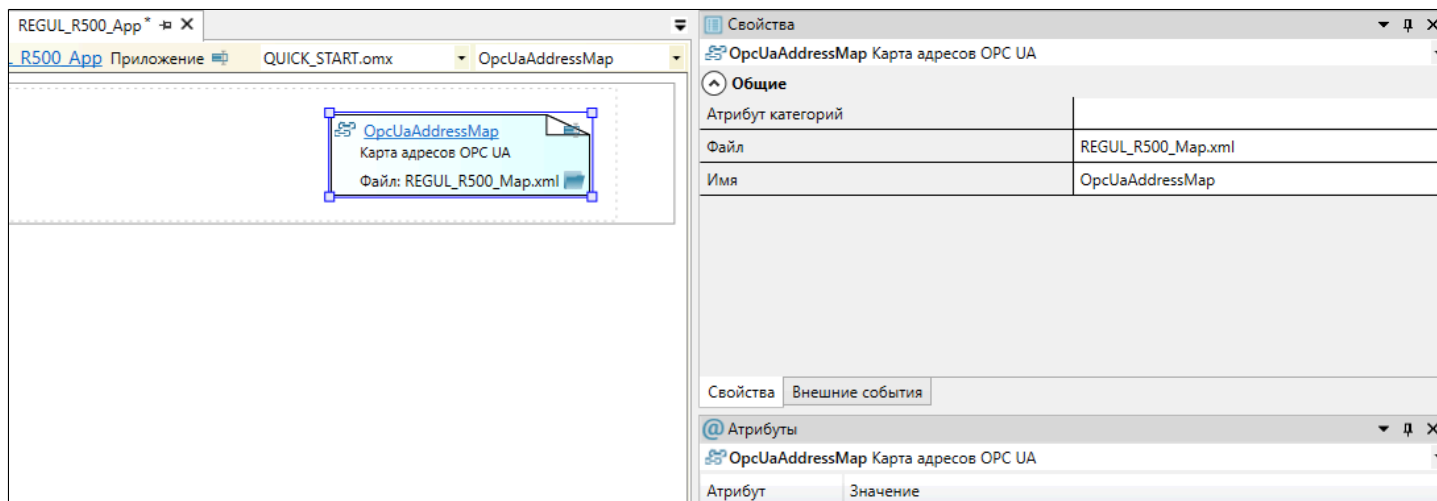
## 1.1.3.2.2.1. Приложение Runtime

**Приложение** – это элемент, в котором описываются данные исполняющего компонента. В любом исполняющем компоненте может быть только одно приложение.



Приложение "REGUL\_R500\_App", компоненты "Карта адресов OPC UA" и "OPC UA Сервер" добавляется в приложение "Runtime" автоматически при выполнении [экспорта проекта в Astra.IDE](#).



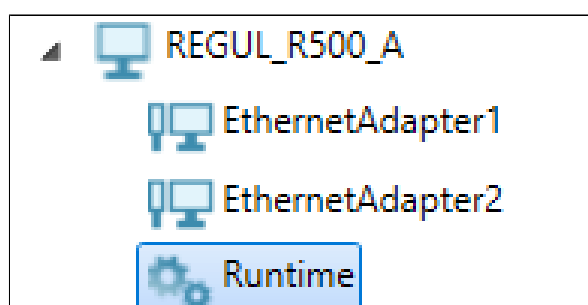


После экспорта необходимо создать [структуру дерева объектов](#) в приложении и [заполнить карту адресов](#). Все остальные компоненты настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление приложения Runtime

Чтобы добавить приложение вручную, выполните следующие действия:

1. Перейдите в компонент "Runtime" по пути "AstraRegul" → "[Имя ПЛК]" → "Runtime".



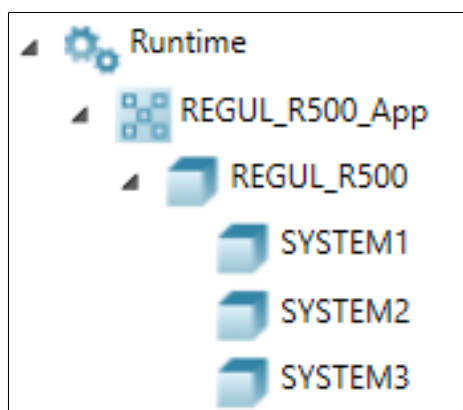
2. Перетащите в рабочую область из "Панели элементов" компонент "Приложение". В окне "Свойства" задайте "Имя".



Рекомендуется имя приложения задавать в формате [Имя ПЛК]\_App.

3. Перейдите в "Приложение" и создайте структуру дерева объектов, соответствующую текущим требованиям проекта.

Например, если на ПЛК приходят данные с трех подсистем, то структуру дерева объектов необходимо сформировать следующим образом:



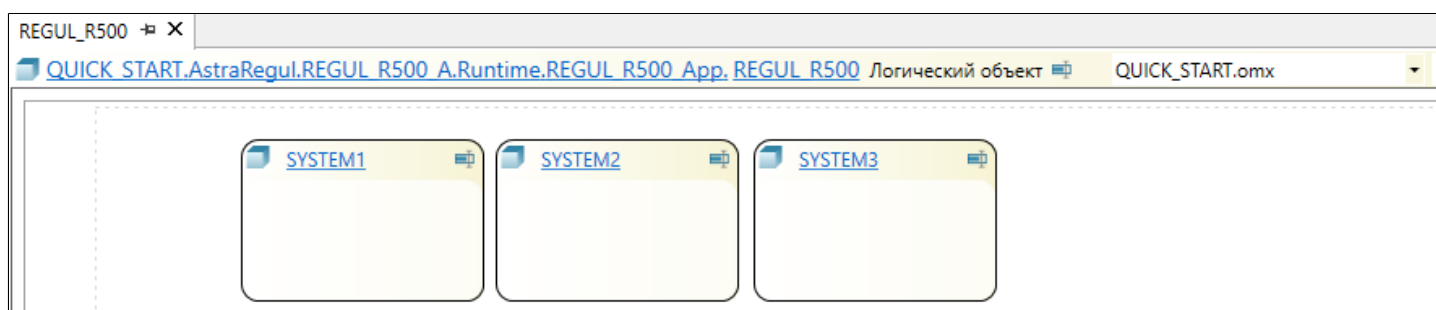
Для формирования структуры объектов необходимо использовать элемент "Логический объект" из "Панели элементов".

Логический объект представляет собой контейнер для других объектов (наподобие папки для файлов в проводнике).



Структура объектов в виде дерева позволяет удобно фильтровать и искать данные в проекте.

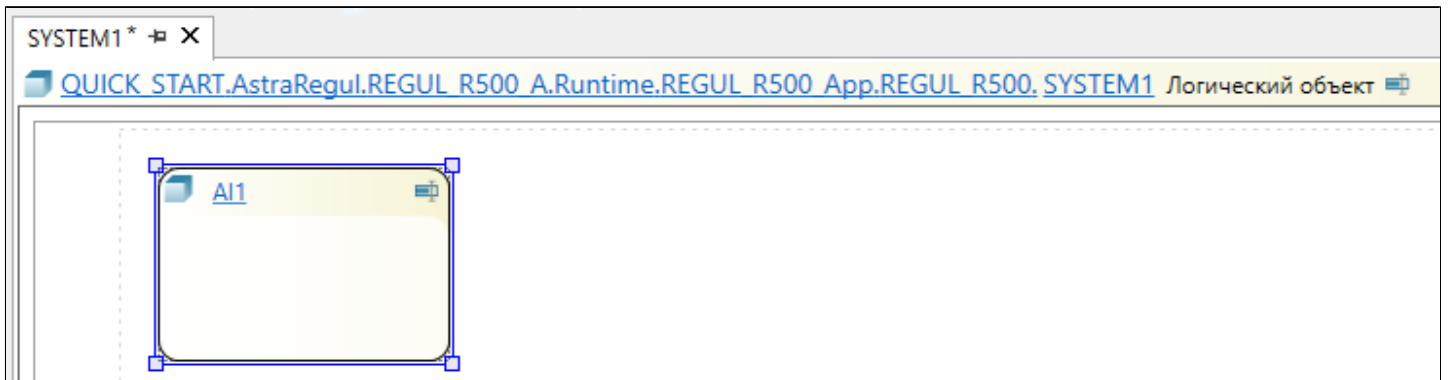
Внутри логического объекта ПЛК создайте логические объекты "SYSTEM1", "SYSTEM2", "SYSTEM3", которые будут обозначать каждую из возможных подсистем.



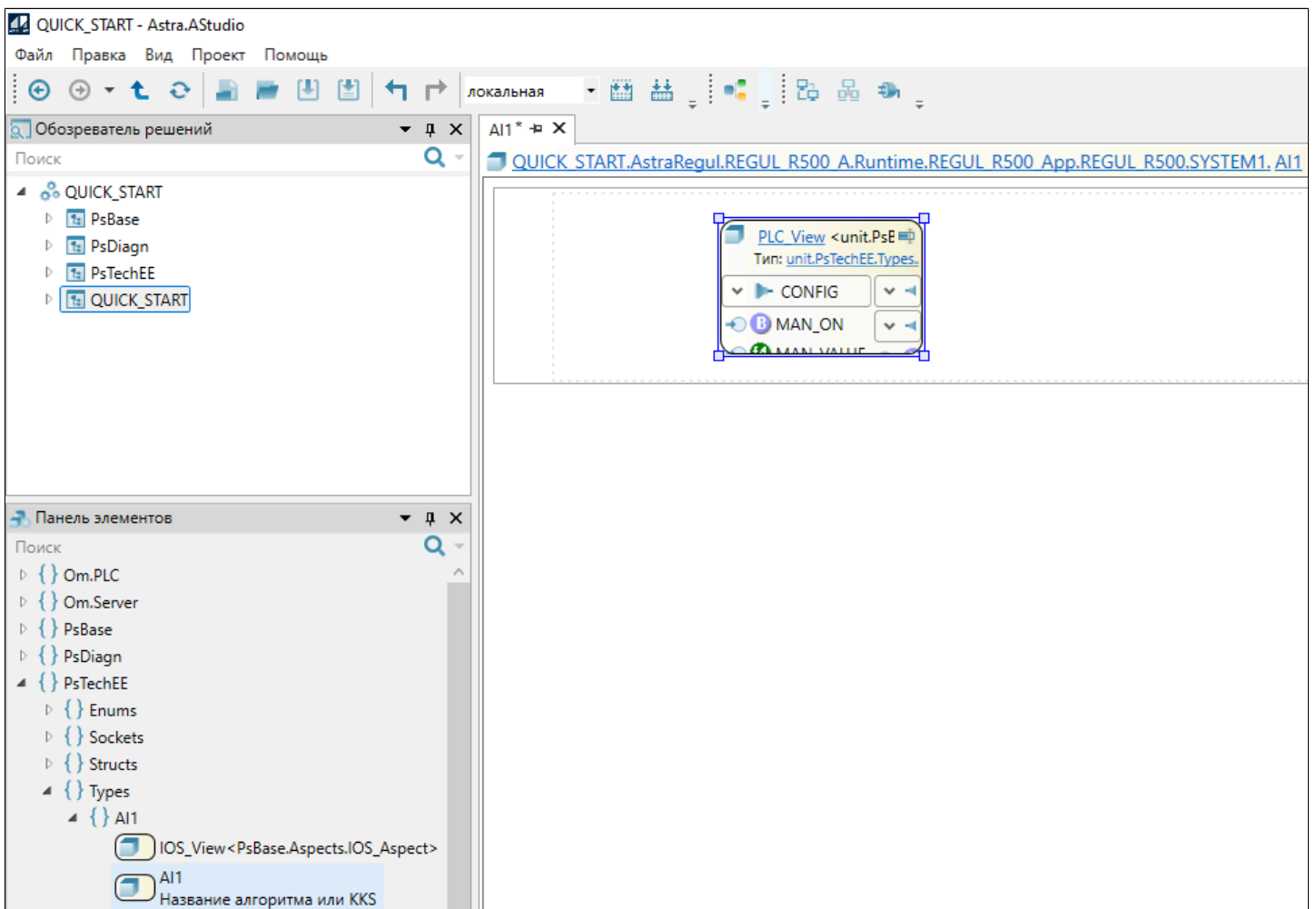
4. Для удобства рекомендуется группировать объекты по типу. Для объединения объектов одного типа необходимо использовать элемент



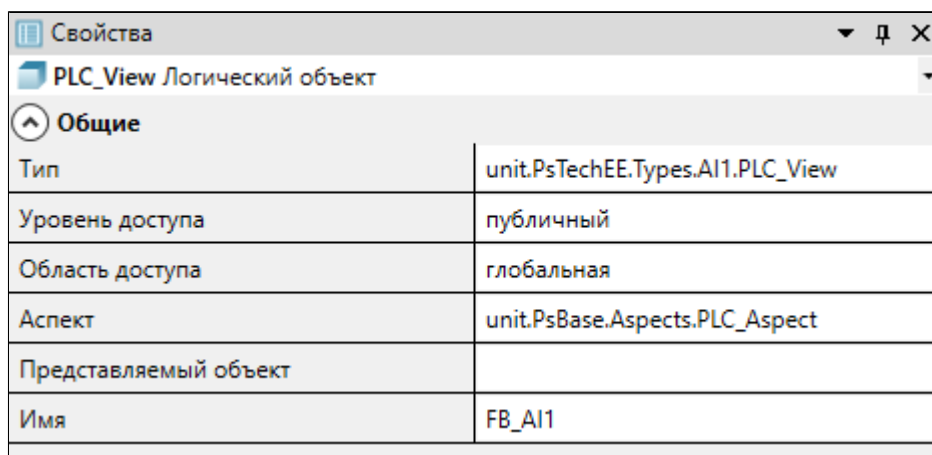
"Логический объект" из панели элементов. Перейдите в логический объект "SYSTEM1" и создайте логический объект "AI1" для всех аналоговых датчиков.



5. Перейдите в логический объект "AI1" и добавьте в рабочую область экземпляр типа "AI1" библиотеки "PsTechEE" из панели элементов.

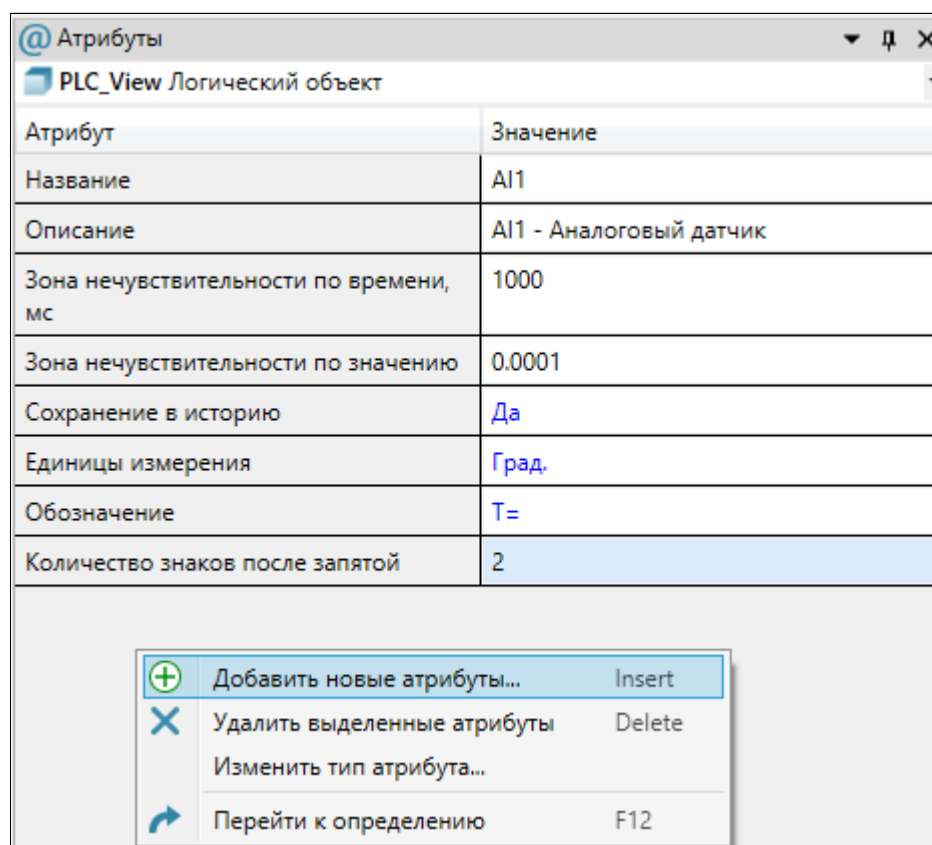


6. Выделите экземпляр и в редакторе "Свойства" задайте свойство "Имя" - "FB\_AI1".



Свойства	
PLC_View Логический объект	
Общие	
Тип	unit.PsTechEE.Types.AI1.PLC_View
Уровень доступа	публичный
Область доступа	глобальная
Аспект	unit.PsBase.Aspects.PLC_Aspect
Представляемый объект	
Имя	FB_AI1

7. В редакторе "Атрибуты" добавьте новый атрибут "KKS", нажав правой кнопкой мыши на пустую область и выбрав "Добавить новые атрибуты".

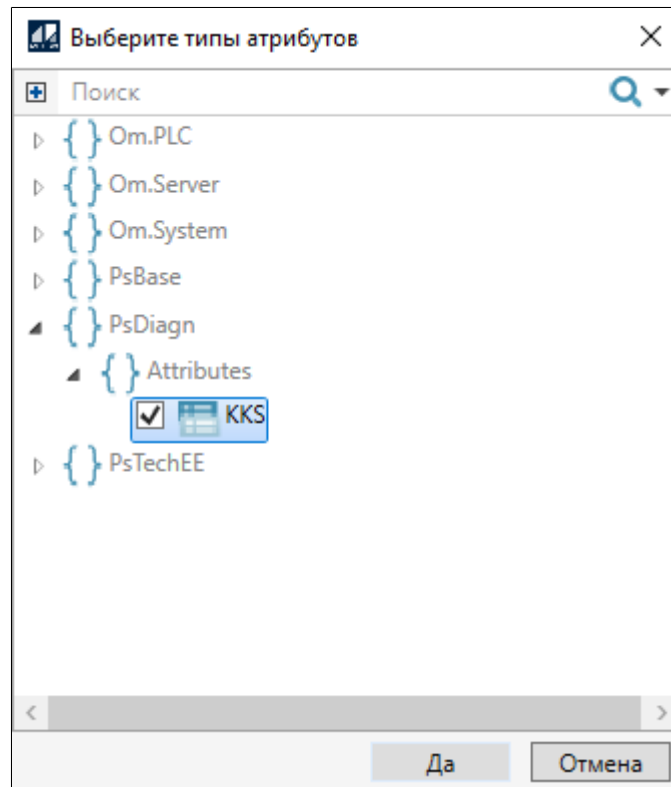


Атрибуты	
PLC_View Логический объект	
Атрибут	Значение
Название	AI1
Описание	AI1 - Аналоговый датчик
Зона нечувствительности по времени, мс	1000
Зона нечувствительности по значению	0.0001
Сохранение в историю	Да
Единицы измерения	Град.
Обозначение	T=
Количество знаков после запятой	2

+	Добавить новые атрибуты...	Insert
X	Удалить выделенные атрибуты	Delete
	Изменить тип атрибута...	
↶	Перейти к определению	F12

8. В открывшемся окне добавьте новый атрибут "KKS" ("PsDiagn" → "Attributes" → "KKS") и нажмите кнопку "Да".



9. В редакторе "Атрибуты" заполните атрибуты "Описание", "Название", "Обозначение", "KKS", "Зона нечувствительности по времени", "Зона нечувствительности по значению", "Единицы измерения" и "Количество знаков после запятой". Если значения атрибутов, заданные по умолчанию, были переопределены, они подсвечиваются синим цветом.

The screenshot displays the 'Свойства' (Properties) window for a 'PLC\_View' object. The window is divided into two main sections: 'Общие' (General) and '@ Атрибуты' (Attributes).

**Общие (General) Properties:**

Тип	unit.PsTechEE.Types.AI1.PLC_View
Уровень доступа	публичный
Область доступа	глобальная
Аспект	unit.PsBase.Aspects.PLC_Aspect

**@ Атрибуты (Attributes) Table:**

Атрибут	Значение
Название	AI1
Описание	AI1 - Аналоговый датчик
Зона нечувствительности по времени, мс	1000
Зона нечувствительности по значению	0.0001
Сохранение в историю	Да
Единицы измерения	Град.
Обозначение	T=
Количество знаков после запятой	2
KKS	KKS

10. Аналогичным образом добавьте другие объекты автоматизации: задвижки, клапаны, насосы и т.д.

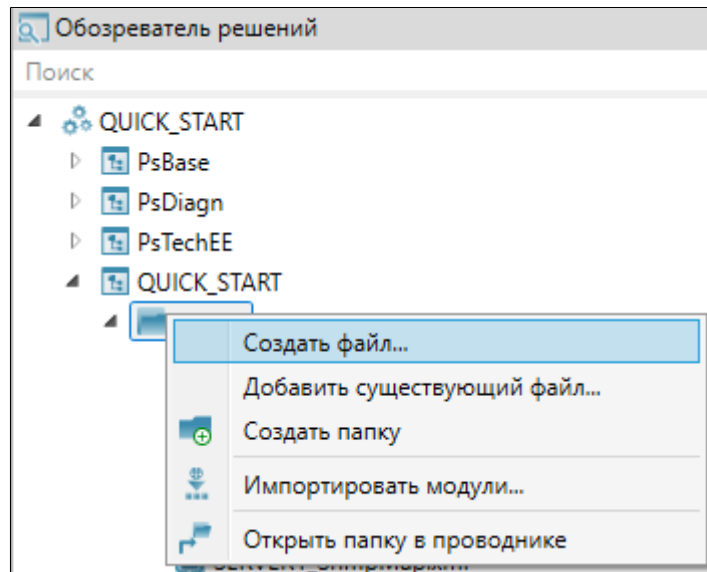


Процесс наполнения дерева объектами можно автоматизировать с помощью макросов.

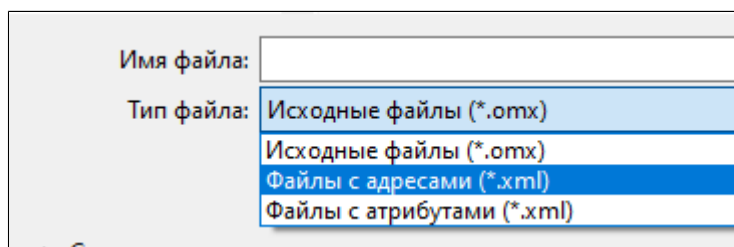
11. Перейдите в приложение и перетащите в рабочую область из "Панели элементов" компонент "Карта адресов OPC UA".

The screenshot shows the 'REGUL\_R500 App' interface. On the left, the 'Панель элементов' (Element Panel) contains a component named 'OpcUaAddressMap1' (Карта адресов OPC UA) with a file icon next to it. This component is highlighted with a red rectangular box. On the right, the main workspace displays a 'REGUL\_R500' object containing three sub-components: 'SYSTEM1', 'SYSTEM2', and 'SYSTEM3'.

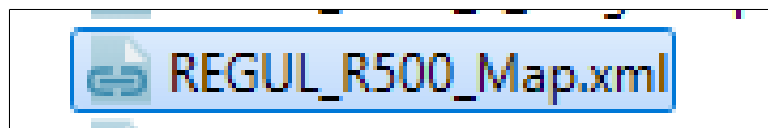
12. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." из контекстного меню объекта "Файлы" в Обозревателе решений.



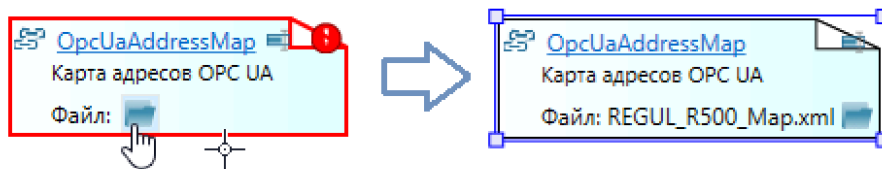
В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".



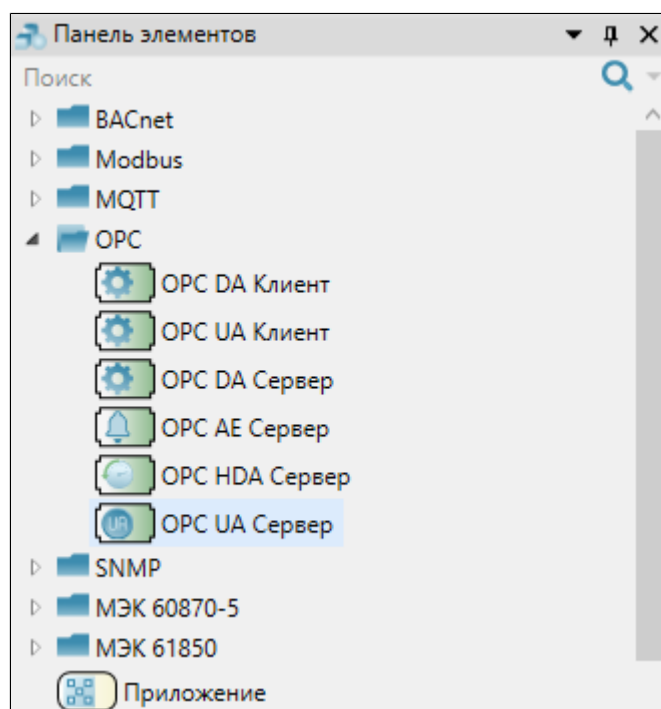
Имя файла рекомендуется задать в формате [Имя ПЛК]\_Map.



13. Перейдите в Приложение. У объекта "Карта адресов OPC UA" укажите созданный файл.



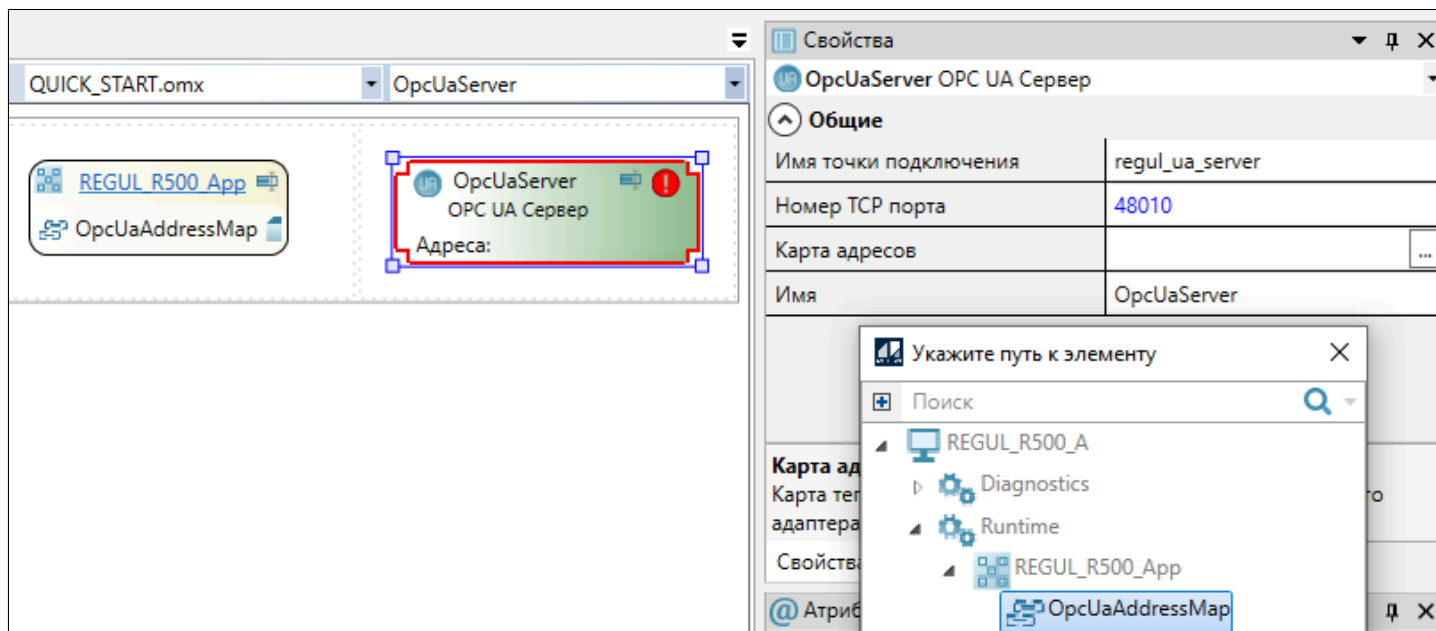
14. Перейдите в исполняющий компонент "Runtime" и добавьте в рабочую область из "Панели элементов" компонент "OPC UA Сервер".



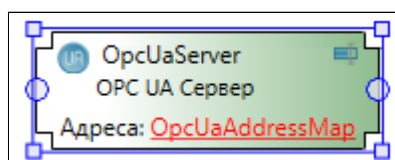
15. Выделите компонент OPC UA Сервер и в окне "Свойства" укажите настройки OPC UA сервера на ПЛК REGUL:

- › Имя точки подключения: regul\_ua\_server.
- › Номер TCP порта: 48010 (порт по умолчанию для ПЛК REGUL).

Для свойства Карта адресов укажите путь к карте адресов приложения.



16. Перейдите по ссылке в редактор карты адресов.

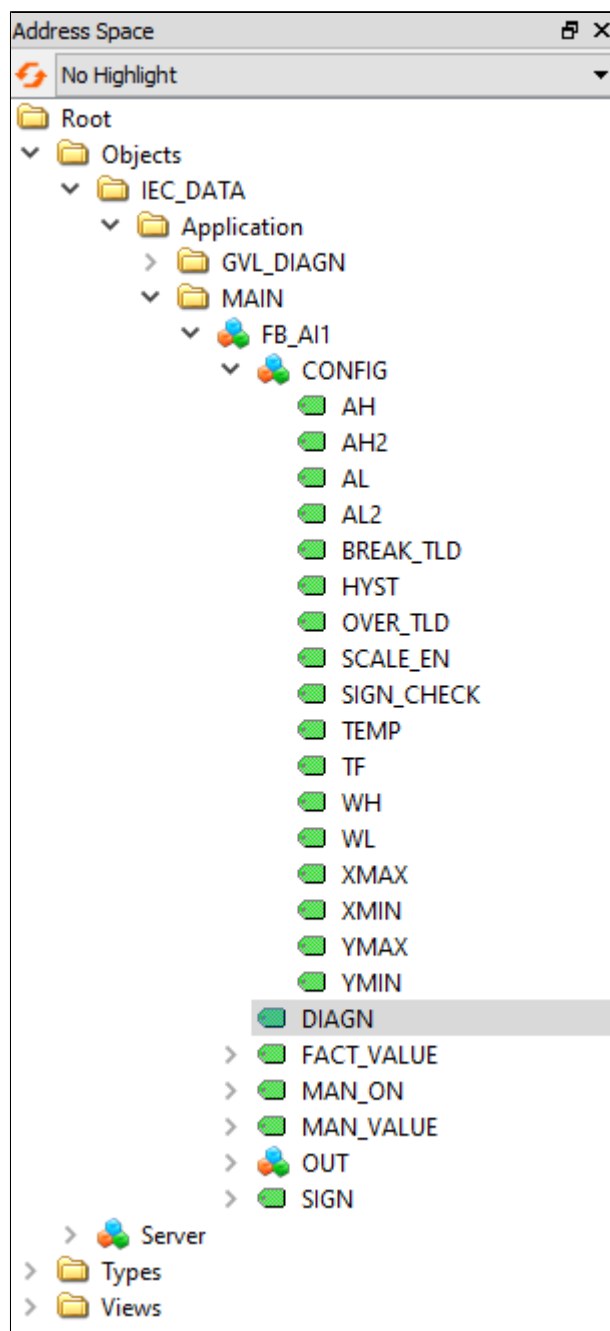


17. Заполните карту адресов.

- Привязка: указывает, происходит ли чтение/запись переменной.
- Адресное пространство ПЛК REGUL: `urn:ProsoftSystems:regul_ua_server:iec_data`
- Тип идентификатора: Строковый.
- Идентификатор узла: полный путь к переменной в адресном пространстве сервера OPC UA (проект Astra.IDE).

Для заполнения поля "Идентификатор узла" необходимо воспользоваться утилитой [UaExpert](#).

18. Откройте утилиту UA Expert и раскройте дерево сигналов: "IEC\_DATA" → "Application" → "MAIN" → "FB\_AI1" .





19. Выберите переменную "YMIN". В окне "Attributes" скопируйте значение поля "Identifier", воспользовавшись контекстным меню или нажмите сочетание клавиш "Ctrl+C".

The screenshot displays the SIMATIC Manager interface with three main panes:

- Project:** Shows a tree view with 'Project' > 'Servers' > 'regul\_ua\_server' > 'Documents' > 'Data Access View'.
- Address Space:** Shows a tree view with 'Root' > 'Objects' > 'IEC\_DATA' > 'Application' > 'GVL\_DIAGN' > 'MAIN' > 'FB\_AI1' > 'CONFIG' > 'YMIN' (highlighted).
- Attributes:** Shows the properties of the selected variable 'YMIN'. The 'Identifier' field is highlighted, and a context menu is open with 'Copy Value' selected.

Attribute	Value
NodeId	ns=2;s=Application.MAIN.FB_AI1.CONFIG.YMIN
NamespaceIndex	2
IdentifierType	String
Identifier	Application.MAIN.FB_AI1.CONFIG.YMIN
NodeClass	Variable
BrowseName	2, "YMIN"
DisplayName	"" , "YMIN"
Description	"" , ""
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeIdInvalid (0x80350000)
UserRolePermissions	BadAttributeIdInvalid (0x80350000)
AccessRestrictions	BadAttributeIdInvalid (0x80350000)
Value	
SourceTimestamp	18.09.2023 16:30:06.766
SourcePicoseconds	0
ServerTimestamp	18.09.2023 16:30:06.766
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	0
DataType	Float
NamespaceIndex	0
IdentifierType	Numeric

Reference	Target DisplayName
HasTypeDefiniti...	DataItem Type

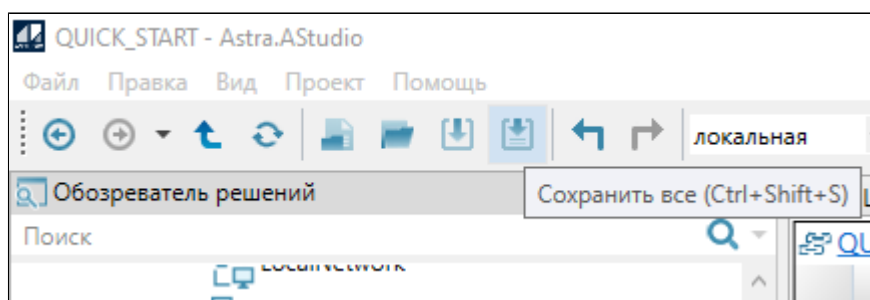
20. Вставьте скопированное значение в поле "Идентификатор узла" карты адресов для переменной "YMIN".

Сигнал	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.YMIN	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.YMIN

21. Аналогичным образом заполните карту адресов для всех сигналов.

Сигнал	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.YMIN	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.YMIN
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.YMAX	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.YMAX
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.AH	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.AH
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.WH	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.WH
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.WL	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.WL
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.AL	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.AL
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.TF	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.TF
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.HYST	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.HYST
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.SIGN_CHECK	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.SIGN_CHECK
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.AH2	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.AH2
REGUL_R500.SYSTEM1.A11.FB_A11.CONFIG.AL2	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.CONFIG.AL2
REGUL_R500.SYSTEM1.A11.FB_A11.OUT.VALUE	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.OUT.VALUE
REGUL_R500.SYSTEM1.A11.FB_A11.OUT.QUALITY	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.OUT.QUALITY
REGUL_R500.SYSTEM1.A11.FB_A11.MAN_ON	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.MAN_ON
REGUL_R500.SYSTEM1.A11.FB_A11.MAN_VALUE	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.MAN_VALUE
REGUL_R500.SYSTEM1.A11.FB_A11.FACT_VALUE	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.FACT_VALUE
REGUL_R500.SYSTEM1.A11.FB_A11.SIGN	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.SIGN
REGUL_R500.SYSTEM1.A11.FB_A11.DIAGN	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_A11.DIAGN

22. Сохраните проект, нажав кнопку "Сохранить (Ctrl+Shift+S)".

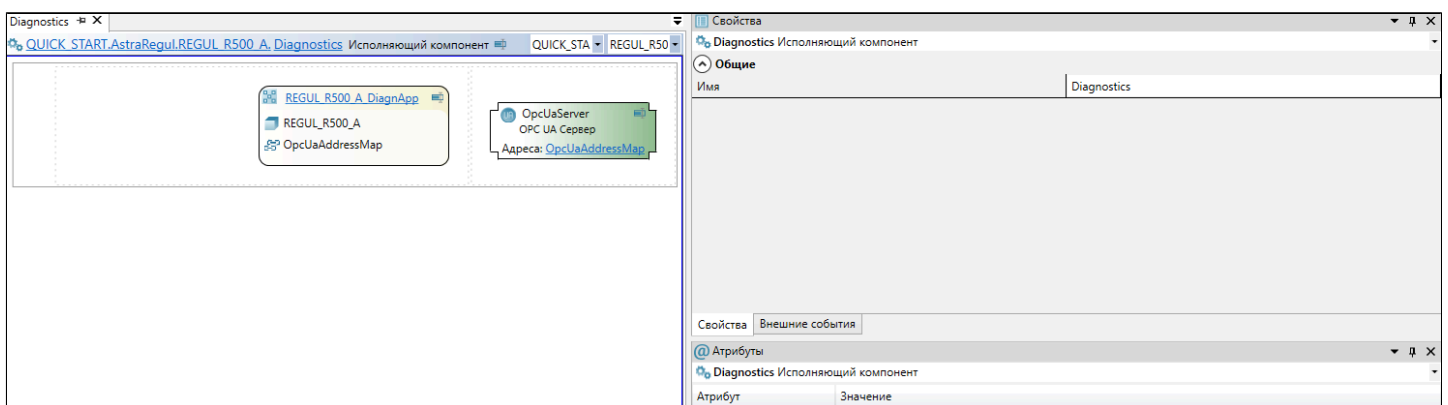
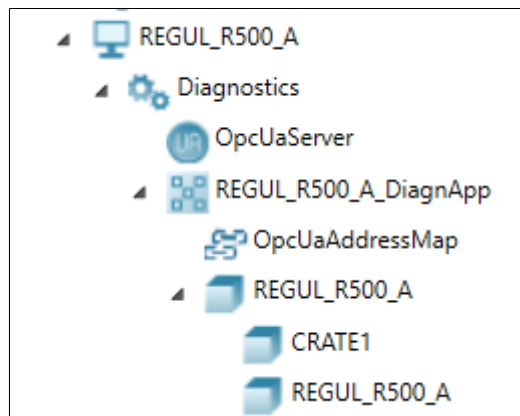


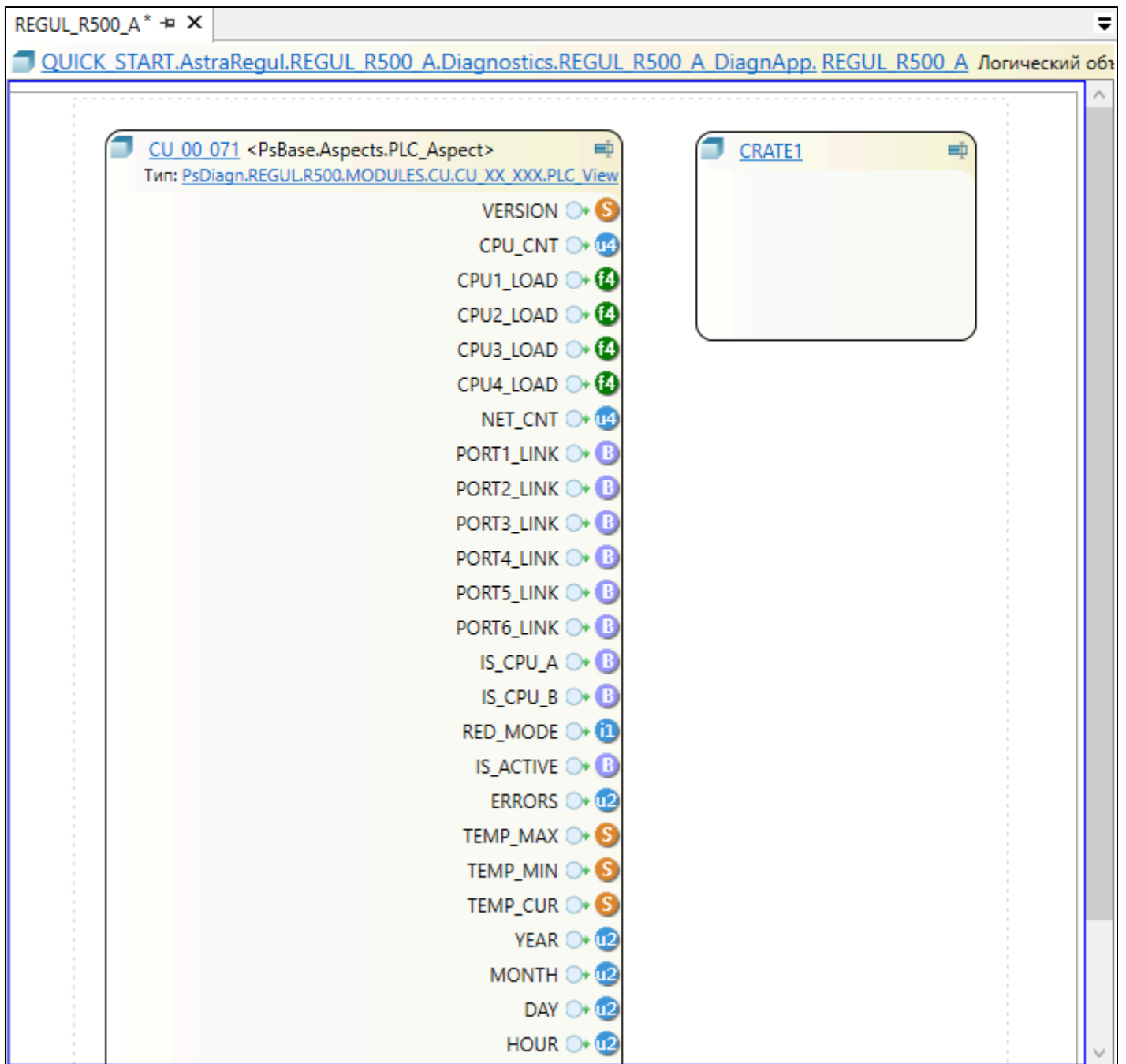
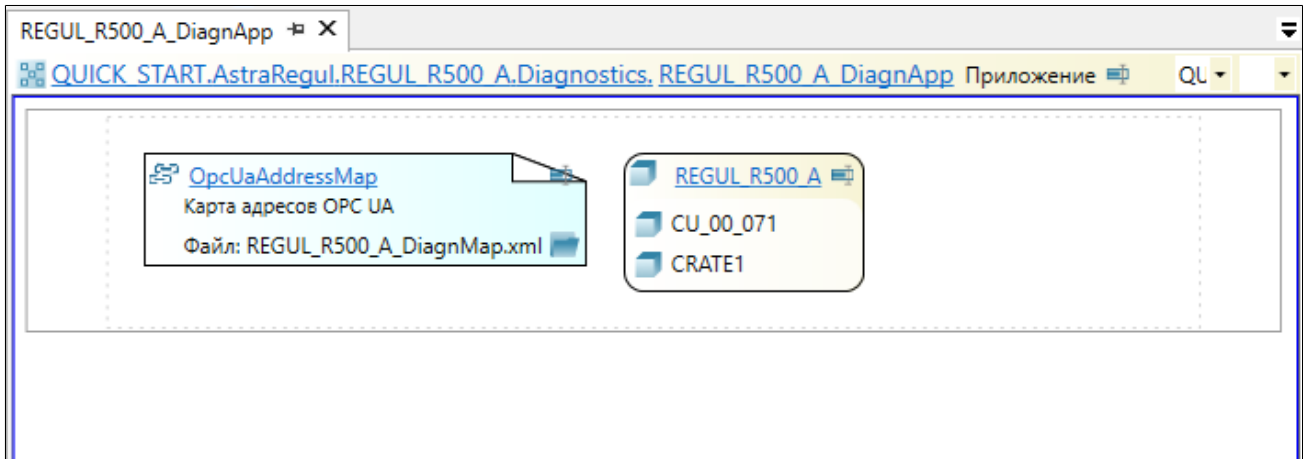
## 1.1.3.2.2.2. Приложение Diagnostics

**Приложение** – это элемент, в котором описываются данные исполняющего компонента. В любом исполняющем компоненте может быть только одно приложение.



Приложение "REGUL\_R500\_[A/B]\_DiagnApp" вместе с объектом контроллера и логическими элементами крейтов, а также компоненты "Карта адресов OPC UA" и "OPC UA Сервер" добавляется в приложение "Diagnostics" автоматически при выполнении [экспорта проекта в Astra.IDE](#).





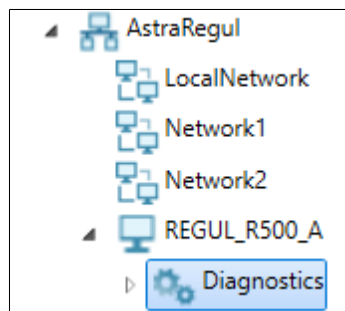


После экспорта необходимо создать структуру дерева объектов в [логических объектах крейтов](#) в приложении и [заполнить карту адресов](#). Все остальные компоненты настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление приложения Diagnostics

Чтобы добавить приложение вручную, необходимо выполнить следующие действия:

1. Перейдите в компонент "Diagnostics" по пути "AstraRegul" → "[Имя ПЛК]" → "Diagnostics".



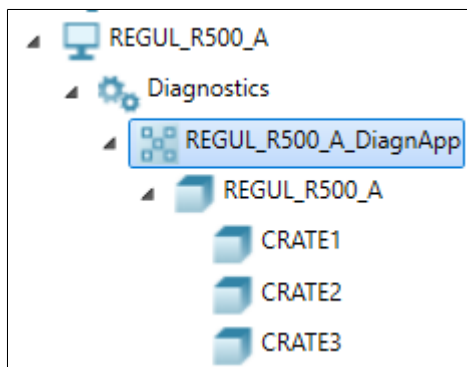
2. Перетащите в рабочую область из Панели элементов компонент "Приложение". В окне "Свойства" задайте "Имя".



Рекомендуется задавать имя приложения в формате [Имя ПЛК]\_[A/B]\_DiagnApp.

3. Перейдите в Приложение и создайте структуру дерева объектов, соответствующую конфигурации ПЛК.

Например, если конфигурация ПЛК состоит из трех крейтов, то структуру дерева объектов необходимо сформировать следующим образом:



Для формирования структуры объектов используйте элемент "Логический объект" из "Панели элементов".

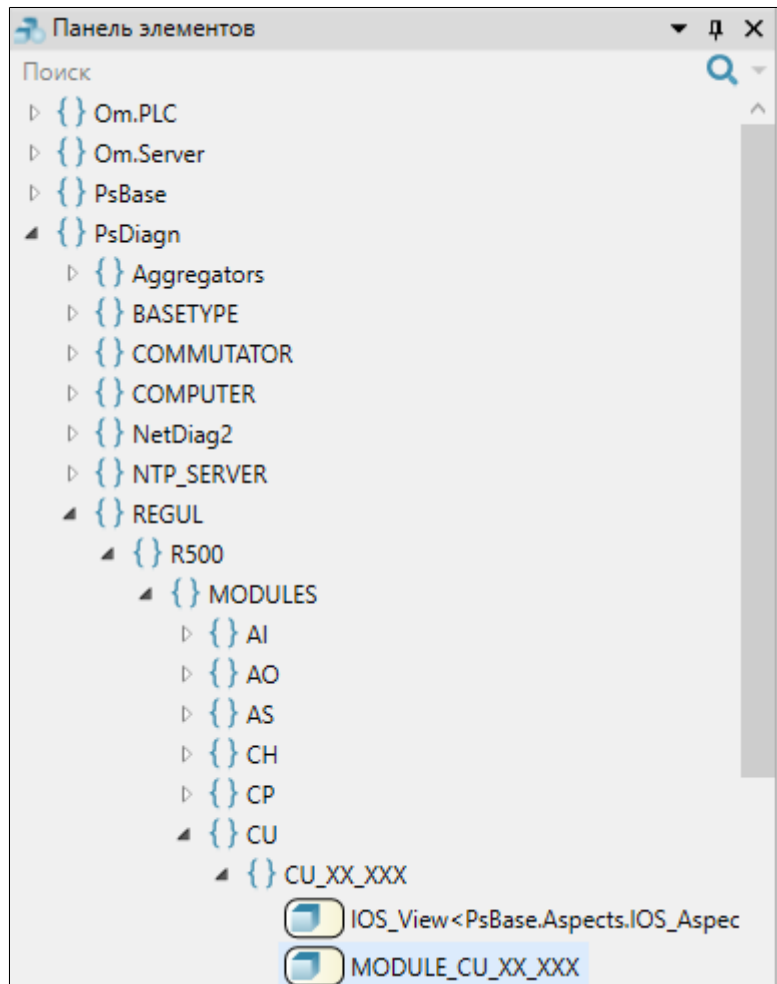
Логический объект представляет собой контейнер для других объектов (наподобие папки для файлов в проводнике).



Структура объектов в виде дерева позволяет удобно фильтровать и искать данные в проекте.

4. Далее необходимо наполнить структуру объектами.

Перетащите объект "MODULE\_CU\_XX\_XXX" (модуль ЦП) из библиотеки "PsDiagn" Панели элементов в рабочую область ПЛК А. Модуль ЦП рекомендуется размещать в структуре объектов отдельно от крейтов.



5. Структура будет выглядеть следующим образом:

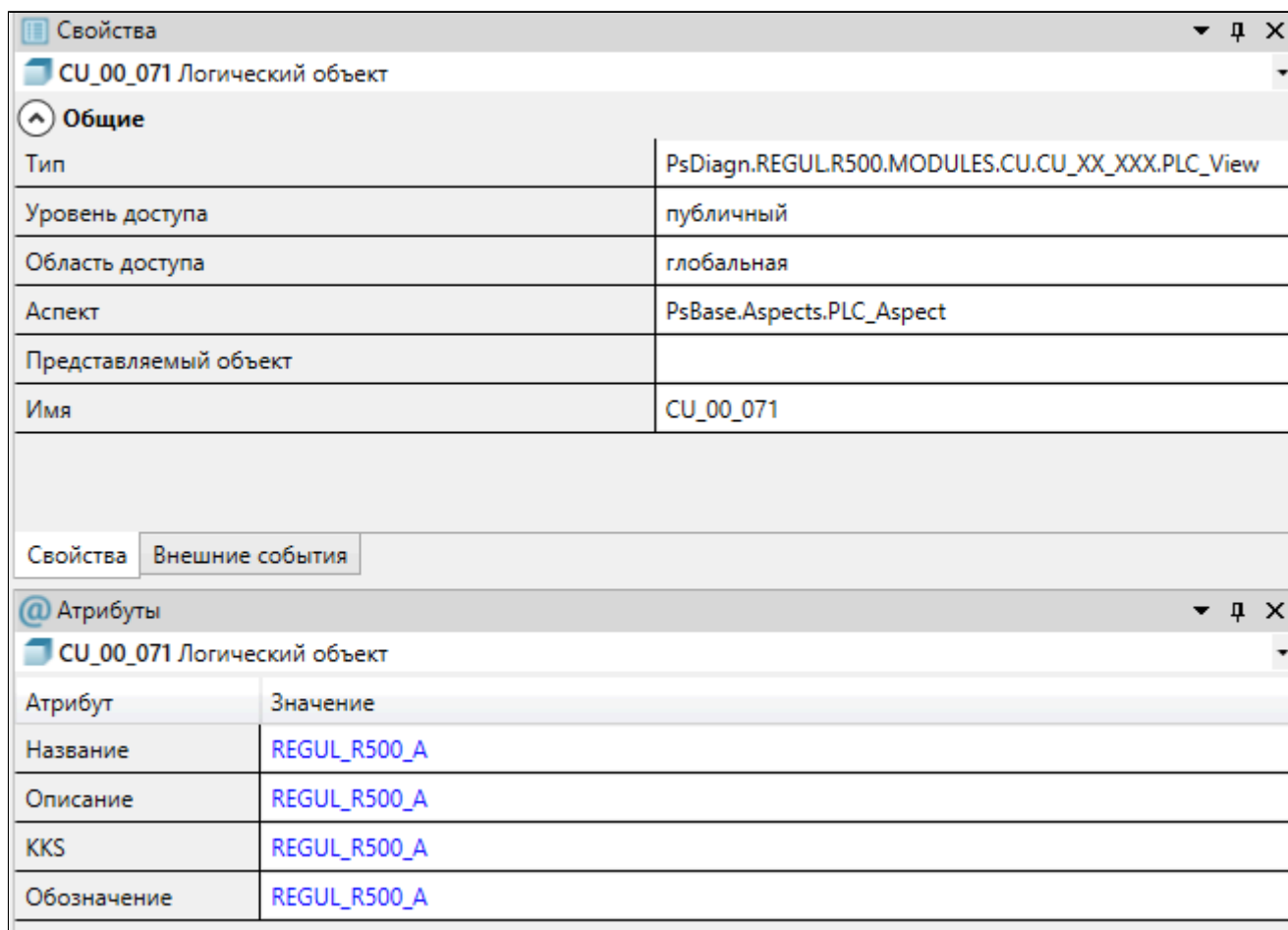
The screenshot displays a software interface with a main window titled "PLC View <unit.PsBase.Aspects.PLC\_Aspect>". Below the title, it specifies the type: "Тип: unit.PsDiagn.REGUL.R500.MODULES.CU.CU XX XXX.PLC View". A list of variables is shown, each with a status indicator (a circle containing a letter or number):

- VERSION (S)
- CPU\_CNT (u4)
- CPU1\_LOAD (f4)
- CPU2\_LOAD (f4)
- CPU3\_LOAD (f4)
- CPU4\_LOAD (f4)
- NET\_CNT (u4)
- PORT1\_LINK (B)
- PORT2\_LINK (B)
- PORT3\_LINK (B)
- PORT4\_LINK (B)
- PORT5\_LINK (B)
- PORT6\_LINK (B)
- IS\_CPU\_A (B)
- IS\_CPU\_B (B)
- RED\_MODE (i1)
- IS\_ACTIVE (B)

To the right of the PLC View window, there are three empty windows labeled "CRATE1", "CRATE2", and "CRATE3".



6. Задайте имя и атрибуты для объекта. Если значения атрибутов, заданные по умолчанию, были переопределены, они подсвечиваются синим цветом.



Свойства

CU\_00\_071 Логический объект

Общие

Тип	PsDiagn.REGULR500.MODULES.CU.CU_XX_XXX.PLC_View
Уровень доступа	публичный
Область доступа	глобальная
Аспект	PsBase.Aspects.PLC_Aspect
Представляемый объект	
Имя	CU_00_071

Свойства Внешние события

Атрибуты

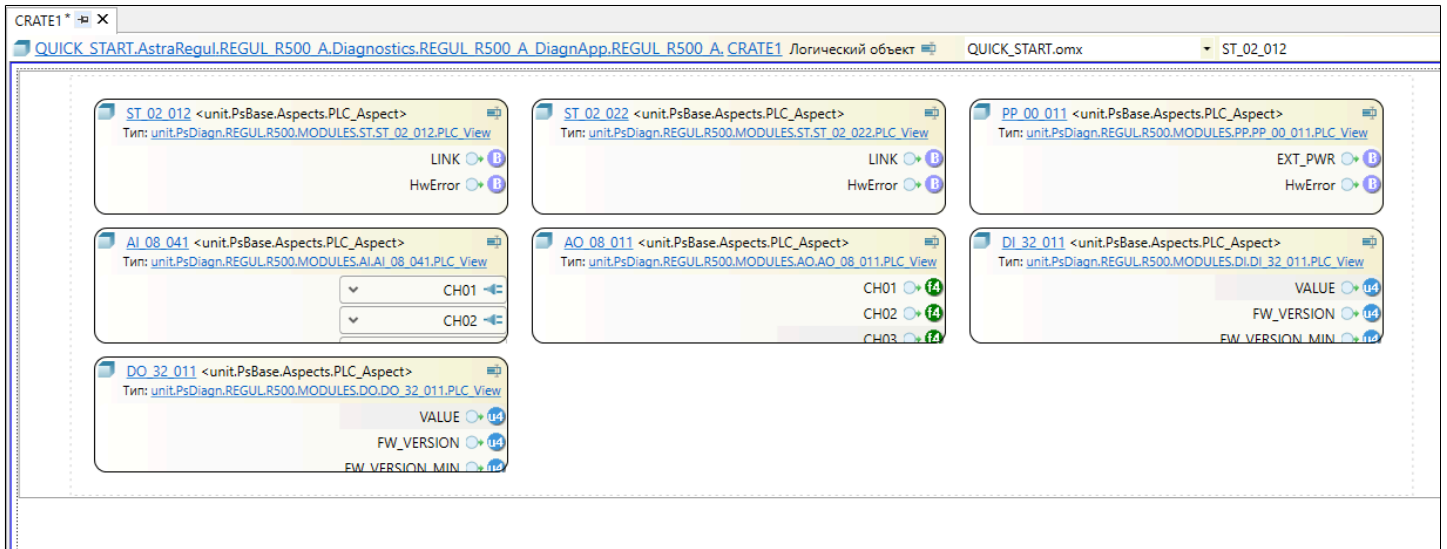
CU\_00\_071 Логический объект

Атрибут	Значение
Название	REGUL_R500_A
Описание	REGUL_R500_A
KKS	REGUL_R500_A
Обозначение	REGUL_R500_A

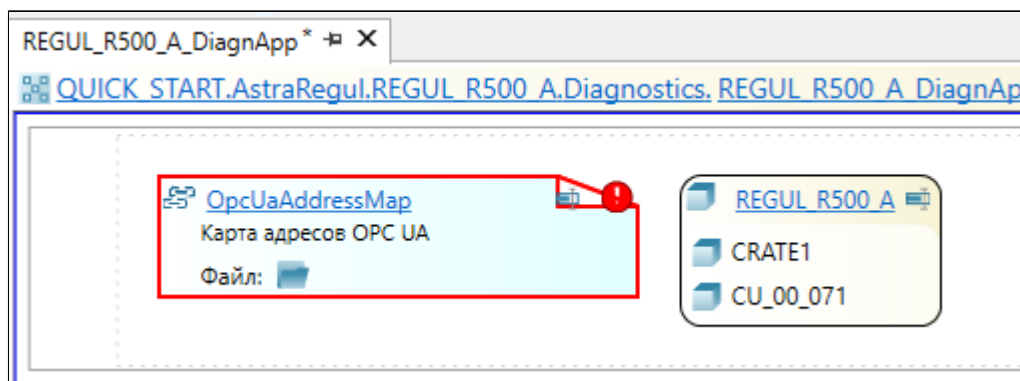


При необходимости атрибуты "Название", "Описание", "Обозначение" и "KKS" можно изменить.

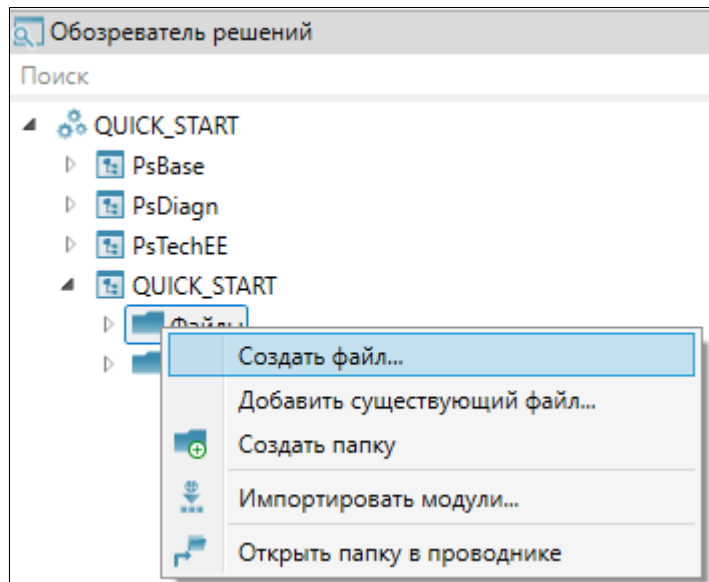
7. Из библиотеки "PsDiagn" Панели элементов перетащите в рабочую область "CRATE1", "CRATE2", "CRATE3" остальные модули в соответствии с конфигурацией ПЛК.



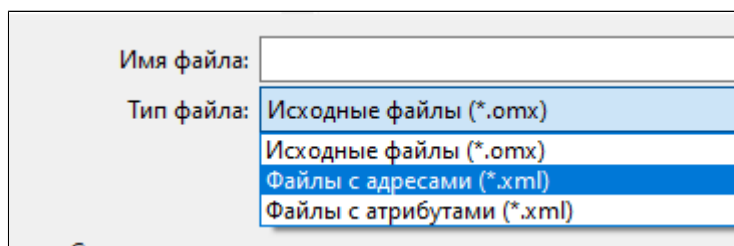
8. После формирования дерева объектов перейдите в "Приложение" и перетащите в рабочую область из "Панели элементов" компонент "Карта адресов OPC UA".



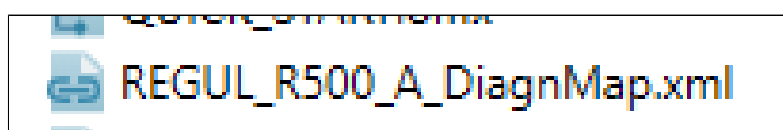
9. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." в контекстном меню объекта "Файлы" в Обозревателе решений.



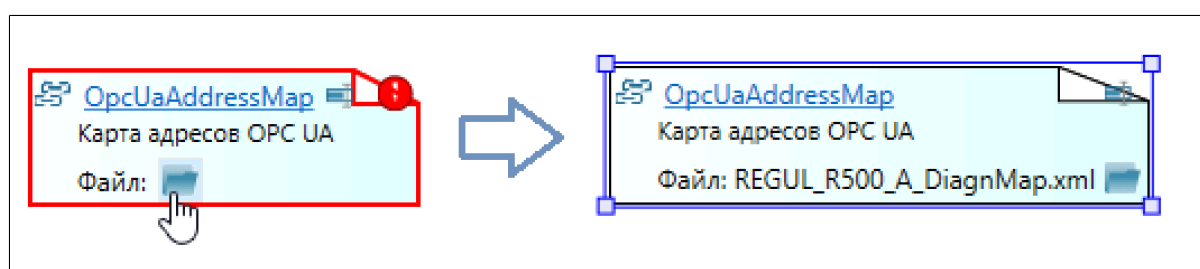
В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".



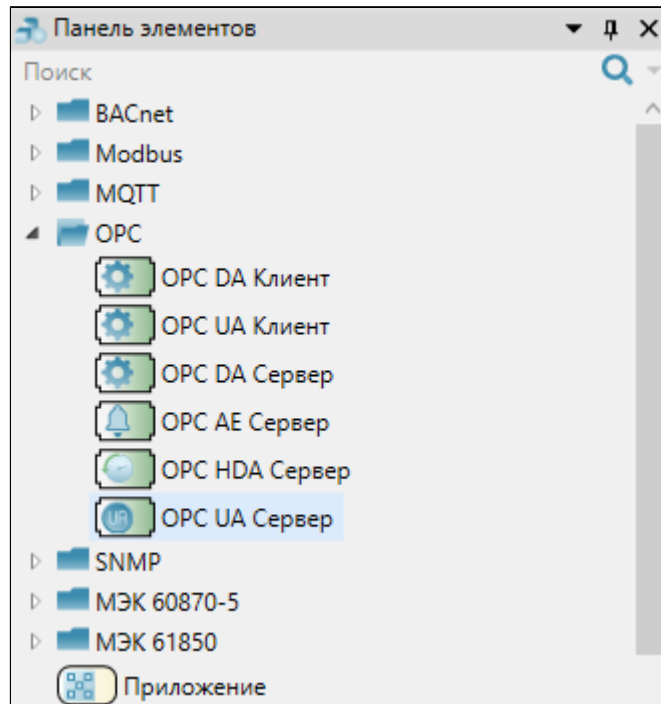
**i** Имя файла рекомендуется задавать в формате [Имя ПЛК]\_[A/B]\_DiagnMap.



10. Перейдите в Приложение. У объекта "Карта адресов OPC UA" укажите созданный файл.



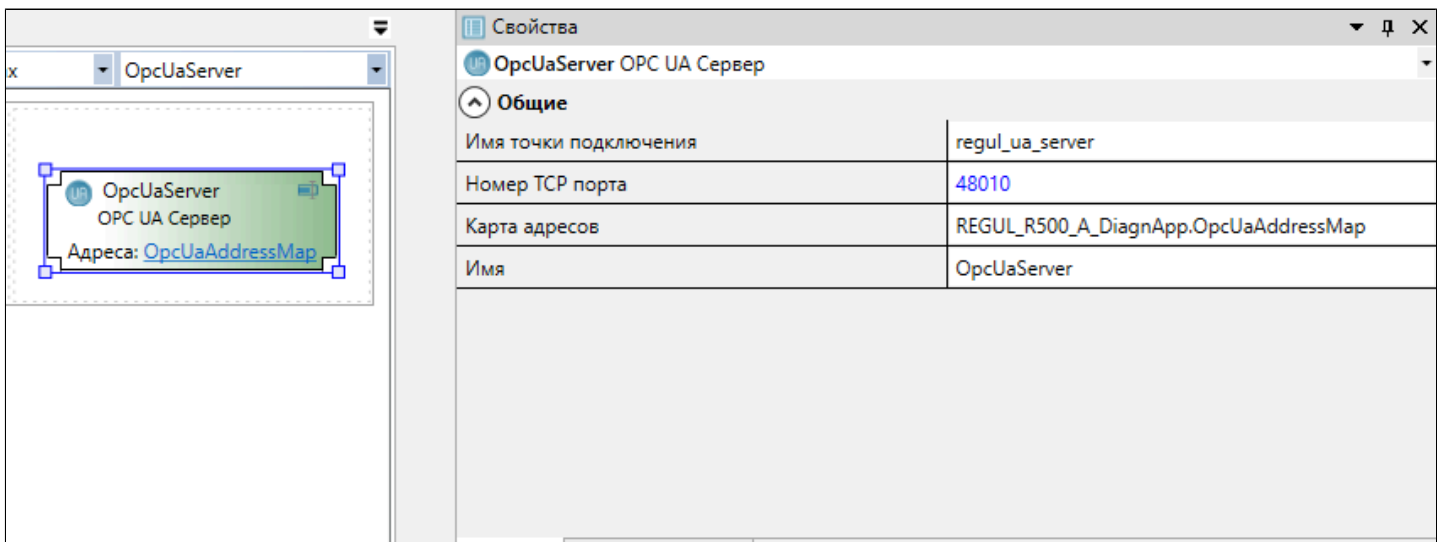
11. Далее перейдите в исполняющий компонент "Runtime" и добавьте в рабочую область из Панели элементов компонент "OPC UA Сервер".



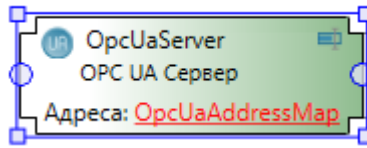
12. Выберите компонент "OPC UA Сервер" и в окне "Свойства" укажите настройки OPC UA сервера на ПЛК REGUL:

- › Имя точки подключения: regul\_ua\_server.
- › Номер TCP порта: 48010 (по умолчанию для ПЛК REGUL).

Для свойства "Карта адресов" укажите путь к карте адресов приложения.



13. Перейдите по ссылке в редактор карты адресов.

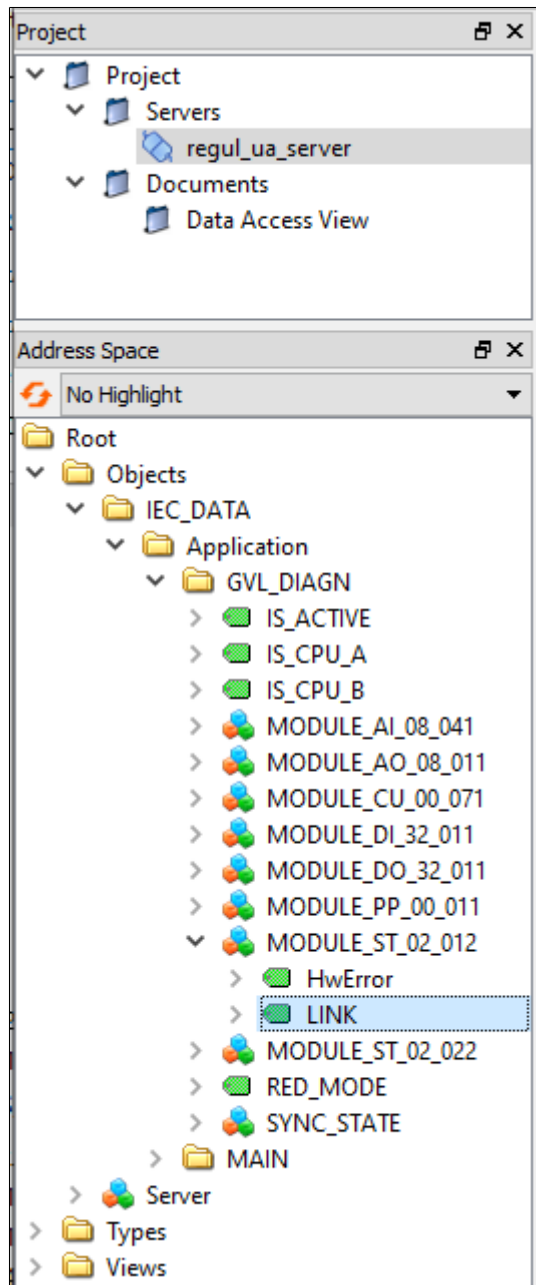


14. Заполните карту адресов.

- Привязка: указывает, происходит ли чтение/запись переменной.
- Адресное пространство ПЛК REGUL: urn:ProsoftSystems:regul\_ua\_server:iec\_data
- Тип идентификатора: Строковый.
- Идентификатор узла: полный путь к переменной в адресном пространстве сервера OPC UA (проект Astra.IDE).

Для заполнения поля "Идентификатор узла" необходимо воспользоваться утилитой [UaExpert](#).

15. Откройте утилиту UA Expert и раскройте дерево сигналов сервера: "IEC\_DATA" → "Application" → "GVL\_DIAGN" → "MODULE\_ST\_02\_12".



16. Выберите любую переменную модуля "ST\_02\_012". В окне "Attributes" скопируйте значение поля "Identifier" с помощью контекстного меню или нажмите сочетание клавиш "Ctrl+C".

The screenshot displays the SIMATIC Manager interface with three main panes:

- Project:** Shows a tree view with 'Servers' containing 'regul\_Lua\_server' and 'Documents' containing 'Data Access View'.
- Address Space:** Shows a tree view starting with 'Root', followed by 'Objects', 'IEC\_DATA', 'Application', and 'GVL\_DIAGN'. Under 'GVL\_DIAGN', there are several variables including 'IS\_ACTIVE', 'IS\_CPU\_A', 'IS\_CPU\_B', and 'MODULE\_ST\_02\_012'. The 'LINK' variable under 'MODULE\_ST\_02\_012' is selected.
- Attributes:** A table showing the properties of the selected variable. The 'Identifier' field is highlighted.

Attribute	Value
NodeId	ns=2;s=Application.GVL_DIAGN.MODULE_ST_02_012.LINK
NamespaceIndex	2
IdentifierType	String
Identifier	Application.GVL_DIAGN.MODULE_ST_02_012.LINK
NodeClass	Variable
BrowseName	2, "LINK"
DisplayName	"" , "LINK"
Description	"" , ""
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeValueInvalid (0x80350000)
UserRolePermissions	BadAttributeValueInvalid (0x80350000)
AccessRestrictions	BadAttributeValueInvalid (0x80350000)
Value	
SourceTimestamp	18.09.2023 17:38:55.015
SourcePicoseconds	0
ServerTimestamp	18.09.2023 17:38:55.015
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	false
DataType	Boolean
NamespaceIndex	0
IdentifierType	Numeric

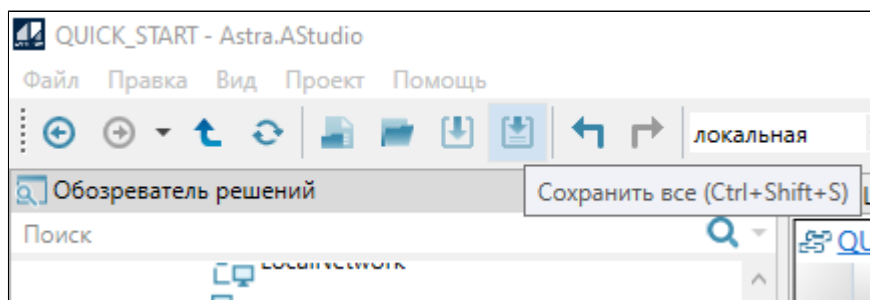
17. Вставьте скопированное значение в поле "Идентификатор узла" карты адресов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентик.	Идентификатор узла
REGUL_R500_A.CRATE1.ST_02_012.LINK	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_012.LINK

18. Аналогичным образом заполните поля у всех сигналов карты адресов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентик.	Идентификатор узла
REGUL_R500_A.CRATE1.ST_02_012.LINK	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_012.LINK
REGUL_R500_A.CRATE1.ST_02_012.HwError	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_012.HwError
REGUL_R500_A.CRATE1.ST_02_022.LINK	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_022.LINK
REGUL_R500_A.CRATE1.ST_02_022.HwError	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_022.HwError
REGUL_R500_A.CRATE1.PP_00_011.EXT_PWR	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_PP_00_011.EXT_PWR
REGUL_R500_A.CRATE1.PP_00_011.HwError	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_PP_00_011.HwError
REGUL_R500_A.CRATE1.AI_08_041.CH01.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH01.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH01.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH01.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH02.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH02.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH02.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH02.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH03.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH03.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH03.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH03.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH04.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH04.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH04.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH04.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH05.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH05.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH05.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH05.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH06.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH06.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH06.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH06.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH07.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH07.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH07.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH07.STATUS
REGUL_R500_A.CRATE1.AI_08_041.CH08.VALUE	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH08.VALUE
REGUL_R500_A.CRATE1.AI_08_041.CH08.STATUS	uint1	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.CH08.STATUS
REGUL_R500_A.CRATE1.AI_08_041.FW_VERSION	uint4	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.FW_VERSION
REGUL_R500_A.CRATE1.AI_08_041.FW_VERSION_MIN	uint4	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.FW_VERSION_MIN
REGUL_R500_A.CRATE1.AI_08_041.HwError	bool	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_041.HwError
REGUL_R500_A.CRATE1.AO_08_011.CH01	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AO_08_011.CH01
REGUL_R500_A.CRATE1.AO_08_011.CH02	float	непосредственно	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AO_08_011.CH02

19. Сохраните проект, нажав кнопку "Сохранить (Ctrl+Shift+S)".



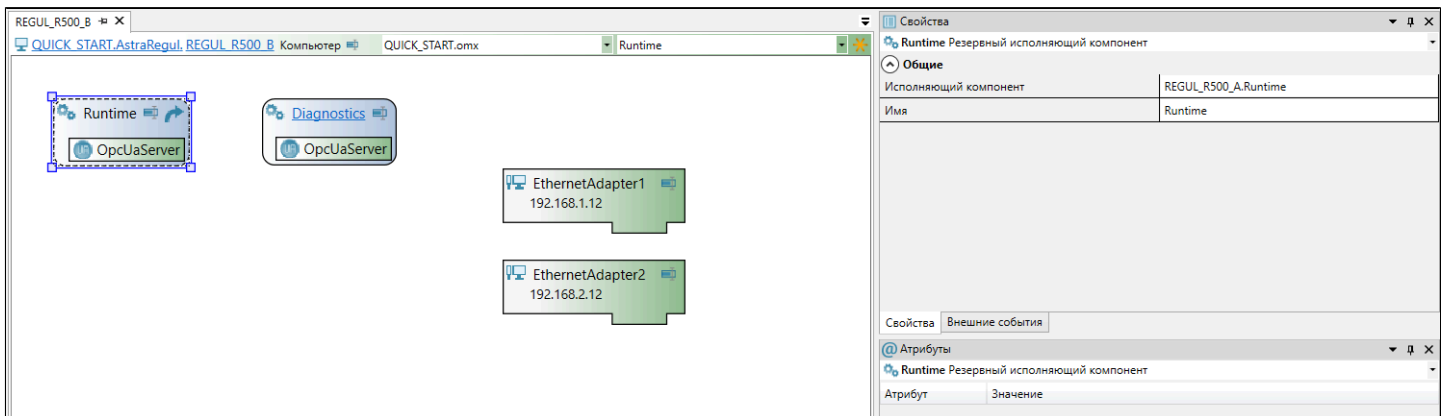


Аналогичным образом добавьте приложение диагностики для "REGUL\_R500\_B".

## 1.1.3.2.2.2.3. Резервирование



Резервный ПЛК с резервным исполняющим компонентом "Runtime" добавляется в топологию автоматически при выполнении [экспорта проекта в Astra.IDE](#).



После экспорта необходимо создать структуру дерева объектов в [приложении "Diagnostics"](#), аналогично с основным ПЛК, и [заполнить карту адресов](#). Все остальные компоненты резервного ПЛК настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление резервного ПЛК

Чтобы описать резервный ПЛК в топологии вручную, выполните следующие действия:

1. Перейдите в компонент "Astra.Domain" и перетащите в рабочую область из "Панели элементов" компонент "Компьютер".
2. Выделите элемент "Компьютер" и в окне "Свойства" задайте ему свойство "Имя".

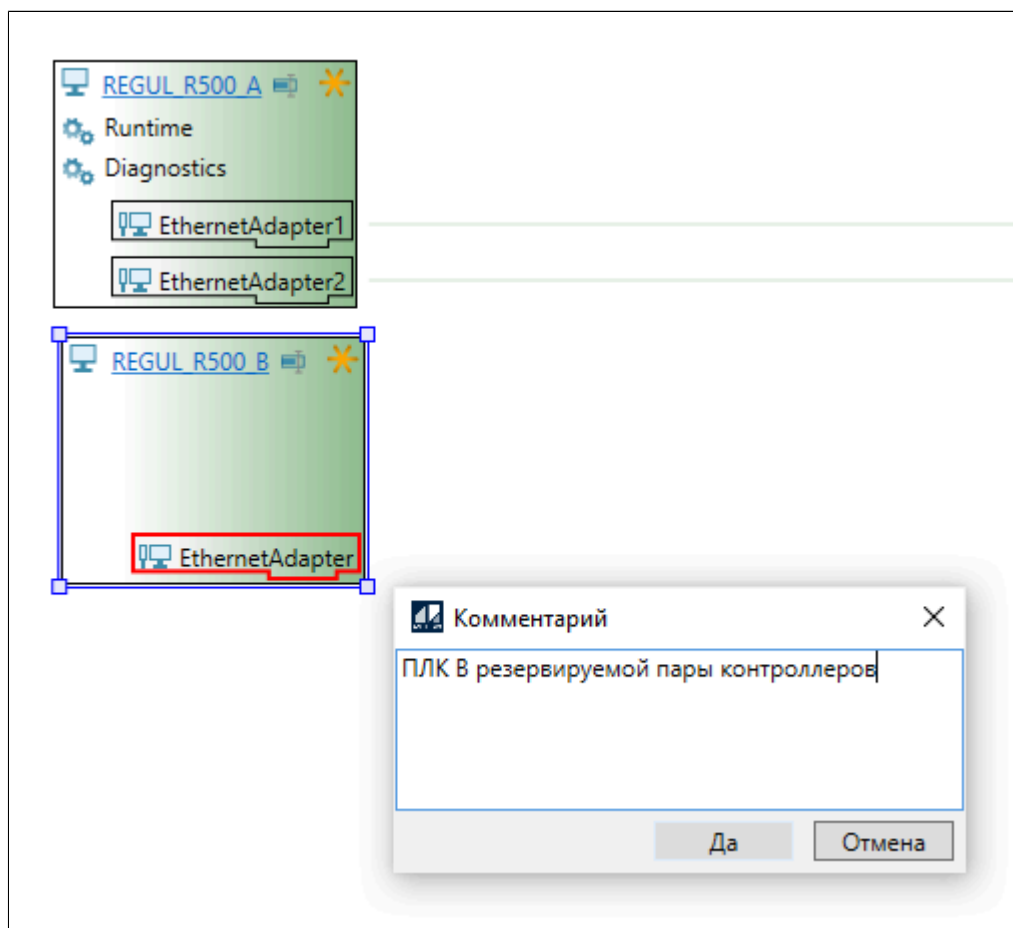


Имя элемента "Компьютер" должно соответствовать структурной схеме проекта (например, KKS).

### 3. Добавьте комментарий у объекта "Компьютер".



Комментарии к объектам добавляются с помощью контекстного меню "Комментарий".

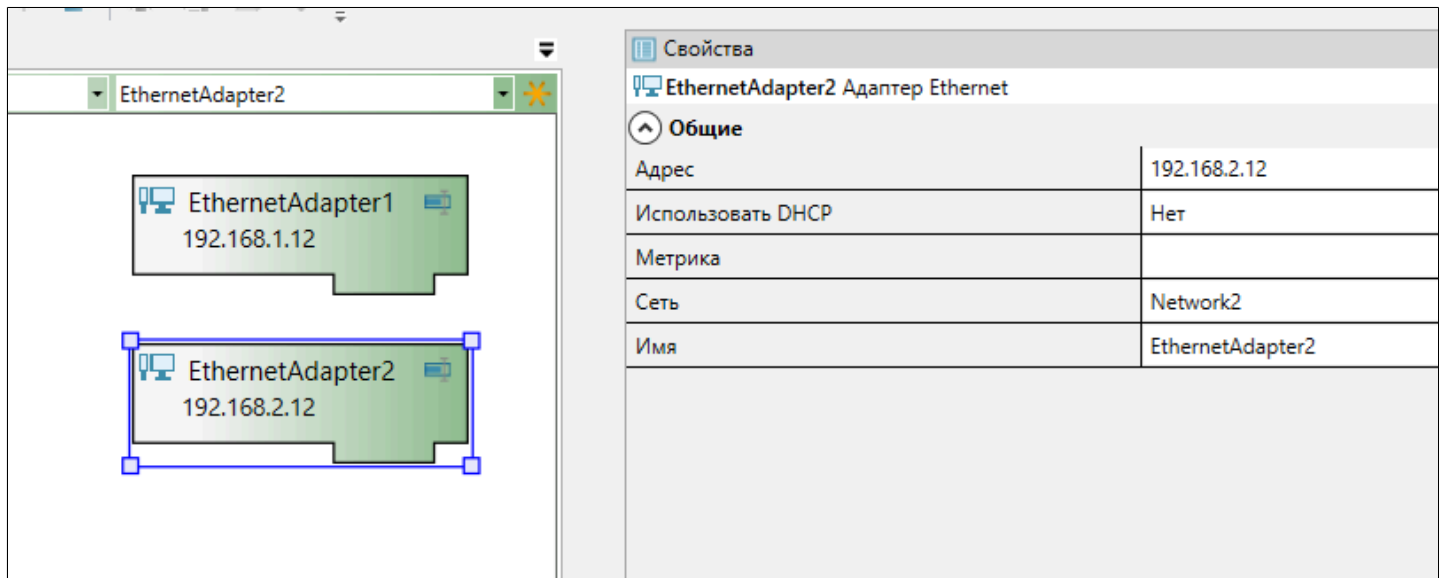


### 4. Перейдите в компонент "Компьютер" и перетащите в рабочую область из "Панели элементов" необходимое количество адаптеров Ethernet.



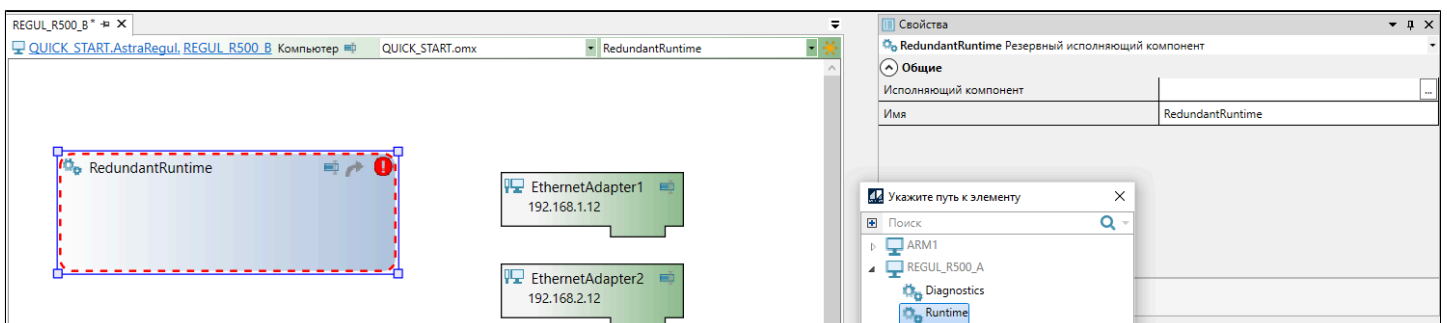
Рекомендуется использовать резервируемую линию связи (два адаптера Ethernet).

У каждого адаптера необходимо настроить свойства "Имя", "Адрес" и "Сеть". В свойстве "Адрес" укажите IP-адрес порта ПЛК. Сеть выбирается из выпадающего списка: для каждого адаптера своя сеть.



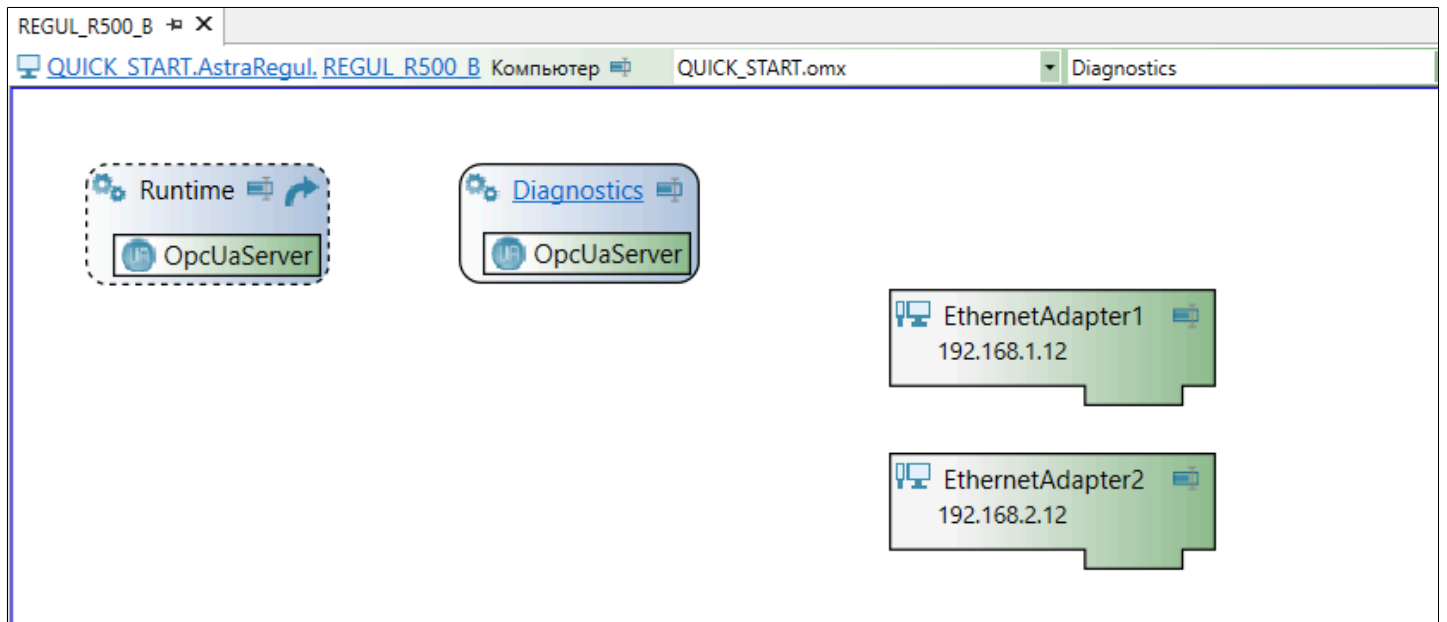
5. Перетащите в рабочую область элемент "Резервный исполняющий компонент" из "Панели элементов".

Выделите элемент и в окне "Свойства" задайте имя "Runtime", укажите путь к исполняющему компоненту ПЛК А резервной пары.

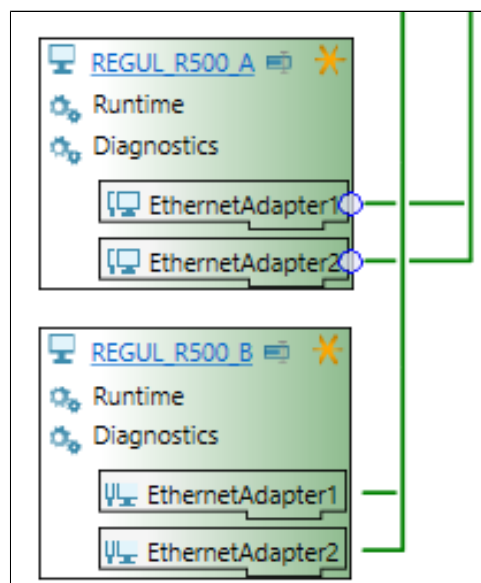


6. Перетащите в рабочую область элемент "Исполняющий компонент" из "Панели элементов".

Выделите элемент и в окне "Свойства" задайте имя "Diagnostics". Настройте компонент аналогично ПЛК А.



Описание резервного ПЛК в топологии завершено.

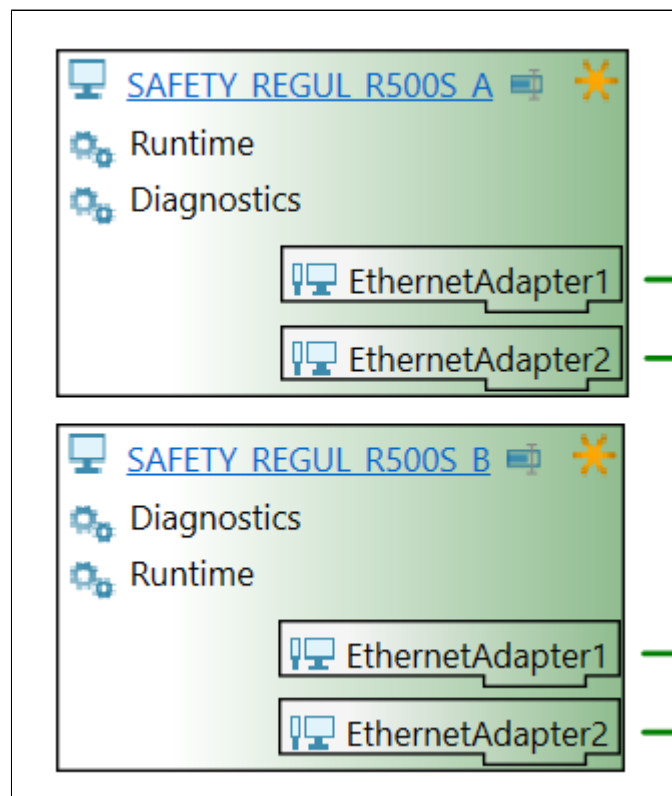


## 1.1.3.2.2.3. ПЛК SAFETY REGUL R500S

В качестве источника данных будет использоваться ПЛК SAFETY REGUL R500S.



ПЛК SAFETY REGUL R500S вместе с адаптерами Ethernet и исполняющими компонентами (Runtime и Diagnostics) добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).



### Ручное добавление ПЛК

Чтобы описать ПЛК REGUL в топологии вручную, необходимо выполнить следующие действия:

1. Перейдите в компонент "Astra.Domain" и перетащите в рабочую область из Панели элементов компонент "Компьютер".



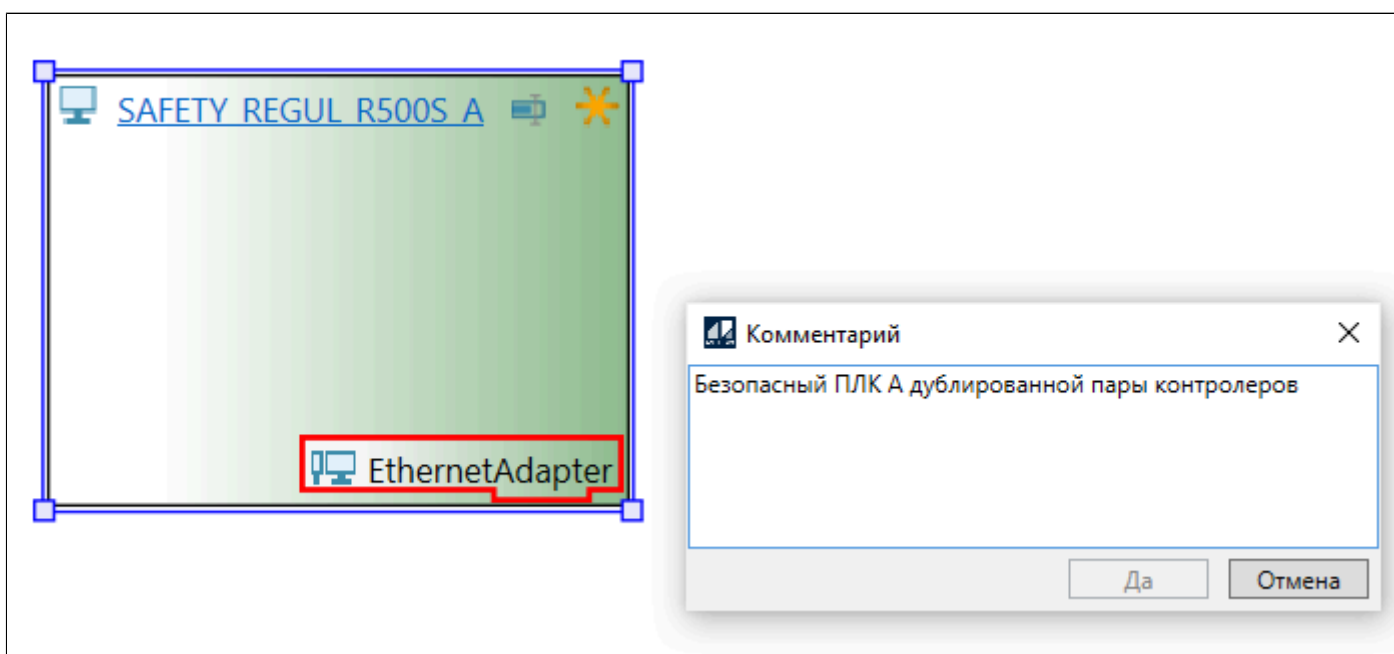
Компонент "Компьютер" - это компьютер или иное сетевое устройство (например, ПЛК), на котором нет компонентов AstraRegul, но с которыми они взаимодействуют.

2. Выделите элемент "Компьютер" и в окне "Свойства" задайте ему свойство "Имя".



Имя элемента "Компьютер" должно соответствовать структурной схеме проекта (например, KKS).

3. Добавьте комментарий у объекта "Компьютер".

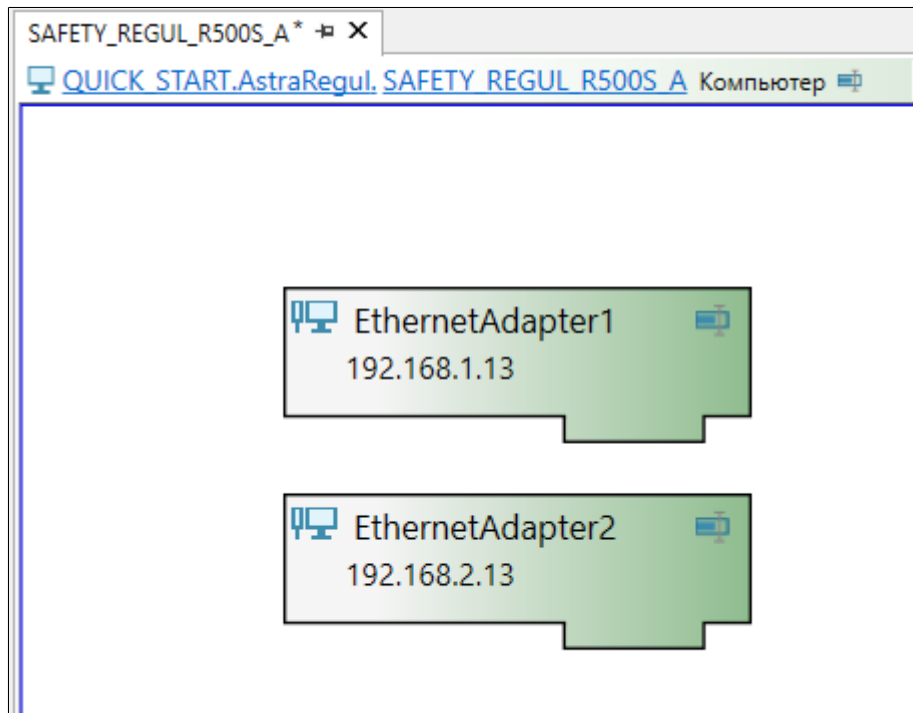


4. Перейдите в компонент "Компьютер" и перетащите в рабочую область из "Панели элементов" необходимое количество адаптеров "Ethernet".



Рекомендуется использовать резервируемую линию связи (два адаптера Ethernet).

У каждого адаптера необходимо настроить свойства "Имя", "Адрес" и "Сеть". В свойстве "Адрес" укажите IP-адрес порта ПЛК. Сеть выбирается из выпадающего списка: для каждого адаптера - своя сеть.



5. Перетащите в рабочую область два элемента "Исполняющий компонент" из "Панели элементов".

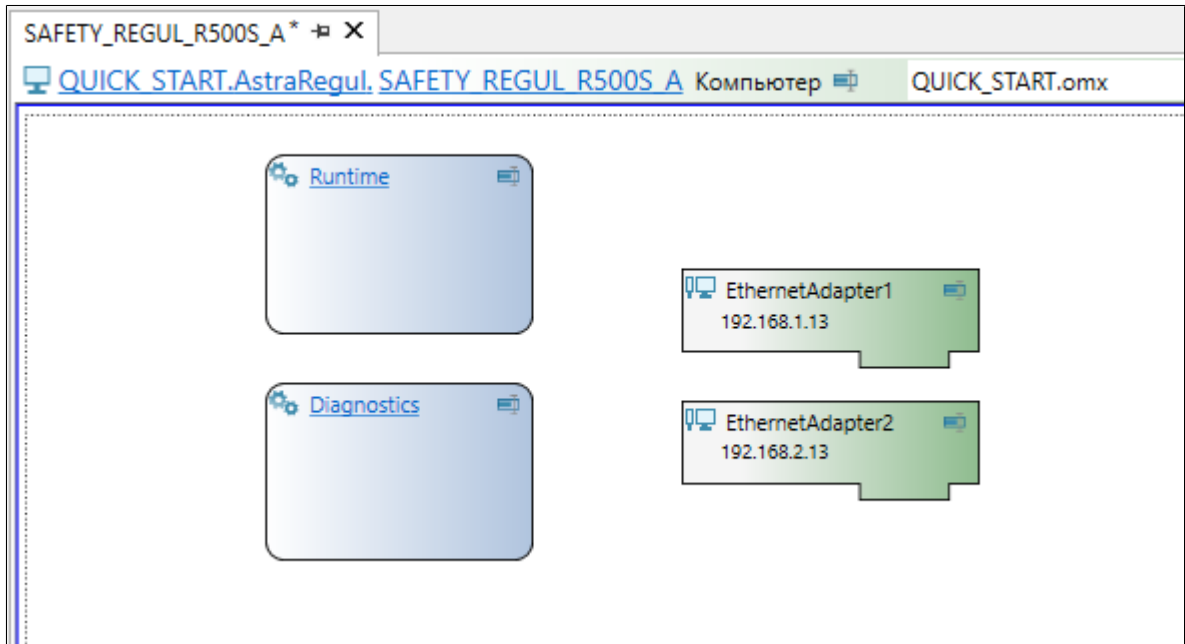
**Исполняющий компонент** – это любая сторонняя информационная система. Может иметь как программную реализацию (например, служба на компьютере), так и физическую (датчик или другое устройство).



Astra.AStudio не конфигурирует сторонние системы, но использует их описание при создании конфигураций для экземпляров Astra.Server, которые с ними взаимодействуют.

Первому исполняющему компоненту в окне Свойства задайте имя "Runtime" (для технологических данных), второму - "Diagnostics" (для диагностики ПЛК).





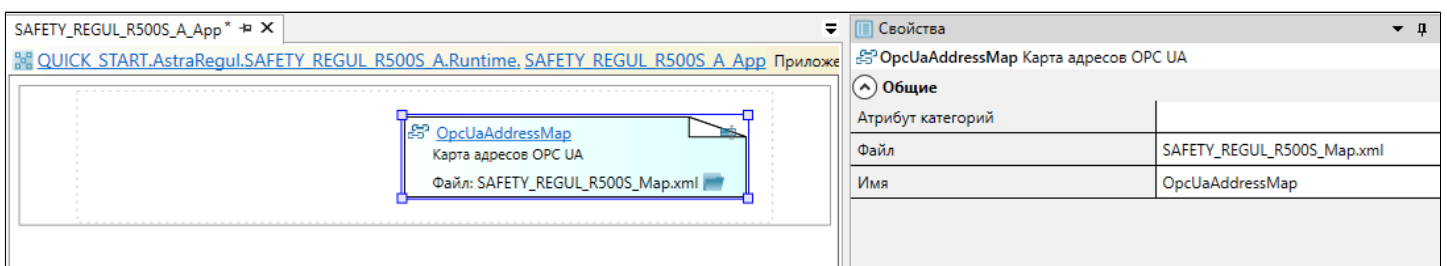
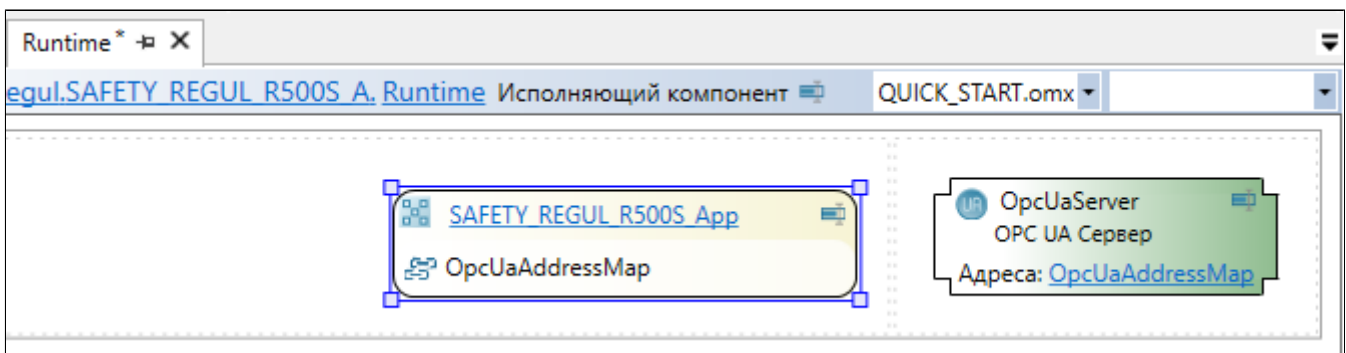
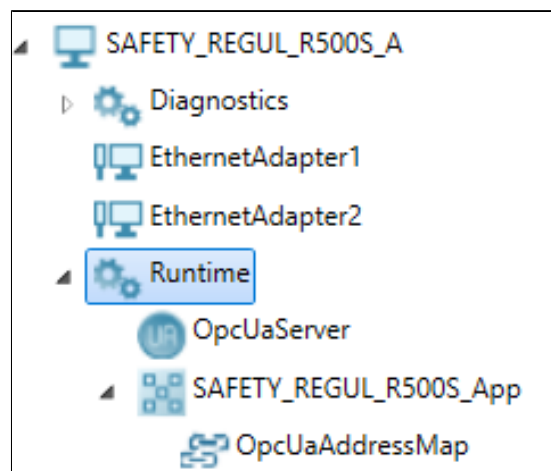
Далее необходимо создать приложения для каждого исполняющего компонента ПЛК.

## 1.1.3.2.2.3.1. Приложение Runtime

**Приложение** – это элемент, в котором описываются данные исполняющего компонента. В любом исполняющем компоненте может быть только одно приложение.



Приложение "SAFETY\_REGUL\_R500S\_App", компоненты "Карта адресов OPC UA" и "OPC UA Сервер" добавляется в приложение "Runtime" автоматически при выполнении [экспорта проекта в Astra.IDE](#).



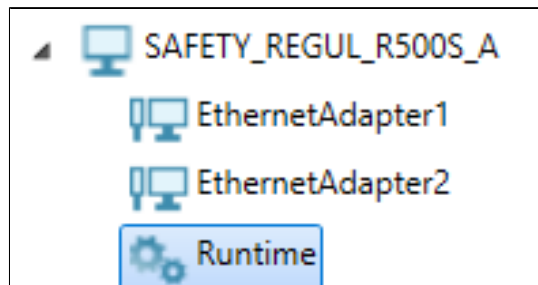


После экспорта необходимо создать [структуру дерева объектов](#) в приложении и [заполнить карту адресов](#). Все остальные компоненты настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление приложения Runtime

Чтобы добавить приложение вручную, выполните следующие действия:

1. Перейдите в компонент "Runtime" по пути "AstraRegul" → "[Имя ПЛК]" → "Runtime".



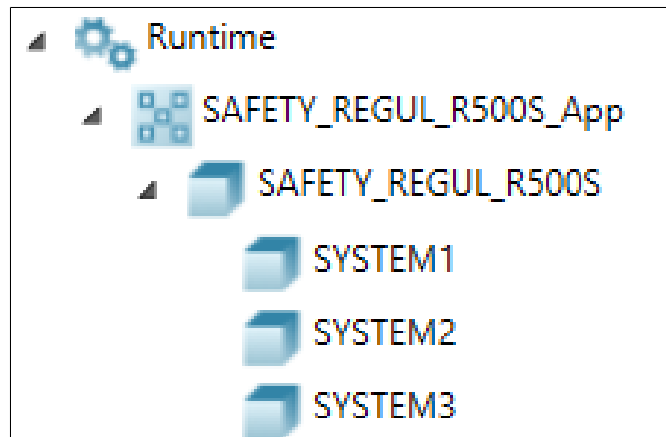
2. Перетащите в рабочую область из "Панели элементов" компонент "Приложение". В окне "Свойства" задайте "Имя".



Рекомендуется имя приложения задавать в формате [Имя ПЛК]\_App.


3. Перейдите в "Приложение" и создайте структуру дерева объектов, соответствующую текущим требованиям проекта.

Например, если на ПЛК приходят данные с трех подсистем, то структуру дерева объектов необходимо сформировать следующим образом:



Для формирования структуры объектов необходимо использовать элемент "Логический объект" из "Панели элементов".

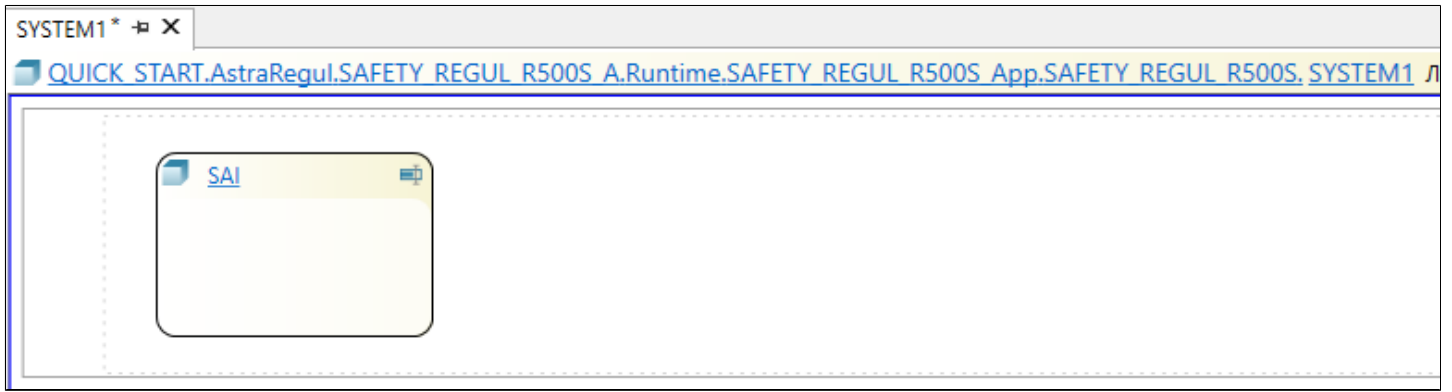
Логический объект представляет собой контейнер для других объектов (наподобие папки для файлов в проводнике).

 Структура объектов в виде дерева позволяет удобно фильтровать и искать данные в проекте.

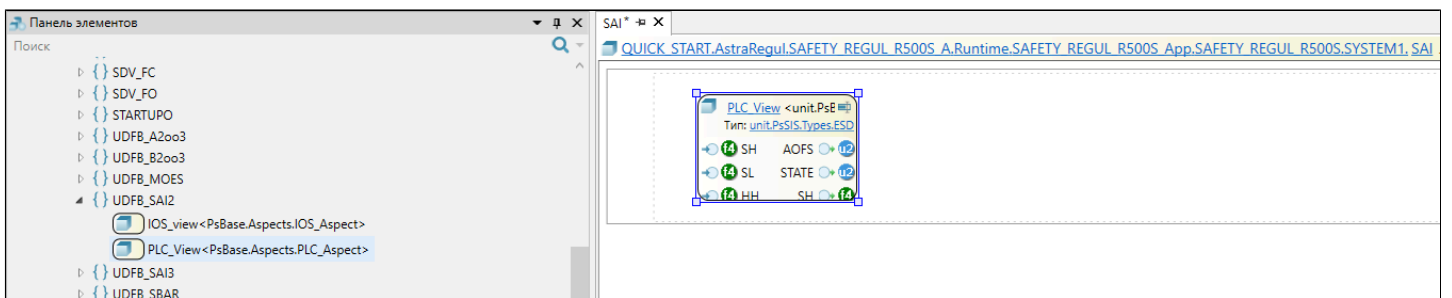
Внутри логического объекта ПЛК создайте логические объекты "SYSTEM1", "SYSTEM2", "SYSTEM3", которые будут обозначать каждую из возможных подсистем.



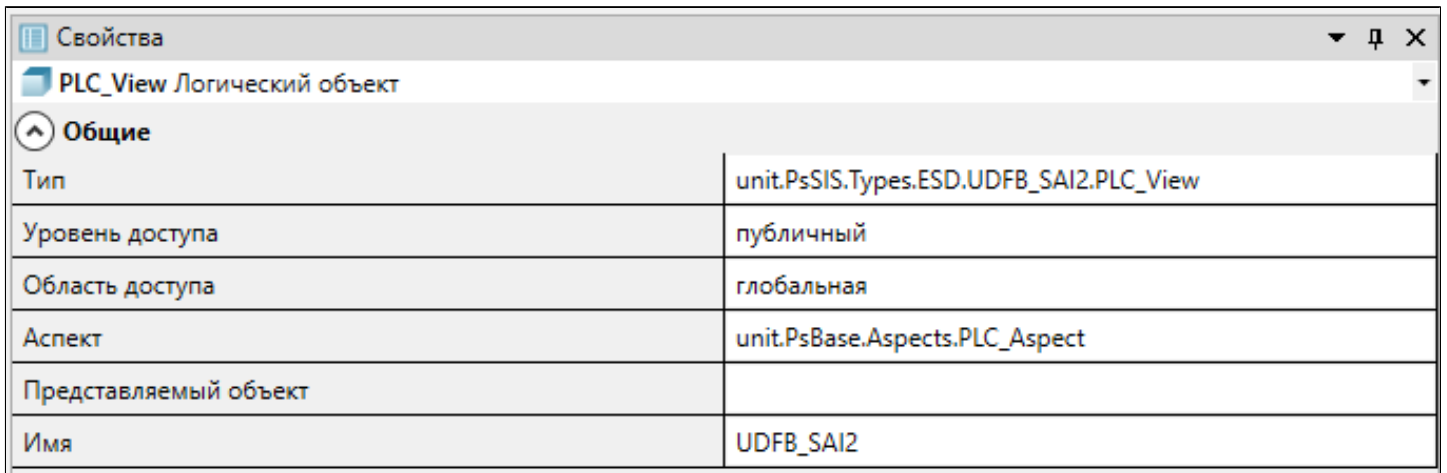
4. Для удобства рекомендуется группировать объекты по типу. Для объединения объектов одного типа необходимо использовать элемент "Логический объект" из панели элементов. Перейдите в логический объект "SYSTEM1" и создайте логический объект "SAI" для всех аналоговых входов.



5. Перейдите в логический объект "UDFB\_SAI2" и добавьте в рабочую область экземпляр типа "PLC\_View" библиотеки "PsSIS" из "Панели элементов".



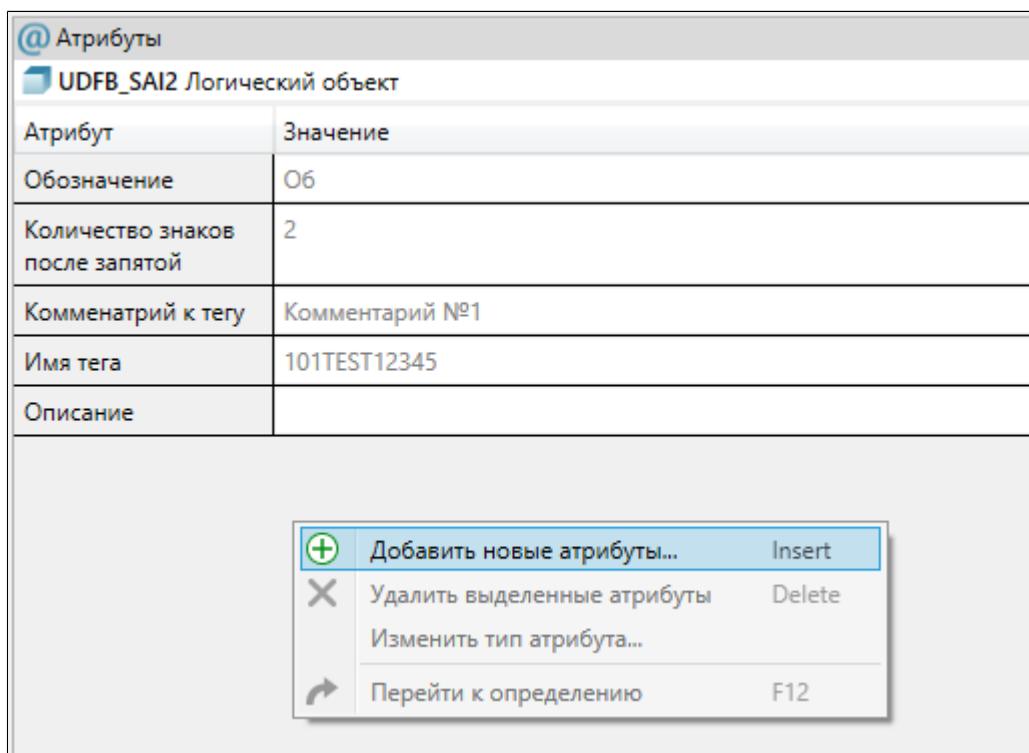
6. Выделите экземпляр и в редакторе "Свойства" задайте свойство "Имя" - "UDFB\_SAI2".



The screenshot shows a window titled 'Свойства' (Properties) for a 'PLC\_View Логический объект' (PLC\_View Logical object). The 'Общие' (General) tab is active, displaying a table of properties.

Property	Value
Тип	unit.PsSIS.Types.ESD.UDFB_SAI2.PLC_View
Уровень доступа	публичный
Область доступа	глобальная
Аспект	unit.PsBase.Aspects.PLC_Aspect
Представляемый объект	
Имя	UDFB_SAI2

7. В редакторе "Атрибуты" добавьте новый атрибут "KKS", нажав правой кнопкой мыши на пустую область и выбрав "Добавить новые атрибуты".



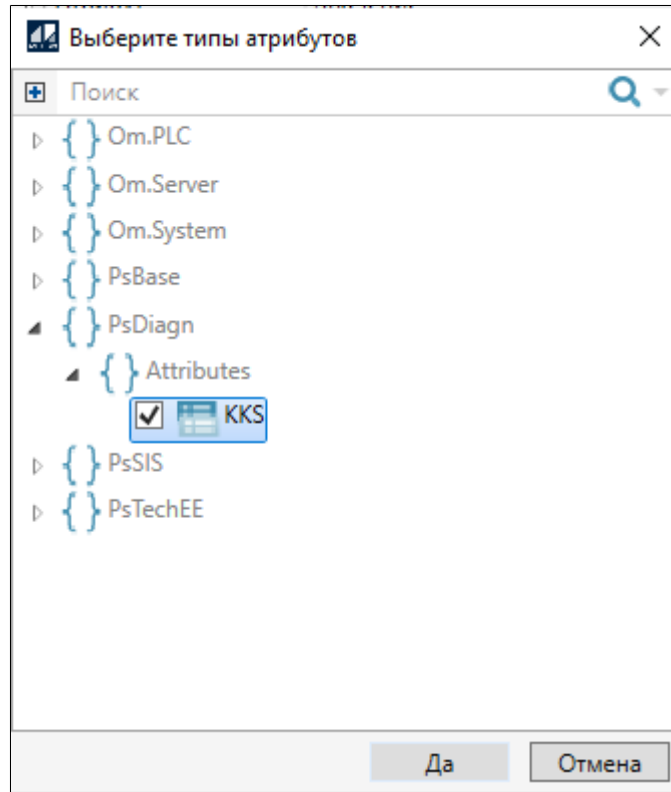
The screenshot shows the 'Атрибуты' (Attributes) editor for the 'UDFB\_SAI2 Логический объект' (UDFB\_SAI2 Logical object). It displays a table of attributes and a context menu.

Атрибут	Значение
Обозначение	Об
Количество знаков после запятой	2
Комментарий к тегу	Комментарий №1
Имя тега	101TEST12345
Описание	

Context menu options:

- + Добавить новые атрибуты... (Insert)
- X Удалить выделенные атрибуты (Delete)
- Изменить тип атрибута...
- ↶ Перейти к определению (F12)

8. В открывшемся окне добавьте новый атрибут "KKS" ("PsDiagn" → "Attributes" → "KKS") и нажмите кнопку "Да".



9. В редакторе "Атрибуты" заполните атрибуты "Описание", "Обозначение", "KKS", "Комментарий к тегу", "Имя тега" и "Количество знаков после запятой". Если значения атрибутов, заданные по умолчанию, были переопределены, они подсвечиваются синим цветом.

SAI\* X

QUICK START.AstraRegul.SAFETY\_REGUL\_R500S\_A.Runtime.SAFETY\_REGUL

UDFB\_SAI2 <unit.P...

Тип: unit.PsSIS.Types.ESD

SH AOFS

SL STATE

HH SH

Свойства

UDFB\_SAI2 Логический объект

Общие

Тип	unit.PsSIS.Types.ESD.UDFB_SAI2.PLC_View
Уровень доступа	публичный
Область доступа	глобальная
Аспект	unit.PsBase.Aspects.PLC_Aspect
Представляемый объект	
Имя	UDFB_SAI2

Свойства Внешние события

Атрибуты

UDFB\_SAI2 Логический объект

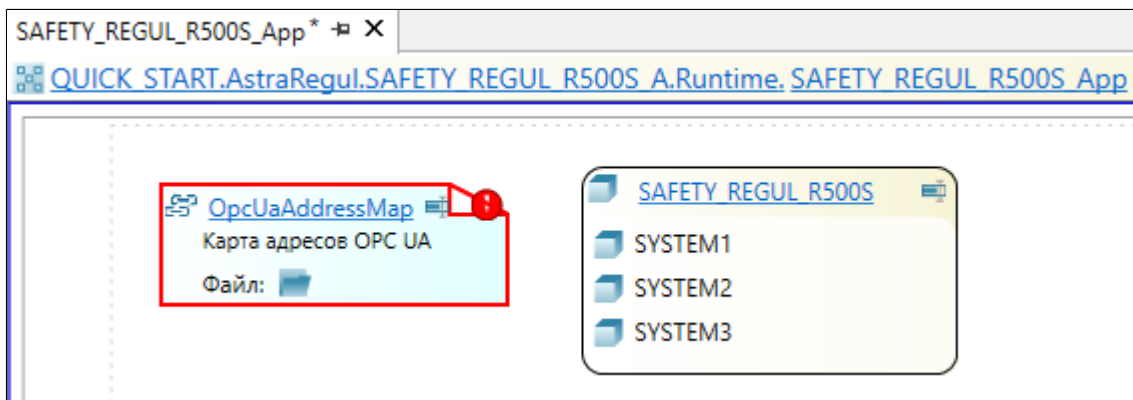
Атрибут	Значение
KKS	KKS
Обозначение	Град.
Количество знаков после запятой	2
Комментарий к тегу	Аналоговый вход с двумя порогами сигнализации
Имя тега	SAI2
Описание	UDFB_SAI2 - Аналоговый вход с двумя порогами сигнализации

10. Аналогичным образом добавьте другие объекты автоматизации: задвижки, клапаны, насосы и т.д.

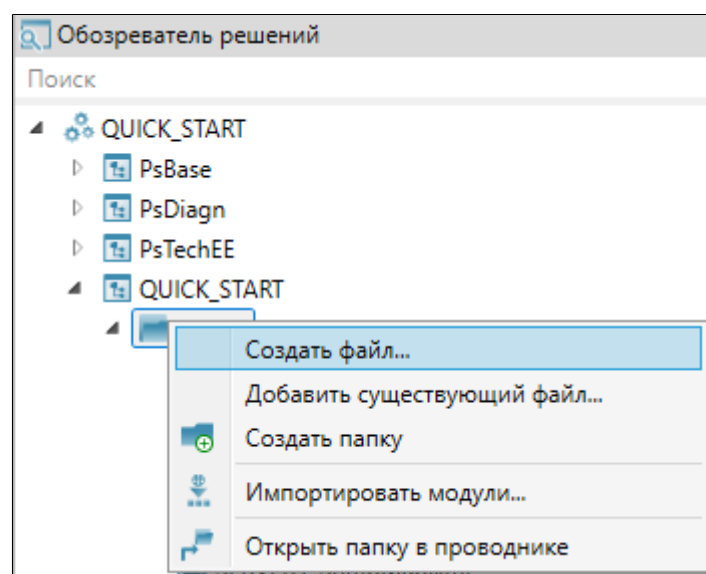


Процесс наполнения дерева объектами можно автоматизировать с помощью макросов.

11. Перейдите в приложение и перетащите в рабочую область из "Панели элементов" компонент "Карта адресов OPC UA".

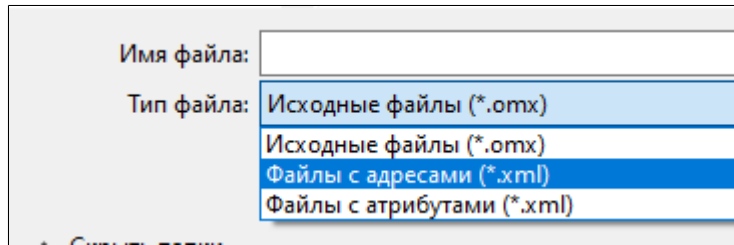


12. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." из контекстного меню объекта "Файлы" в Обозревателе решений.

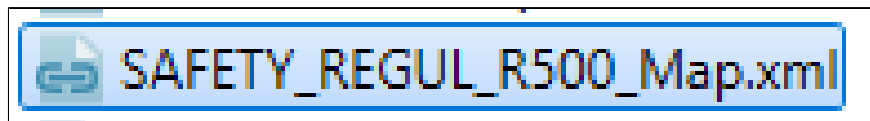


В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".

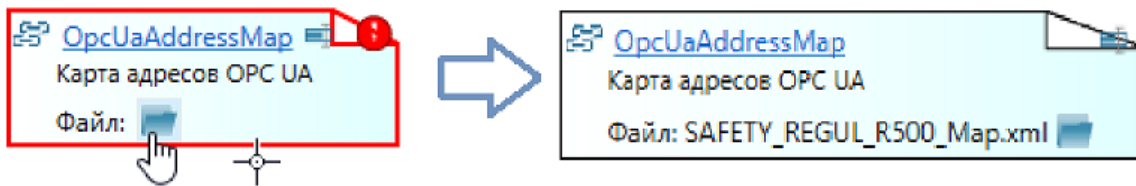




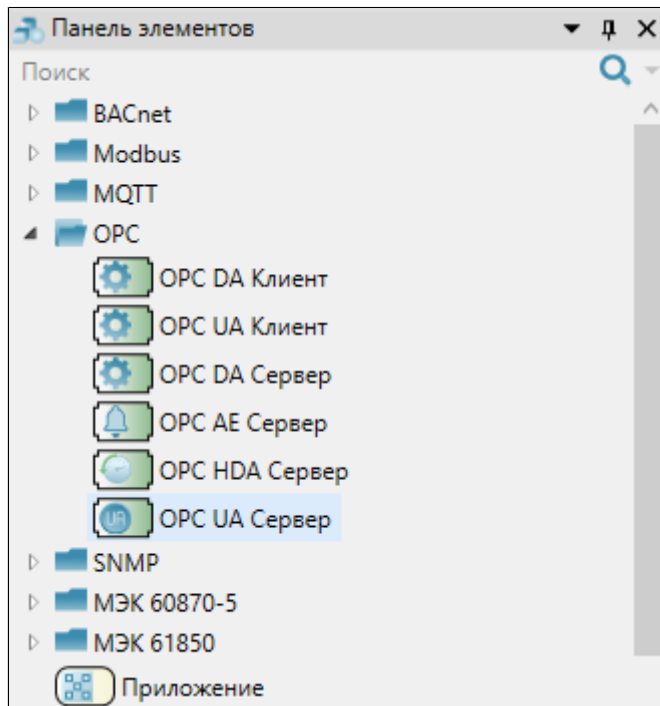
Имя файла рекомендуется задать в формате [Имя ПЛК]\_Map.



13. Перейдите в Приложение. У объекта "Карта адресов OPC UA" укажите созданный файл.



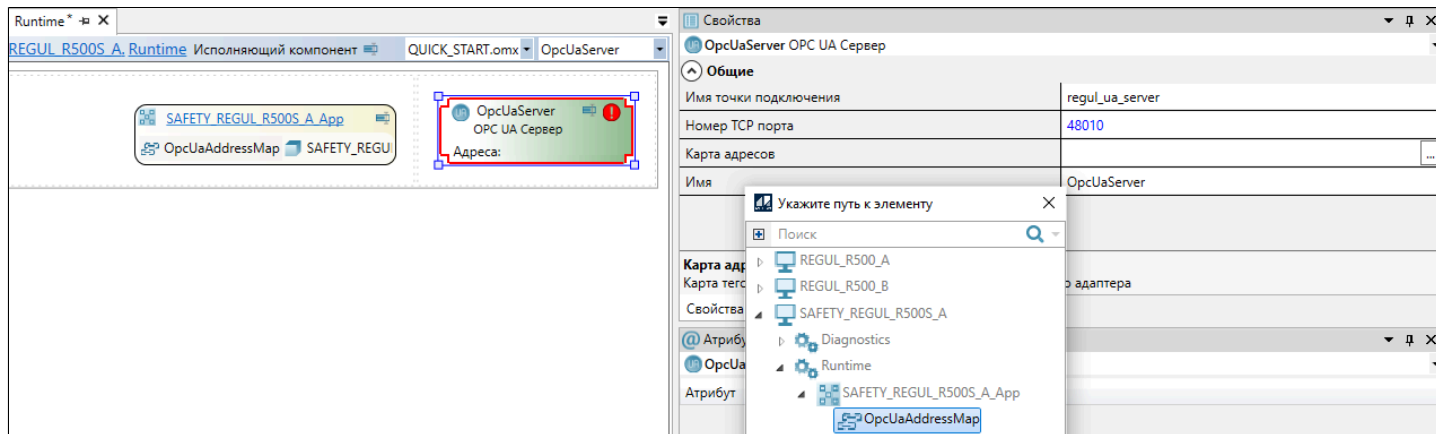
14. Перейдите в исполняющий компонент "Runtime" и добавьте в рабочую область из "Панели элементов" компонент "OPC UA Сервер".



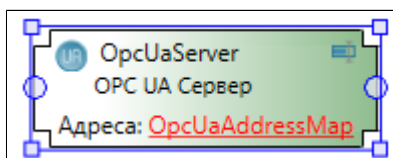
15. Выделите компонент OPC UA Сервер и в окне "Свойства" укажите настройки OPC UA сервера на ПЛК SAFETY REGUL:

- Имя точки подключения: regul\_ua\_server.
- Номер TCP порта: 48010 (порт по умолчанию для ПЛК SAFETY REGUL).

Для свойства Карта адресов укажите путь к карте адресов приложения.



16. Перейдите по ссылке в редактор карты адресов.



## 17. Заполните карту адресов.

› Привязка: указывает, происходит ли чтение/запись переменной.

› Адресное пространство ПЛК REGUL:

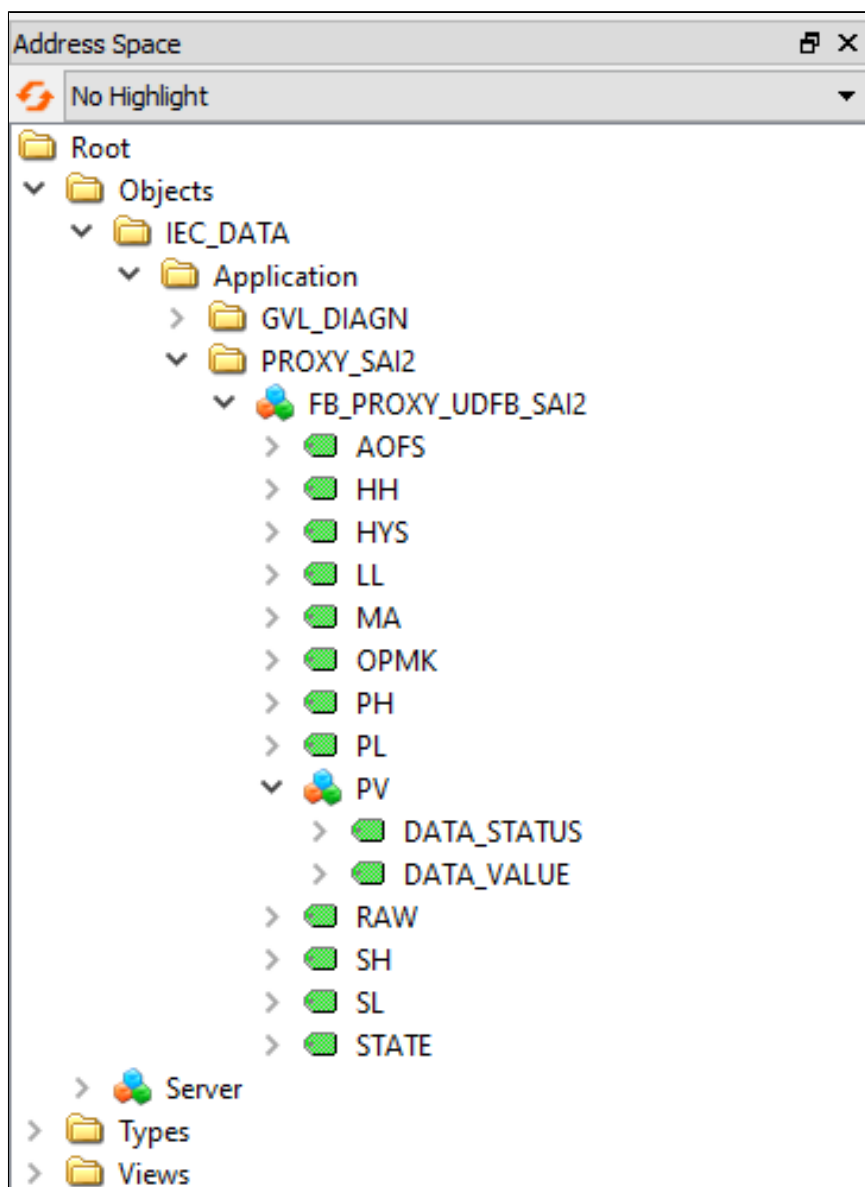
urn:ProsoftSystems:regul\_ua\_server:iec\_data

› Тип идентификатора: Строковый.

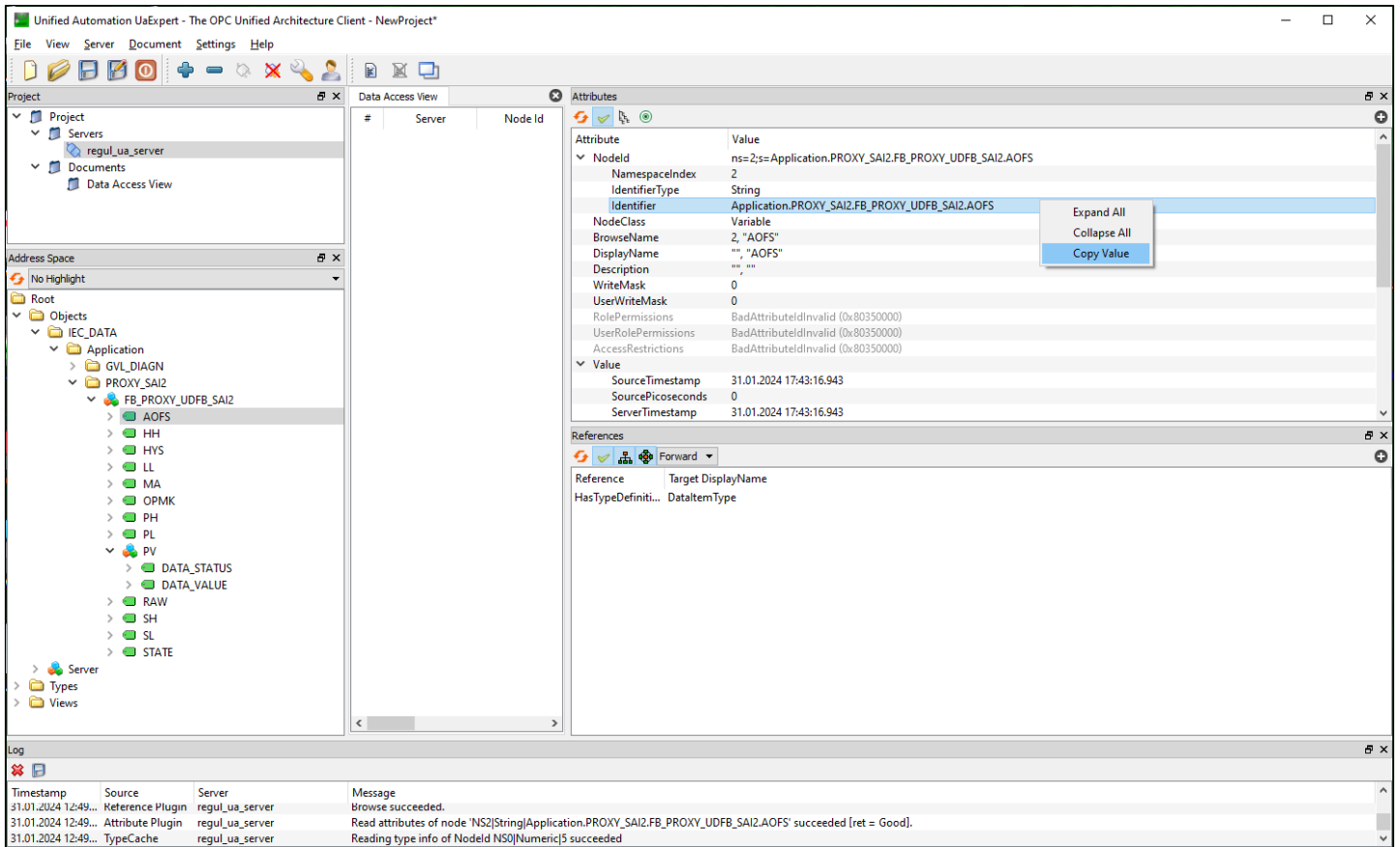
› Идентификатор узла: полный путь к переменной в адресном пространстве сервера OPC UA (проект Astra.IDE).

Для заполнения поля "Идентификатор узла" необходимо воспользоваться утилитой [UaExpert](#).

18. Откройте утилиту UA Expert и раскройте дерево сигналов: "IEC\_DATA" → "Application" → "PROXY\_SAI2" → "FB\_PROXY\_UDFB\_SAI2" .



19. Выберите переменную "AOFS". В окне "Attributes" скопируйте значение поля "Identifier", воспользовавшись контекстным меню или нажмите сочетание клавиш "Ctrl+C".



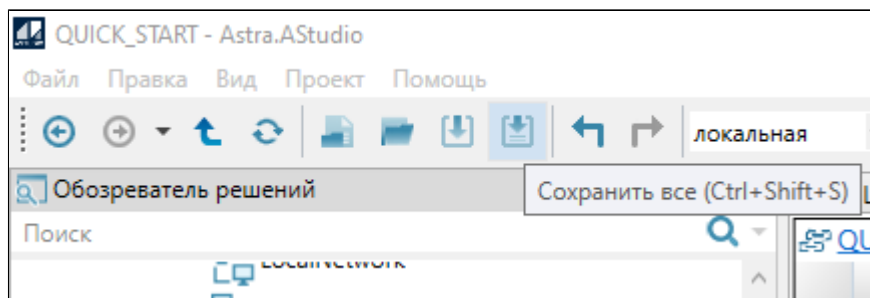
20. Вставьте скопированное значение в поле "Идентификатор узла" карты адресов для переменной "AOFS".

Сигнал	Тип	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.AOFS	uint2	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.MAIN.FB_PROXY_UDFB_SAI2.AOFS

21. Аналогичным образом заполните карту адресов для всех сигналов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.AOFS	uint2	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.AOFS
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.STATE	uint2	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.STATE
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.SH	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.SH
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.SL	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.SL
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.PV.DATA_VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.PV.DATA_VALUE
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.PV.DATA_STATUS	int4	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.PV.DATA_STATU
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.RAW	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.RAW
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.OPMK	int4	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.OPMK
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.HH	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.HH
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.MA	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.MA
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.PH	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.PH
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.PL	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.PL
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.LL	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.LL
SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2.HYS	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.PROXY_SAI2.FB_PROXY_UDFB_SAI2.HYS

22. Сохраните проект, нажав кнопку "Сохранить (Ctrl+Shift+S)".

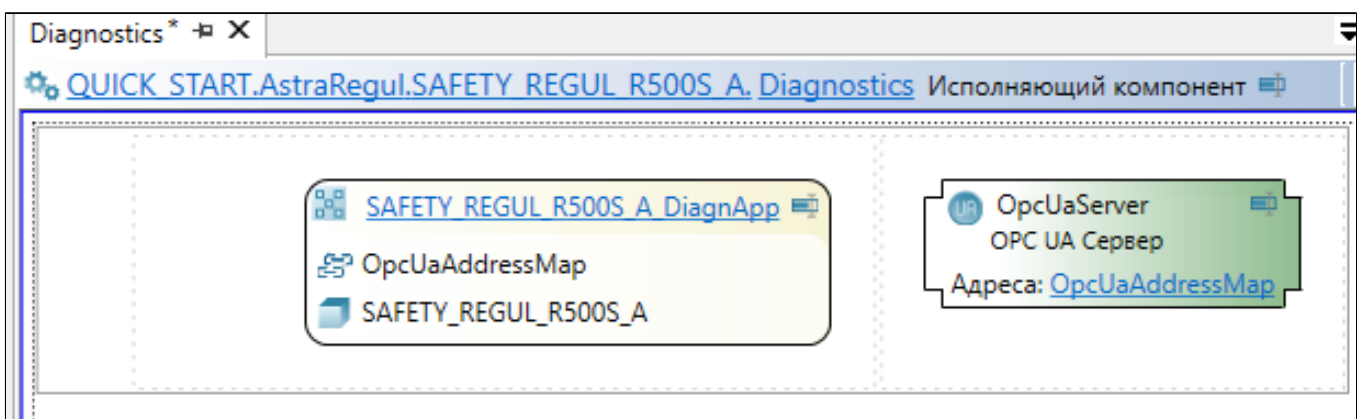
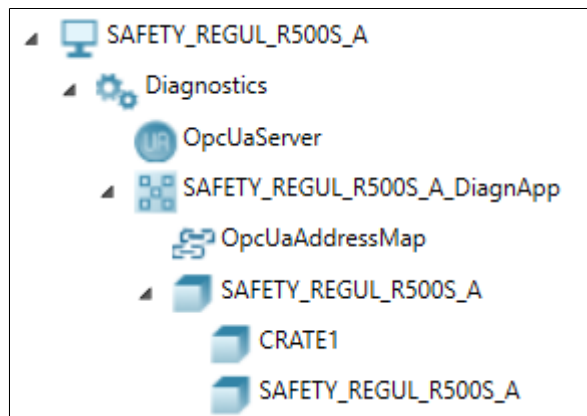


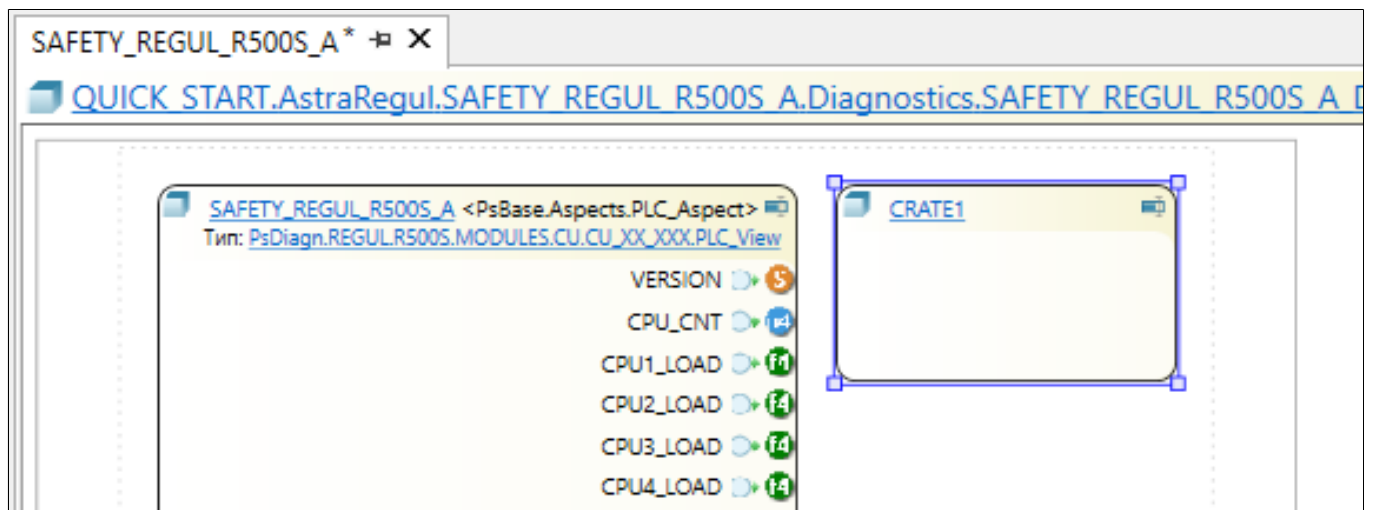
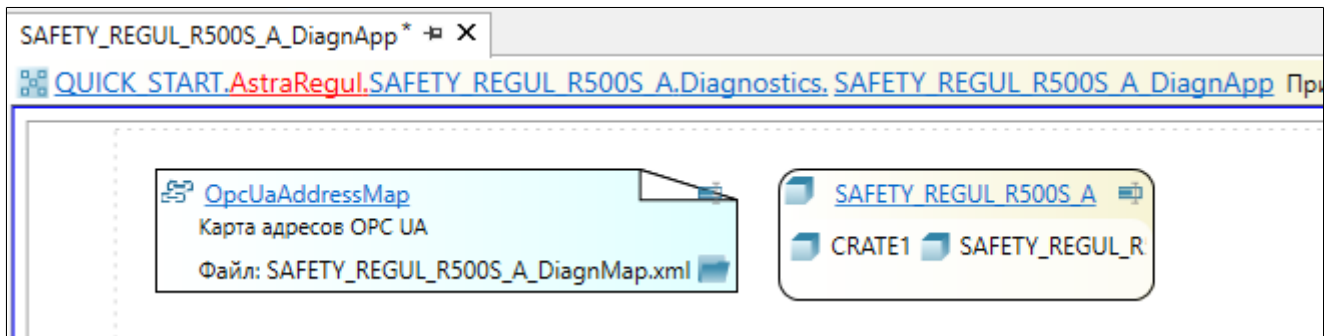
## 1.1.3.2.2.3.2. Приложение Diagnostics


**Приложение** – это элемент, в котором описываются данные исполняющего компонента. В любом исполняющем компоненте может быть только одно приложение.



Приложение "SAFETY\_REGUL\_R500S\_[A/B]\_DiagnApp" вместе с объектом контроллера и логическими элементами крейтов, а также компоненты "Карта адресов OPC UA" и "OPC UA Сервер" добавляется в приложение "Diagnostics" автоматически при выполнении [экспорта проекта в Astra.IDE](#).



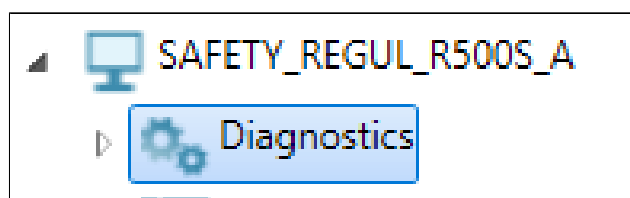



 После экспорта необходимо создать структуру дерева объектов в [логических объектах крейтов](#) в приложении и [заполнить карту адресов](#). Все остальные компоненты настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление приложения Diagnostics

Чтобы добавить приложение вручную, необходимо выполнить следующие действия:

1.Перейдите в компонент "Diagnostics" по пути "AstraRegul" → "[Имя ПЛК]" → "Diagnostics".





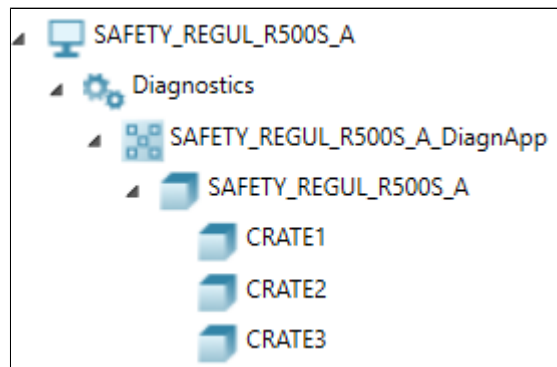
2. Перетащите в рабочую область из Панели элементов компонент "Приложение". В окне "Свойства" задайте "Имя".



Рекомендуется задавать имя приложения в формате [Имя ПЛК]\_[A/B]\_DiagnApp.

3. Перейдите в Приложение и создайте структуру дерева объектов, соответствующую конфигурации ПЛК.

Например, если конфигурация ПЛК состоит из трех крейтов, то структуру дерева объектов необходимо сформировать следующим образом:



Для формирования структуры объектов используйте элемент "Логический объект" из "Панели элементов".

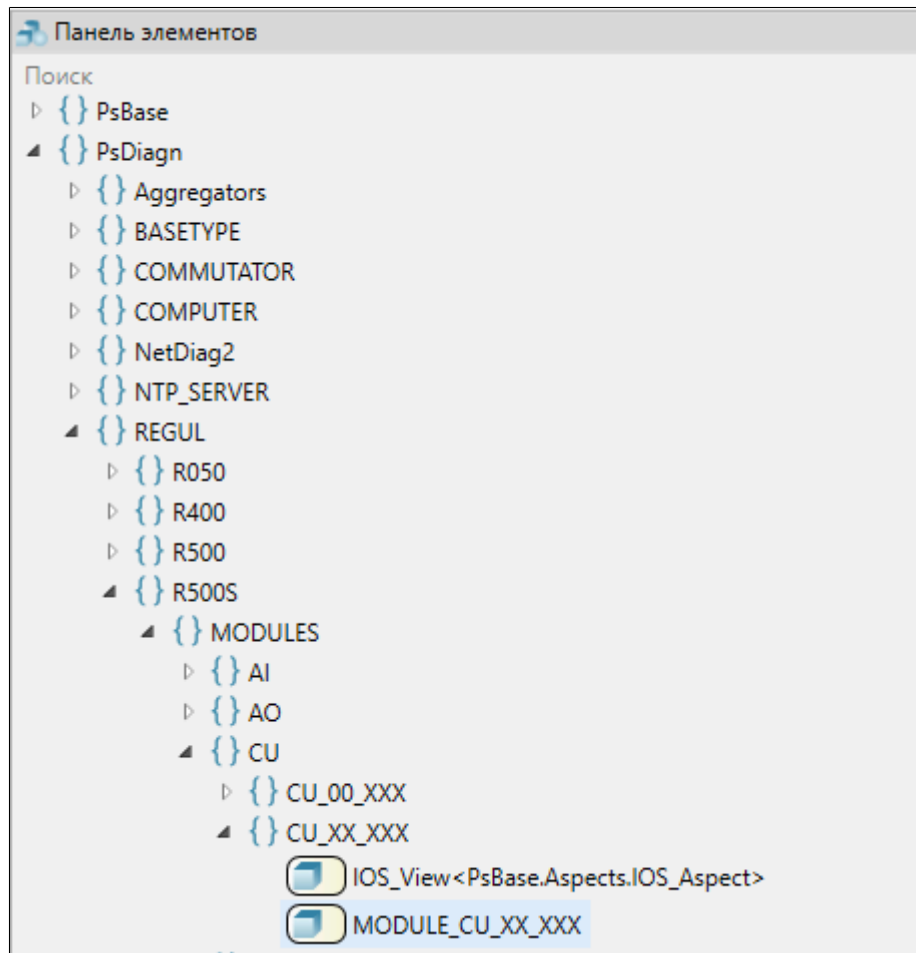
Логический объект представляет собой контейнер для других объектов (наподобие папки для файлов в проводнике).



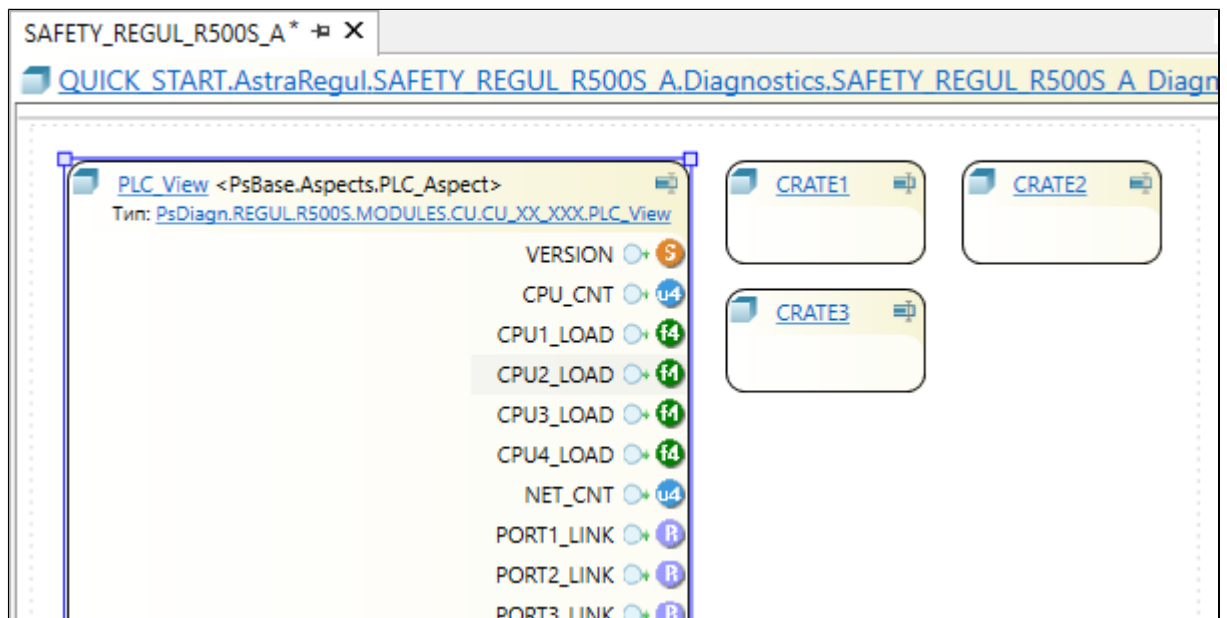
Структура объектов в виде дерева позволяет удобно фильтровать и искать данные в проекте.

4. Далее необходимо наполнить структуру объектами.

Перетащите объект "MODULE\_CU\_XX\_XXX" (модуль ЦП) из библиотеки "PsDiagn" "Панели элементов" в рабочую область ПЛК А. Модуль ЦП рекомендуется размещать в структуре объектов отдельно от крейтов.



5. Структура будет выглядеть следующим образом:



6. Задайте имя и атрибуты для объекта. Если значения атрибутов, заданные по умолчанию, были переопределены, они подсвечиваются синим цветом.

Свойства

CU\_00\_851 Логический объект

Общие

Тип	PsDiagn.REGUL.R500S.MODULES.CU.CU_XX_XXX.PLC_View
Уровень доступа	публичный
Область доступа	глобальная
Аспект	PsBase.Aspects.PLC_Aspect
Представляемый объект	
Имя	CU_00_851

Свойства | Внешние события

Атрибуты

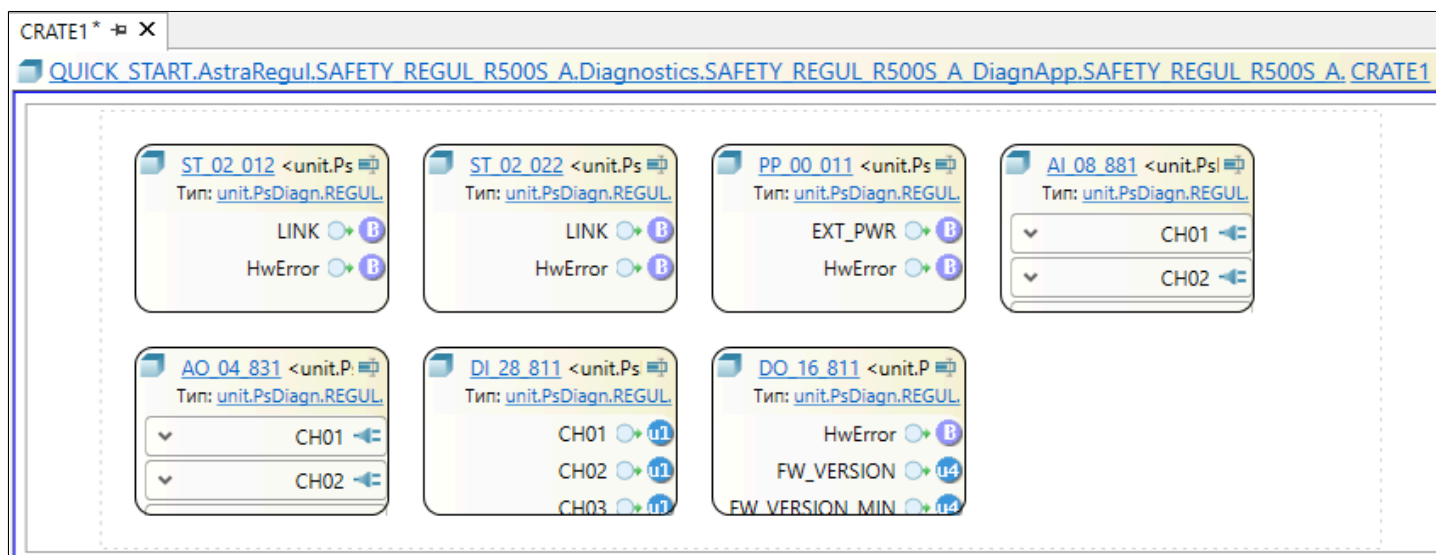
CU\_00\_851 Логический объект

Атрибут	Значение
Название	CU_00_851
Описание	CU_00_851
KKS	CU_00_851
Обозначение	CU_00_851

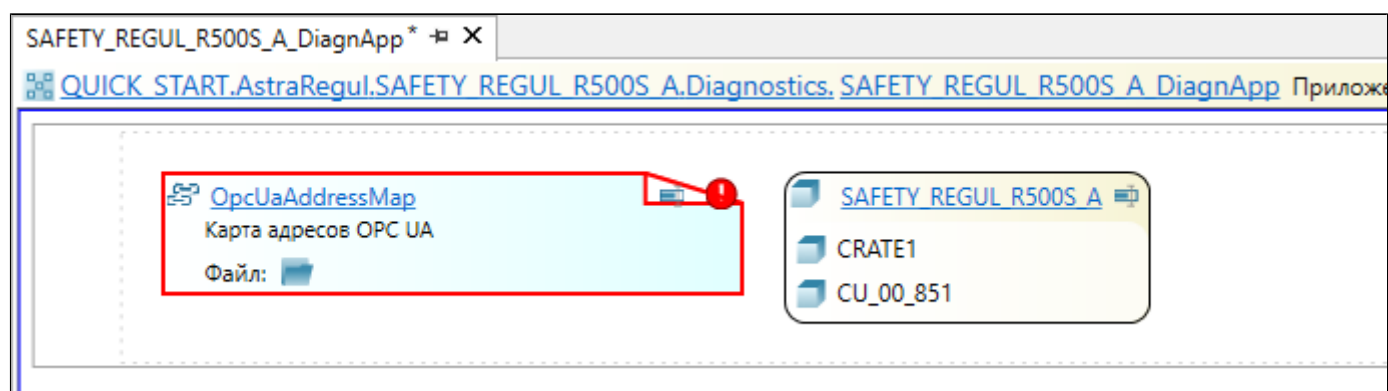


При необходимости атрибуты "Название", "Описание", "Обозначение" и "KKS" можно изменить.

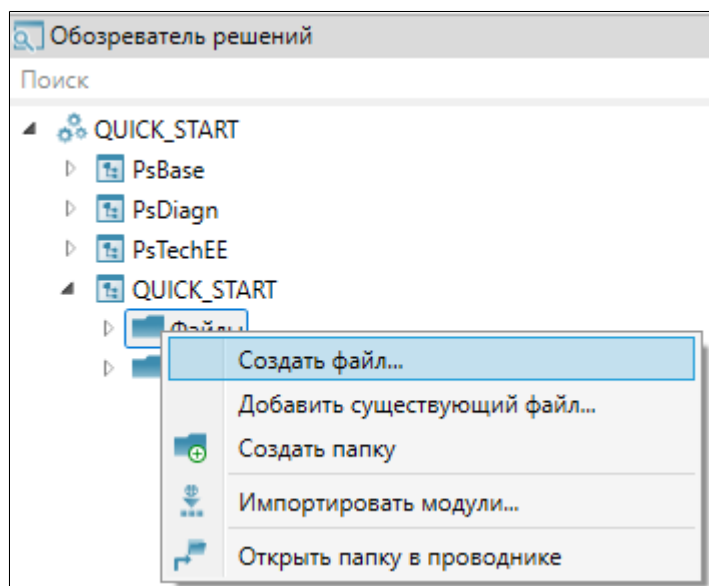
7. Из библиотеки "PsDiagn" Панели элементов перетащите в рабочую область "CRATE1", "CRATE2", "CRATE3" остальные модули в соответствии с конфигурацией ПЛК.



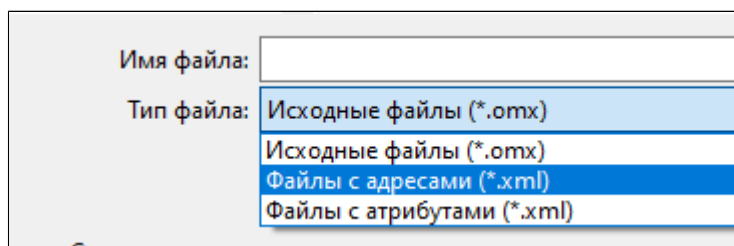
8. После формирования дерева объектов перейдите в "Приложение" и перетащите в рабочую область из "Панели элементов" компонент "Карта адресов OPC UA".



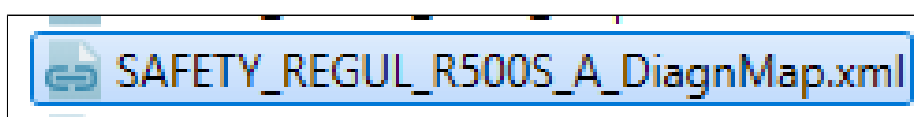
9. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." в контекстном меню объекта "Файлы" в Обозревателе решений.



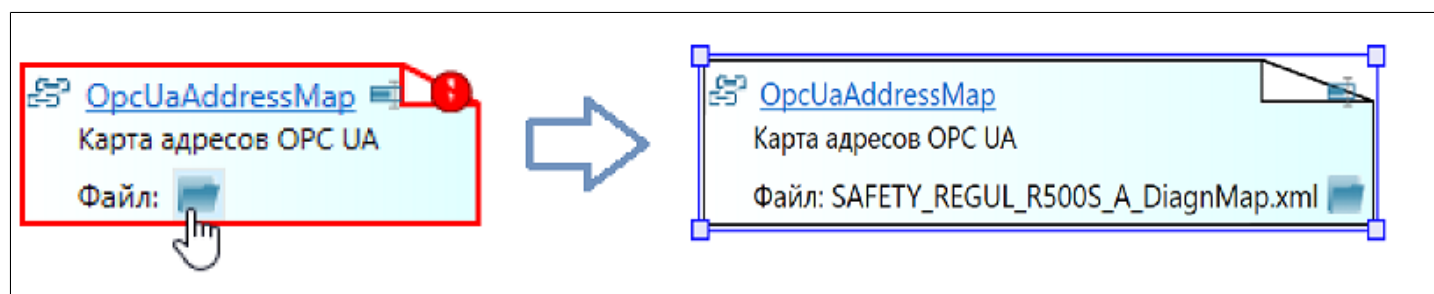
В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".



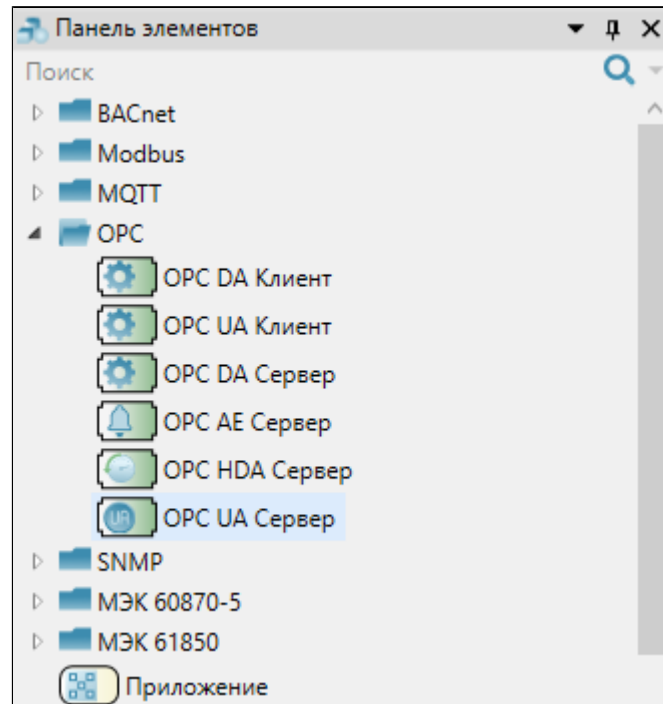
Имя файла рекомендуется задавать в формате [Имя ПЛК]\_[A/B]\_DiagnMap.



10. Перейдите в Приложение. У объекта "Карта адресов OPC UA" укажите созданный файл.



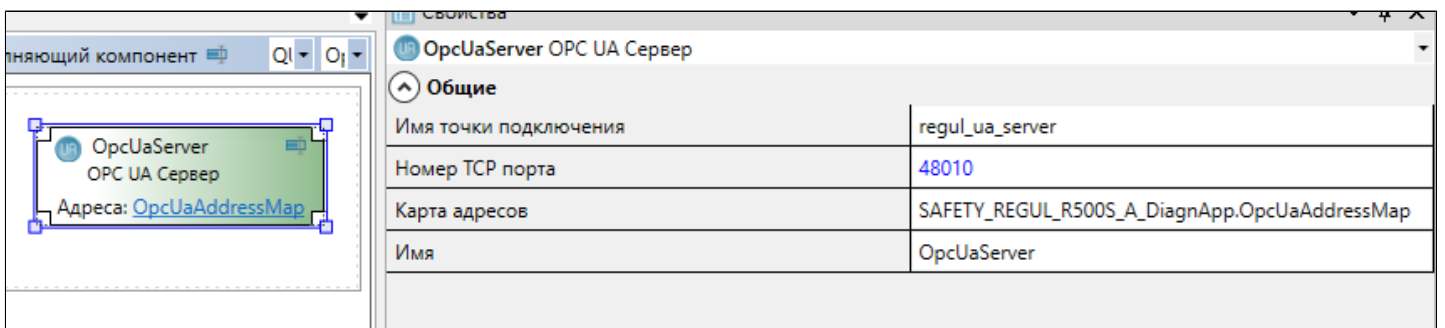
11. Далее перейдите в исполняющий компонент "Runtime" и добавьте в рабочую область из Панели элементов компонент "OPC UA Сервер".



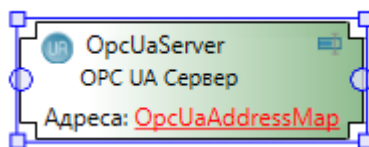
12. Выберите компонент "OPC UA Сервер" и в окне "Свойства" укажите настройки OPC UA сервера на ПЛК SAFETY REGUL:

- › Имя точки подключения: regul\_ua\_server.
- › Номер TCP порта: 48010 (по умолчанию для ПЛК SAFETY REGUL).

Для свойства "Карта адресов" укажите путь к карте адресов приложения.



13. Перейдите по ссылке в редактор карты адресов.

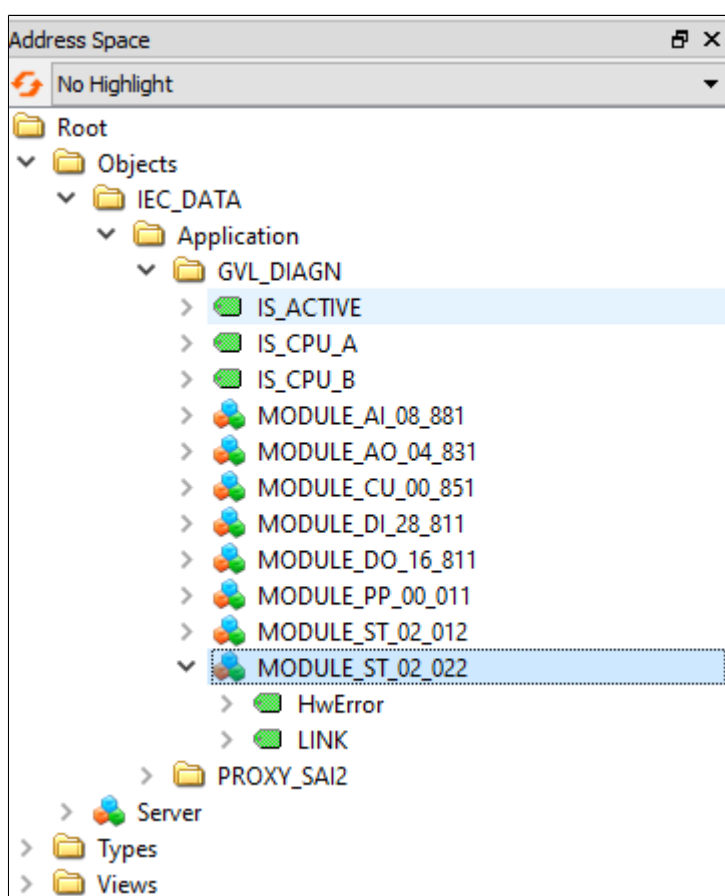


14. Заполните карту адресов.

- › Привязка: указывает, происходит ли чтение/запись переменной.
- › Адресное пространство ПЛК REGUL: `urn:ProsoftSystems:regul_ua_server:iec_data`
- › Тип идентификатора: Строковый.
- › Идентификатор узла: полный путь к переменной в адресном пространстве сервера OPC UA (проект Astra.IDE).

Для заполнения поля "Идентификатор узла" необходимо воспользоваться утилитой [UaExpert](#).

15. Откройте утилиту UA Expert и раскройте дерево сигналов сервера: "IEC\_DATA" → "Application" → "GVL\_DIAGN" → "MODULE\_ST\_02\_22".



16. Выберите любую переменную модуля "ST\_02\_012". В окне "Attributes" скопируйте значение поля "Identifier" с помощью контекстного меню или нажмите сочетание клавиш "Ctrl+C".

The screenshot shows the SIMATIC Manager interface with three main panes:

- Project:** Shows a tree structure under 'Servers' > 'regul\_ua\_server' > 'Documents' > 'Data Access View'.
- Address Space:** Shows a tree structure under 'Objects' > 'IEC\_DATA' > 'Application' > 'GVL\_DIAGN' > 'MODULE\_ST\_02\_022' > 'LINK'.
- Data Access View:** A table with columns '#', 'Server', and 'Node Id'.
- Attributes:** A table showing the properties of the selected variable. The 'Identifier' is 'Application.GVL\_DIAGN.MODULE\_ST\_02\_022.LINK'.

Attribute	Value
NodeId	ns=2;s=Application.GVL_DIAGN.MODULE_ST_02_022.LINK
NamespaceIndex	2
IdentifierType	String
Identifier	Application.GVL_DIAGN.MODULE_ST_02_022.LINK
NodeClass	Variable
BrowseName	2, "LINK"
DisplayName	""; "LINK"
Description	""; ""
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeValue (0x80350000)
UserRolePermissions	BadAttributeValue (0x80350000)
AccessRestrictions	BadAttributeValue (0x80350000)
Value	
SourceTimestamp	31.01.2024 18:15:55.886
SourcePicoseconds	0
ServerTimestamp	31.01.2024 18:15:55.886

Reference	Target DisplayName
HasTypeDefiniti...	DatatermType



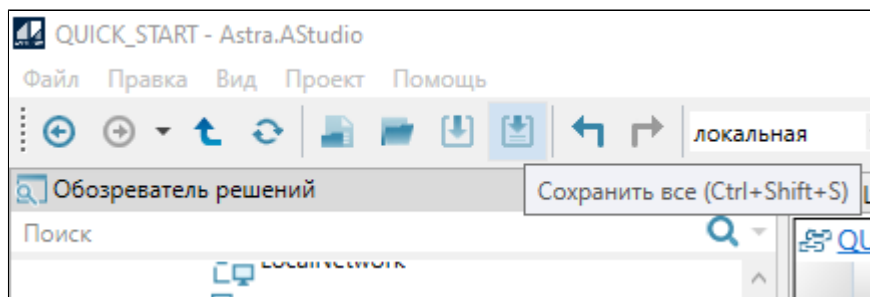
17. Вставьте скопированное значение в поле "Идентификатор узла" карты адресов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла
SAFETY_REGUL_R500S_A_CRATE1.ST_02_022.LINK	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_022.LINK

18. Аналогичным образом заполните поля у всех сигналов карты адресов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла
SAFETY_REGUL_R500S_A_CRATE1.ST_02_022.LINK	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_022.LINK
SAFETY_REGUL_R500S_A_CRATE1.ST_02_022.HwError	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_022.HwError
SAFETY_REGUL_R500S_A_CRATE1.ST_02_012.LINK	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_012.LINK
SAFETY_REGUL_R500S_A_CRATE1.ST_02_012.HwError	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_ST_02_012.HwError
SAFETY_REGUL_R500S_A_CRATE1.PP_00_011.EXT_PWR	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_PP_00_011.EXT_PWR
SAFETY_REGUL_R500S_A_CRATE1.PP_00_011.HwError	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_PP_00_011.HwError
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH01.VALID	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH01.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH02.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH02.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH02.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH02.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH03.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH03.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH03.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH03.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH04.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH04.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH04.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH04.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH05.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH05.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH05.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH05.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH06.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH06.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH06.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH06.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH07.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH07.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH07.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH07.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH08.VALUE	float	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH08.VALUE
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.CH08.VALID	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.CH08.VALID
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.HwError	bool	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.HwError
SAFETY_REGUL_R500S_A_CRATE1.AI_08_881.FW_VERSION	uint4	непосредствен	urn:ProsoftSystems:regul_ua_serveriec_data	Строковый	Application.GVL_DIAGN.MODULE_AI_08_881.FW_VERSION

19. Сохраните проект, нажав кнопку "Сохранить (Ctrl+Shift+S)".

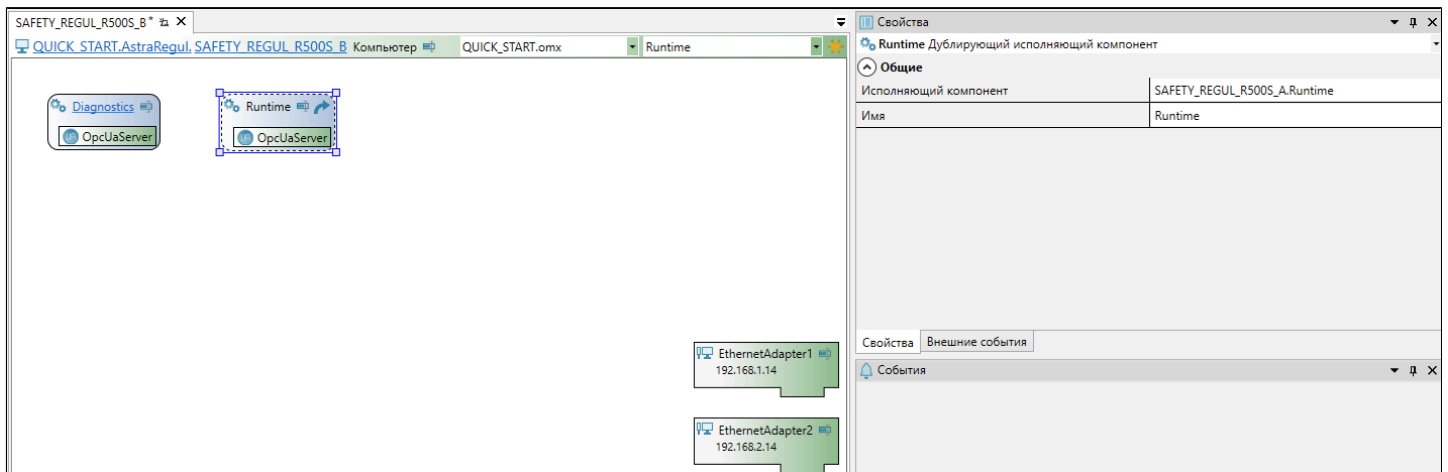


Аналогичным образом добавьте приложение диагностики для SAFETY\_REGUL\_R500S\_B.

## 1.1.3.2.2.3.3. Резервирование



Резервный ПЛК SAFETY REGUL R500S с дублирующим исполняющим компонентом "Runtime" добавляется в топологию автоматически при выполнении [экспорта проекта в Astra.IDE](#).



После экспорта необходимо создать структуру дерева объектов в [приложении "Diagnostics"](#), аналогично с основным ПЛК, и [заполнить карту адресов](#). Все остальные компоненты резервного ПЛК настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление резервного ПЛК

Чтобы описать резервный ПЛК в топологии вручную, выполните следующие действия:

1. Перейдите в компонент "Astra.Domain" и перетащите в рабочую область из "Панели элементов" компонент "Компьютер".
2. Выделите элемент "Компьютер" и в окне "Свойства" задайте ему свойство "Имя".

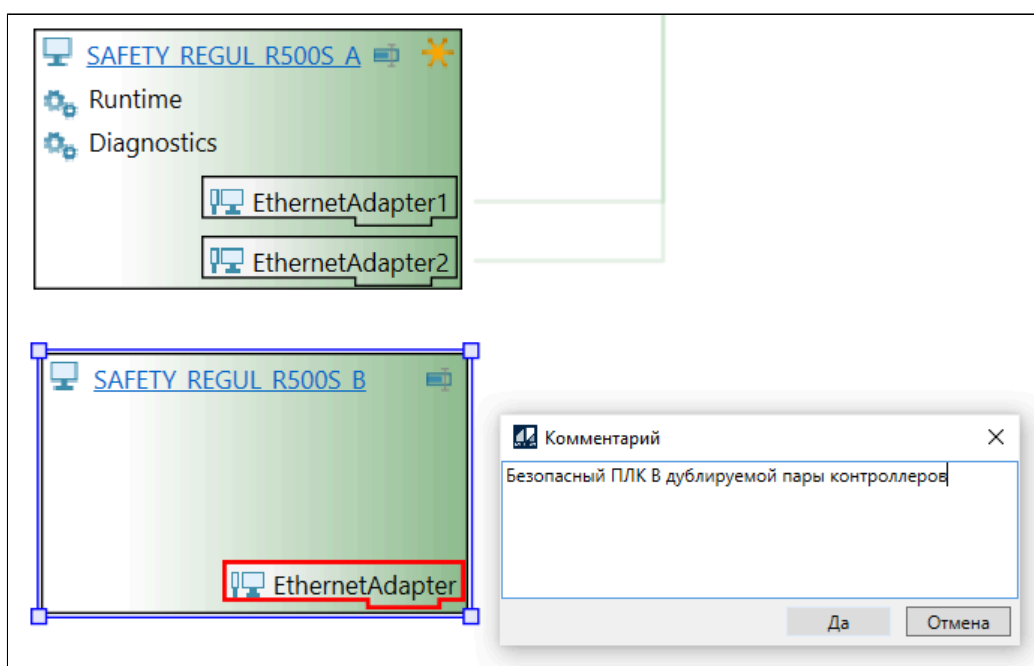


Имя элемента "Компьютер" должно соответствовать структурной схеме проекта (например, KKS).

### 3. Добавьте комментарий у объекта "Компьютер".



Комментарии к объектам добавляются с помощью контекстного меню "Комментарий".

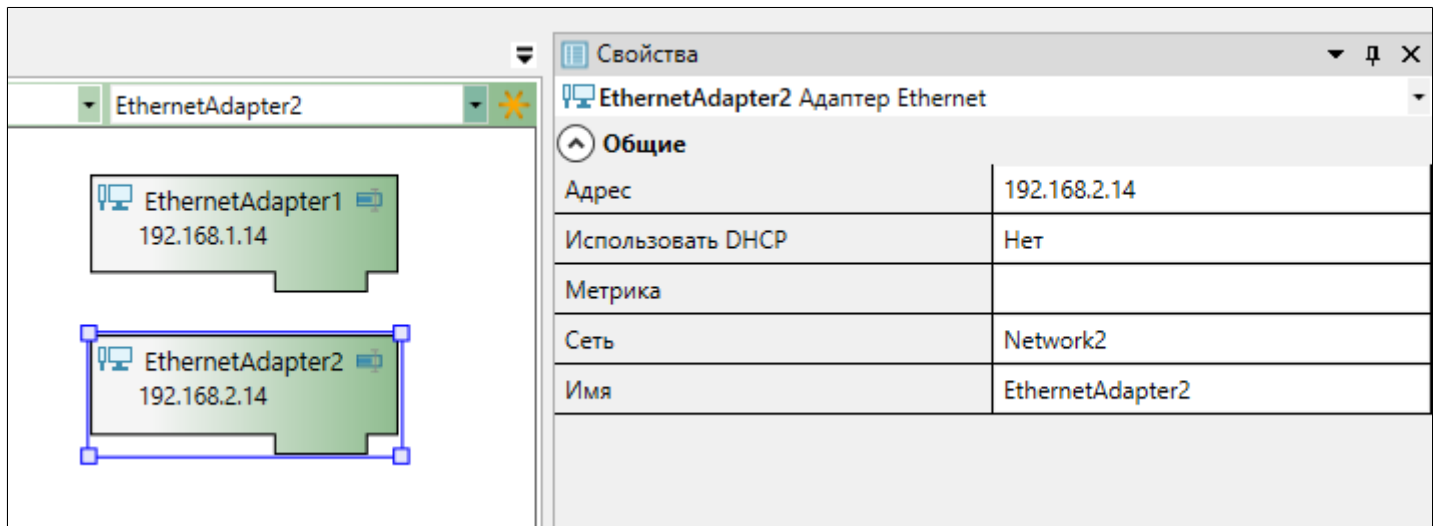


### 4. Перейдите в компонент "Компьютер" и перетащите в рабочую область из "Панели элементов" необходимое количество адаптеров Ethernet.



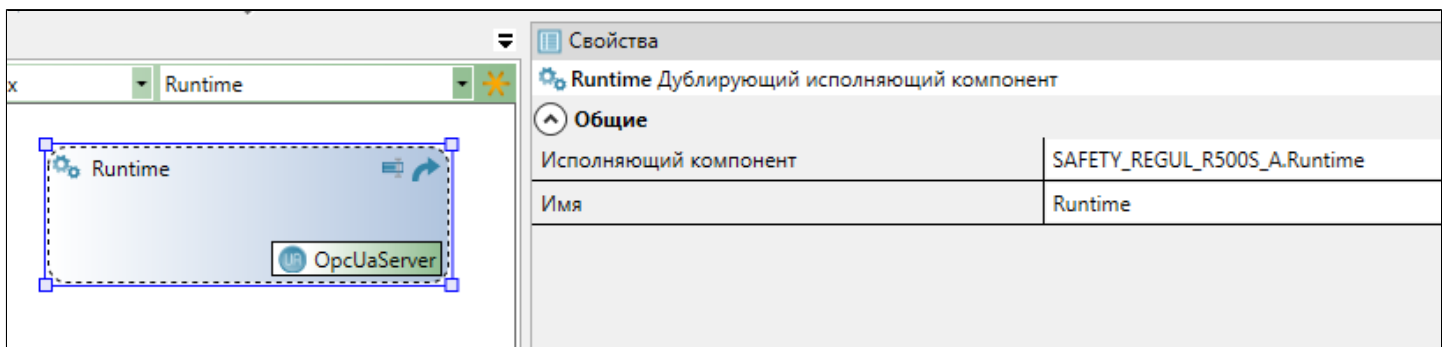
Рекомендуется использовать резервируемую линию связи (два адаптера Ethernet).

У каждого адаптера необходимо настроить свойства "Имя", "Адрес" и "Сеть". В свойстве "Адрес" укажите IP-адрес порта ПЛК. Сеть выбирается из выпадающего списка: для каждого адаптера своя сеть.



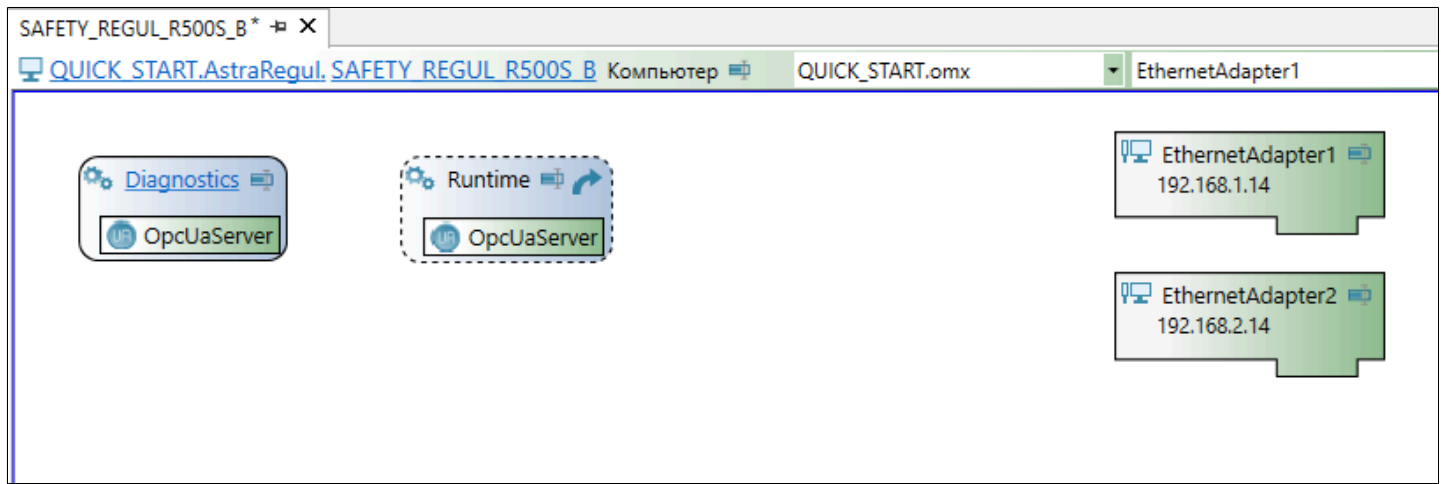
5. Перетащите в рабочую область элемент "Дублирующий исполняющий компонент" из "Панели элементов".

Выделите элемент и в окне "Свойства" задайте имя "Runtime", укажите путь к исполняющему компоненту ПЛК А резервной пары.

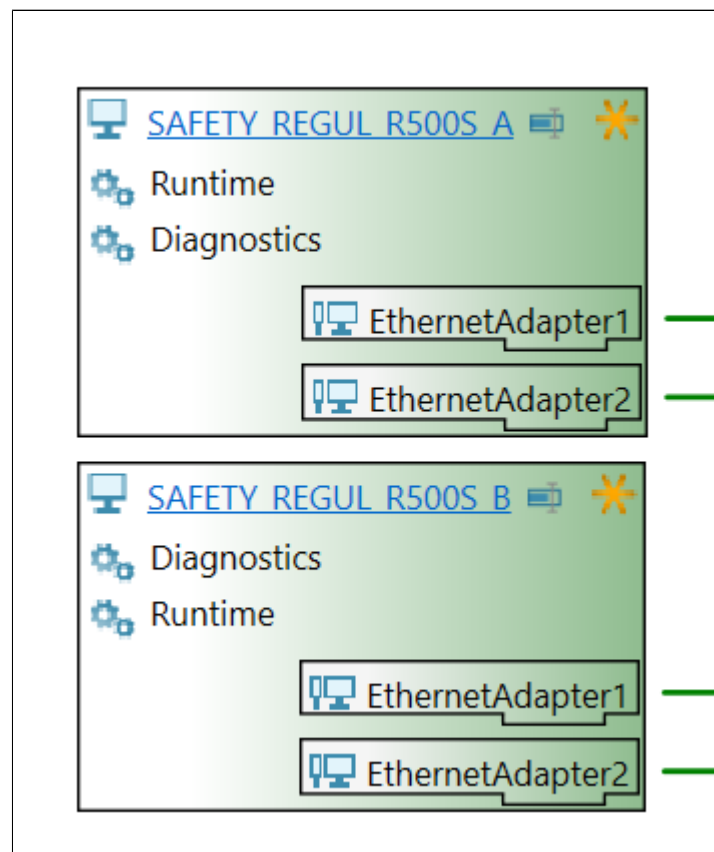


6. Перетащите в рабочую область элемент "Исполняющий компонент" из "Панели элементов".

Выделите элемент и в окне "Свойства" задайте имя "Diagnostics". Настройте компонент аналогично ПЛК А.



Описание резервного ПЛК в топологии завершено.

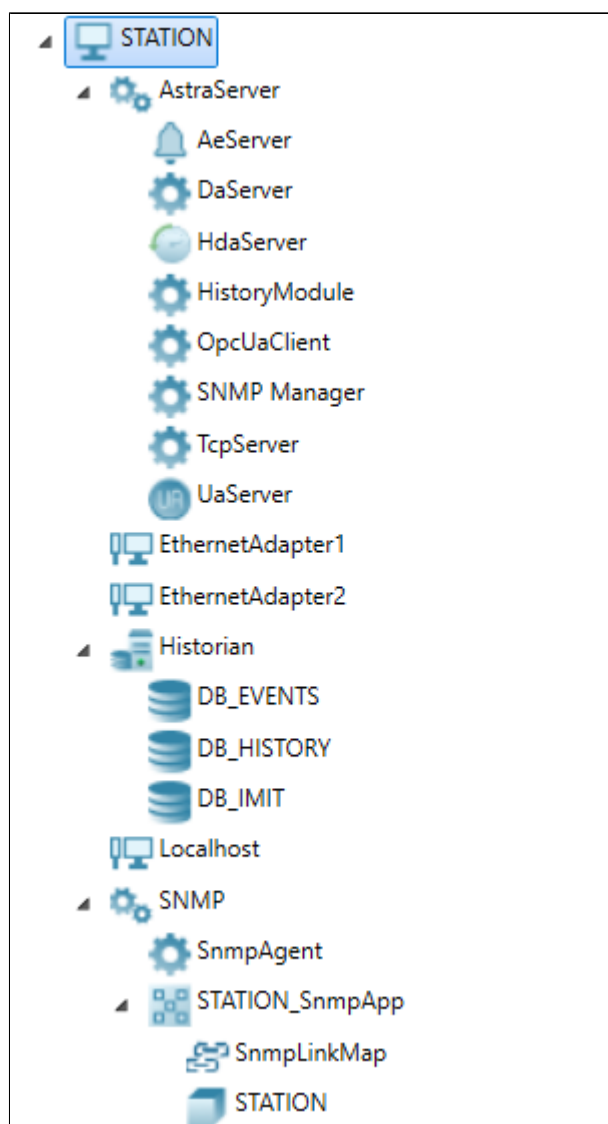


## 1.1.3.2.2.4. Инженерная станция

**Инженерная станция** - компьютер, с которого будет происходить разработка и развертывание конфигурации.



Инженерная станция вместе с адаптерами Ethernet и компонентами "Astra.Server", "Astra.Historian" и исполняющим компонентом "SNMP" добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).



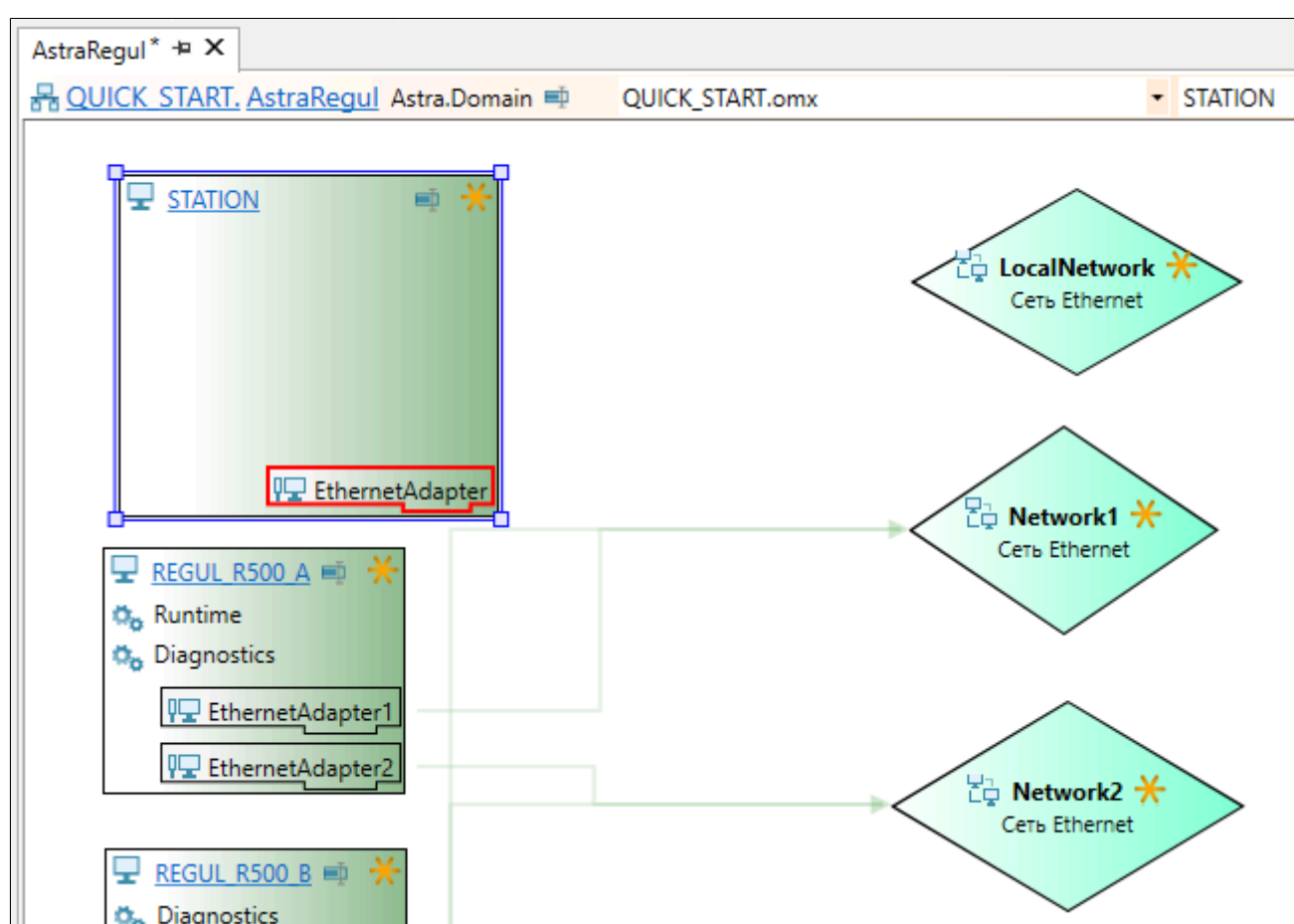
## Ручное добавление инженерной станции

Чтобы описать инженерную станцию в топологии вручную, необходимо выполнить следующие действия:

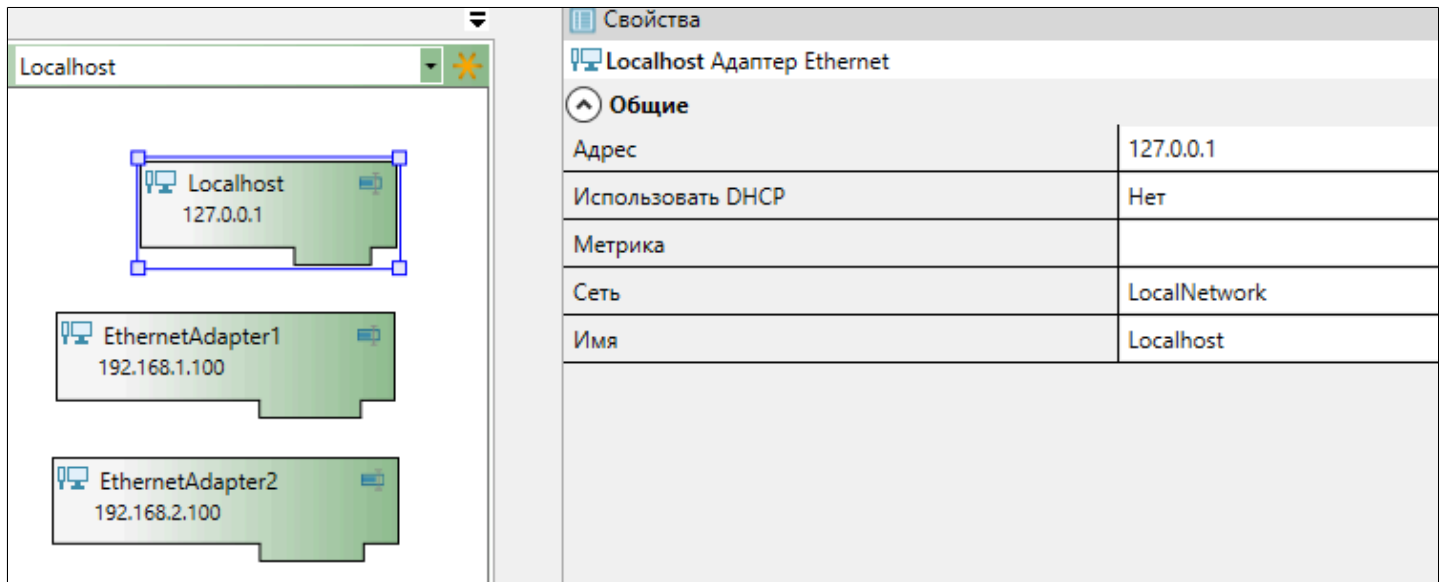
1. Перейдите в компонент "Astra.Domain" и перетащите в рабочую область из "Панели элементов" компонент "Узел Astra.Domain". В конфигурации проекта Astra.AStudio инженерная станция представлена в виде компонента "Узел Astra.Domain".



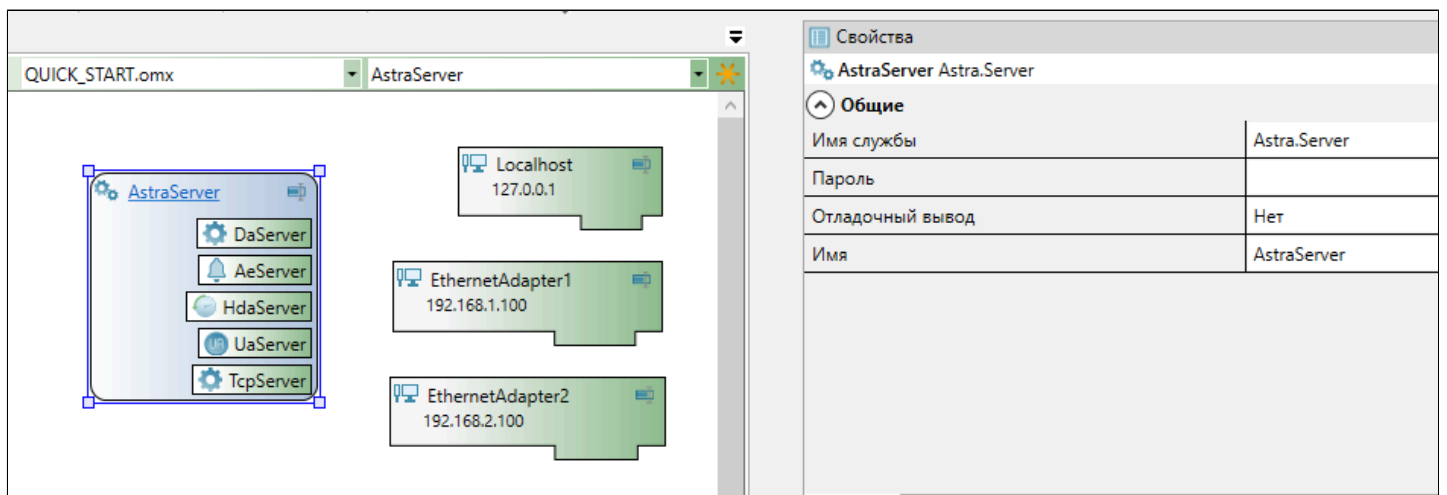
Имя инженерной станции в конфигурации должно совпадать с сетевым именем компьютера.



2. Перейдите в "Узел Astra.Domain" и перетащите в рабочую область из "Панели элементов" необходимое количество Ethernet-адаптеров в соответствии с физическими адаптерами компьютера. Дополнительно добавьте адаптер для локальной сети с IP-адресом 127.0.0.1.



3. Перетащите в рабочую область из Панели элементов компонент "Astra.Server". В окне "Свойства" укажите имя "AstraServer".



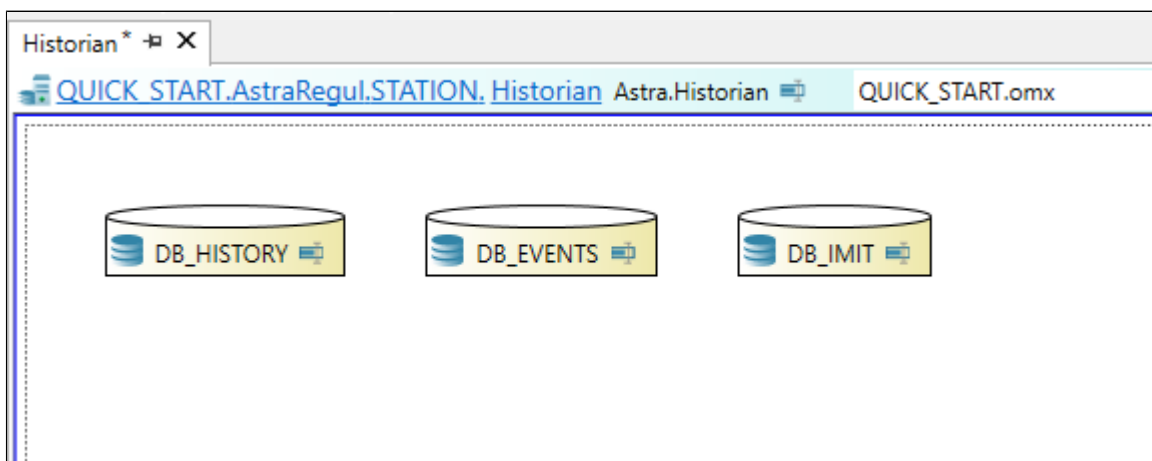
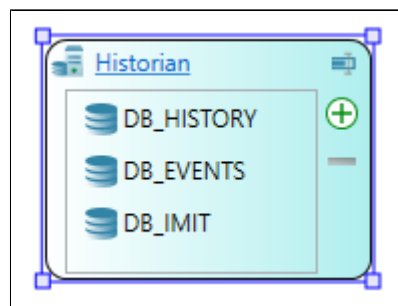


## 1.1.3.2.2.4.1. База данных

**Сервер истории** – это компонент, в котором хранятся сохранённые данные.

**i** Для сохранения исторических данных рекомендуется использовать БД "Astra.Historian".

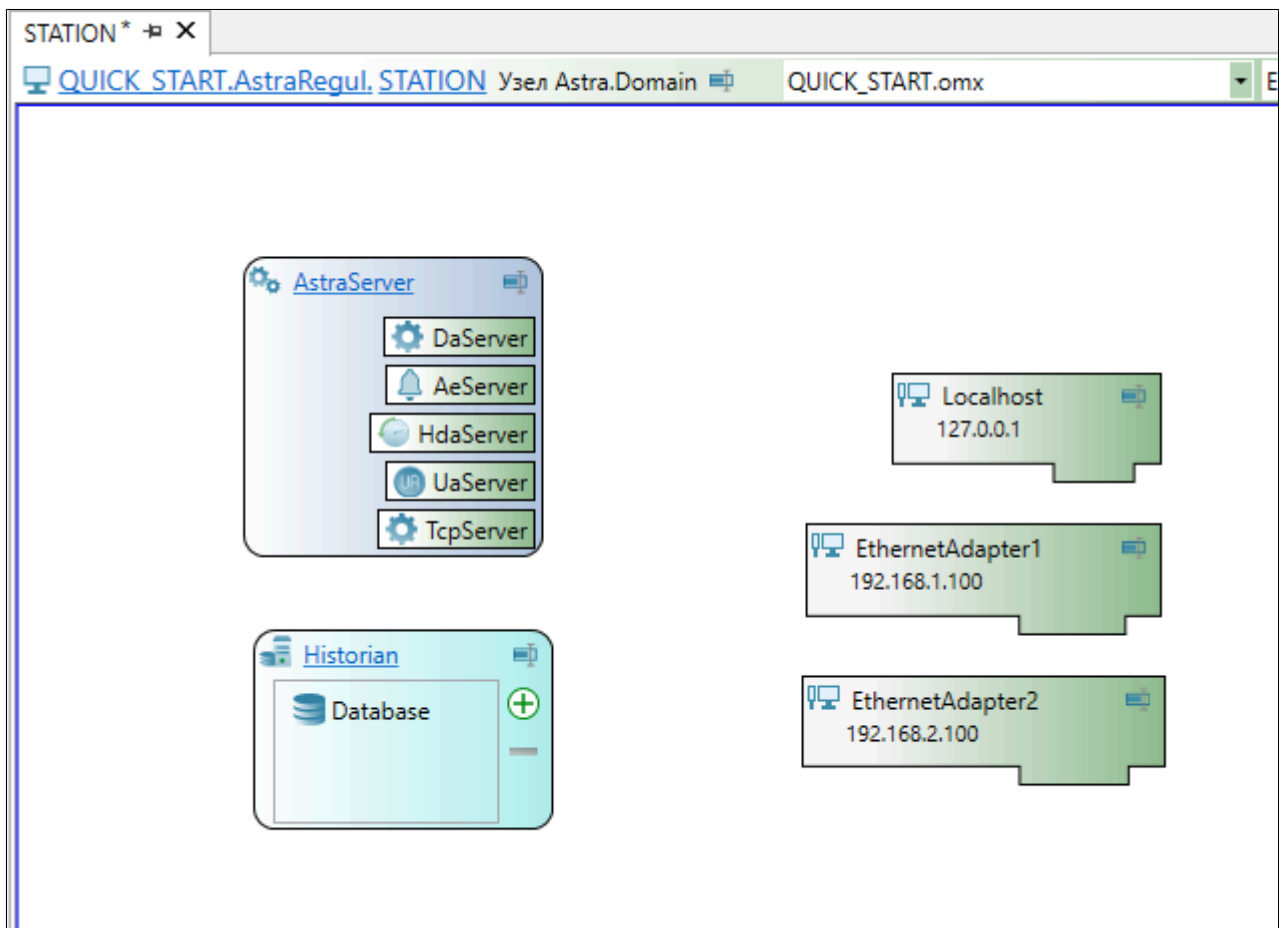
**i** Компонент "Astra.Historian" для инженерной станции добавляется и настраивается автоматически при выполнении [экспорта проекта в Astra.IDE](#).



### Ручное добавление сервера истории

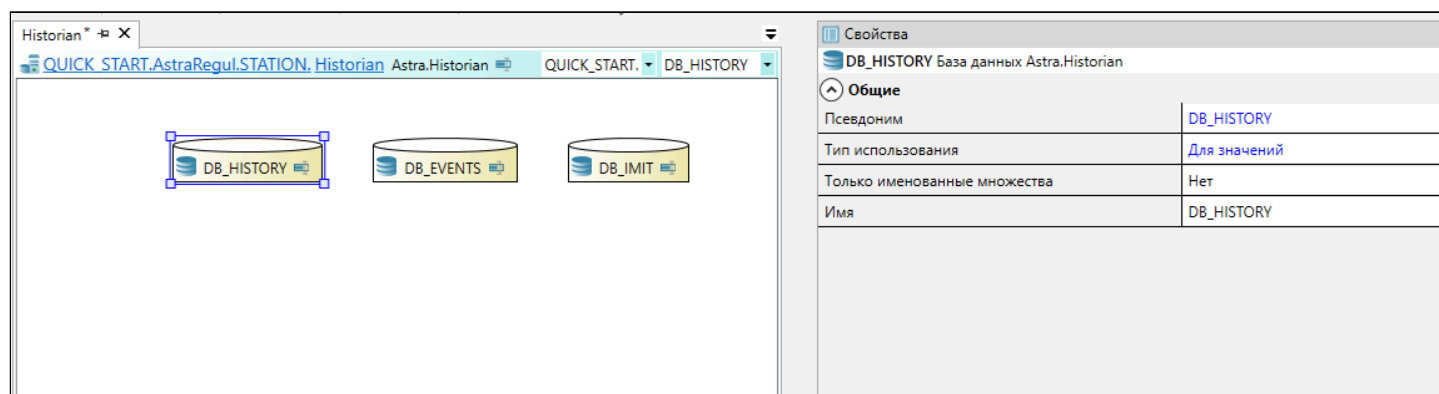
Чтобы добавить сервер истории вручную, выполните следующие действия:

1. Перетащите в рабочую область узла домена из "Панели элементов" компонент "Astra.Historian".

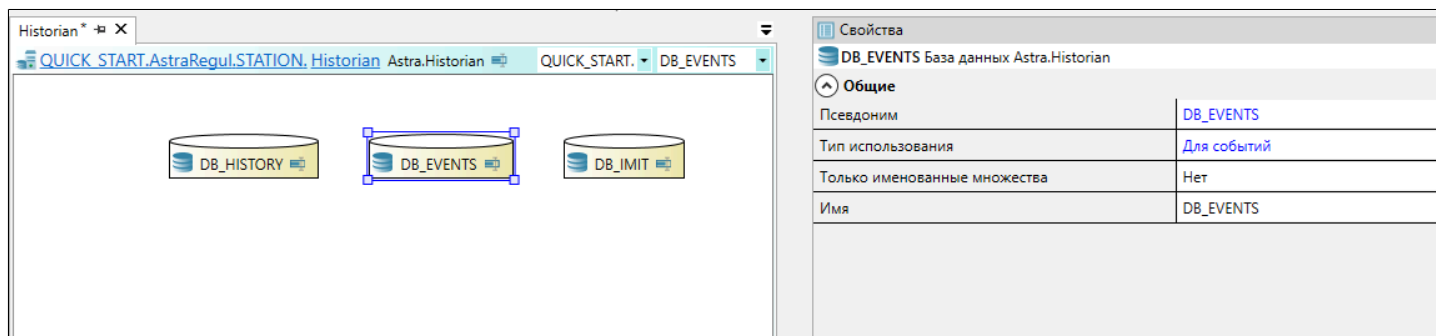


2. Перейдите в компонент "Historian" и перетащите из Панели элементов два дополнительных элемента "База данных Astra.Historian".

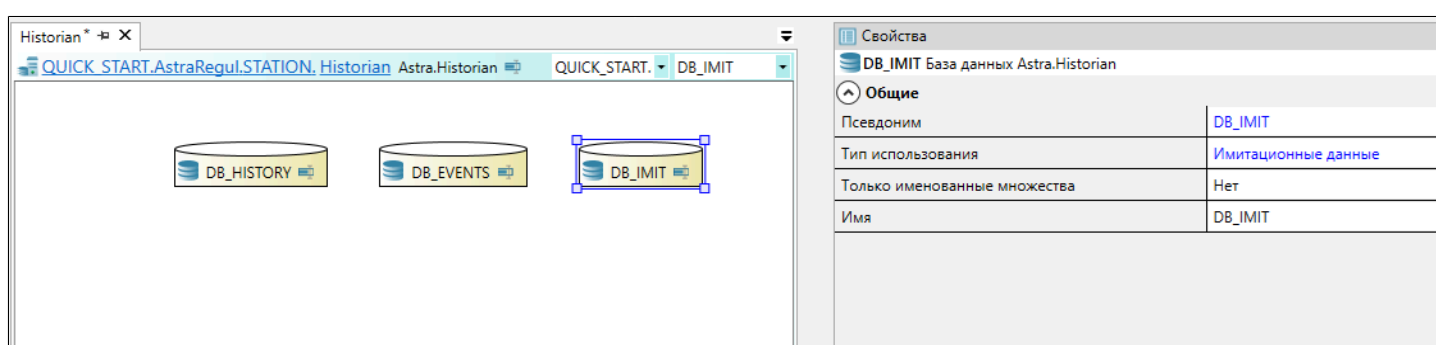
Для первой базы данных задайте имя "DB\_HISTORY" и назначьте тип использования "Для значений".



Для второй базы данных задайте имя "DB\_EVENTS" и назначьте тип использования "Для событий".

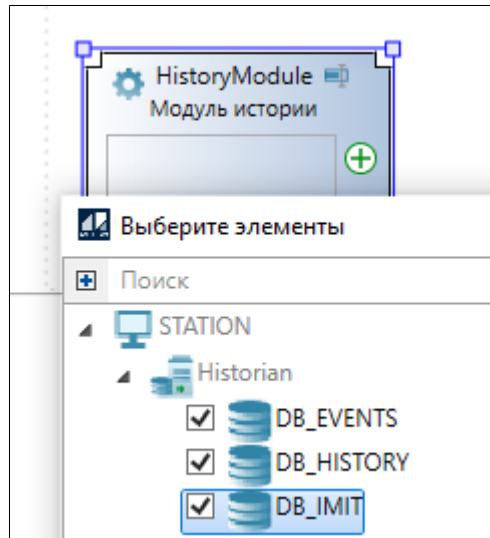


Для третьей базы данных задайте имя "DB\_IMIT" и назначьте тип использования "Имитационные данные".

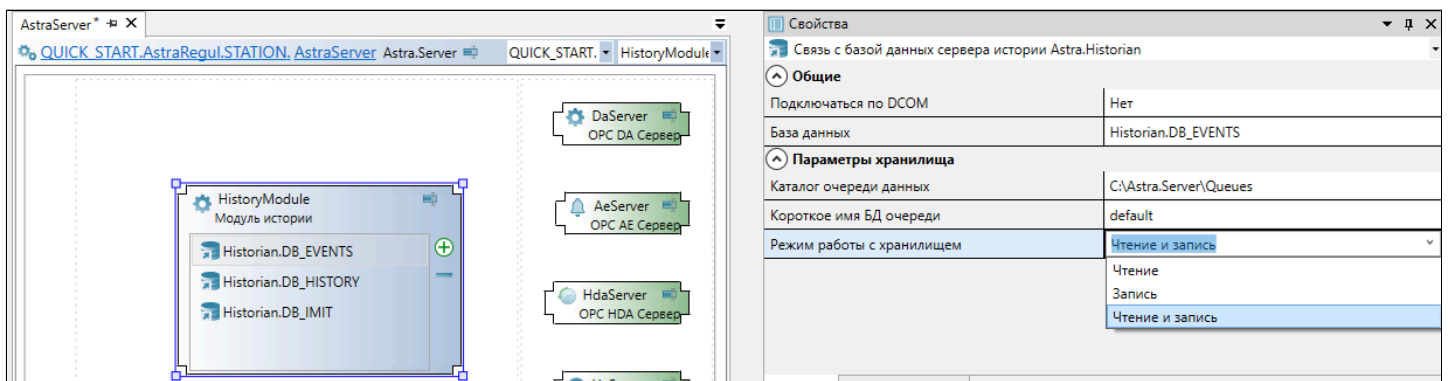


Аналогично создайте базы данных на всех компьютерах домена, где это требуется.

3. Перейдите в компонент "Astra.Server" и добавьте в Модуль истории необходимые БД. В списке выбора отображаются как локальные БД, так и БД с других серверов.

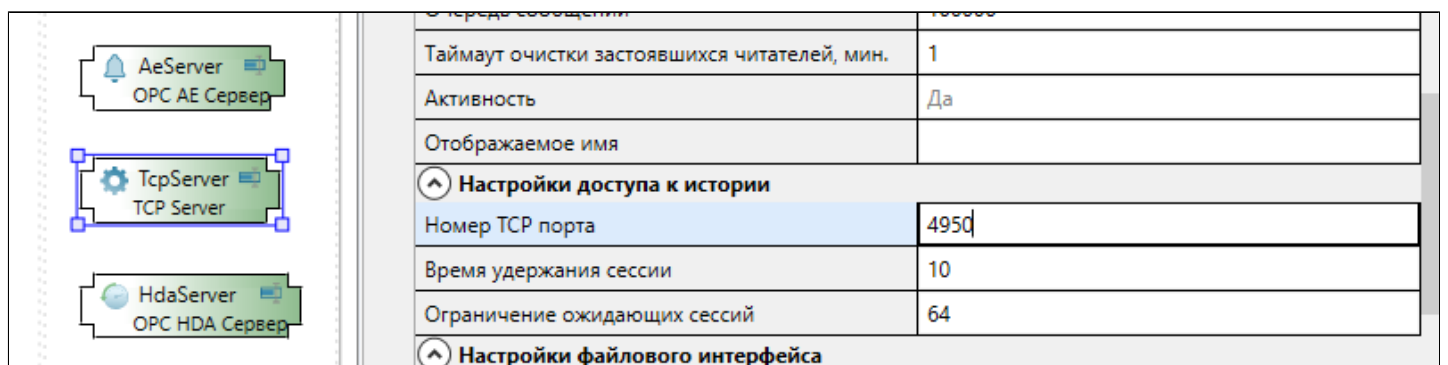


4. У каждой БД настройте параметры хранилища.



При возможности выбирайте для сохранения данных несистемный диск (диск, на котором установлена ОС).

5. Исторические данные передаются по проприетарному протоколу на базе TCP. Выделите TCP-сервер и в окне "Свойства" выставите порт 4950.



## 1.1.3.2.2.4.2. SNMP-диагностика

Для диагностики компьютеров применяется протокол [SNMP](#).

Протокол SNMP используется для диагностики сетевых устройств и управления их параметрами работы. При использовании протокола SNMP данные передаются между менеджером и агентом.

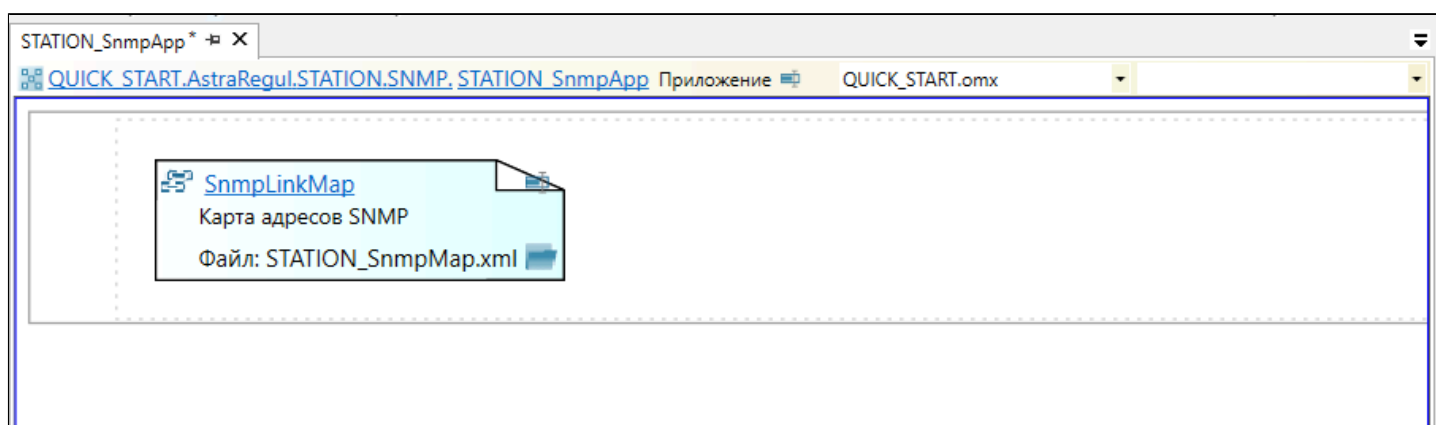
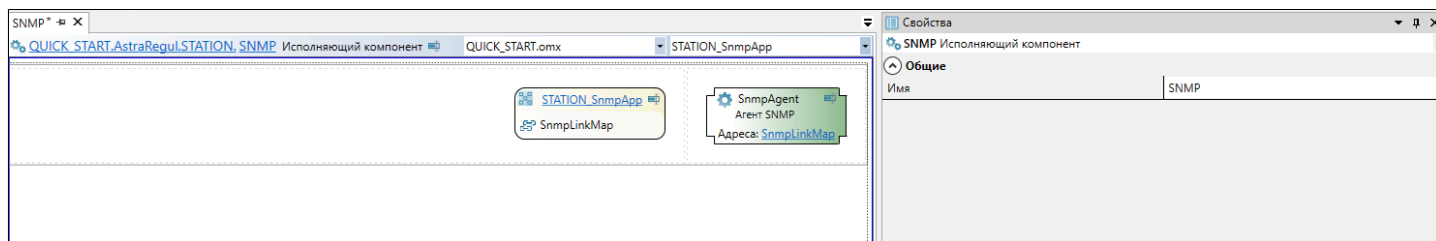
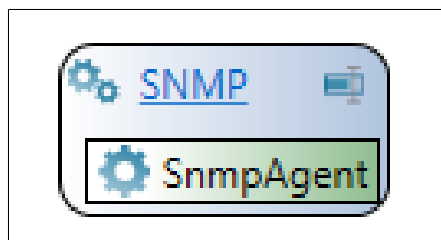
**Менеджер** – расположен на компьютере, который выполняет диагностику. Отправляет агентам запросы на получение или изменение параметров, получает от агентов значения запрошенных параметров и уведомления. В AstraRegul роль менеджера выполняет модуль SNMP Manager в составе Astra.Server.

**Агент** – расположен на диагностируемом устройстве или компьютере. Принимает запросы от менеджеров, передаёт им значения запрошенных параметров или, если запрос был на изменение значения, изменяет их. При возникновении в устройстве события (критическое значение температуры, изменение состояния порта и пр.) агент отправляет менеджерам уведомление со значением параметра, вызвавшего событие.

Значения параметров, предоставляемые агентом, хранятся в устройстве в виртуальной базе данных MIB (Management Information Base). Список параметров в MIB задаётся разработчиком устройства и не меняется в процессе работы устройства. Параметры в MIB хранятся в древовидной структуре, у каждого параметра есть уникальный OID (Object Identifier) – его положение в дереве MIB. OID используется в качестве идентификатора параметра в запросах и в уведомлениях.



Исполняющий компонент "SNMP" для инженерной станции вместе с приложением "STATION\_SnmpApp", компонентом "SnmpAgent" и "Карта адресов SNMP" добавляется автоматически при выполнении [экспорта проекта в Astra.IDE](#).

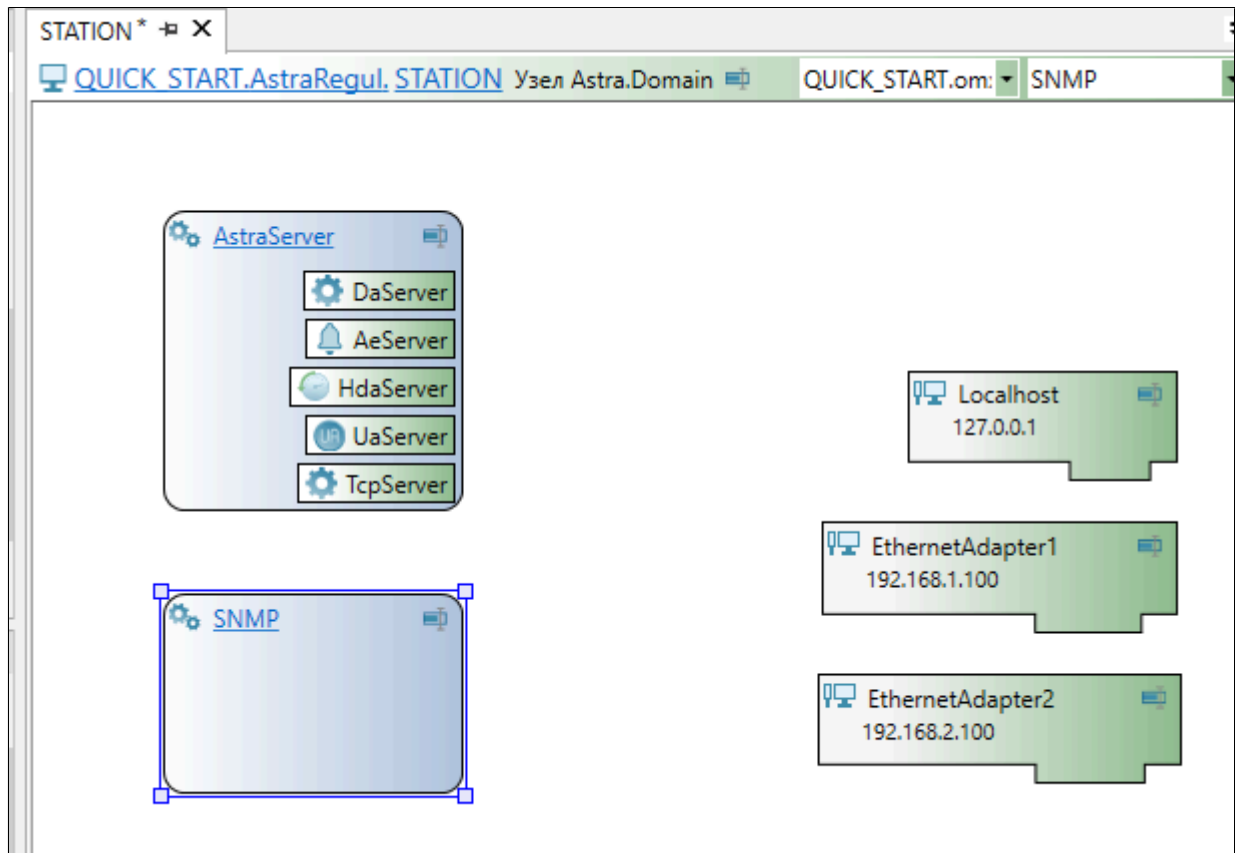


Для диагностики станции необходимо добавить в приложении "STATION\_SnmpApp" тип "COMPUTER" из библиотеки "PsDiagn" и заполнить карту адресов. Все остальные компоненты инженерной станции настраиваются автоматически с помощью плагина AstraRegul.

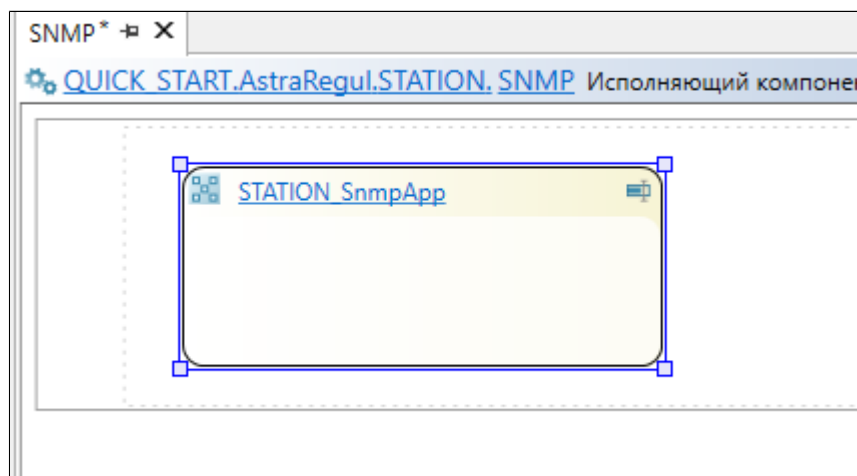
## Ручная настройка SNMP диагностики для инженерной станции

Для настройки диагностики инженерной станции по SNMP вручную, выполните следующие действия:

1. Перейдите в узел домена рабочей станции и перетащите в рабочую область узла домена из Панели элементов "Исполняющий компонент". Задайте ему имя "SNMP".



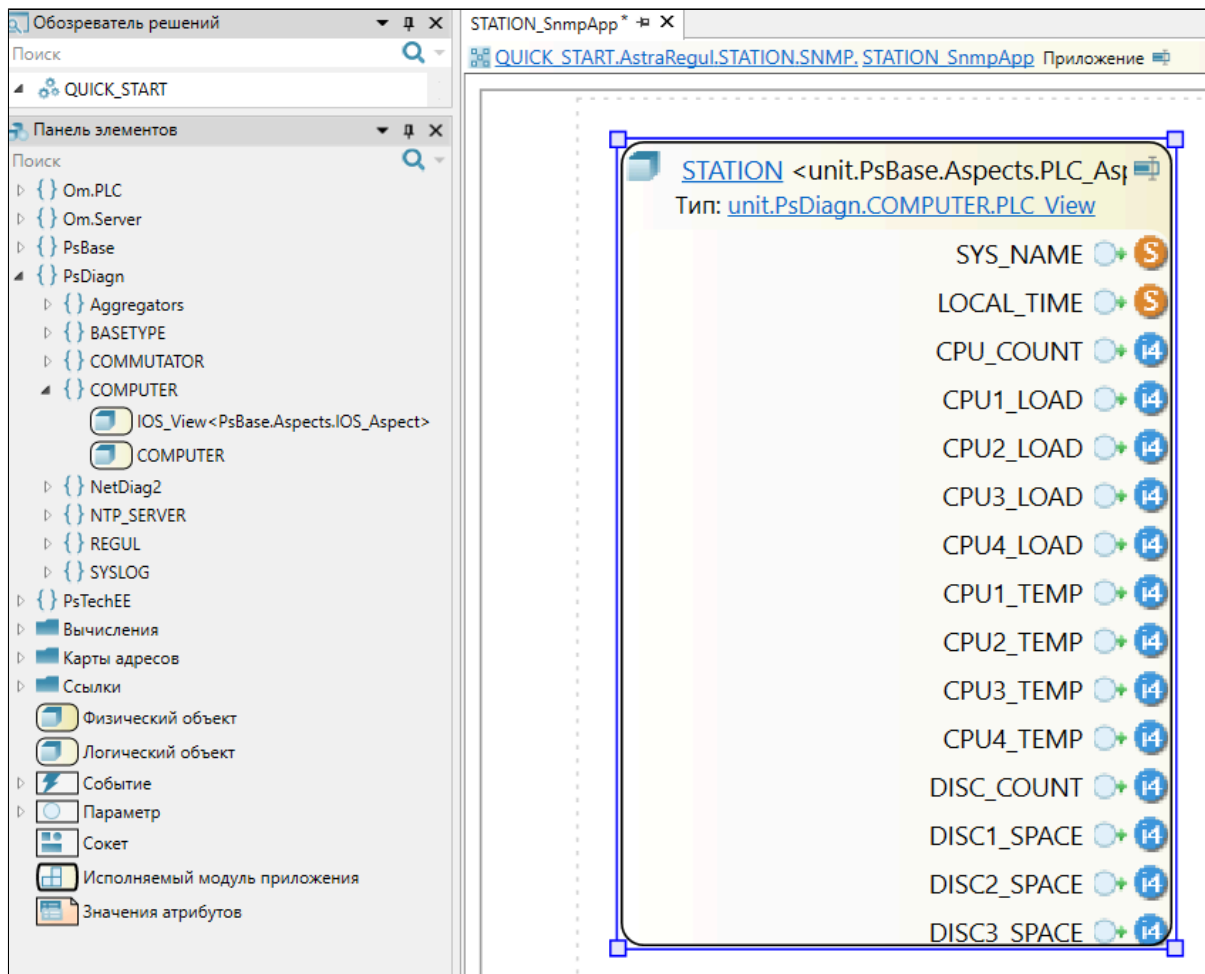
2. Перейдите в исполняющий компонент "SNMP" и перетащите в рабочую область из "Панели элементов" компонент "Приложение".





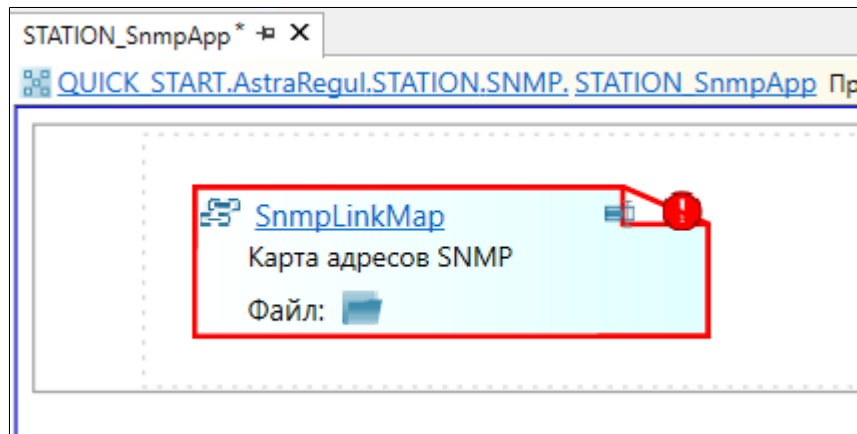
Рекомендуется имя приложения задавать в формате [Имя компьютера]\_SnmpApp.

3. Перейдите в Приложение и перетащите в рабочую область из библиотеки "PsDiagn" Панели элементов тип "COMPUTER". Имя экземпляра типа должно соответствовать имени компьютера.

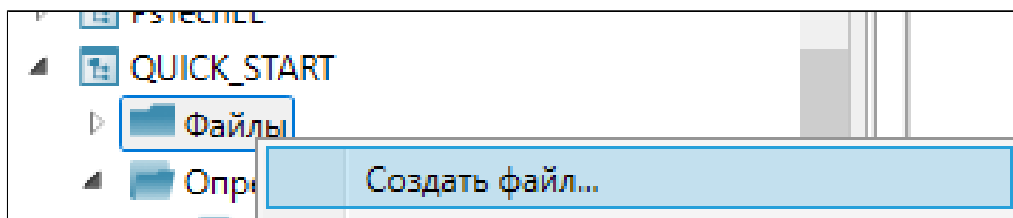


4. Перейдите в приложение и перетащите в рабочую область из Панели элементов компонент "Карта адресов SNMP".

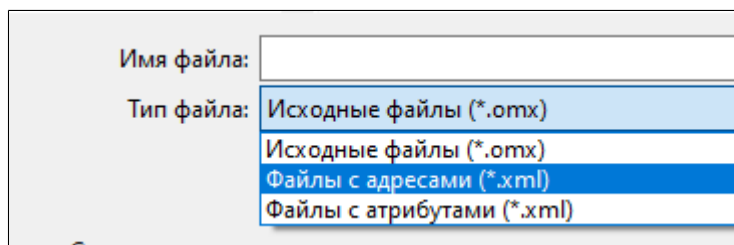




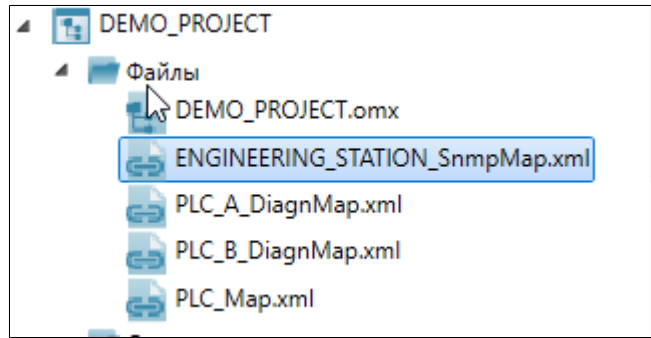
5. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." в контекстном меню объекта "Файлы" в Обозревателе решений.



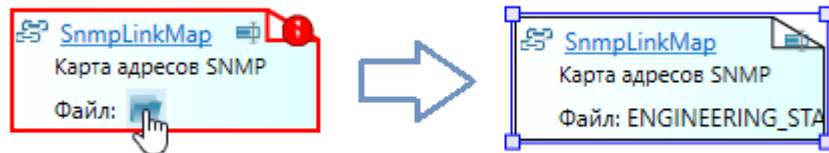
В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".



Имя файла рекомендуется задавать в формате [Имя компьютера]\_SnmpMap.

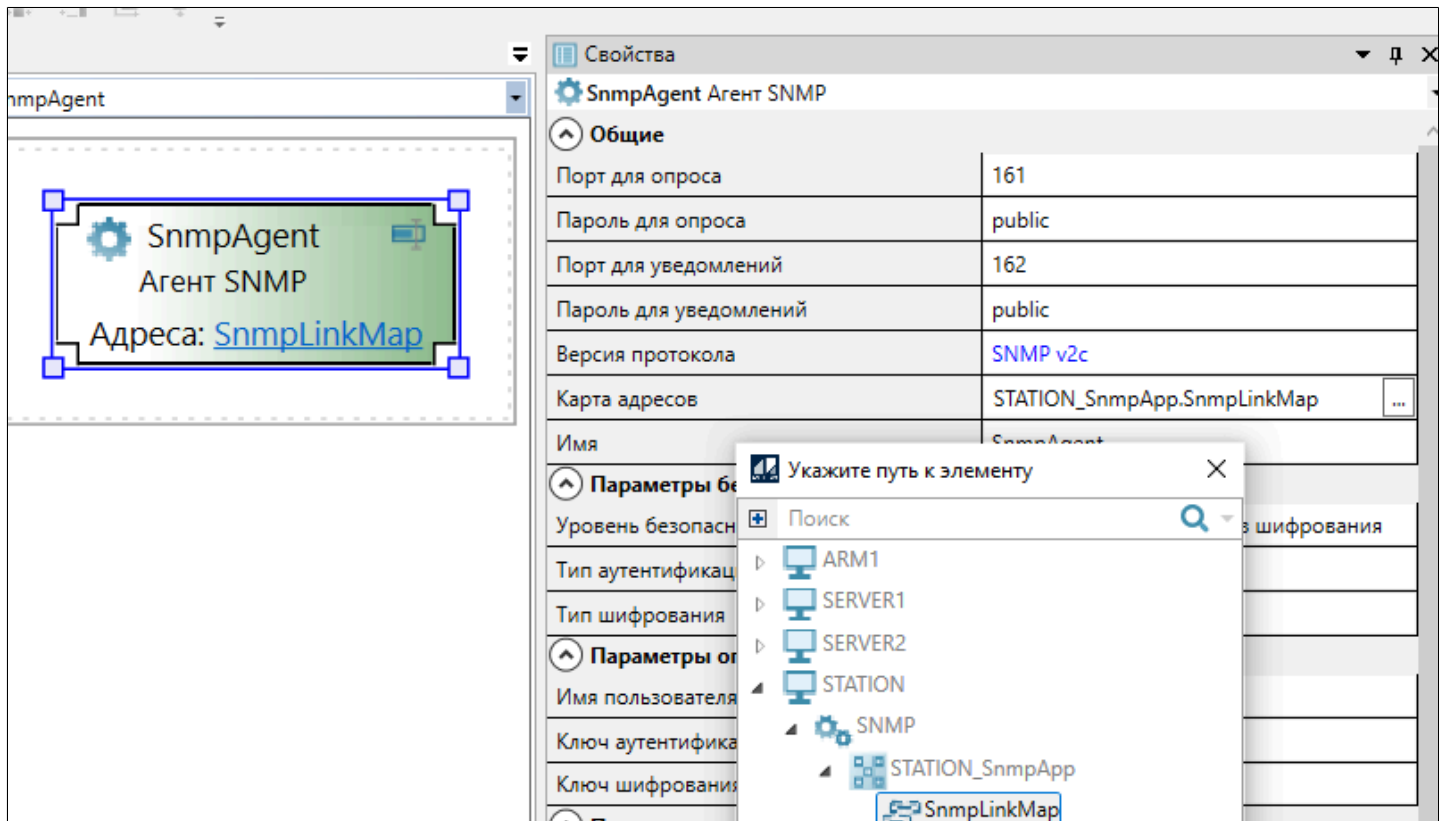


6. Перейдите в Приложение исполняющего компонента SNMP. У объекта "Карта адресов SNMP" укажите созданный файл.

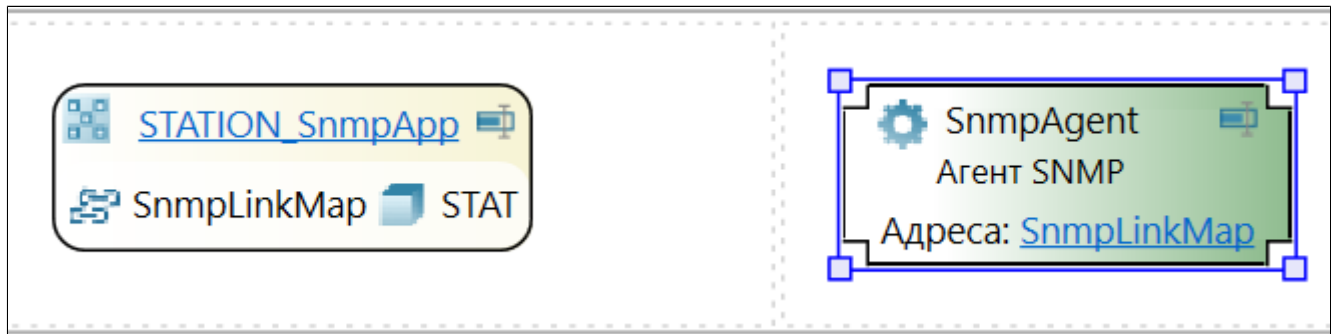


7. Перейдите в исполняющий компонент "SNMP" и перетащите из Панели элементов в рабочую область элемент "Агент SNMP".

8. Выделите Агент SNMP. В свойствах элемента укажите добавленную карту адресов и параметры агента: порты для опроса и уведомлений, версию протокола и т.д.



9. Откройте карту адресов с помощью ссылки.



10. Заполните карту адресов.

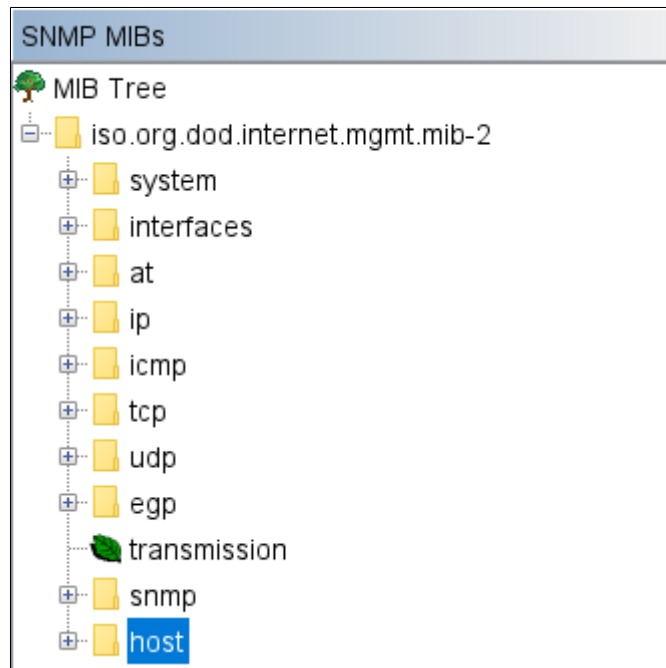
Поле	Значение
Привязка	Укажите "Непосредственно", если параметры адреса заданы в данной карте адресов.
OID элемента	Укажите OID параметра. OID можно узнать в документации на устройство или извлечь из MIB устройства с помощью утилит (например, <a href="#">MIB Browser</a> ).

Индекс элемента массива	Индекс элемента в массиве. Указывается, если OID указывает на массив.
Композитный тип объекта	Указывается только для сложных типов данных: <ul style="list-style-type: none"> <li>› Дата и время.</li> <li>› Физический адрес.</li> <li>› IP-адрес.</li> </ul> Для остальных типов оставьте поле пустым.
Способ получения данных	Возможные значения: <ul style="list-style-type: none"> <li>› Только через опрос.</li> <li>› Только через уведомления.</li> <li>› Через опрос и уведомления.</li> <li>› Пусто – равносильно значению Через опрос и уведомления.</li> </ul>

Для заполнения поля "OID элемента" необходимо воспользоваться MIB Browser.

11. Запустите MIB Browser, в дереве перейдите к элементу "mib-2" (iso → org → dod → internet → mgmt → mib-2):

- › в ветви "system" содержится информация о системе;
- › в ветви "host" содержится информация о машине (хосте);
- › в ветви "ip" содержится информация о портах машины (хоста).



Для получения информации об SNMP диагностике об устройстве используйте, указанный в свойстве "Порт для опроса" (раздел "Диагностика") соответствующего устройства, при конфигурировании топологии системы.

По умолчанию номер порта для опроса - 161.

12. Найдите соответствующий сигнал из карты адресов SNMP в дереве MIB Browser, нажмите левой кнопкой мыши на сигнал, скопируйте свойство "OID" и вставьте в поле "OID элемента" карты адресов (без первой точки).

SNMP MIBs

MIB Tree

- iso.org.dod.internet.mgmt.mib-2
  - system
    - sysDescr
    - sysObjectID
    - sysUpTime
    - sysContact
    - sysName
    - sysLocation
    - sysServices
  - interfaces
    - at
    - ip
    - icmp
    - tcp
    - udp
    - egp
  - transmission
  - snmp
  - host

Name	sysName
OID	.1.3.6.1.2.1.1.5
MIB	RFC1213-MIB
Syntax	DisplayString (OCTET STRING) (SI...
Access	read-write
Status	mandatory

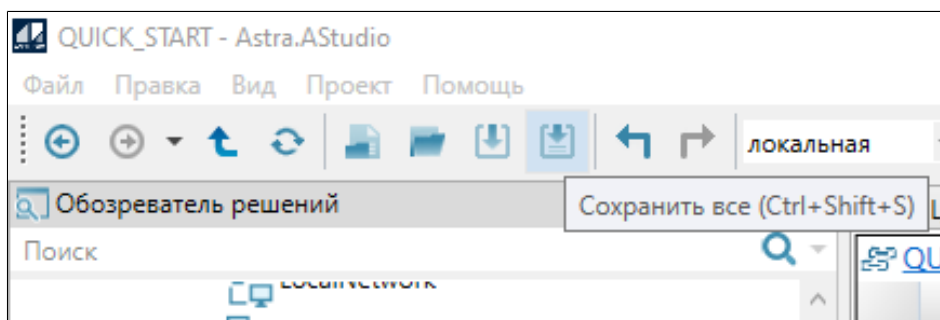
QUICK START.AstraRegul.STATION.SNMP.STATION SnmpApp, SnmpLinkMap Карта адресов SNMP

Сигнал	Тип	Привязка	OID элемента	Индекс элемента массива	Композитный тип объекта	Способ получения данных
STATION.SYS_NAME	string	непосредственно	1.3.6.1.2.1.1.5			

13. Аналогичным образом заполните OID элементов для всех необходимых параметров.

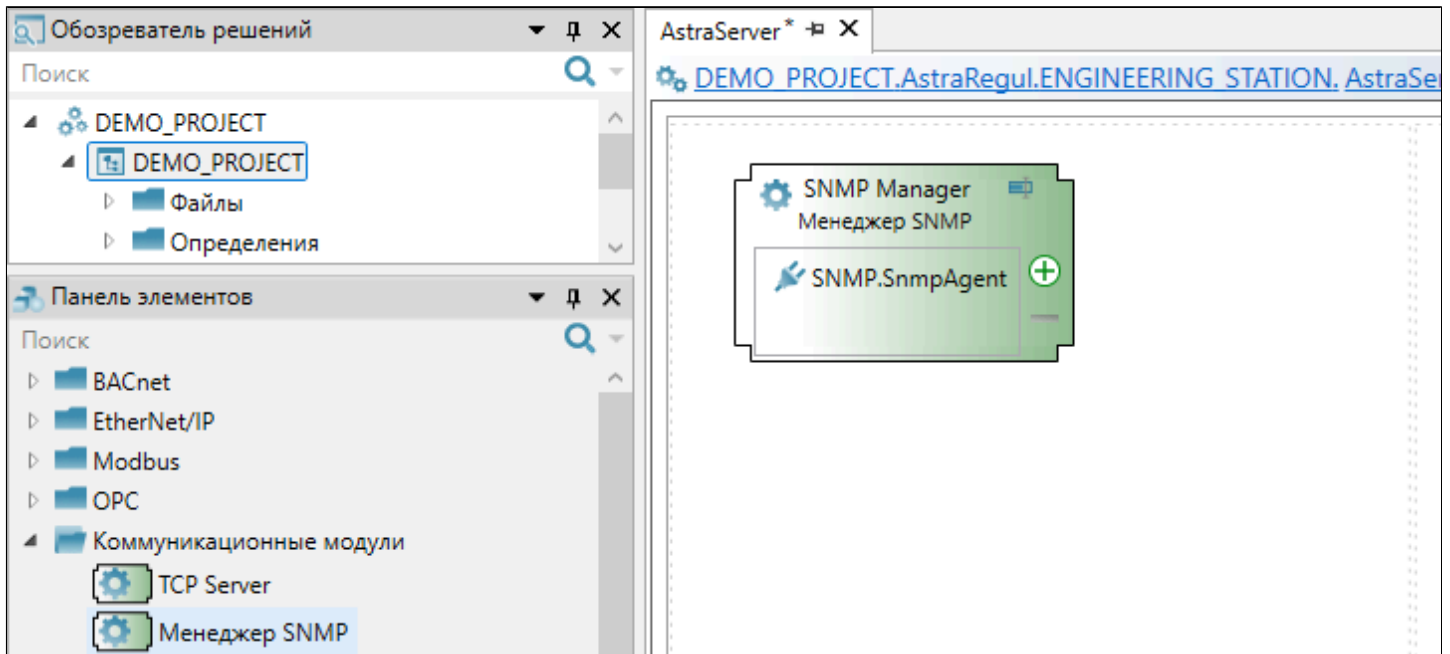
Сигнал	Тип	Привязка	OID элемента
STATION.SYS_NAME	string	непосредственно	1.3.6.1.2.1.1.5
STATION.LOCAL_TIME	string	непосредственно	1.3.6.1.2.1.25.1.2.0
STATION.CPU_COUNT	int4	не привязан	
STATION.CPU1_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.1
STATION.CPU2_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.2
STATION.CPU3_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.3
STATION.CPU4_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.4
STATION.CPU1_TEMP	int4	не привязан	
STATION.CPU2_TEMP	int4	не привязан	
STATION.CPU3_TEMP	int4	не привязан	
STATION.CPU4_TEMP	int4	не привязан	
STATION.DISC_COUNT	int4	не привязан	
STATION.DISC1_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.1
STATION.DISC2_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.2
STATION.DISC3_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.3
STATION.DISC4_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.4
STATION.DISC1_NAME	string	непосредственно	1.3.6.1.2.1.25.2.3.1.3.1
STATION.DISC2_NAME	string	непосредственно	1.3.6.1.2.1.25.2.3.1.3.2
STATION.DISC3_NAME	string	непосредственно	1.3.6.1.2.1.25.2.3.1.3.3
STATION.DISC4_NAME	string	непосредственно	1.3.6.1.2.1.25.2.3.1.3.4
STATION.PORT1_IP	string	непосредственно	1.3.6.1.2.1.4.20.1.1.192.168.1.100
STATION.PORT2_IP	string	непосредственно	1.3.6.1.2.1.4.20.1.1.192.168.2.100
STATION.PORT3_IP	string	не привязан	
STATION.PORT4_IP	string	не привязан	
STATION.PORT5_IP	string	не привязан	

14. Сохраните проект, нажав кнопку "Сохранить (Ctrl+Shift+S)".

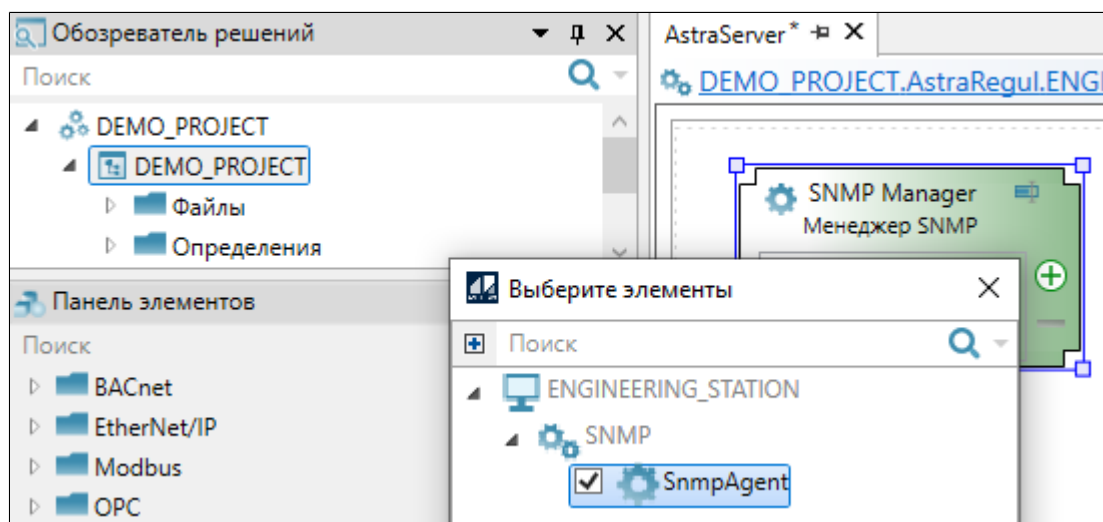


В результате инженерная станция сможет отдавать диагностические параметры по протоколу SNMP.

15. Чтобы инженерная станция могла собирать диагностические данные со всех устройств по SNMP, перейдите в "Astra.Server" инженерной станции и перетащите из "Панели элементов" в рабочую область компонент "Менеджер SNMP".



16. Добавьте в "Менеджер SNMP" необходимые агенты из списка. В списке отображаются все агенты SNMP, созданные в проекте.



17. Настройте каждый агент SNMP с учетом требований проекта.



AstraServer\* X

DEMO PROJECT.AstraRegul.ENGINEER

**Свойства**

Параметры взаимодействия с агентом

**Общие**

SNMP Агент	SNMP.SnmpAgent
------------	----------------

**Параметры агентов**

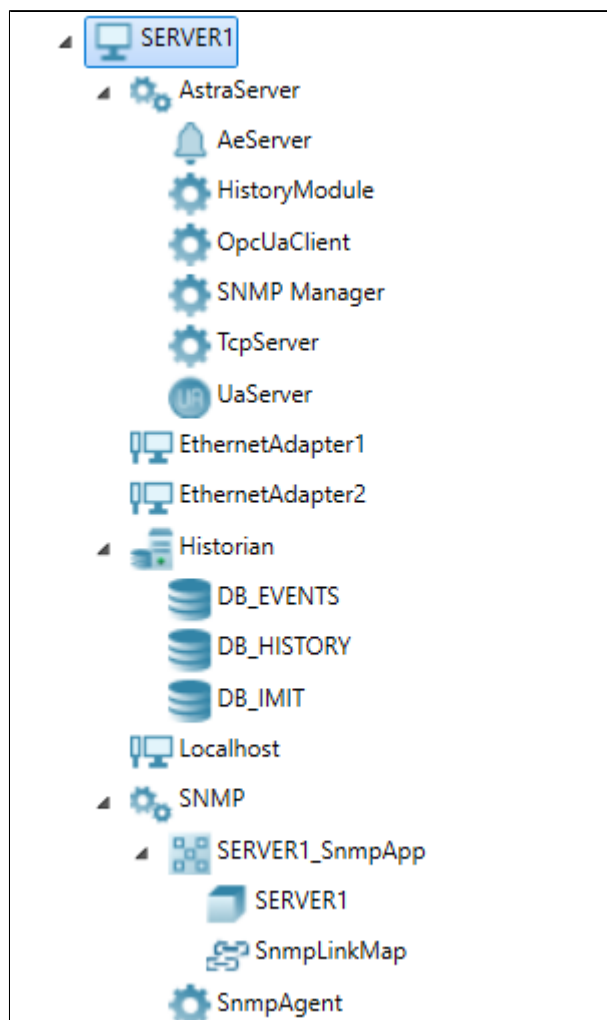
Интервал запроса данных, секунд	1
Интервал отклика на запрос, мс	500
Таймаут потери связи, сек.	3
Таймаут потери связи по каналу, секунд	3
OID для тестового запроса	

## 1.1.3.2.2.5. Сервер ввода/вывода

**Сервер ввода/вывода** - выделенный или специализированный компьютер, который служит для опроса данных с различных устройств и передачи информации на тонкие клиенты (APM).



Сервер ввода/вывода вместе с адаптерами Ethernet и компонентами ("Astra.Server", "Astra.Historian" и исполняющий компонент "SNMP") добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).

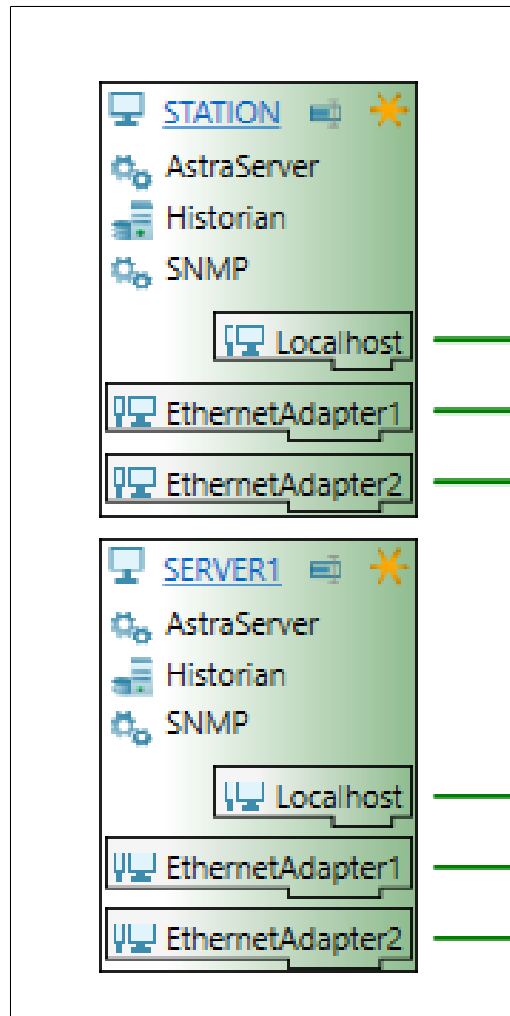


Для диагностики сервера ввода/вывода после экспорта необходимо добавить в приложении "SERVER1\_SnmpApp" тип "COMPUTER" из библиотеки "PsDiagn" и [заполнить карту адресов](#). Все остальные

компоненты резервного сервера настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление сервера ввода/вывода

Одиночный сервер вручную настраивается по аналогии с [инженерной станцией](#).



## 1.1.3.2.2.5.1. Резервирование

Чтобы повысить надёжность проекта автоматизации необходимо использовать резервированную пару серверов ввода/вывода. При прекращении работы основного компонента, резервный компонент продолжит работу и будет выполнять его функции в проекте.



Для резервирования сервера ввода/вывода применяется горячее резервирование.

При горячем резервировании основной и резервный компоненты образуют резервную пару. В резервной паре основной компонент работает в полнофункциональном режиме, а резервный компонент находится в режиме ожидания. При прекращении работы основного компонента автоматически выполнится резервный переход: резервный компонент переключится в полнофункциональный режим и будет выполнять функции основного компонента. Клиенты в процессе работы подключаются к тому компоненту резервной пары, который является основным.



Резервный сервер ввода/вывода с резервным компонентом "Astra.Server" добавляется в топологию автоматически при выполнении [экспорта проекта в Astra.IDE](#).

The screenshot displays the Astra IDE interface. On the left, a topology diagram shows a server node labeled 'AstraServer' containing sub-components: 'TcpServer', 'SNMP Manager', 'AeServer', 'OpCuaClient', and 'UaServer'. Other nodes include 'SNMP' with 'SnmpAgent', 'Localhost' (127.0.0.1), 'EthernetAdapter1' (192.168.1.202), 'EthernetAdapter2' (192.168.2.202), and 'Historian' with 'DB\_HISTORY', 'DB\_EVENTS', and 'DB\_IMIT'. On the right, the 'Свойства' (Properties) window is open for 'SERVER2 Узел Astra.Domain'. It shows general information: 'Адрес' (Address) is 'SERVER2', 'Конфигурации' (Configuration) is 'C:\DomainStorage\cache', and 'Имя' (Name) is 'SERVER2'. Below, the 'Атрибуты' (Attributes) section shows a table with 'SERVER2 Узел Astra.Domain' and a 'Резервный сервер' (Backup server) attribute.

Атрибут	Значение
SERVER2 Узел Astra.Domain	Резервный сервер



Для диагностики резервного сервера ввода/вывода после экспорта необходимо добавить в приложении "SERVER2\_SnmpApp" тип "COMPUTER" из библиотеки "PsDiagn" и [заполнить карту адресов](#). Все остальные компоненты резервного сервера настраиваются автоматически с помощью плагина AstraRegul.

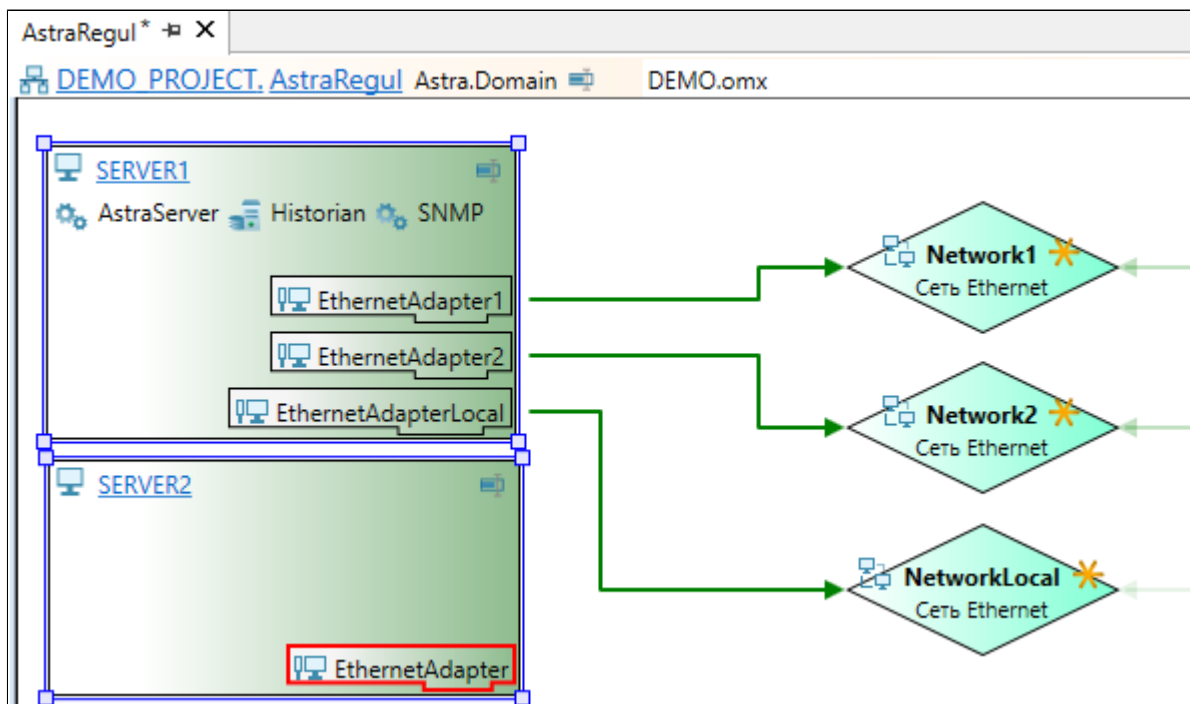
## Ручное добавление резервного сервера ввода/вывода

Чтобы описать резервный сервер ввода/вывода в топологии вручную, необходимо выполнить следующие действия:

1. Перейдите в "Astra.Domain" и перетащите в рабочую область домена из Панели элементов компонент "Узел Astra.Domain".

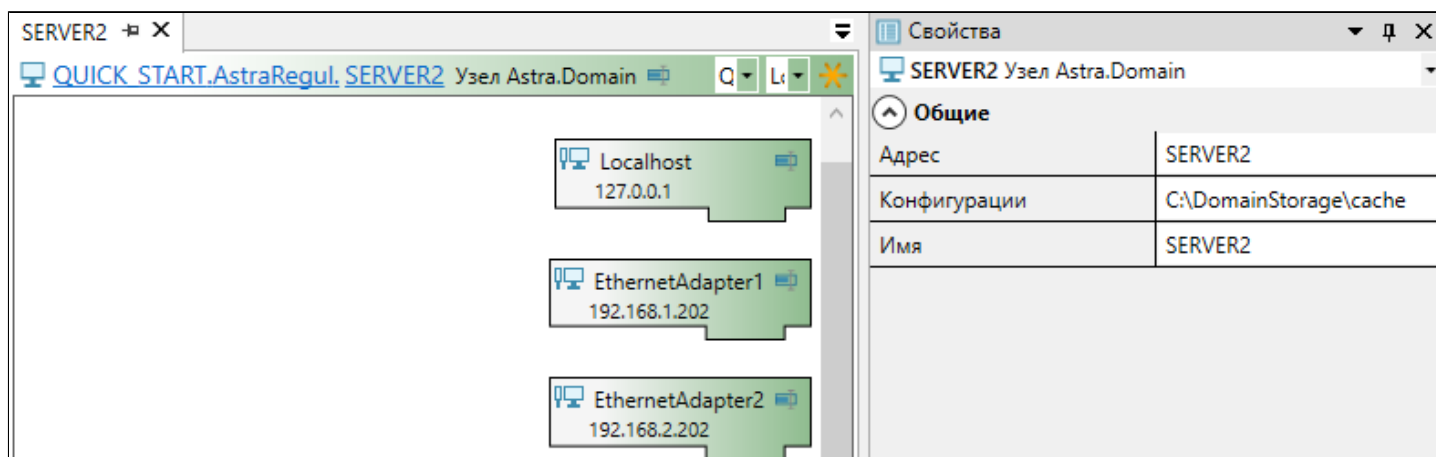


Имя узла домена должно совпадать с сетевым именем компьютера.

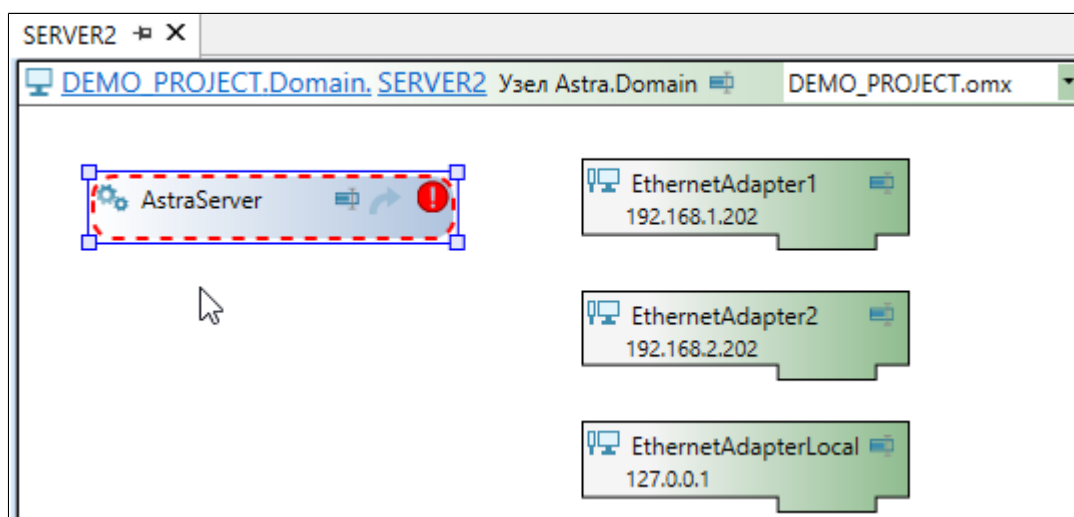


2. Перейдите в "Узел Astra.Domain" и перетащите в рабочую область из Панели элементов необходимое количество Ethernet-адаптеров в соответствии с

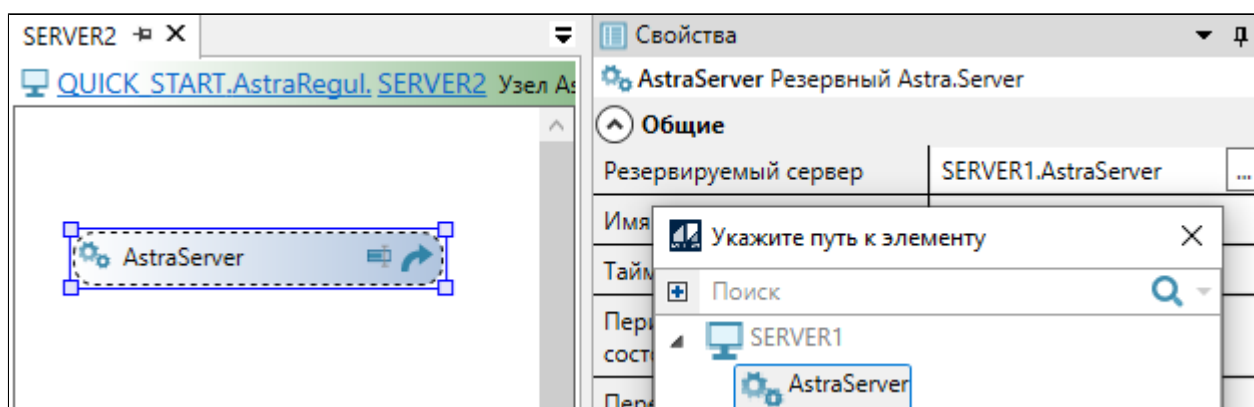
физическими адаптерами компьютера. Дополнительно добавьте адаптер для локальной сети с IP-адресом 127.0.0.1.



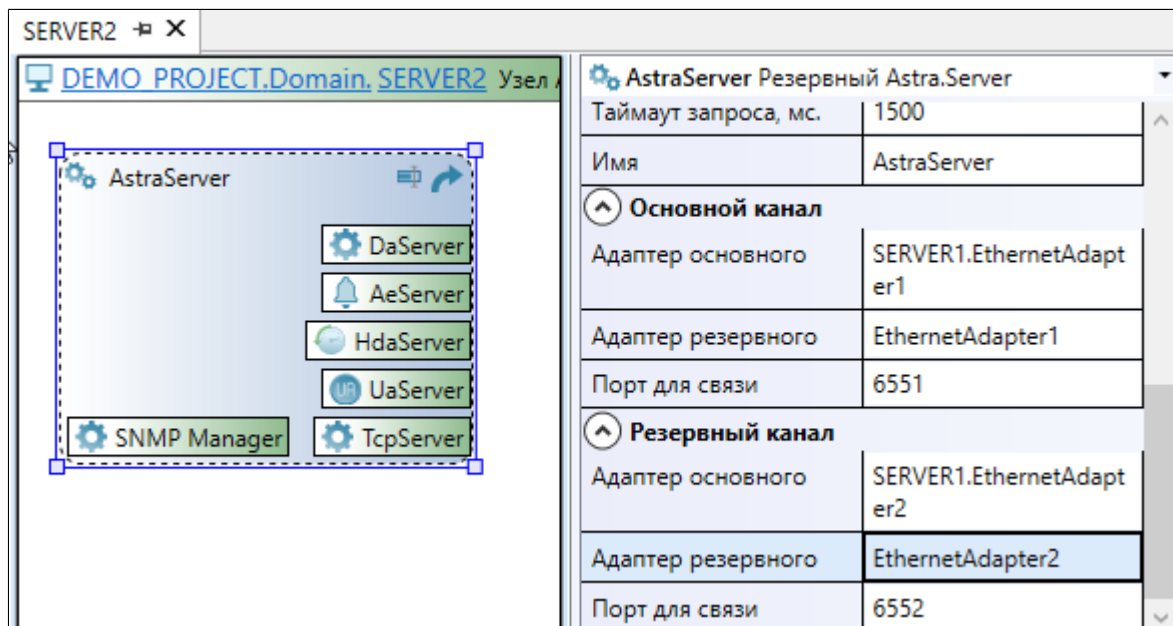
3. Перетащите в рабочую область из Панели элементов компонент "Резервный Astra.Server". В окне "Свойства" укажите имя "AstraServer".



4. В окне "Свойства" элемента укажите "Резервируемый сервер".

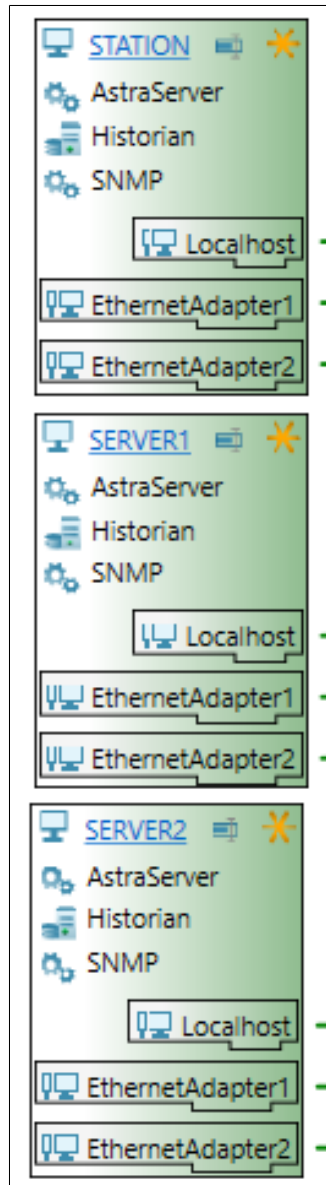


5. В окне "Свойства" элемента укажите параметры основного и резервного канала связи (порты, адаптеры) для связи с основным сервером.



6. БД резервного сервера настраивается так же, как и [для основного сервера](#).

7. Диагностика резервного сервера настраивается так же, как и [для основного сервера](#).



Для основного и резервного Astra.Server создаётся один файл конфигурации, который при развёртывании применяется к обоим экземплярам.

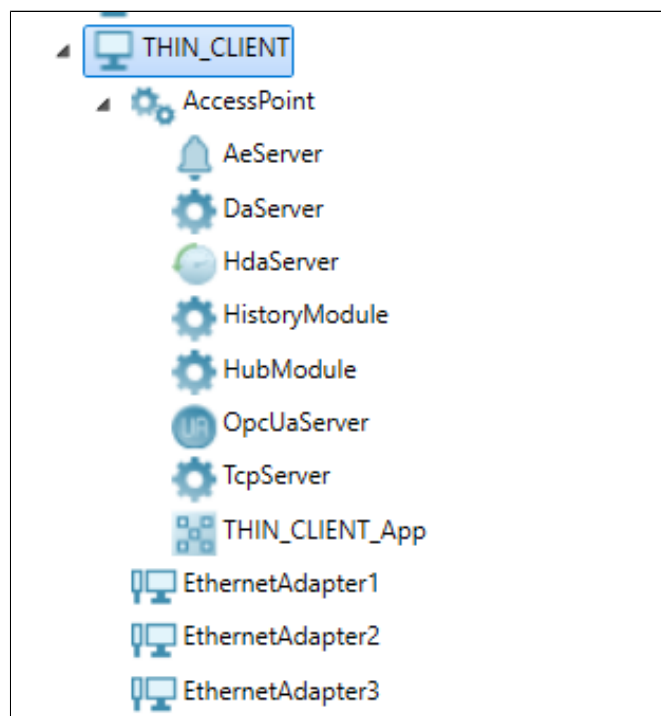


## 1.1.3.2.2.6. APM

**APM** - это автоматизированное рабочее место оператора (тонкий клиент).



APM оператора вместе с адаптерами Ethernet и компонентом "Astra.AccessPoint" добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).

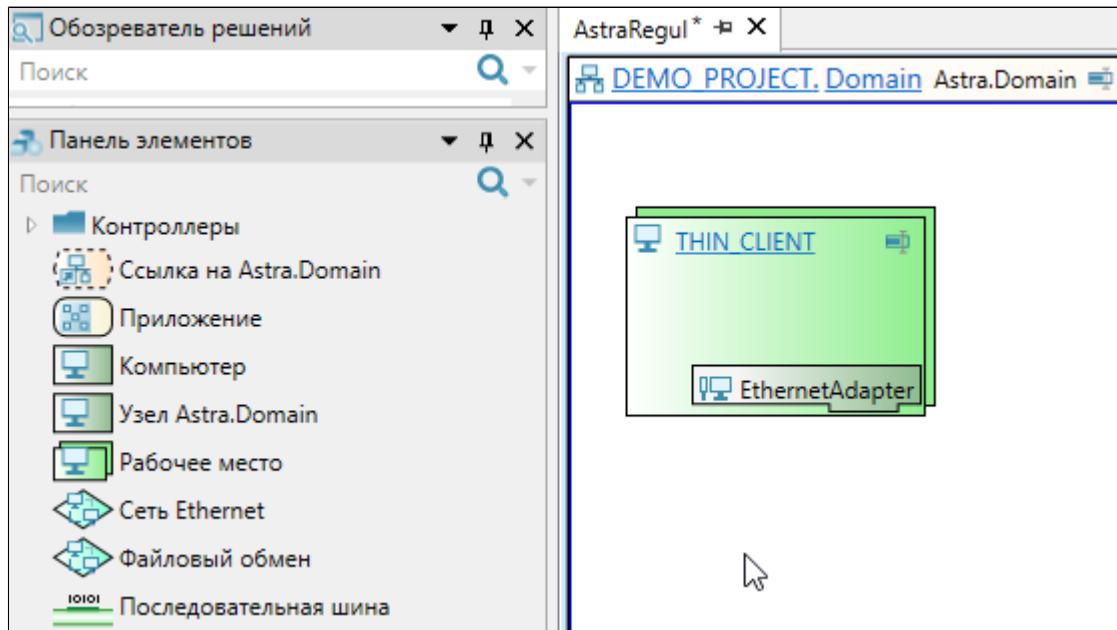


Все компоненты APM настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление APM

Чтобы описать APM в топологии вручную, необходимо выполнить следующие действия:

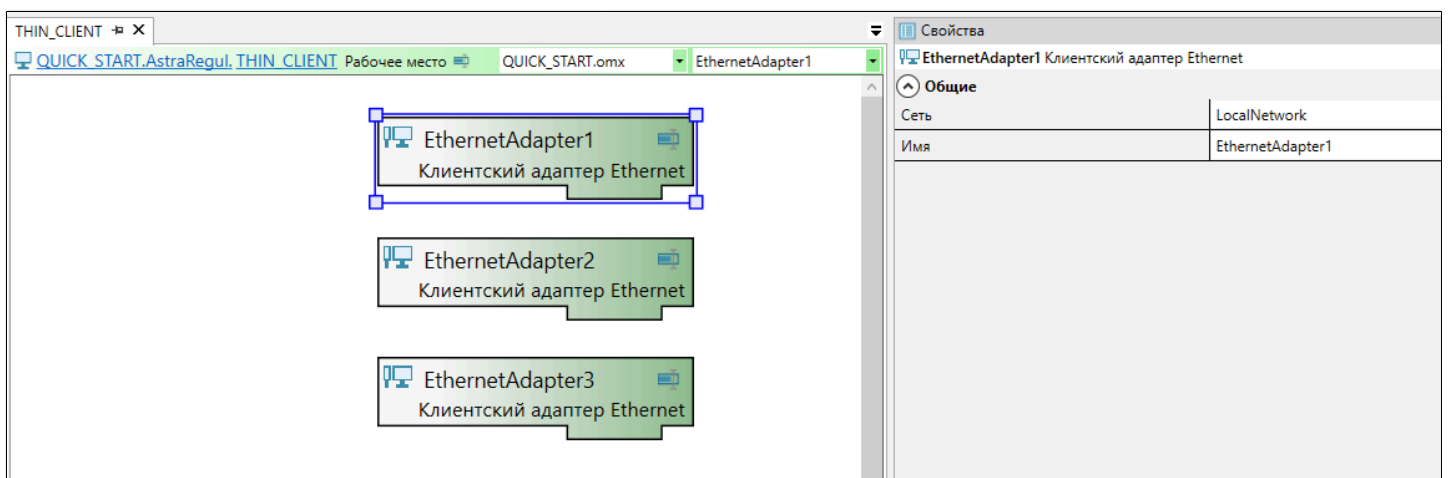
1. Перетащите в рабочую область домена из "Панели элементов" компонент "Рабочее место". Выделите элемент и в окне "Свойства" задайте имя "THIN\_CLIENT" (тонкий клиент).



Одному элементу "Рабочее место" соответствует любое количество АРМ.

Все АРМ получают одинаковые данные и выполняют одни и те же функции.

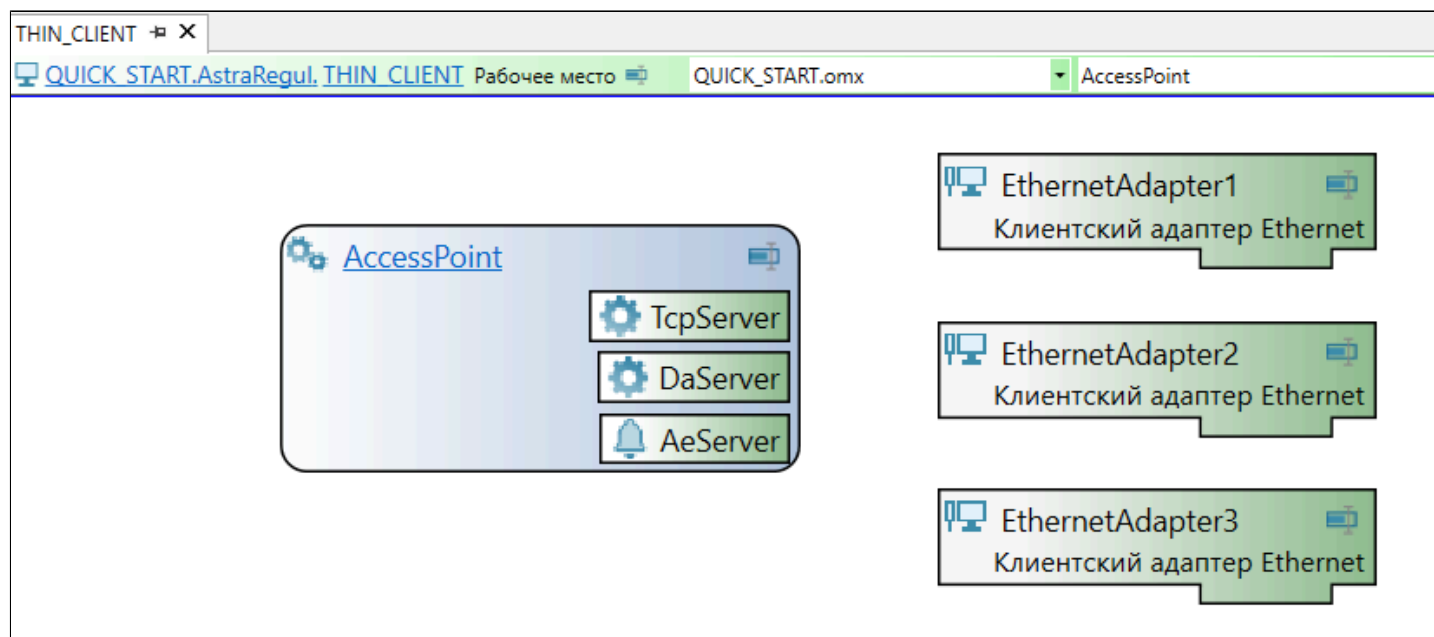
2. Перейдите в элемент "Рабочее место" и перетащите в рабочую область из Панели элементов необходимое количество Клиентских адаптеров Ethernet в соответствии с физическими адаптерами компьютера. Каждому адаптеру необходимо указать соответствующую ему сеть.



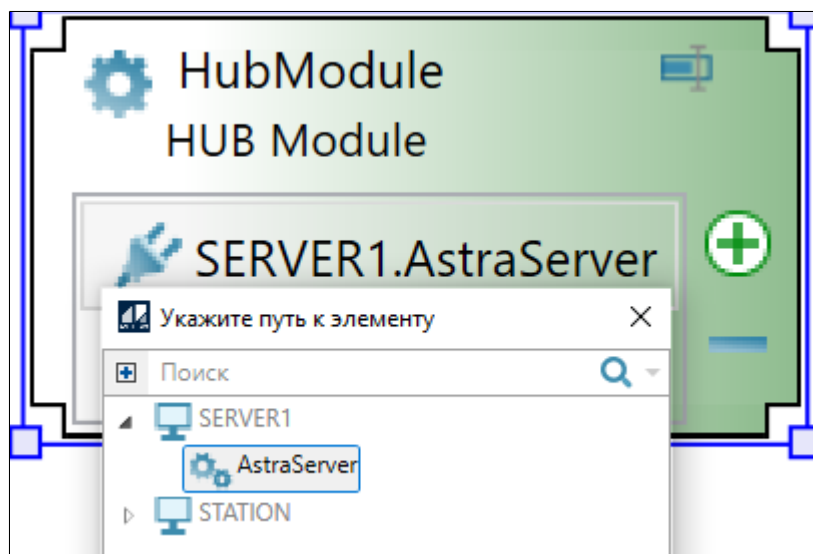


У элемента "Клиентский адаптер Ethernet" свойство "Адрес" отсутствует, поскольку элементу может соответствовать любое количество АРМ.

3. Перетащите в рабочую область из Панели элементов компонент "Astra.AccessPoint".



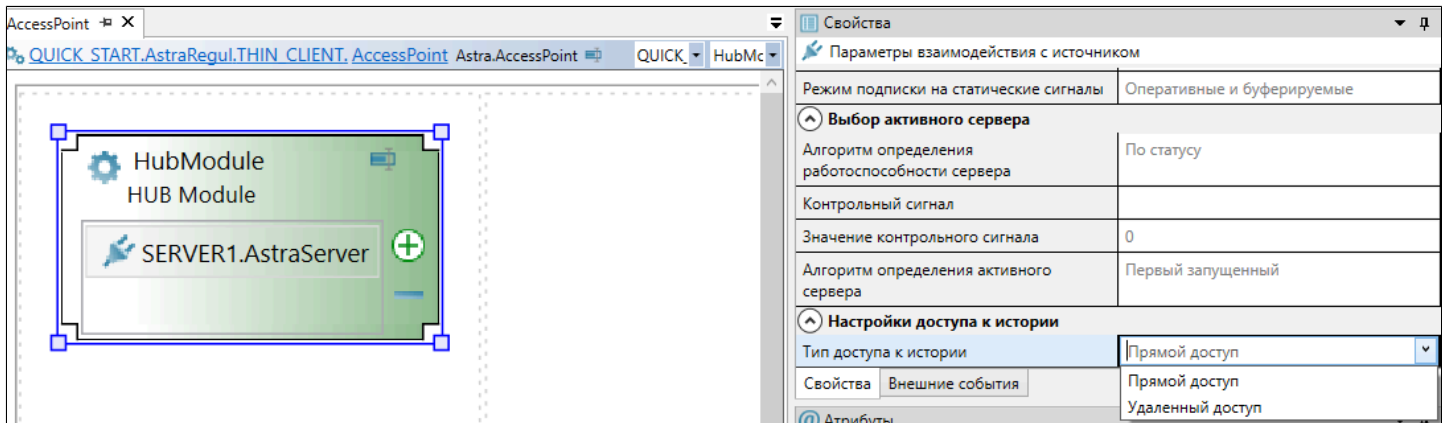
4. Перейдите в компонент "AccessPoint" и добавьте в "HubModule" серверы, с которыми будет осуществляться взаимодействие.



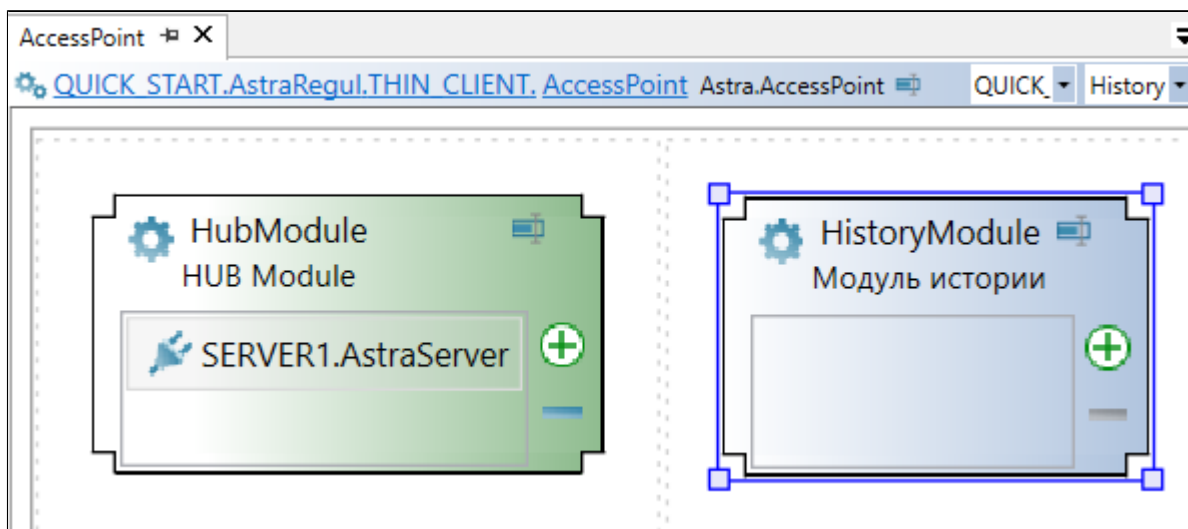
5. Для каждого сервера настройте параметры.



Рекомендуется выставлять "Тип доступа" к истории: "Удаленный доступ".

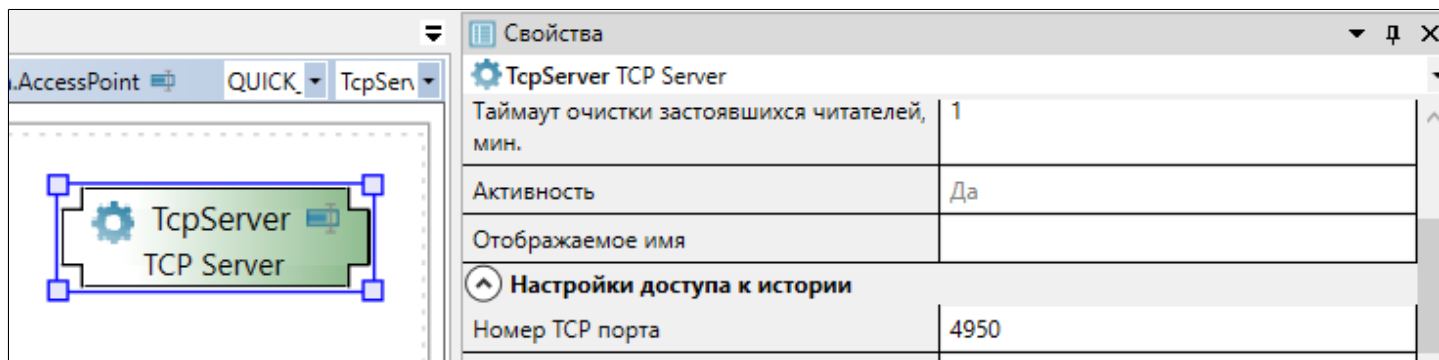


6. Перетащите в рабочую область из Панели элементов компонент "Модуль истории".

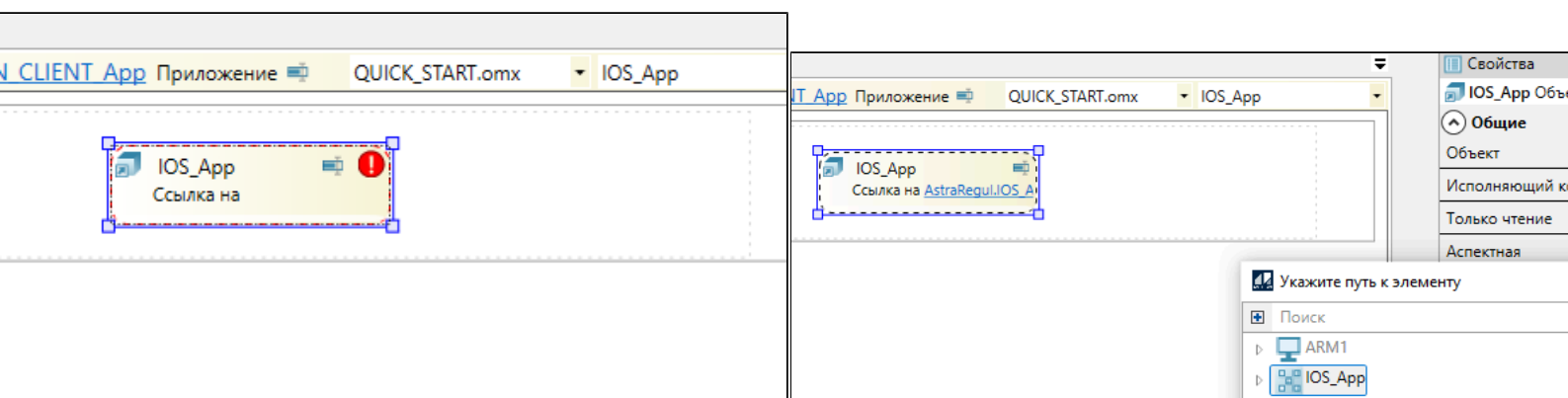


7. Перетащите в рабочую область из Панели элементов компонент "TCP-сервер".

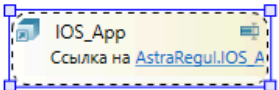
Выделите элемент и в окне "Свойства" укажите "Номер TCP-порта": 4950.



8. Добавьте ссылку на приложение "IOS\_App", размещённое в "Astra.Domain". Укажите имя ссылки – оно станет именем папки, в которой будет создаваться дерево сигналов.



9. Для ссылки добавьте атрибут "Раскрывать ссылку динамически" из библиотеки Om.Server.



**Общие**

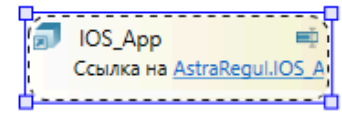
Объект	AstraRegul.IOS_App
Исполняющий компонент	
Только чтение	Нет
Аспектная	Нет
Имя	IOS_App

Свойства | Внешние события

**@ Атрибуты**

**IOS\_App** Объектная ссылка

Атрибут	Значение
<div style="border: 1px solid gray; padding: 5px; margin: 5px auto; width: fit-content;"> <p>+ Добавить новые атрибуты... Insert</p> <p>✕ Удалить выделенные атрибуты Delete</p> <p>Изменить тип атрибута...</p> <p>↩ Перейти к определению F12</p> </div>	



**Общие**

Объект	
Исполняющий компонент	
Только чтение	
Аспектная	
Имя	

Свойства | Внешние события

**@ Атрибуты**

**IOS\_App** Объектная ссылка

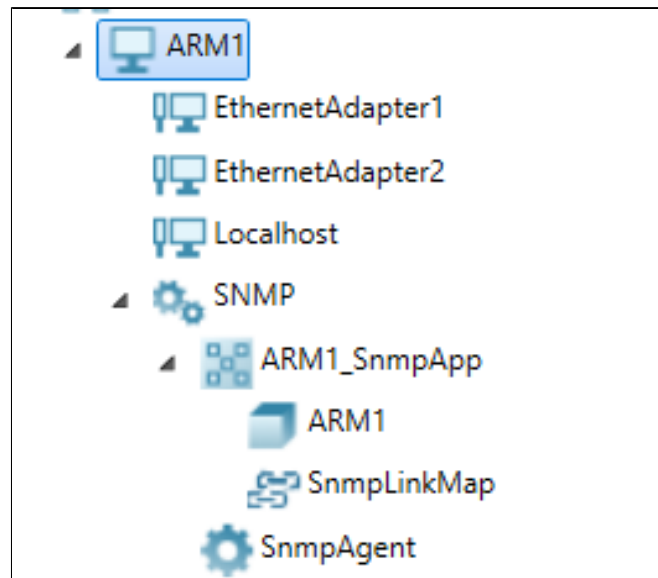
Атрибут	Значение
Раскрывать ссылку динамически	

В процессе работы `Astra.AccessPoint` ссылка динамически раскрывается: в дереве сигналов `Astra.AccessPoint` создаётся папка, в которой создаётся копия дерева сигналов объекта или приложения, на которое указывает привязка. При изменении значений сигналов в `Astra.Server`, новые значения автоматически передаются и записываются в соответствующие сигналы в `Astra.AccessPoint`. Таким образом `Astra.AccessPoint` может получать данные с одного или нескольких экземпляров `Astra.Server` и предоставлять их клиентским приложениям.

## 1.1.3.2.2.6.1. SNMP-диагностика



Для диагностики APM после [экспорта проекта в Astra.IDE](#) в топологию автоматически добавляется компонент "Компьютер" с компонентом "SNMP".



После экспорта необходимо добавить в [приложении "ARM1\\_SnmpApp"](#) тип "COMPUTER" из библиотеки "PsDiagn" и [заполнить карту адресов](#). Все остальные компоненты ARM настраиваются автоматически с помощью плагина AstraRegul.

## Ручная настройка SNMP диагностики для APM

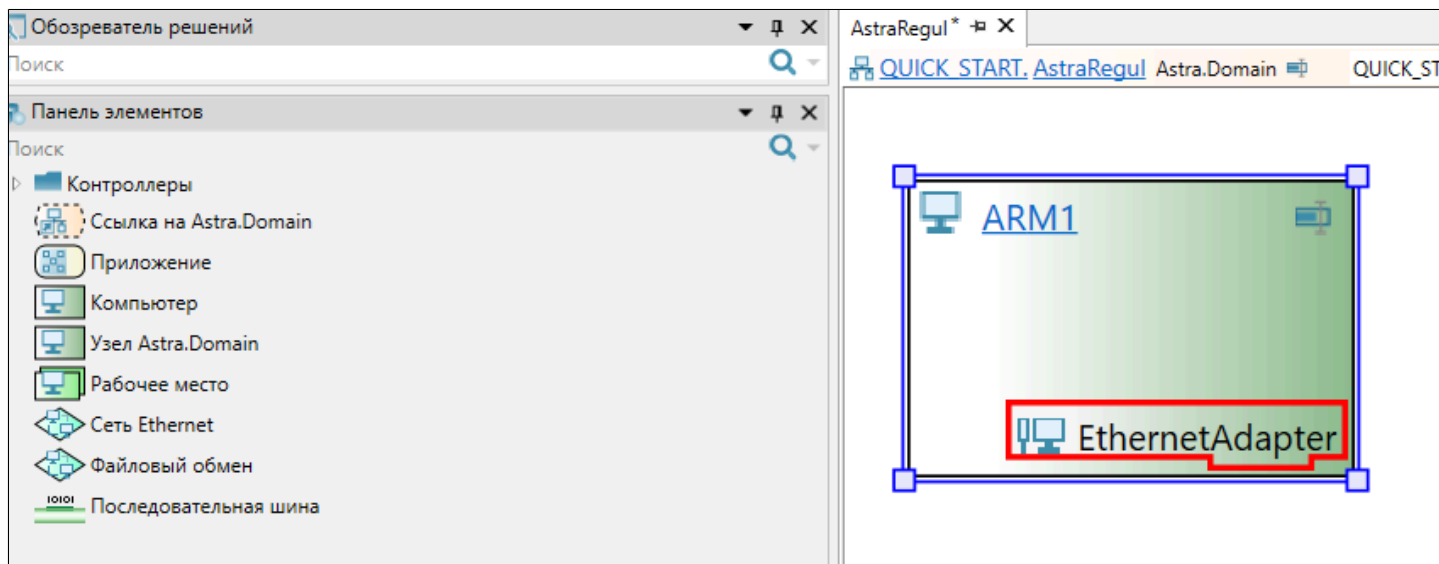
Для ручной настройки диагностики APM по SNMP:

1. Перейдите в домен и перетащите в рабочую область компонент "Компьютер" из Панели элементов.



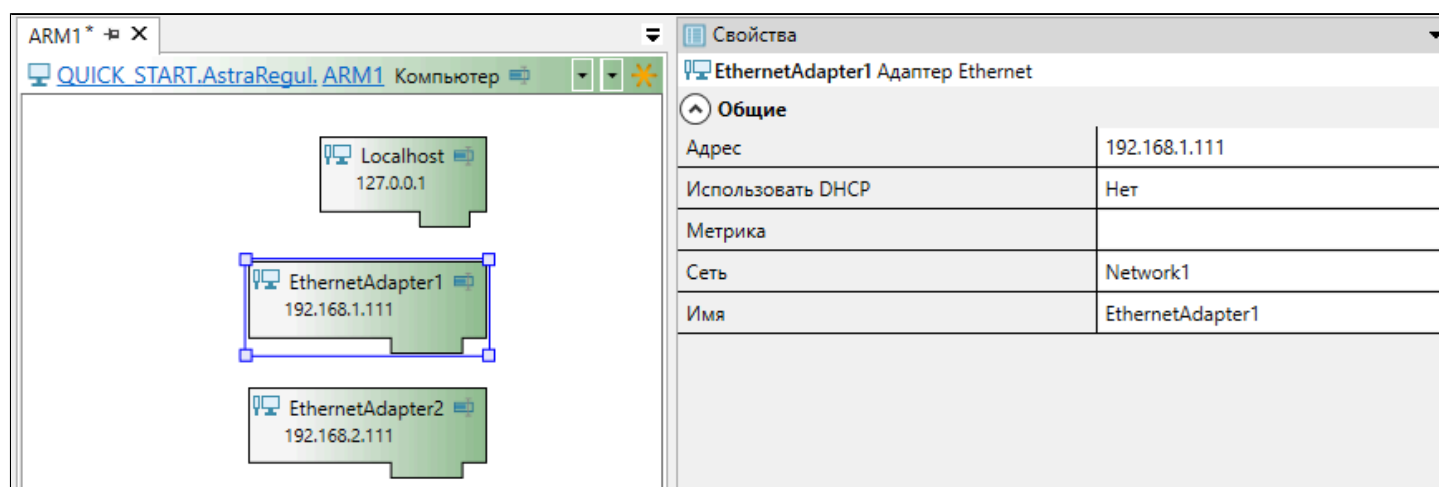
Имя элемента "Компьютер" должно совпадать с сетевым именем компьютера.



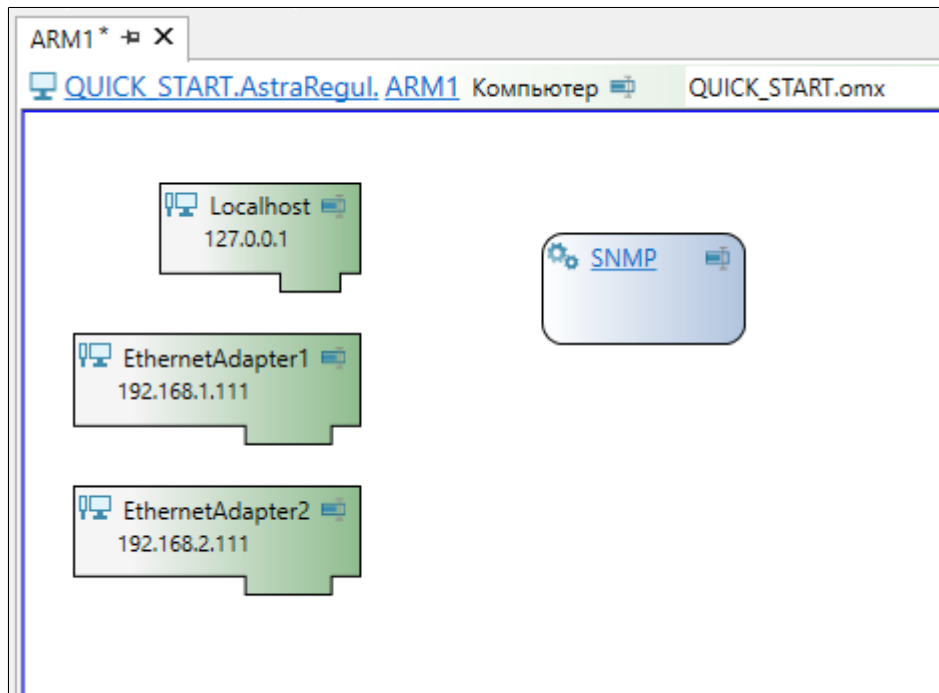


2. Перейдите в элемент "Компьютер" и перетащите в рабочую область из "Панели элементов" необходимое количество Ethernet-адаптеров в соответствии с физическими адаптерами АРМ.

Для каждого адаптера укажите IP-адрес и соответствующую сеть.

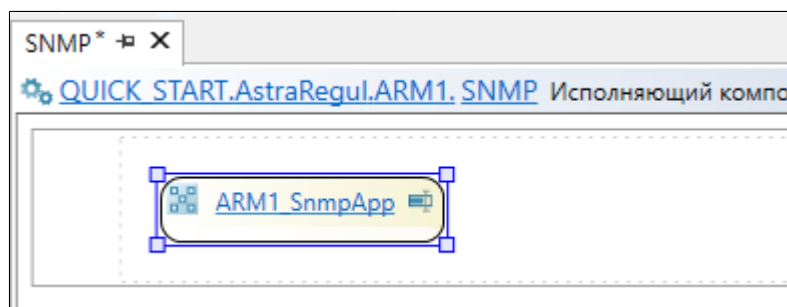


3. Перетащите в рабочую область из Панели элементов "Исполняющий компонент". Задайте ему имя "SNMP".

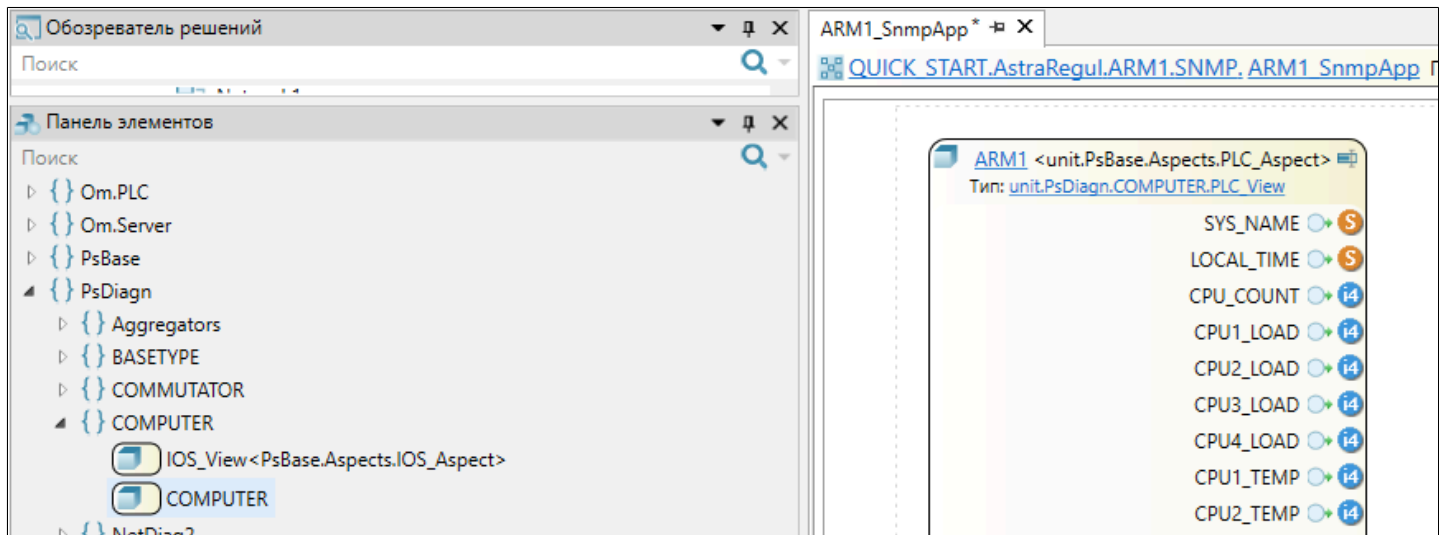


4. Перейдите в исполняющий компонент SNMP и перетащите в рабочую область из Панели элементов компонент "Приложение".

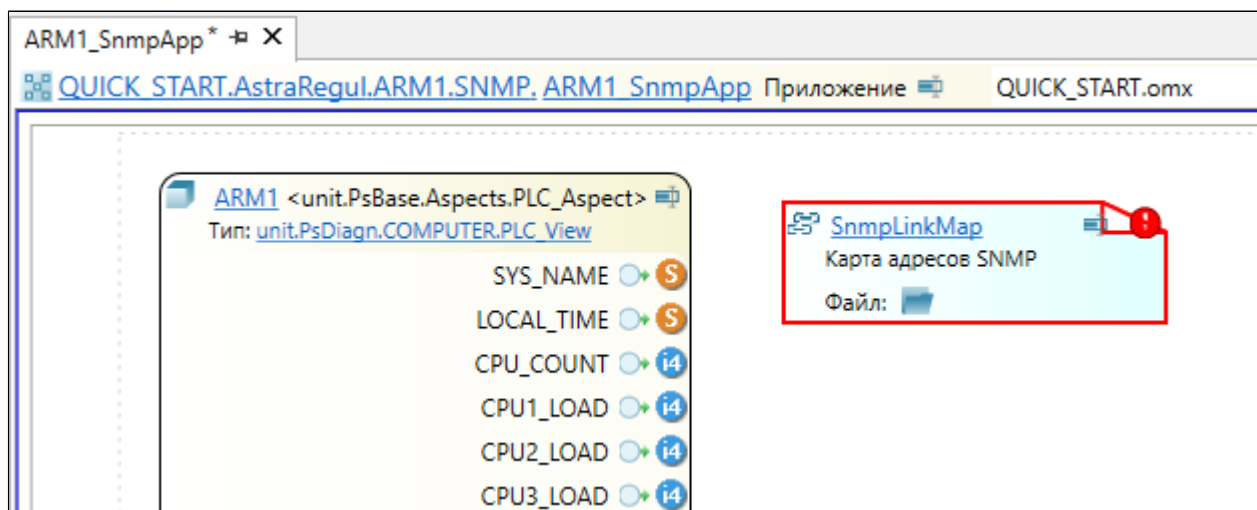
**i** Рекомендуется имя приложения задавать в формате [Имя компьютера]\_SnmpApp.



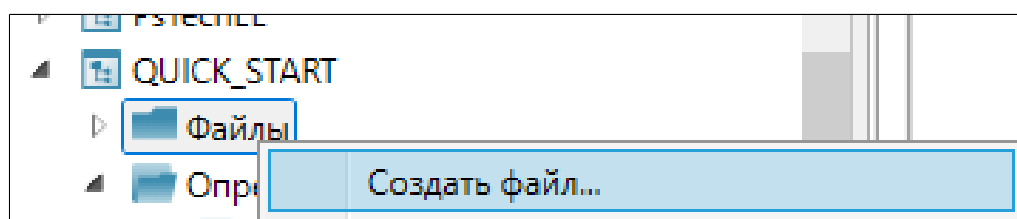
5. Перейдите в приложение и перетащите в рабочую область из библиотеки "PsDiagn" Панели элементов тип "COMPUTER". Имя экземпляра типа должно соответствовать имени компьютера.



6. Перейдите в Приложение и перетащите в рабочую область из Панели элементов компонент "Карта адресов SNMP".



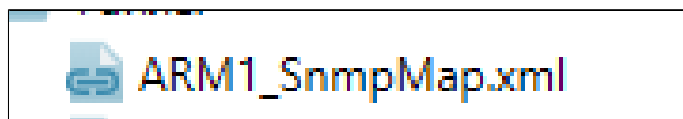
7. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." в контекстном меню объекта "Файлы" в Обозревателе решений.



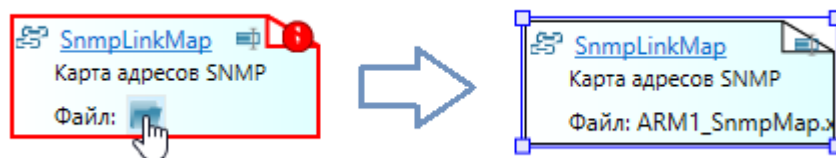
В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".



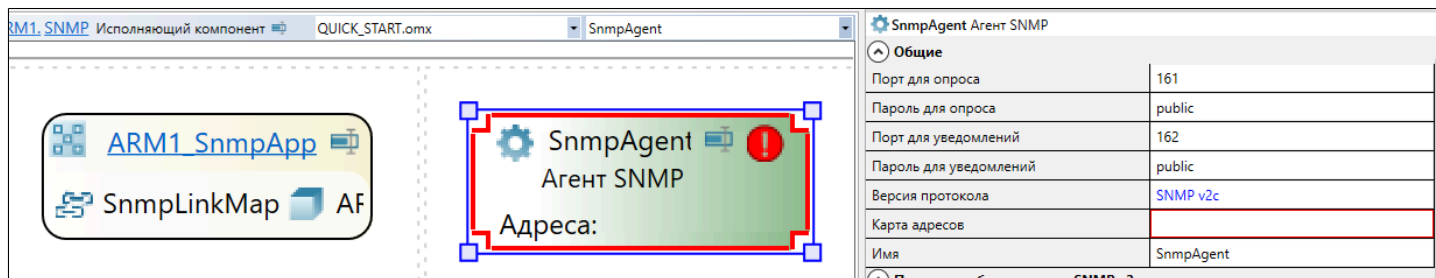
Имя файла рекомендуется задавать в формате [Имя компьютера]\_SnmpMap.



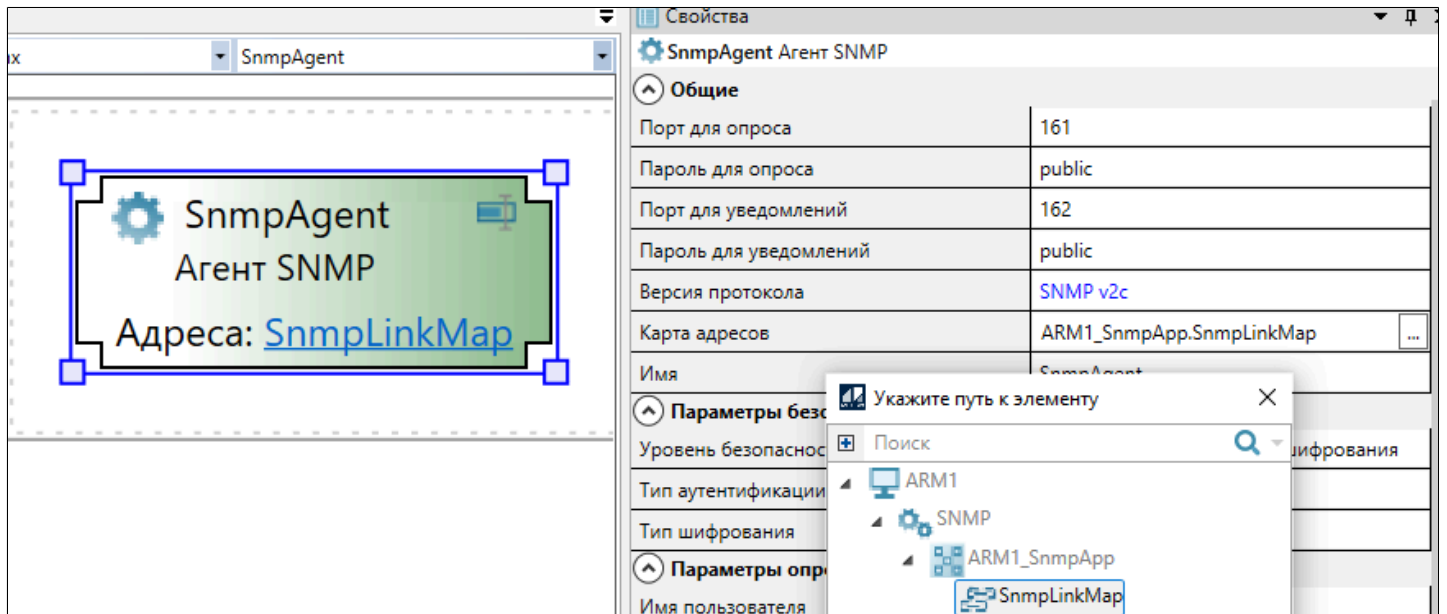
8. Перейдите в Приложение исполняющего компонента SNMP. У объекта "Карта адресов SNMP" укажите созданный файл.



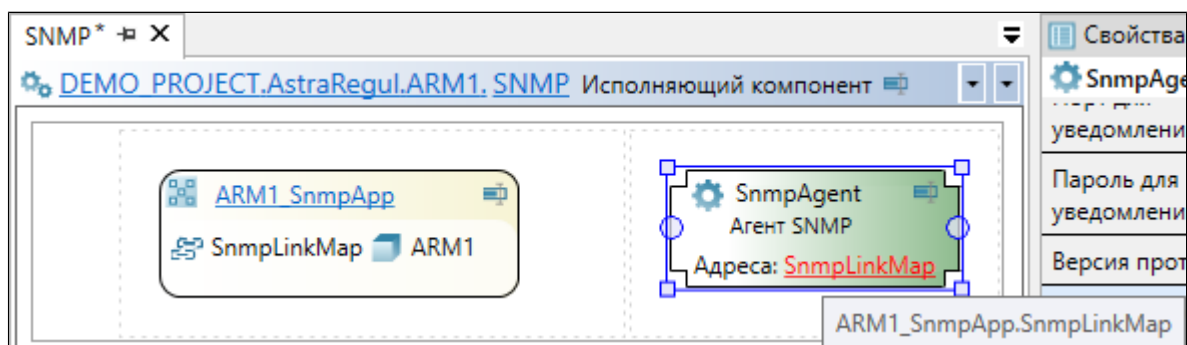
9. Перейдите в исполняющий компонент SNMP и перетащите из панели элементов в рабочую область элемент "Агент SNMP".



10. Выделите "Агент SNMP". В свойствах элемента укажите добавленную карту адресов и параметры агента: порты для опроса и уведомлений, версию протокола и т.д.



11. Откройте карту адресов с помощью ссылки.



12. Заполните карту адресов.

Поле	Значение
Привязка	Укажите "Непосредственно", если параметры адреса заданы в данной карте адресов.
OID элемента	Укажите OID параметра. OID можно узнать в документации на устройство или извлечь из MIB устройства с помощью утилит (например, <a href="#">MIB Browser</a> ).
Индекс элемента массива	Индекс элемента в массиве. Указывается только если OID указывает на массив.
Композитный тип объекта	Указывается только для сложных типов данных: <ul style="list-style-type: none"> <li>➤ Дата и время.</li> <li>➤ Физический адрес.</li> </ul>

	<ul style="list-style-type: none"> <li>› IP-адрес.</li> </ul> <p>Для всех остальных типов оставьте поле пустым.</p>
Способ получения данных	<p>Возможные варианты:</p> <ul style="list-style-type: none"> <li>› Только через опрос.</li> <li>› Только через уведомления.</li> <li>› Через опрос и уведомления.</li> <li>› Пусто – равносильно значению Через опрос и уведомления.</li> </ul>

Сигнал	Тип	Привязка	OID элемента
ARM1.SYS_NAME	string	непосредственно	1.3.6.1.2.1.1.5
ARM1.LOCAL_TIME	string	непосредственно	1.3.6.1.2.1.25.1.2.0
ARM1.CPU_COUNT	int4	не привязан	
ARM1.CPU1_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.1
ARM1.CPU2_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.2
ARM1.CPU3_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.3
ARM1.CPU4_LOAD	int4	непосредственно	1.3.6.1.2.1.25.3.3.1.2.4
ARM1.CPU1_TEMP	int4	не привязан	
ARM1.CPU2_TEMP	int4	не привязан	
ARM1.CPU3_TEMP	int4	не привязан	
ARM1.CPU4_TEMP	int4	не привязан	
ARM1.DISC_COUNT	int4	не привязан	
ARM1.DISC1_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.1
ARM1.DISC2_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.2
ARM1.DISC3_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.3
ARM1.DISC4_SPACE	int4	непосредственно	1.3.6.1.2.1.25.2.3.1.6.4
ARM1.DISC1_NAME	string	непосредственно	1.3.6.1.2.1.25.2.3.1.3.1
ARM1.DISC2_NAME	string	непосредственно	1.3.6.1.2.1.25.2.3.1.3.2

После настройки АРМ сможет отдавать диагностические параметры по SNMP. Аналогично настраиваются все АРМ.

## 1.1.3.2.2.6.2. Резервирование

Для резервирования АРМов используется дублирование: в проект автоматизации добавляются несколько АРМов, которые получают одинаковые данные и выполняют одни и те же функции.



Одному элементу Рабочее место, могут соответствовать любое количество АРМов.

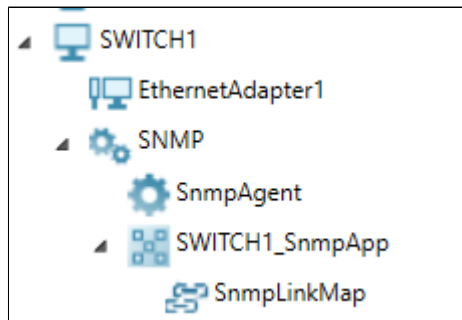
Чтобы получать диагностику по SNMP с каждого АРМ, необходимо выполнить пункт [SNMP-диагностика](#) для каждого АРМ.

## 1.1.3.2.2.7. Коммутатор

**Коммутатор** - это прибор, который объединяет устройства в одну сеть для обмена данными.



Коммутатор вместе с исполняющим компонентом "SNMP" добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).



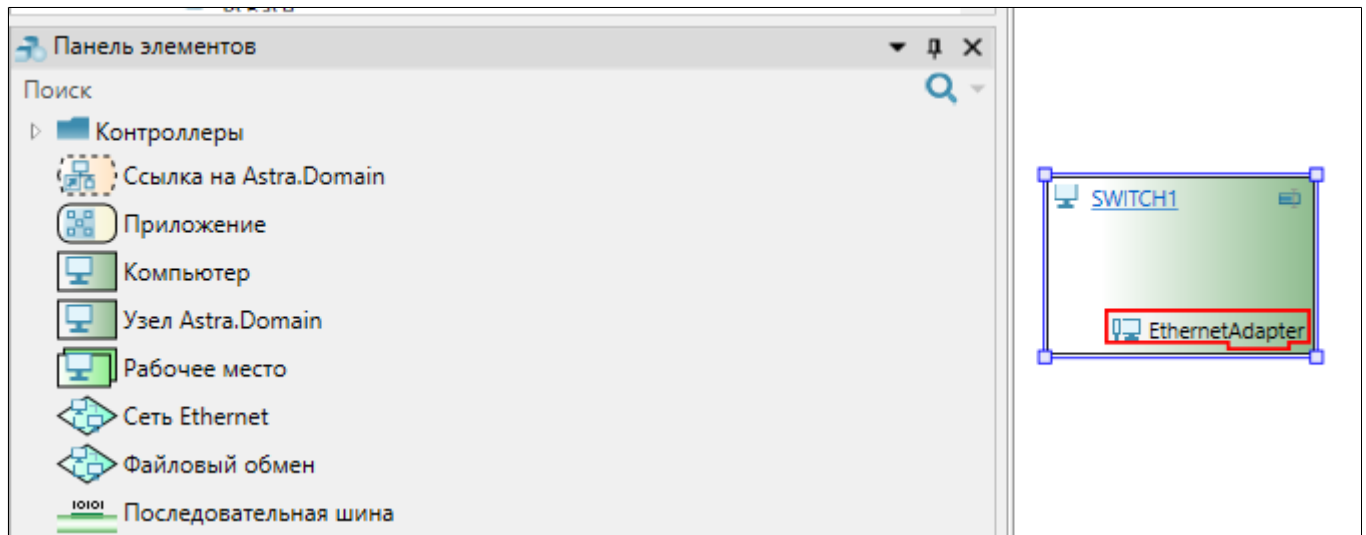
После экспорта необходимо добавить в приложении "SWITCH1\_SnmpApp" тип "COMMUTATOR" из библиотеки "PsDiagn" и заполнить карту адресов. Все остальные компоненты коммутатора настраиваются автоматически с помощью плагина AstraRegul.

## Ручное добавление коммутатора

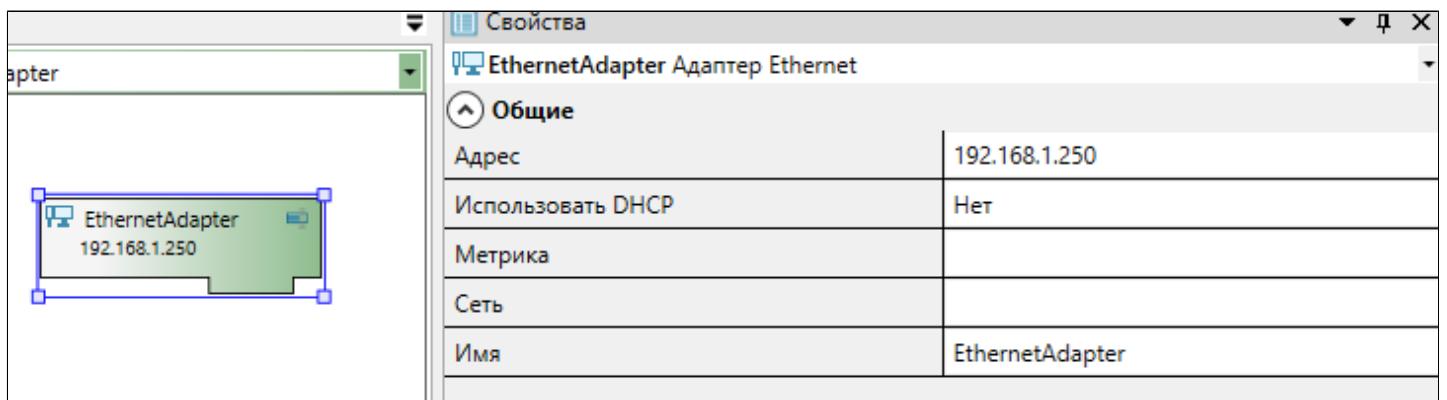
Чтобы описать коммутатор в топологии вручную, необходимо выполнить следующие действия:

1. Перейдите в компонент "Astra.Domain" и перетащите в рабочую область из "Панели элементов" компонент "Компьютер".





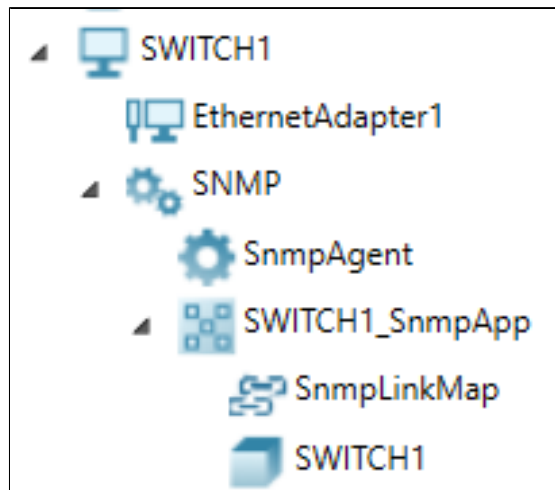
2. Перейдите в элемент "Компьютер" и задайте адрес коммутатора в свойствах компонента "EthernetAdapter".



## 1.1.3.2.2.7.1. SNMP-диагностика



Для диагностики коммутатора после [экспорта проекта в Astra.IDE](#) в топологию автоматически добавляется компонент "Компьютер" с компонентом "SNMP".

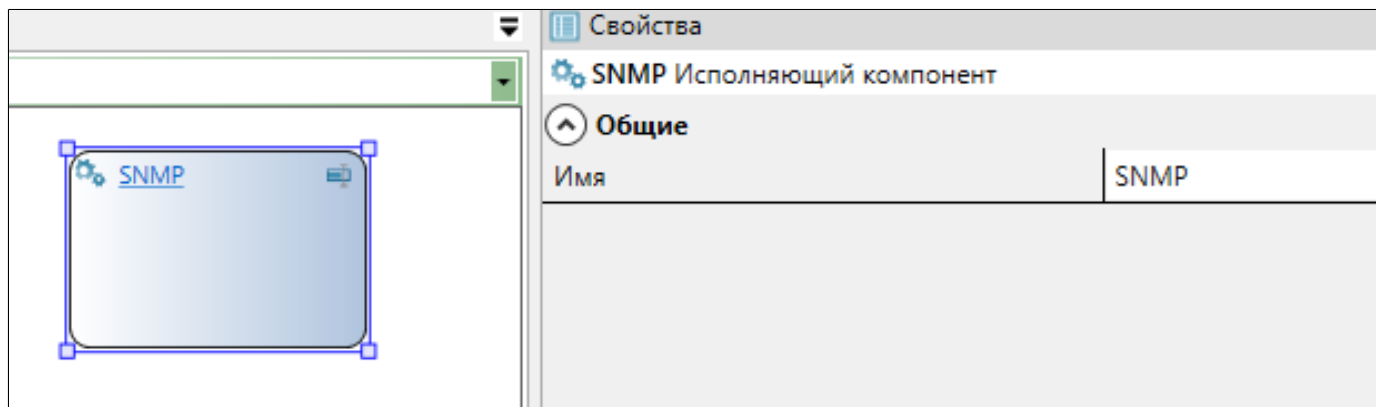


После экспорта необходимо добавить в [приложении "SWITCH1\\_SnmpApp"](#) тип "COMMUTATOR" из библиотеки "PsDiagn" и [заполнить карту адресов](#). Все остальные компоненты коммутатора настраиваются автоматически с помощью плагина AstraRegul.

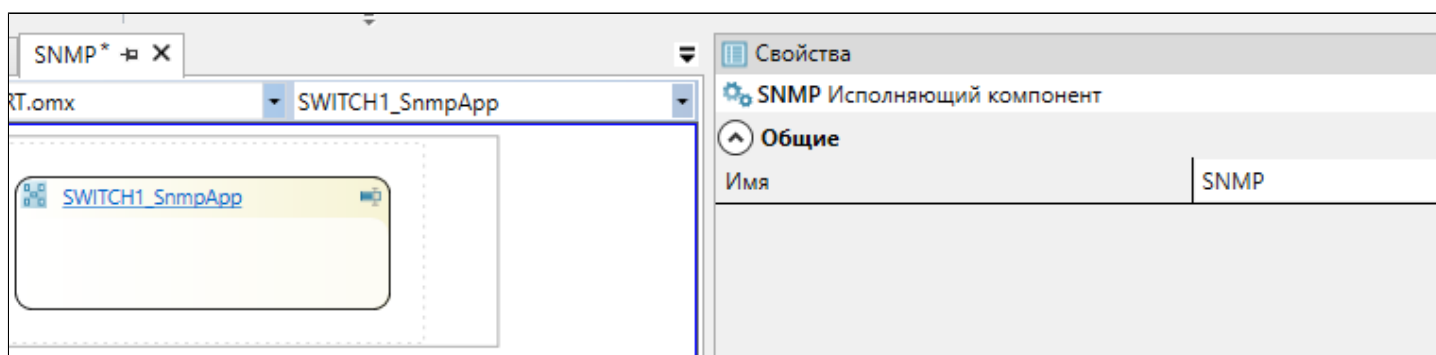
## Ручная настройка SNMP диагностики для коммутатора

Для настройки диагностики коммутатора по SNMP вручную, выполните следующие действия:

1. Перейдите в компонент "Компьютер" и перетащите в рабочую область из "Панели элементов" "Исполняющий компонент". Задайте ему имя "SNMP".

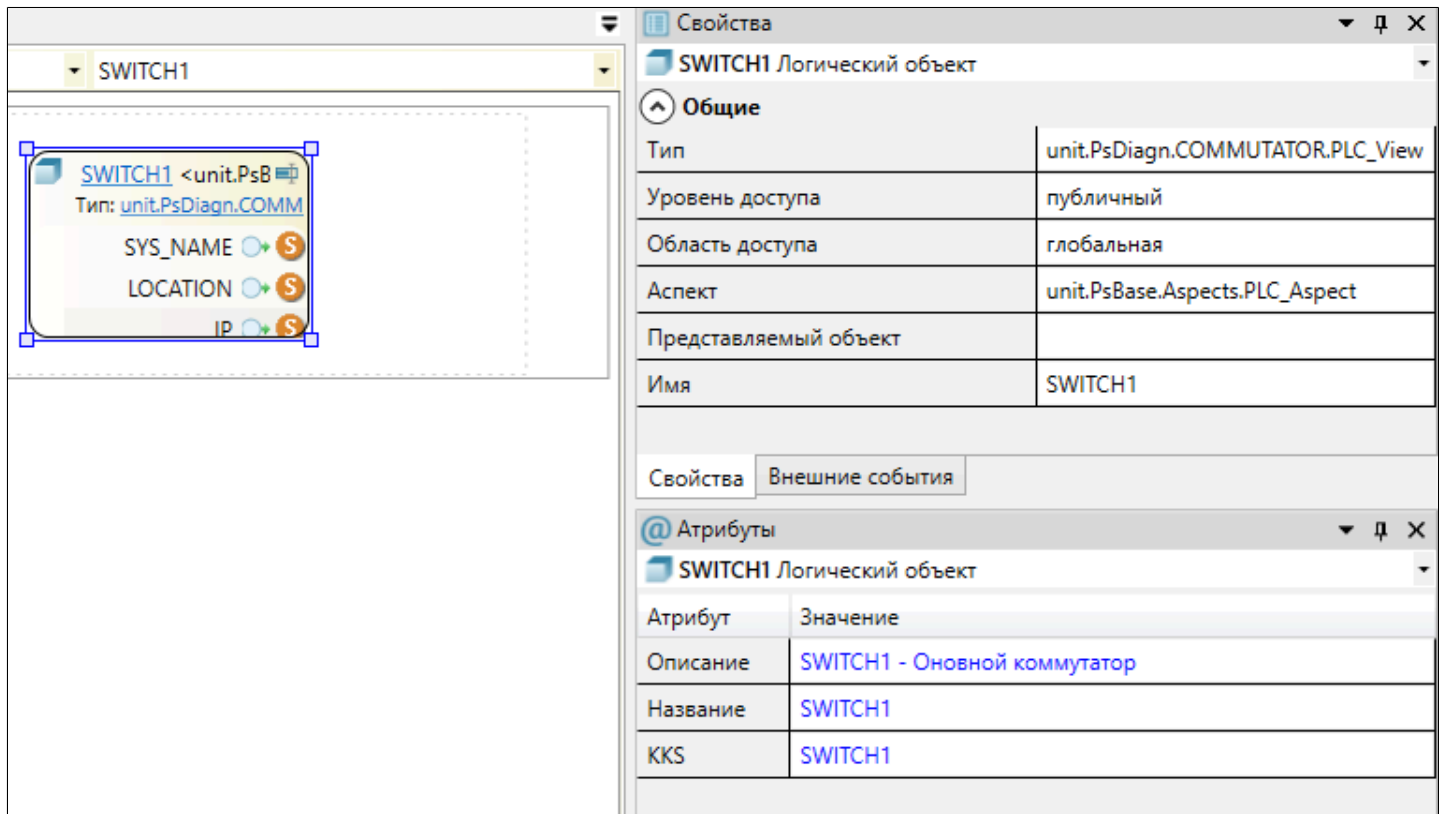


2. Перейдите в исполняющий компонент "SNMP" и перетащите в рабочую область из "Панели элементов" компонент "Приложение".

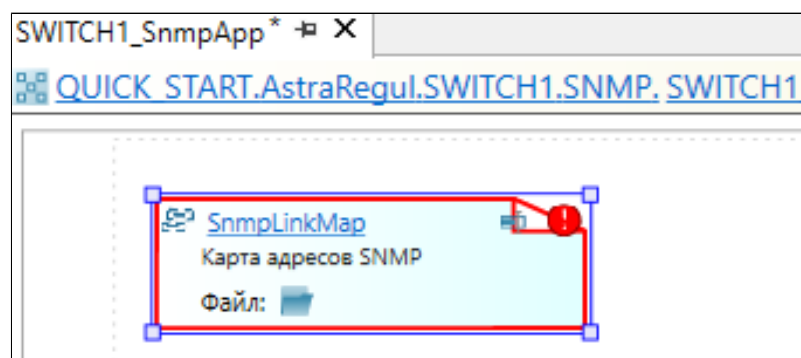


Рекомендуется имя приложения задавать в формате [Имя коммутатора]\_SnmpApp.

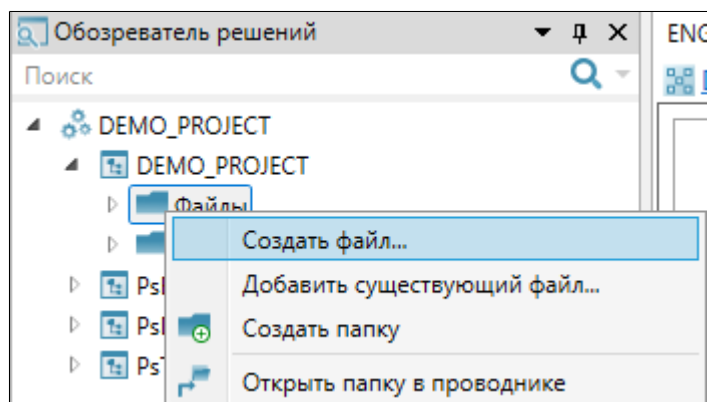
3. Перейдите в Приложение и перетащите в рабочую область из библиотеки "PsDiagn" Панели элементов тип "COMMUTATOR".



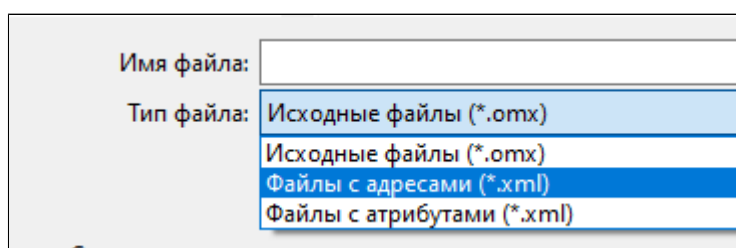
4. Перейдите в Приложение и перетащите в рабочую область из Панели элементов компонент "Карта адресов SNMP".



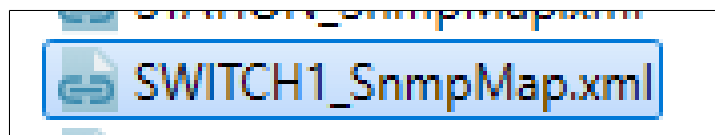
5. Чтобы добавить файл для карты адресов, его необходимо создать. Выберите команду "Создать файл..." в контекстном меню объекта "Файлы" в Обзорателе решений.



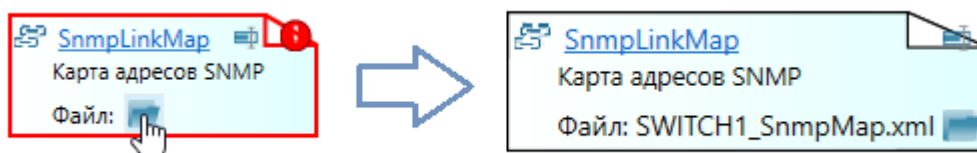
В диалоговом окне выберите тип файла "Файлы с адресами (\*.xml)".



Имя файла рекомендуется задавать в формате [Имя коммутатора]\_SnmpMap.

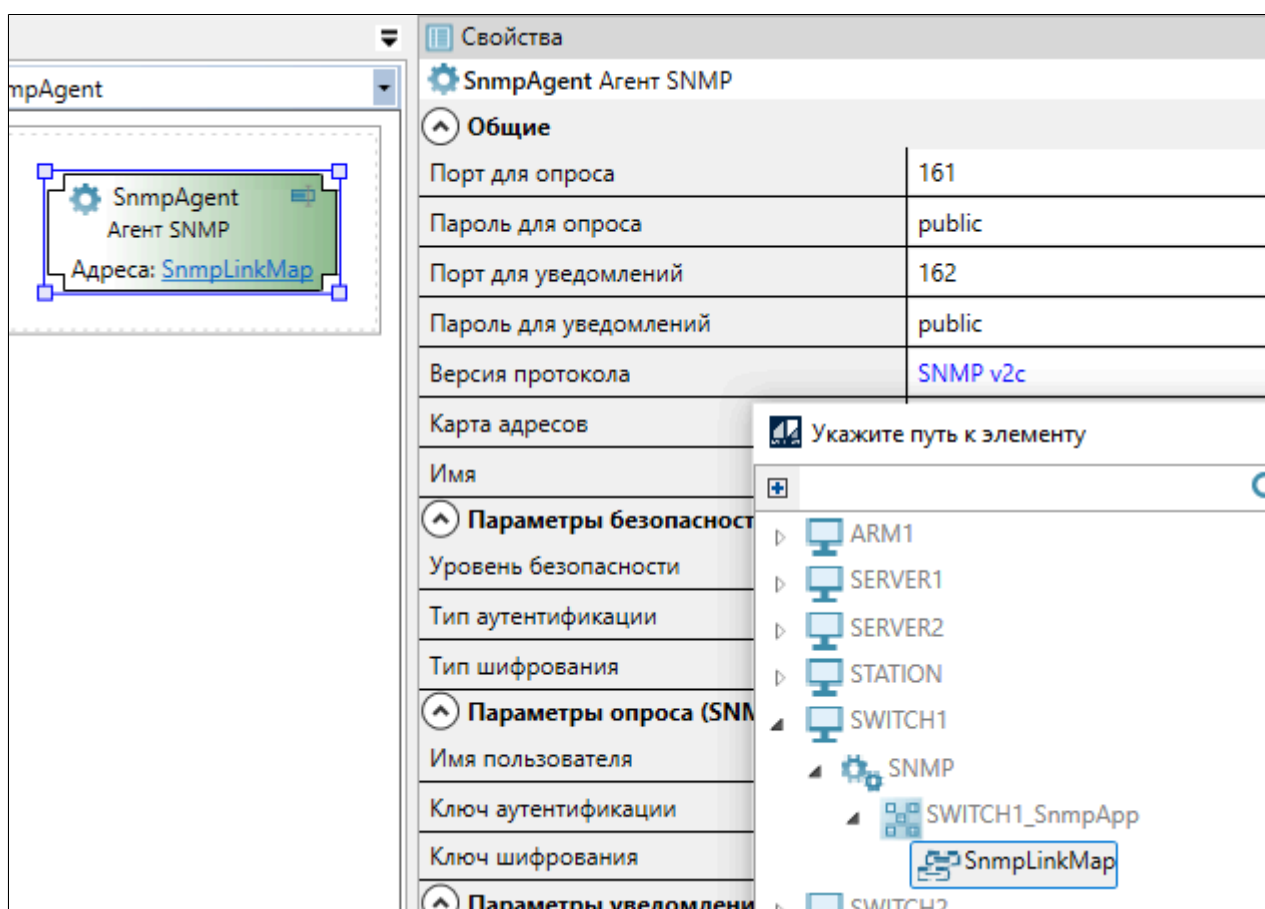


6. Перейдите в Приложение исполняющего компонента SNMP. У объекта "Карта адресов SNMP" укажите созданный файл.

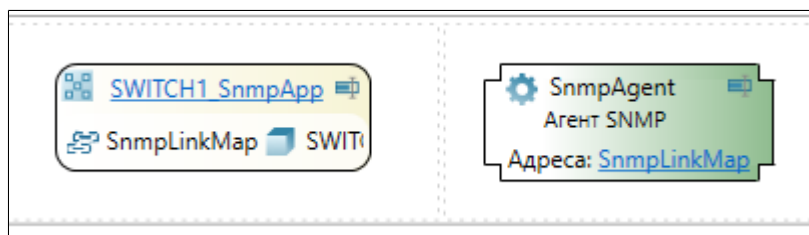


7. Перейдите в исполняющий компонент "SNMP" и перетащите из Панели элементов в рабочую область элемент "Агент SNMP".

8. Выделите Агент SNMP. В свойствах элемента укажите добавленную карту адресов и параметры агента: порты для опроса и уведомлений, версию протокола и т.д.



9. Откройте карту адресов с помощью ссылки.



10. Заполните карту адресов.

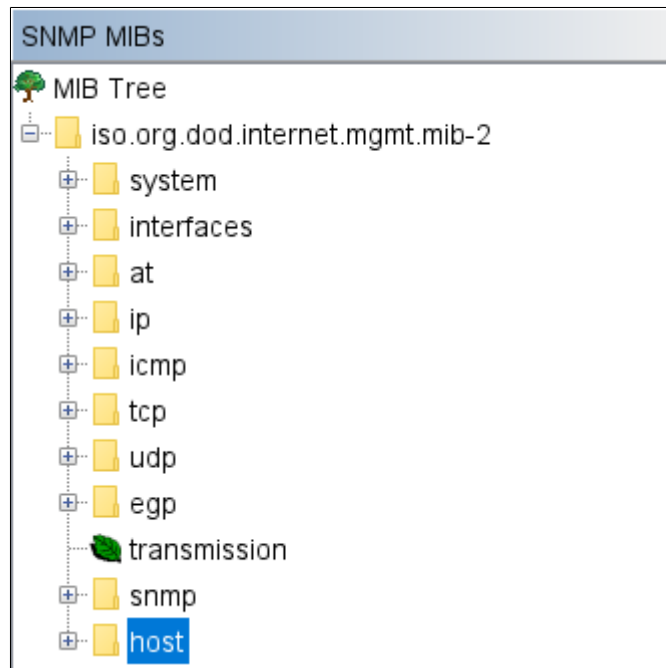
Поле	Значение
Привязка	Укажите "Непосредственно", если параметры адреса заданы в данной карте адресов.

OID элемента	Укажите OID параметра. OID можно узнать в документации на устройство или извлечь из MIB устройства с помощью утилит (например, <a href="#">MIB Browser</a> ).
Индекс элемента массива	Индекс элемента в массиве. Указывается, если OID указывает на массив.
Композитный тип объекта	Указывается только для сложных типов данных: <ul style="list-style-type: none"> <li>➤ Дата и время.</li> <li>➤ Физический адрес.</li> <li>➤ IP-адрес.</li> </ul> Для остальных типов оставьте поле пустым.
Способ получения данных	Возможные значения: <ul style="list-style-type: none"> <li>➤ Только через опрос.</li> <li>➤ Только через уведомления.</li> <li>➤ Через опрос и уведомления.</li> <li>➤ Пусто – равносильно значению Через опрос и уведомления.</li> </ul>

Для заполнения поля "OID элемента" необходимо воспользоваться MIB Browser.

11. Запустите MIB Browser, в дереве перейдите к элементу "mib-2" (iso → org → dod → internet → mgmt → mib-2):

- в ветви "system" содержится информация о системе;
- в ветви "host" содержится информация о машине (хосте);
- в ветви "ip" содержится информация о портах машины (хоста).



Для получения информации об SNMP диагностике об устройстве используйте, указанный в свойстве "Порт для опроса" (раздел "Диагностика") соответствующего устройства, при конфигурировании топологии системы.

По умолчанию номер порта для опроса - 161.

12. Найдите соответствующий сигнал из карты адресов SNMP в дереве MIB Browser, нажмите левой кнопкой мыши на сигнал, скопируйте свойство "OID" и вставьте в поле "OID элемента" карты адресов (без первой точки).



SNMP MIBs

MIB Tree

- iso.org.dod.internet.mgmt.mib-2
  - system
    - sysDescr
    - sysObjectID
    - sysUpTime
    - sysContact
    - sysName
    - sysLocation
    - sysServices
  - interfaces
    - at
    - ip
    - icmp
    - tcp
    - udp
    - egp
  - transmission
  - snmp
  - host

Name	sysName
OID	.1.3.6.1.2.1.1.5
MIB	RFC1213-MIB
Syntax	DisplayString (OCTET STRING) (SI...
Access	read-write
Status	mandatory

QUICK START.AstraRegul.SWITCH1.SNMP.SWITCH1 SnmpApp, SnmpLinkMap Карта адресов SNMP

Сигнал	Тип	Привязка	OID элемента
SWITCH1.SYS_NAME	string	непосредственно	1.3.6.1.2.1.1.5

13. Аналогичным образом заполните OID элементов для всех необходимых параметров.

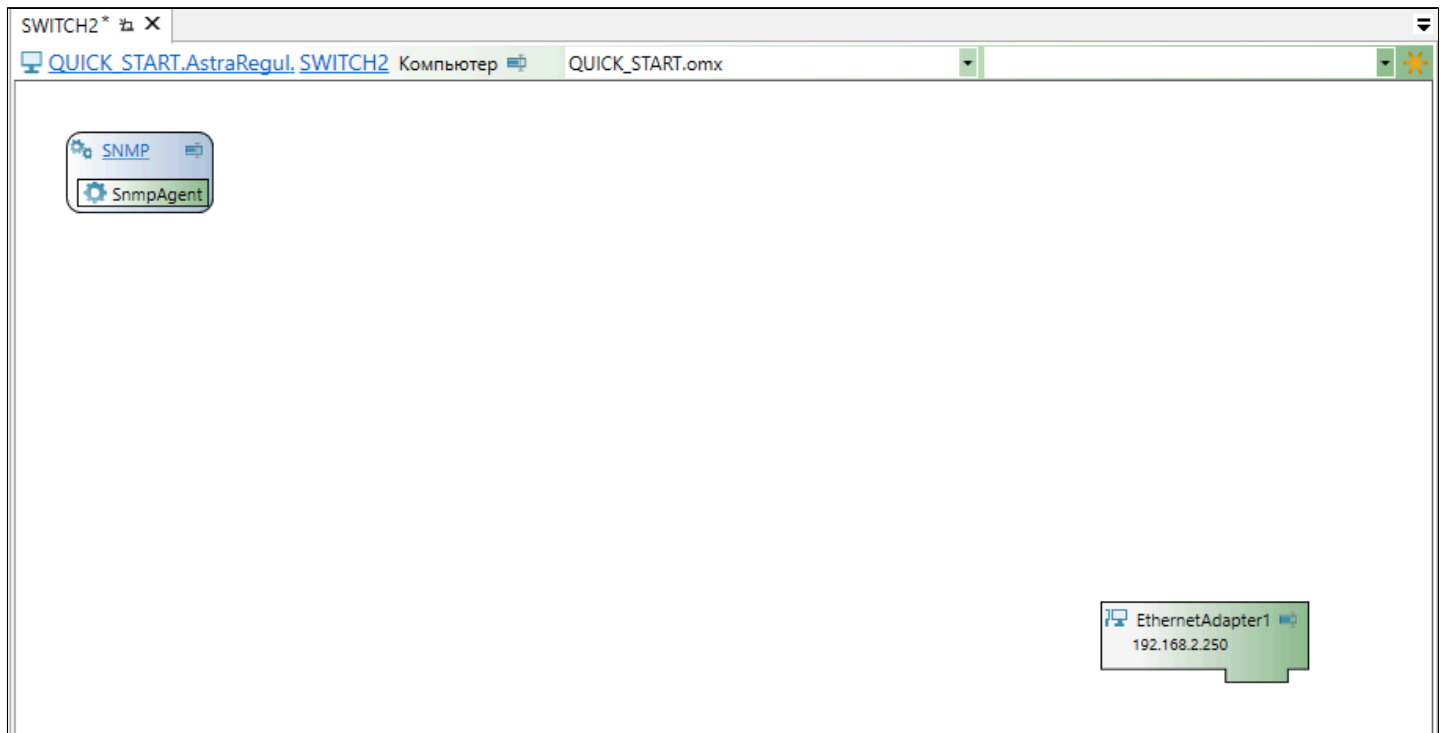
Сигнал	Тип	Привязка	OID элемента	Индекс элемен-	Композитный	Способ получения данных
SWITCH1.SYS_NAME	string	непосредственно	1.3.6.1.2.1.1.5	0		Через опрос и уведомления
SWITCH1.LOCATION	string	непосредственно	1.3.6.1.2.1.1.6	0		Через опрос и уведомления
SWITCH1.IP	string	непосредственно	1.3.6.1.2.1.4.20.1.1	2	IP адрес	Через опрос и уведомления
SWITCH1.SERVICES	int4	непосредственно	1.3.6.1.2.1.1.7	0		Через опрос и уведомления
SWITCH1.START_TIME	uint4	непосредственно	1.3.6.1.2.1.1.3	0	Дата и время	Через опрос и уведомления
SWITCH1.PORT01_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	48		Через опрос и уведомления
SWITCH1.PORT02_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	49		Через опрос и уведомления
SWITCH1.PORT03_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	50		Через опрос и уведомления
SWITCH1.PORT04_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	51		Через опрос и уведомления
SWITCH1.PORT05_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	52		Через опрос и уведомления
SWITCH1.PORT06_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	53		Через опрос и уведомления
SWITCH1.PORT07_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	54		Через опрос и уведомления
SWITCH1.PORT08_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	55		Через опрос и уведомления
SWITCH1.PORT09_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	56		Через опрос и уведомления
SWITCH1.PORT10_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	57		Через опрос и уведомления
SWITCH1.PORT11_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	58		Через опрос и уведомления
SWITCH1.PORT12_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	59		Через опрос и уведомления
SWITCH1.PORT13_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	60		Через опрос и уведомления
SWITCH1.PORT14_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	61		Через опрос и уведомления
SWITCH1.PORT15_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	62		Через опрос и уведомления
SWITCH1.PORT16_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	63		Через опрос и уведомления
SWITCH1.PORT17_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	64		Через опрос и уведомления
SWITCH1.PORT18_STATUS	int4	непосредственно	1.3.6.1.2.1.2.2.1.8	65		Через опрос и уведомления

14. Сохраните проект, нажав кнопку "Сохранить (Ctrl+Shift+S)".

## 1.1.3.2.2.7.2. Резервирование



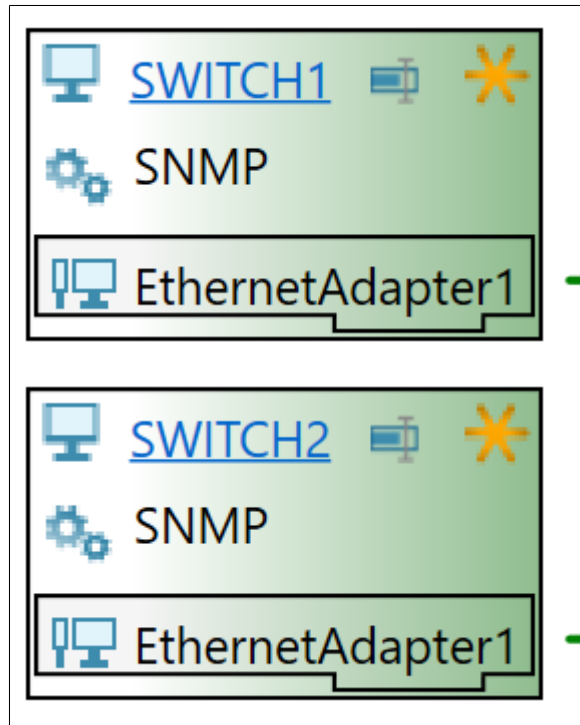
Резервный коммутатор вместе с исполняющим компонентом "SNMP" добавляются в топологию проекта автоматически при выполнении [экспорта проекта в Astra.IDE](#).



Для диагностики резервного коммутатора необходимо добавить в приложении "SWITCH2\_SnmpApp" тип "COMMUTATOR" из библиотеки "PsDiagn" и заполнить карту адресов. Все остальные компоненты коммутатора настраиваются автоматически с помощью плагина AstraRegul.

### Ручное добавление резервного коммутатора

Резервный коммутатор добавляется в топологию и настраивается аналогично с [ОСНОВНЫМ](#).



## 1.1.3.2.2.8. Сервер истории (БД)

Сервер истории (БД) - выделенный специализированный компьютер, который служит для хранения базы данных.

Можно добавить серверы истории следующих типов:

- › [Astra.Historian](#)
- › [Microsoft SQL Server](#)
- › [PostgreSQL](#)



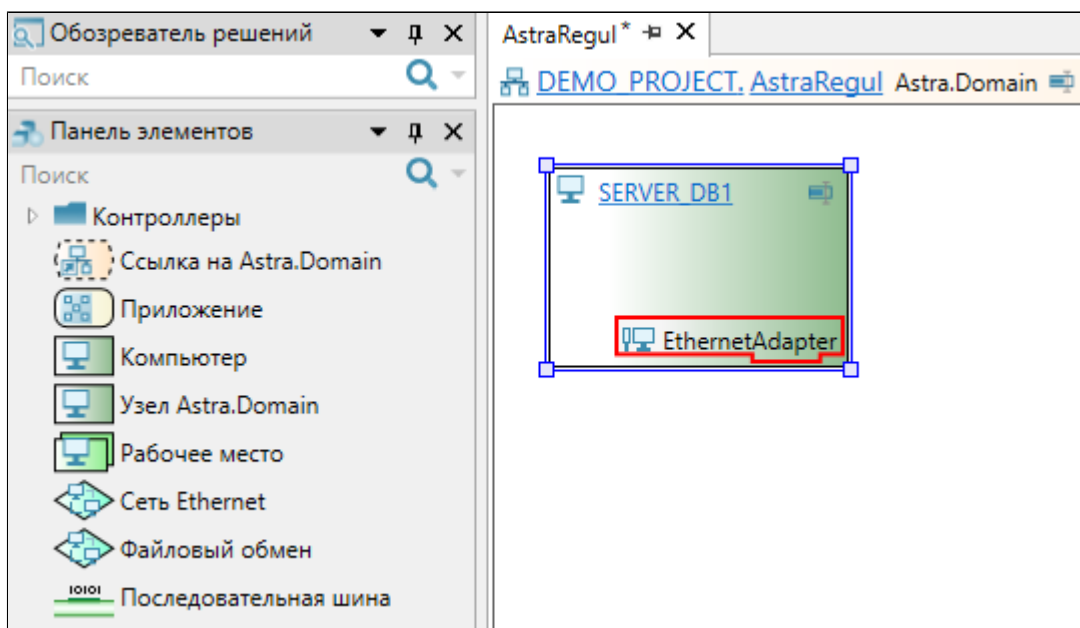
События можно сохранять/запрашивать только из Astra.Historian.

## 1.1.3.2.2.8.1. Astra.Historian

1. Перетащите в рабочую область домена из Панели элементов компонент "Узел Astra.Domain".

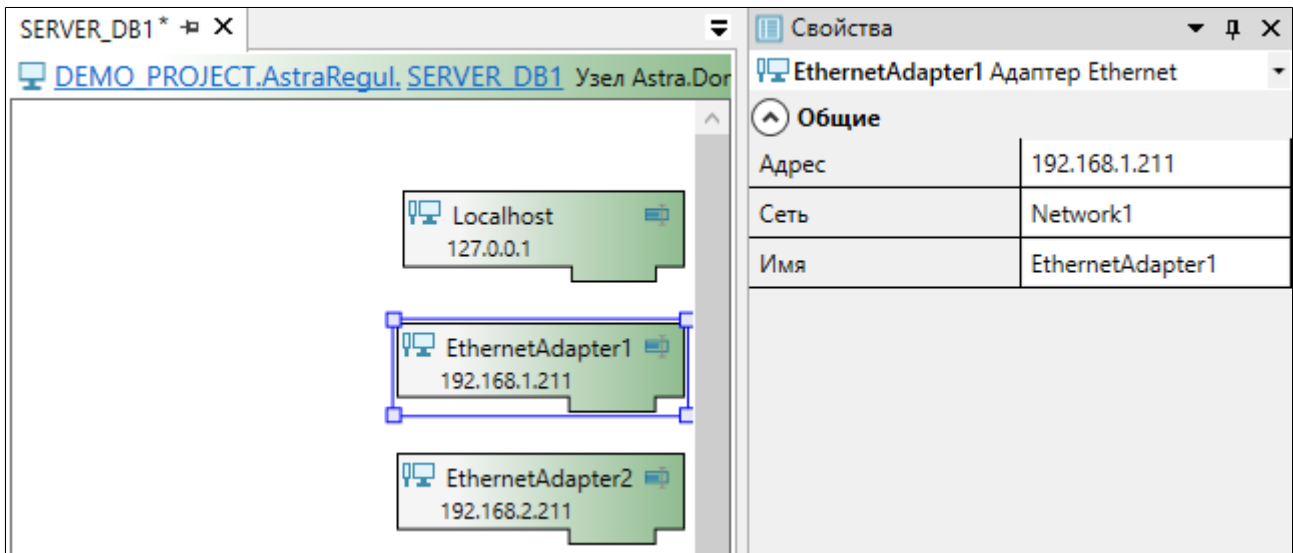


Имя узла домена должно совпадать с сетевым именем компьютера.

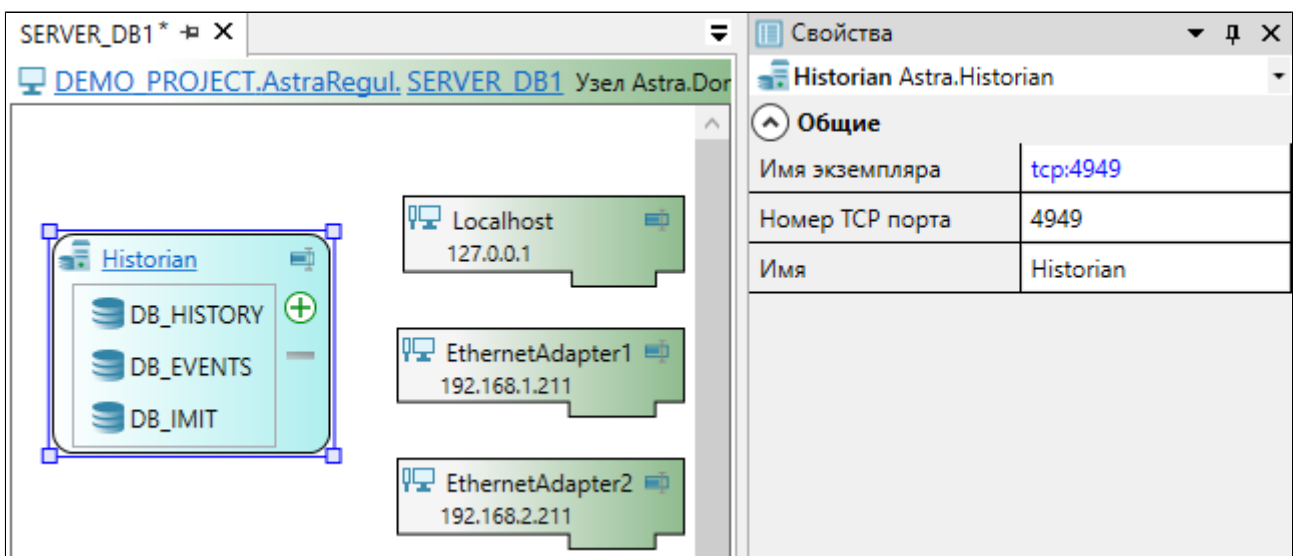


2. Перейдите в элемент "Узел домена" и перетащите в рабочую область из Панели элементов необходимое количество Ethernet-адаптеров в соответствии с физическими адаптерами компьютера.

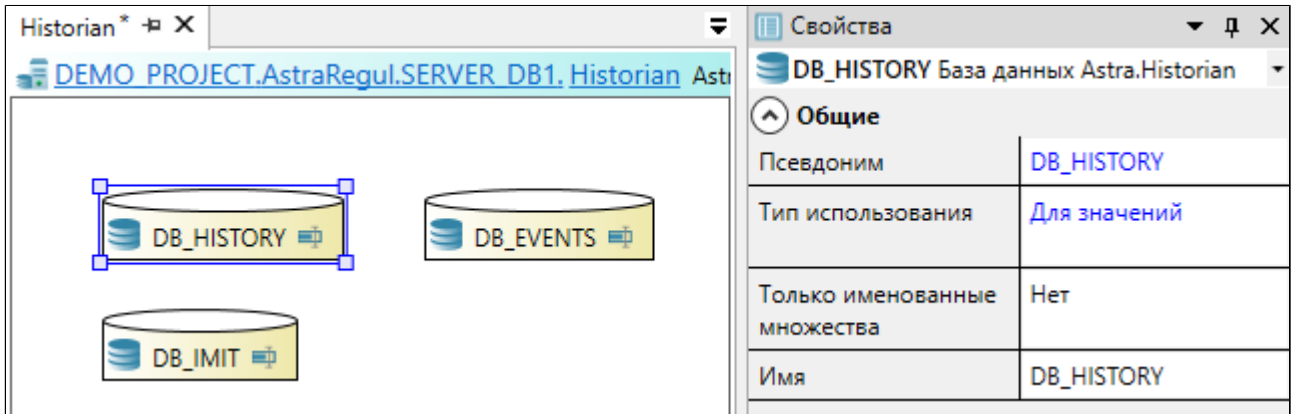
Для каждого адаптера укажите IP-адрес и соответствующую сеть.



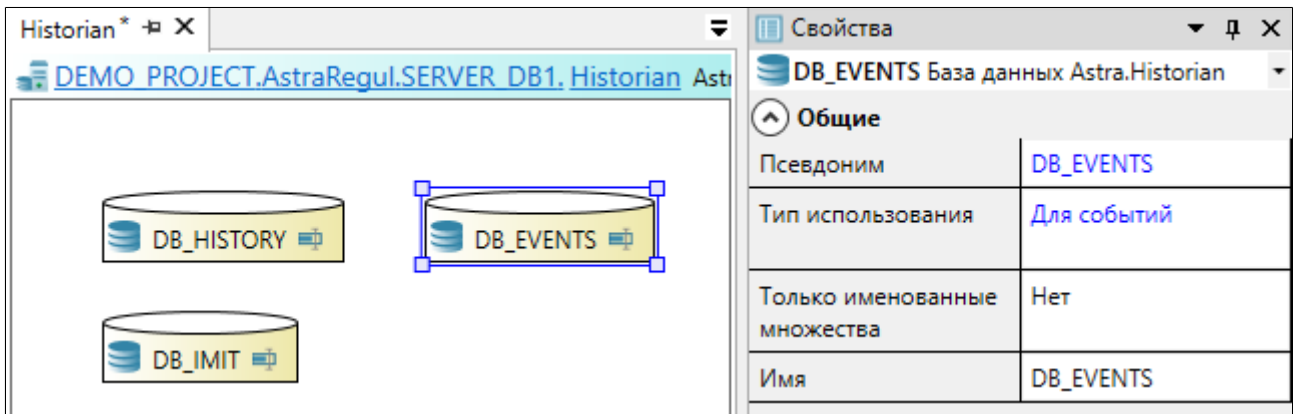
3. Перетащите в рабочую область узла домена из Панели элементов компонент "Astra.Historian".



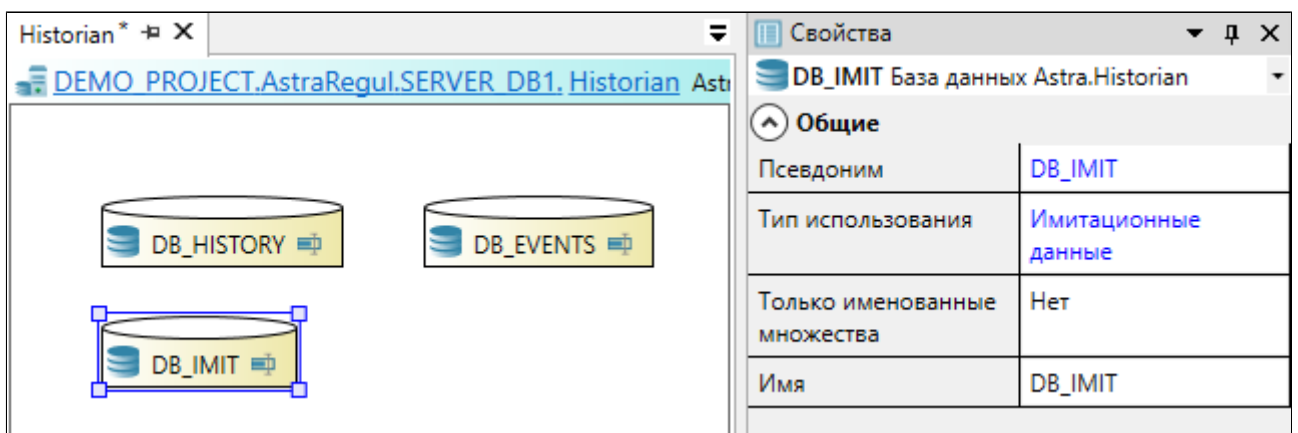
4. Перейдите в компонент "Historian" и перетащите и из панели элементов два дополнительных элемента База данных Astra.Historian. Для первой базы данных задайте имя "DB\_HISTORY" и назначьте тип использования "Для значений".



Для второй базы данных задайте имя "DB\_EVENTS" и назначьте тип использования "Для событий".



Для третьей базы данных задайте имя "DB\_IMIT" и назначьте тип использования "Имитационные данные".





## 1.1.3.2.2.8.1.1. Резервирование

Для резервирования серверов истории используется дублирование: в проект добавляется несколько серверов истории. Резервные серверы истории добавляются так же, как и основной сервер истории.

При дублировании основной и резервный компоненты функционируют одновременно: получают, обрабатывают и передают данные. Клиенты, которые к ним подключаются сами выбирают, к какому компоненту подключиться и от которого получать данные.



Резервные серверы истории добавляются так же, как основной сервер истории.

Данные будут сохраняться во все указанные модулям базы данных. Для запроса сохранённых данных будет использоваться та БД, с которой быстрее всего установлено соединение.

## 1.1.3.2.2.8.2. Microsoft SQL Server

Чтобы добавить сервер истории:

1. Если сервер истории расположен на компьютере, который уже описан в домене (элементы **Узел Astra.Domain** и **Компьютер**), перейдите в него.

Если нет, добавьте в домен компьютер. Тип элемента зависит от типа сервера истории:

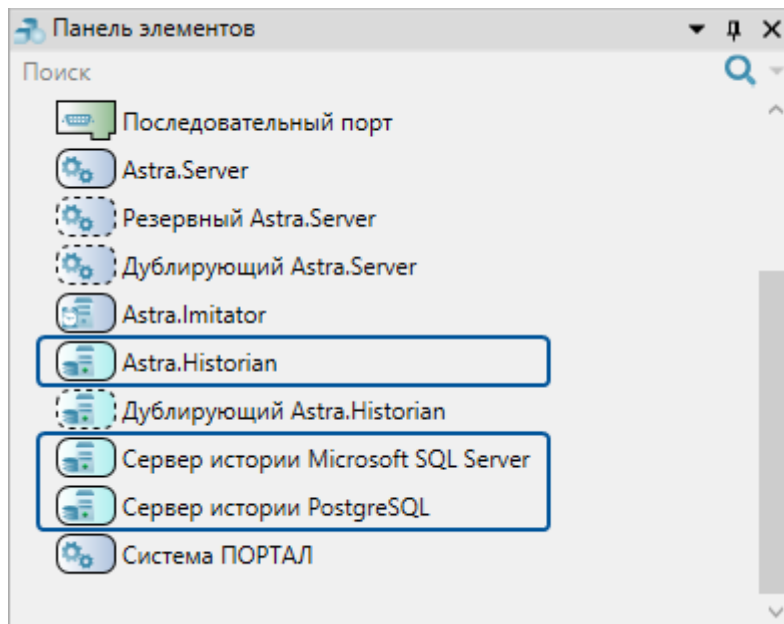
- **Компьютер** - для Microsoft SQL Server или PostgreSQL.
- **Узел Astra.Domain** - для Astra.Historian.

**ПРИМЕЧАНИЕ**  
В **Узел Astra.Domain** можно добавлять серверы истории всех типов. Это нужно, например, если сервер истории расположен на компьютере, где есть **Astra.Server**.

Имя элемента - сетевое имя компьютера.

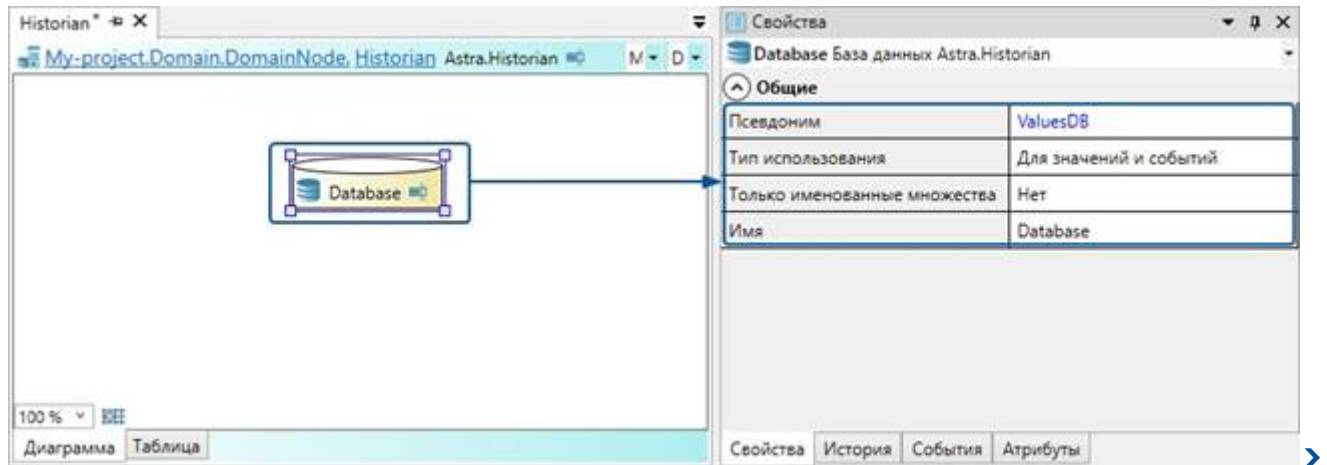
2. Добавьте сервер истории нужного типа:

- **Astra.Historian**
- **Сервер истории Microsoft SQL Server**



**Сервер истории PostgreSQL**

3. Перейдите в добавленный элемент.
4. Добавьте нужное количество баз данных (**База данных Astra.Historian**, **База данных Microsoft SQL Server** или **База данных PostgreSQL** в зависимости от типа сервера истории).
5. Для каждой базы данных в свойствах укажите:
  - **Тип использования** - какие данные хранятся в базе данных: значения и/или события



(только для Astra.Historian) Псевдоним - псевдоним базы данных, должен совпадать с псевдонимом, настроенным в Astra.Historian

## 1.1.3.2.2.8.3. PostgreSQL

Чтобы добавить сервер истории:

1. Если сервер истории расположен на компьютере, который уже описан в домене (элементы **Узел Astra.Domain** и **Компьютер**), перейдите в него.

Если нет, добавьте в домен компьютер. Тип элемента зависит от типа сервера истории:

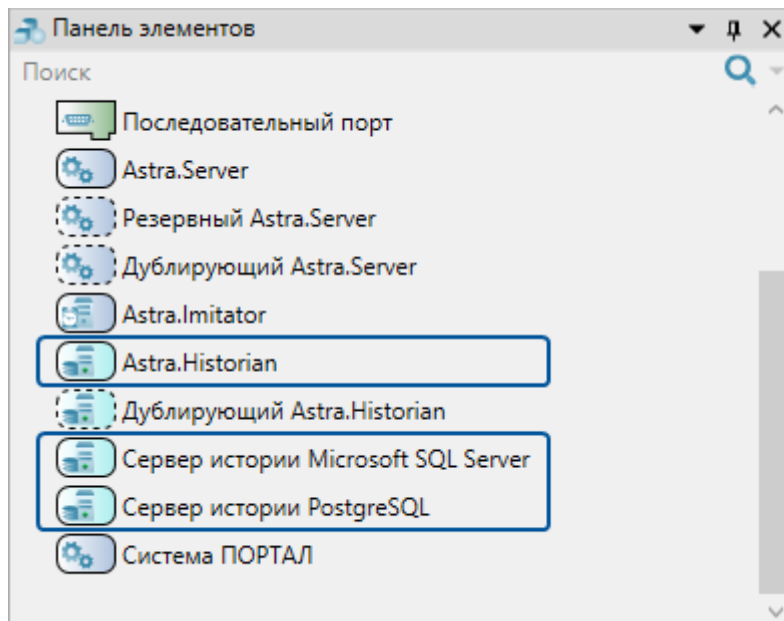
- **Компьютер** - для Microsoft SQL Server или PostgreSQL.
- **Узел Astra.Domain** - для Astra.Historian.

**ПРИМЕЧАНИЕ**  
В **Узел Astra.Domain** можно добавлять серверы истории всех типов. Это нужно, например, если сервер истории расположен на компьютере, где есть **Astra.Server**.

Имя элемента - сетевое имя компьютера.

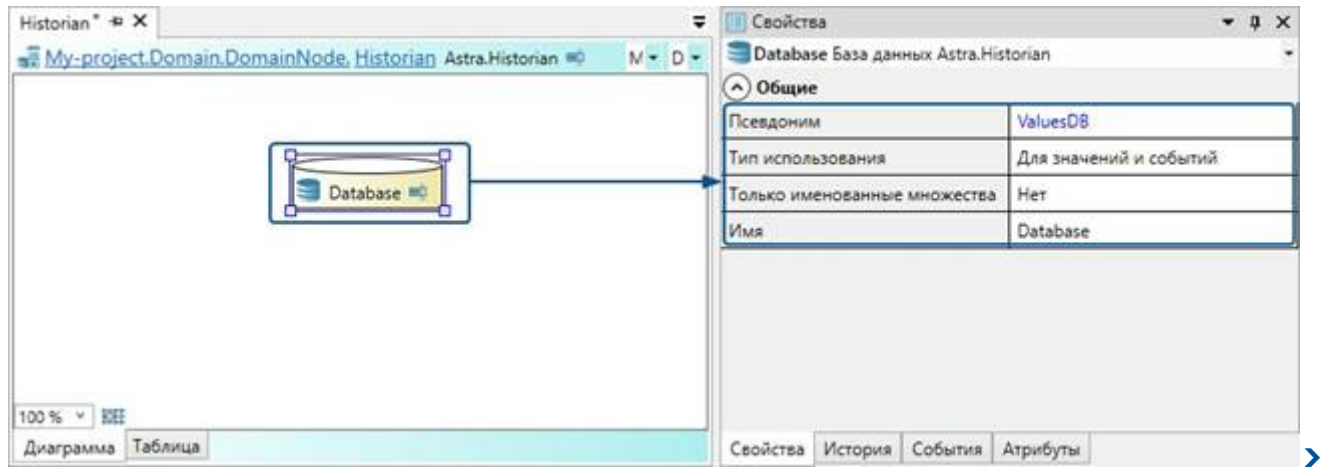
2. Добавьте сервер истории нужного типа:

- **Astra.Historian**
- **Сервер истории Microsoft SQL Server**



**Сервер истории PostgreSQL**


3. Перейдите в добавленный элемент.
4. Добавьте нужное количество баз данных (**База данных Astra.Historian**, **База данных Microsoft SQL Server** или **База данных PostgreSQL** в зависимости от типа сервера истории).
5. Для каждой базы данных в свойствах укажите:
  - **Тип использования** - какие данные хранятся в базе данных: значения и/или события

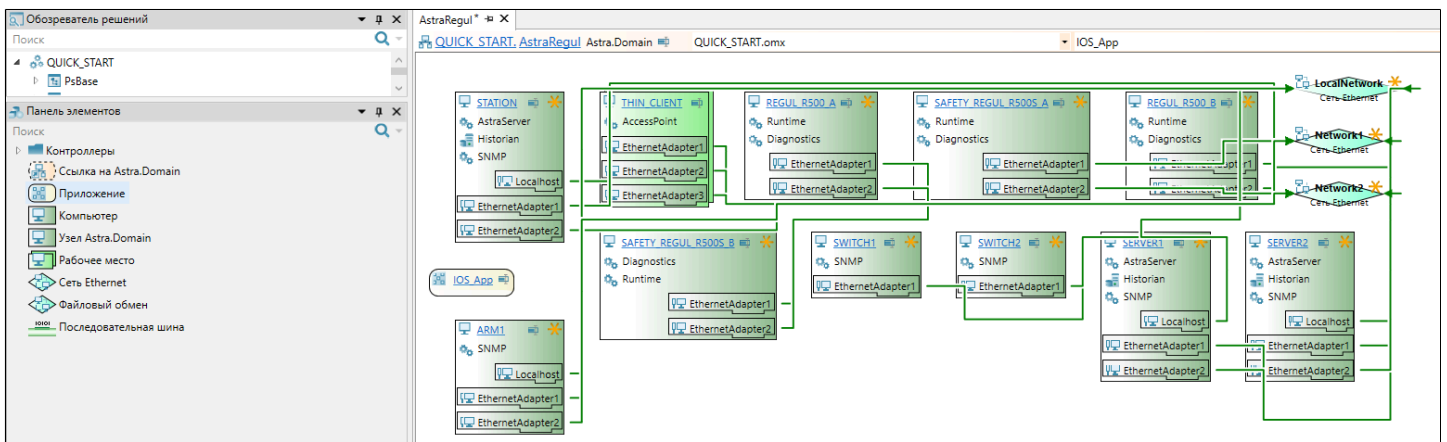


(только для Astra.Historian) Псевдоним - псевдоним базы данных, должен совпадать с псевдонимом, настроенным в Astra.Historian

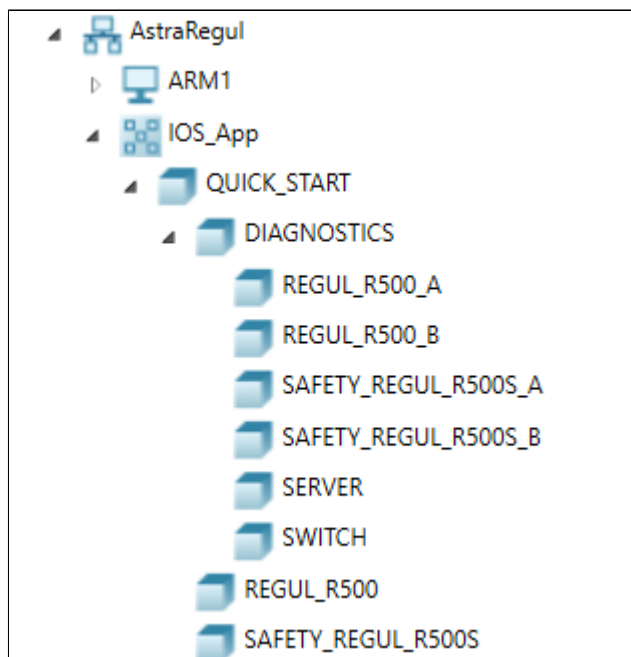
# 1.1.3.2.2.9. Приложение сервера ввода/вывода

1. Перетащите в рабочую область домена из "Панели элементов" компонент "Приложение".

 Рекомендуется задать Имя приложения IOS\_App.



2. В приложении "IOS\_App" создайте логический объект "QUICK\_START" и следующую структуру дерева объектов.



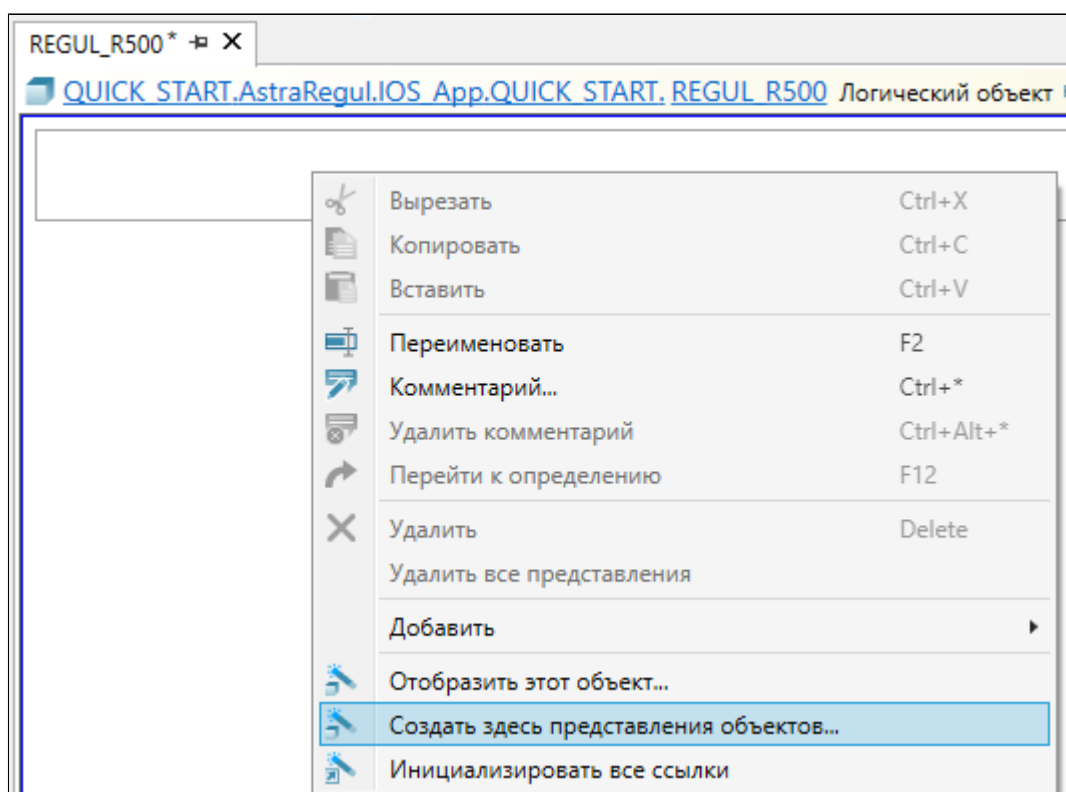
Для формирования структуры объектов необходимо использовать элемент "Логический объект" из панели элементов.

"Логический объект" представляет собой контейнер для других объектов (наподобие папки для файлов в проводнике).

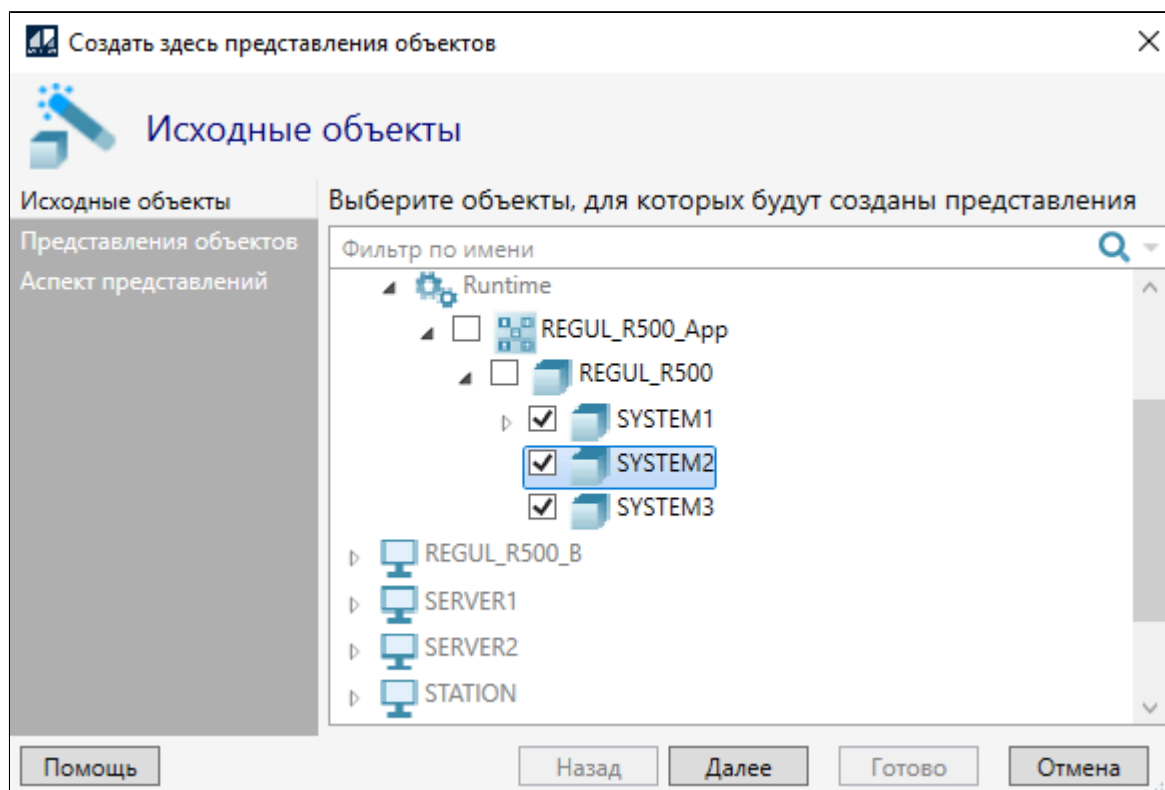


Структура объектов в виде дерева позволит удобно фильтровать и искать данные в проекте.

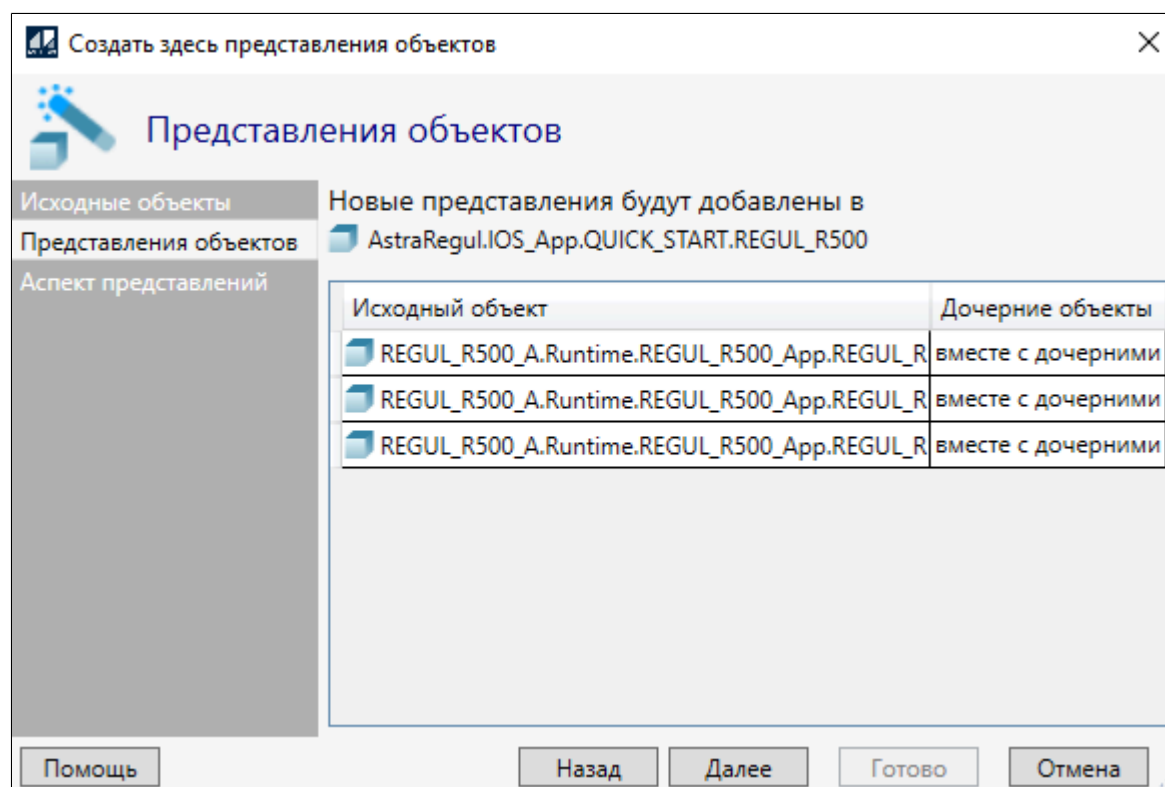
3. Перейдите в логический объект "REGUL\_R500". Кликните правой кнопкой мыши по свободной рабочей области и в контекстном меню выберите команду "Создать здесь представления объектов".



4. Отметьте все логические объекты приложения ПЛК REGUL и нажмите кнопку "Далее".

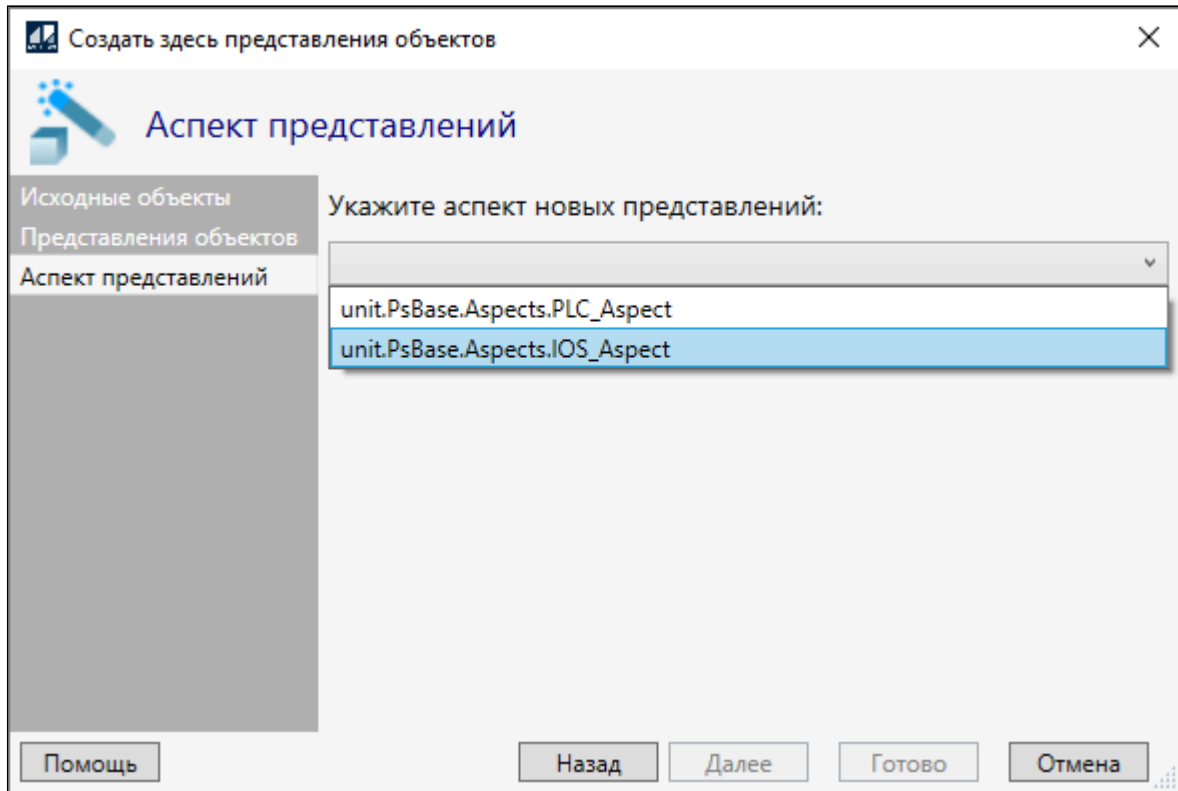


5. В столбце "Дочерние объекты" выберите значение "вместе с дочерними" и нажмите кнопку "Далее".

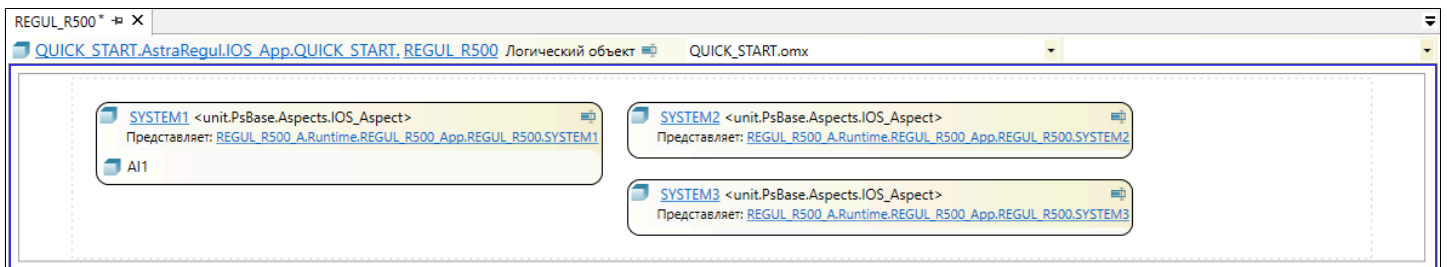




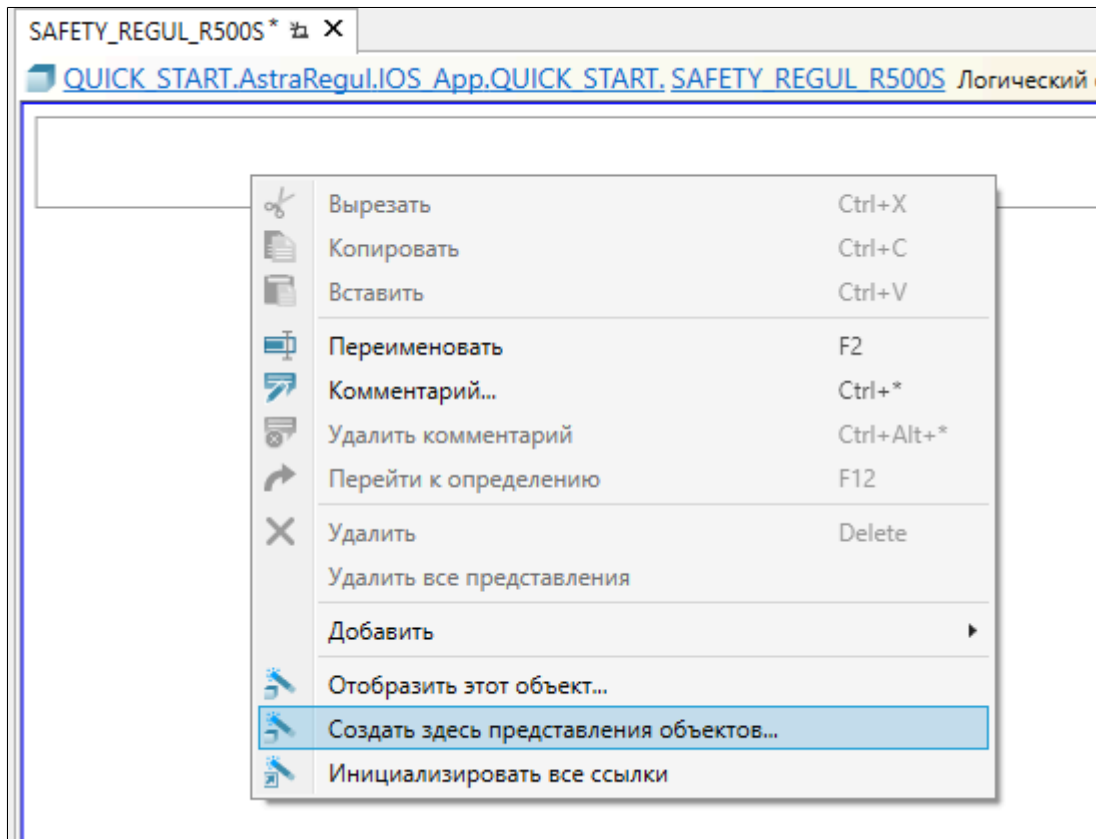
6. Укажите аспект "unit.PsBase.Aspects.IOS\_Aspect" и нажмите кнопку "Готово".



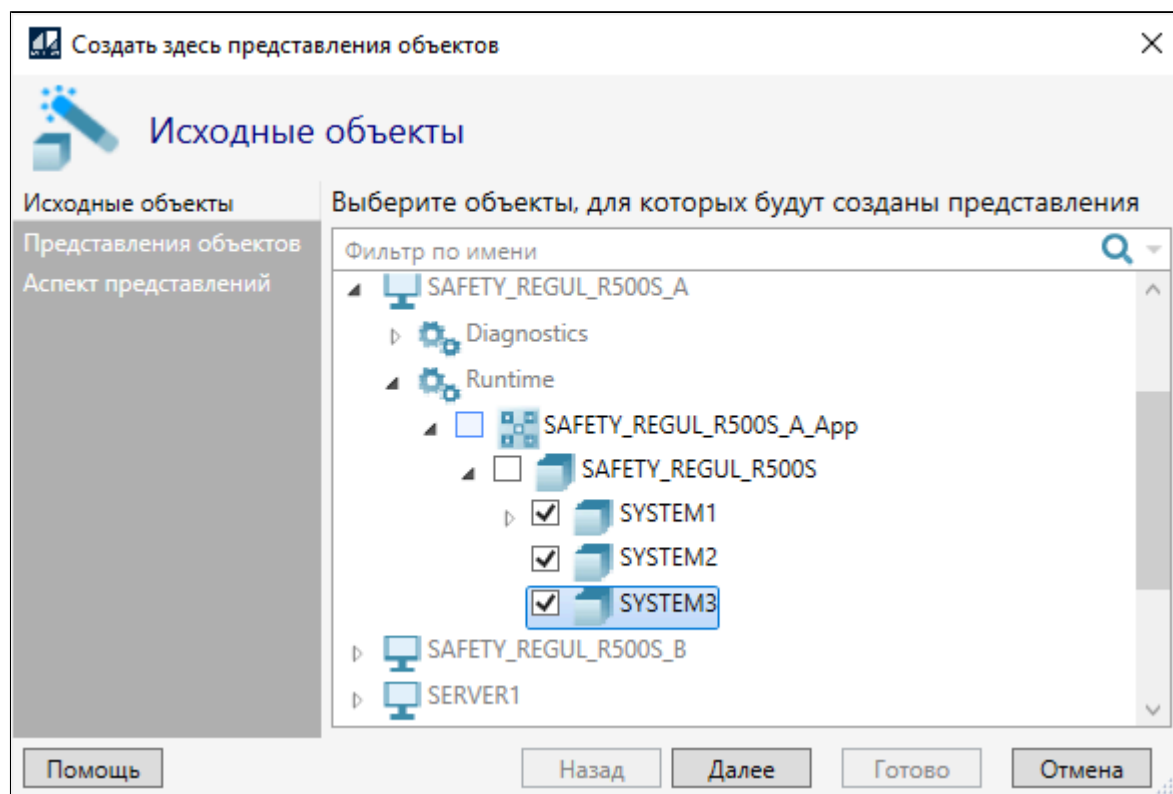
Добавление представления объектов для приложения ПЛК завершено.



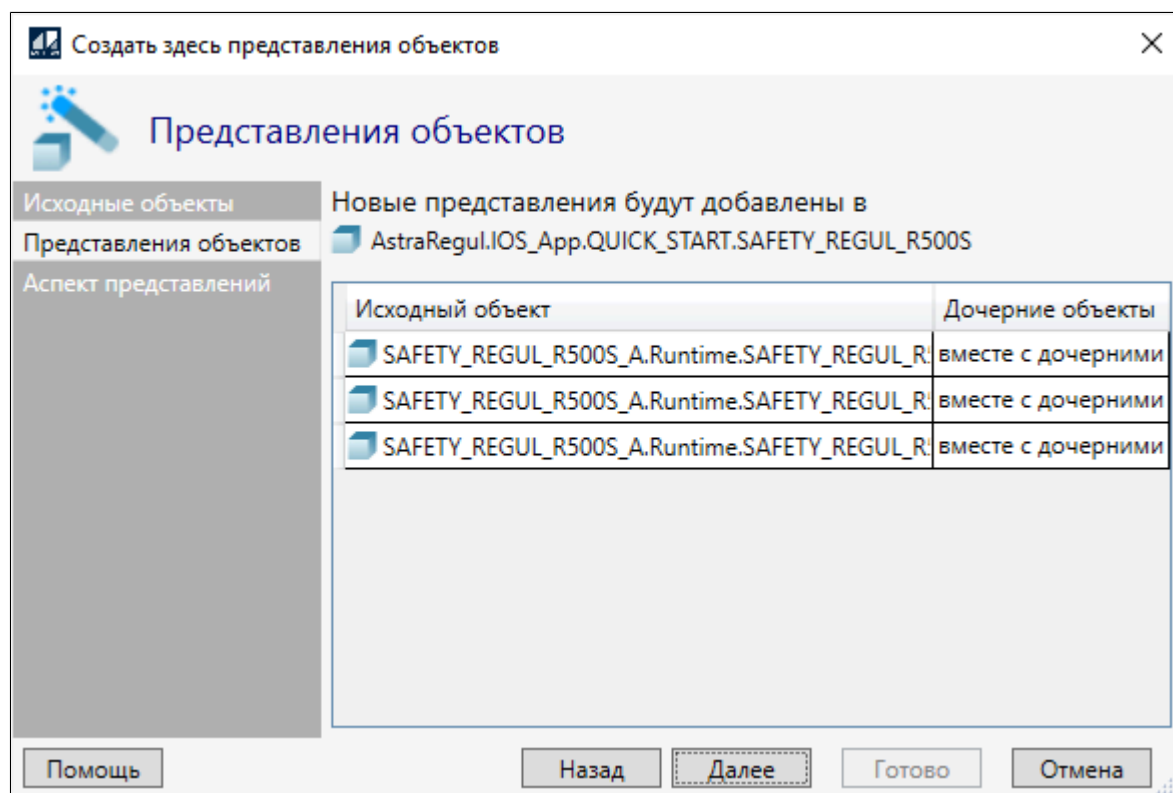
7. Перейдите в логический объект "SAFETY\_REGUL\_R500S". Кликните правой кнопкой мыши по свободной рабочей области и в контекстном меню выберите команду "Создать здесь представления объектов".



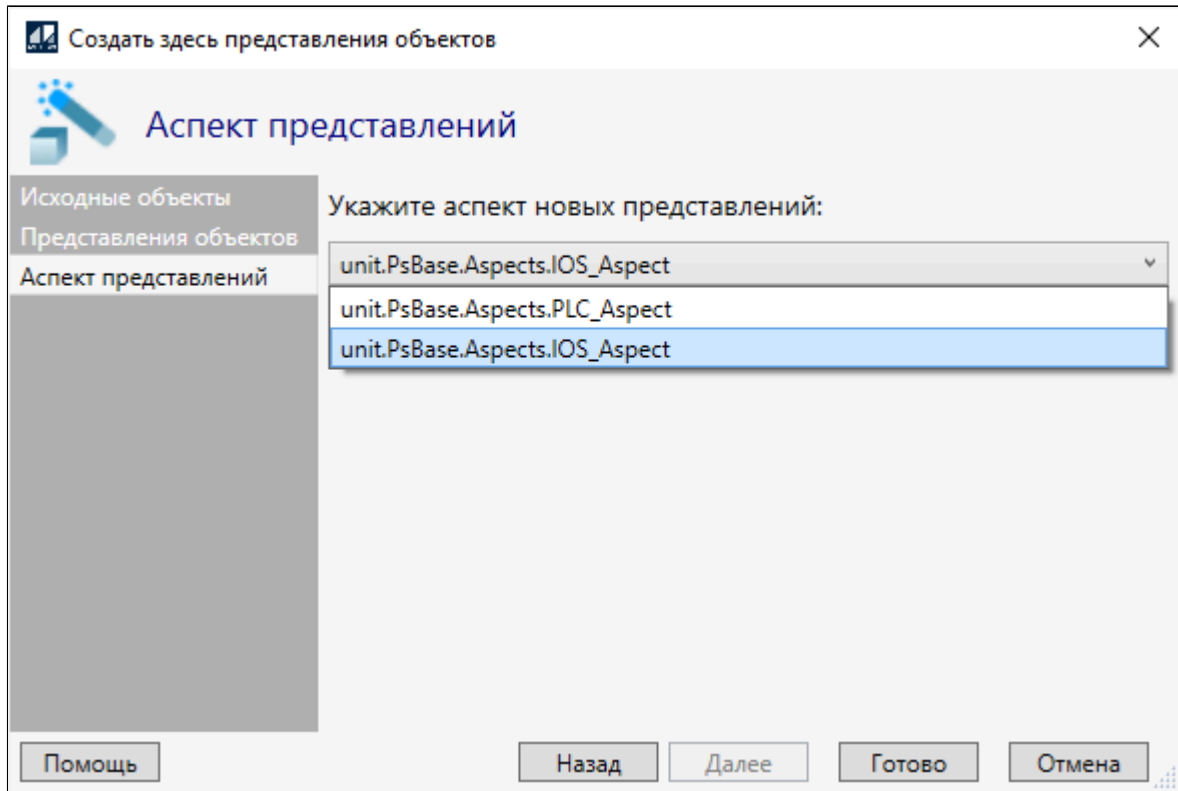
8. Отметьте все логические объекты приложения ПЛК SAFETY REGUL и нажмите кнопку "Далее".



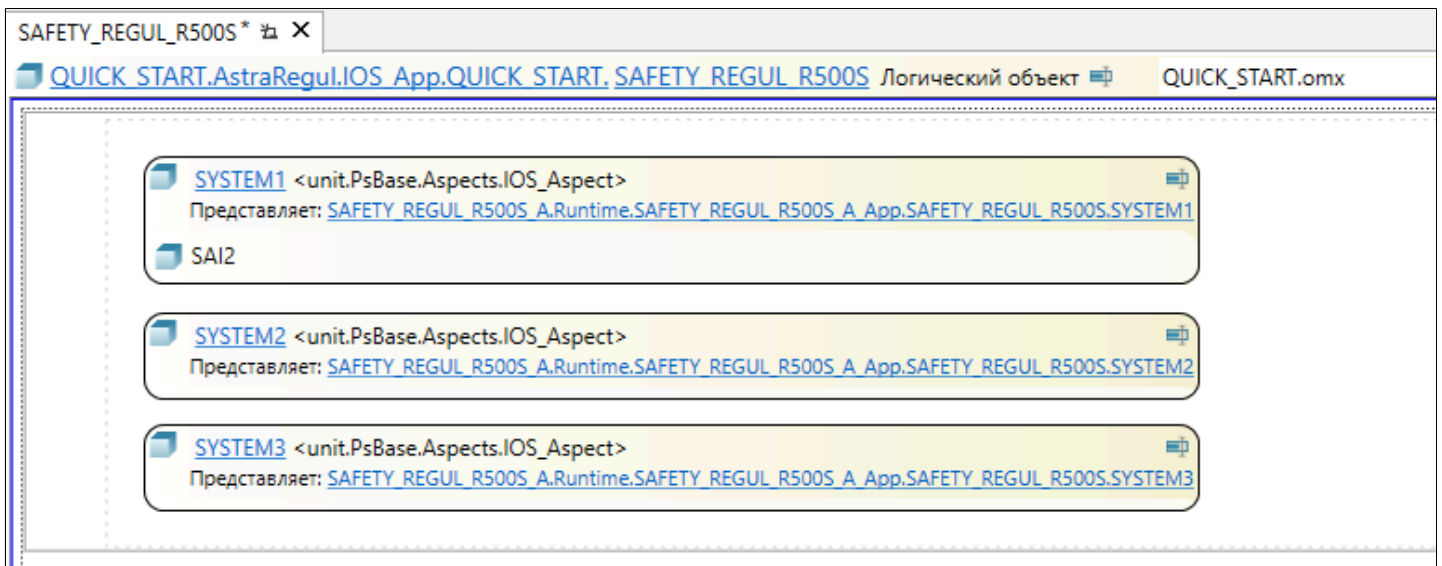
9. В столбце "Дочерние объекты" выберите значение "вместе с дочерними" и нажмите кнопку "Далее".



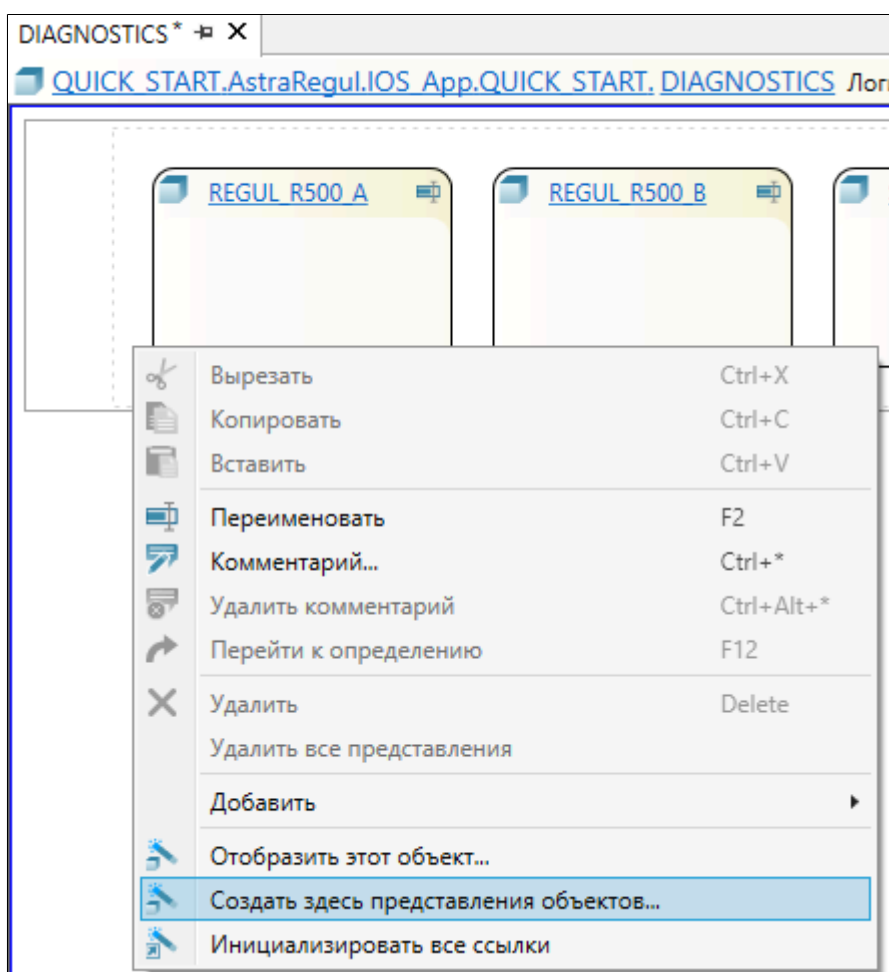
10. Укажите аспект "unit.PsBase.Aspects.IOS\_Аспект" и нажмите кнопку "Готово".



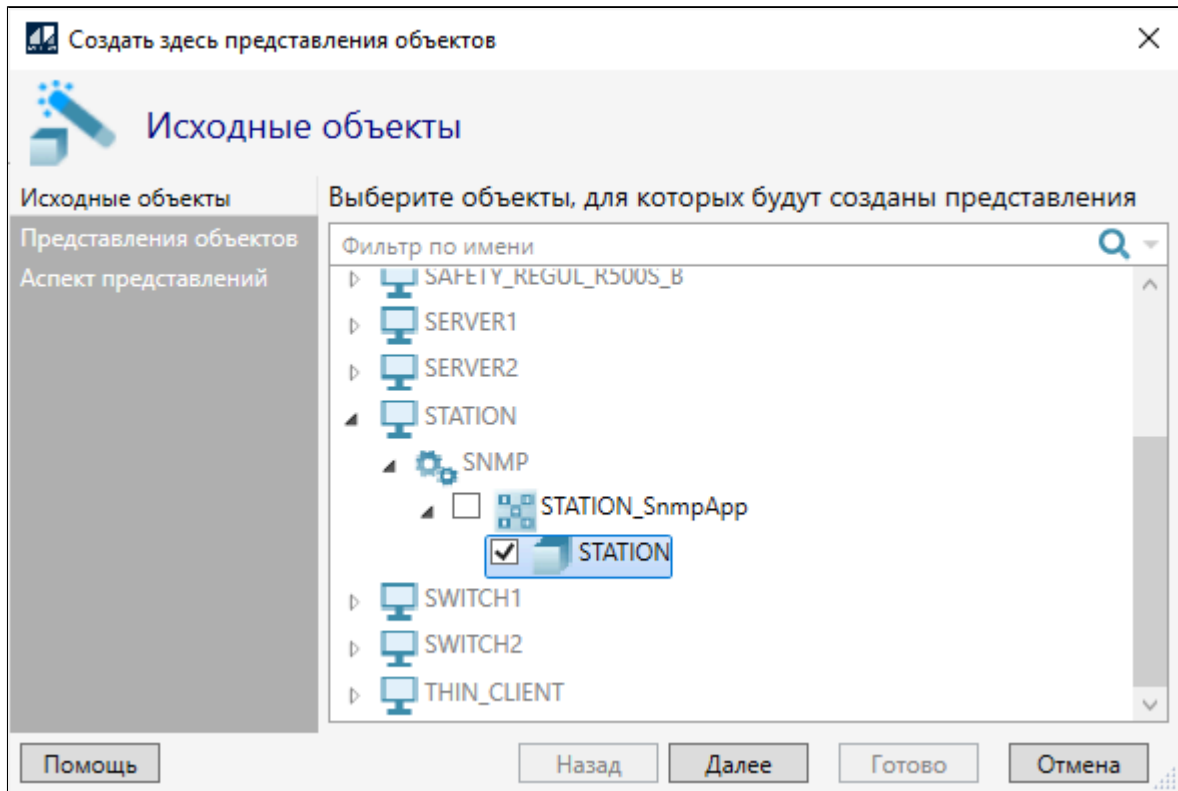
Добавление представления объектов для приложения ПЛК завершено.



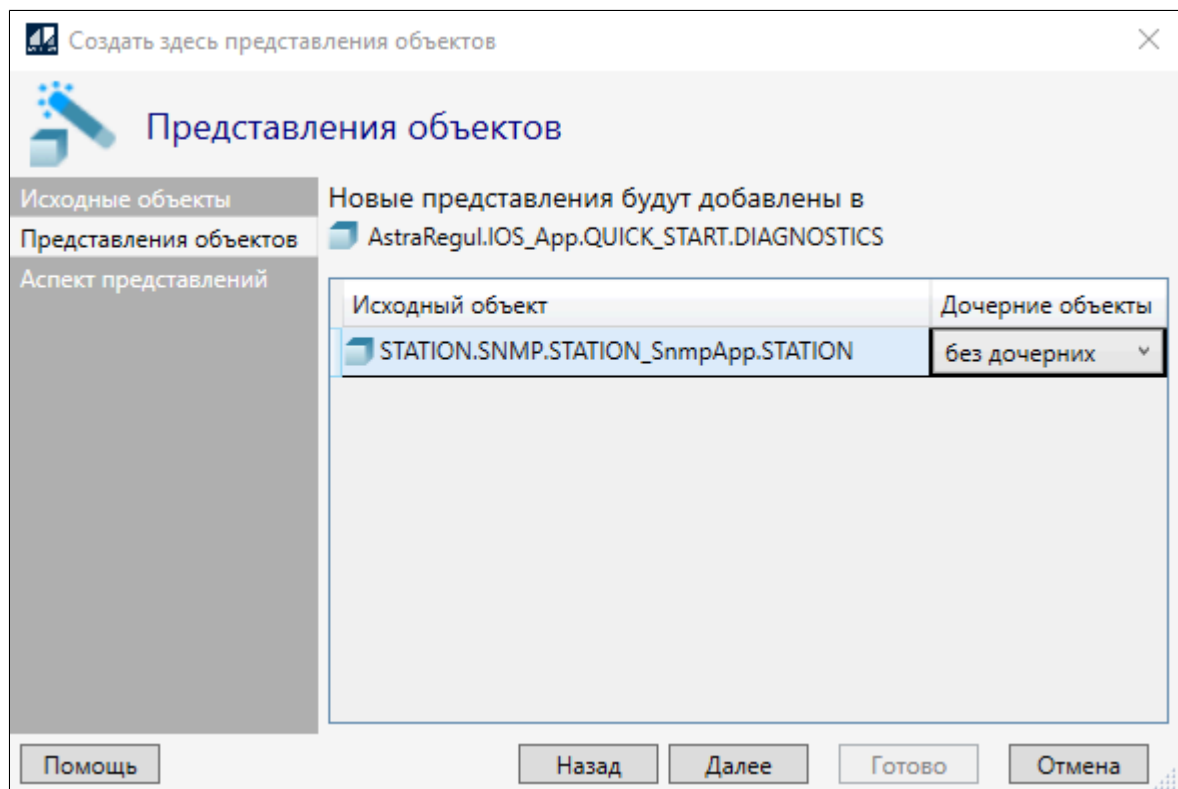
11. Перейдите в логический объект "DIAGNOSTICS". Кликните правой кнопкой мыши по свободной рабочей области и в контекстном меню выберите команду "Создать здесь представления объектов".



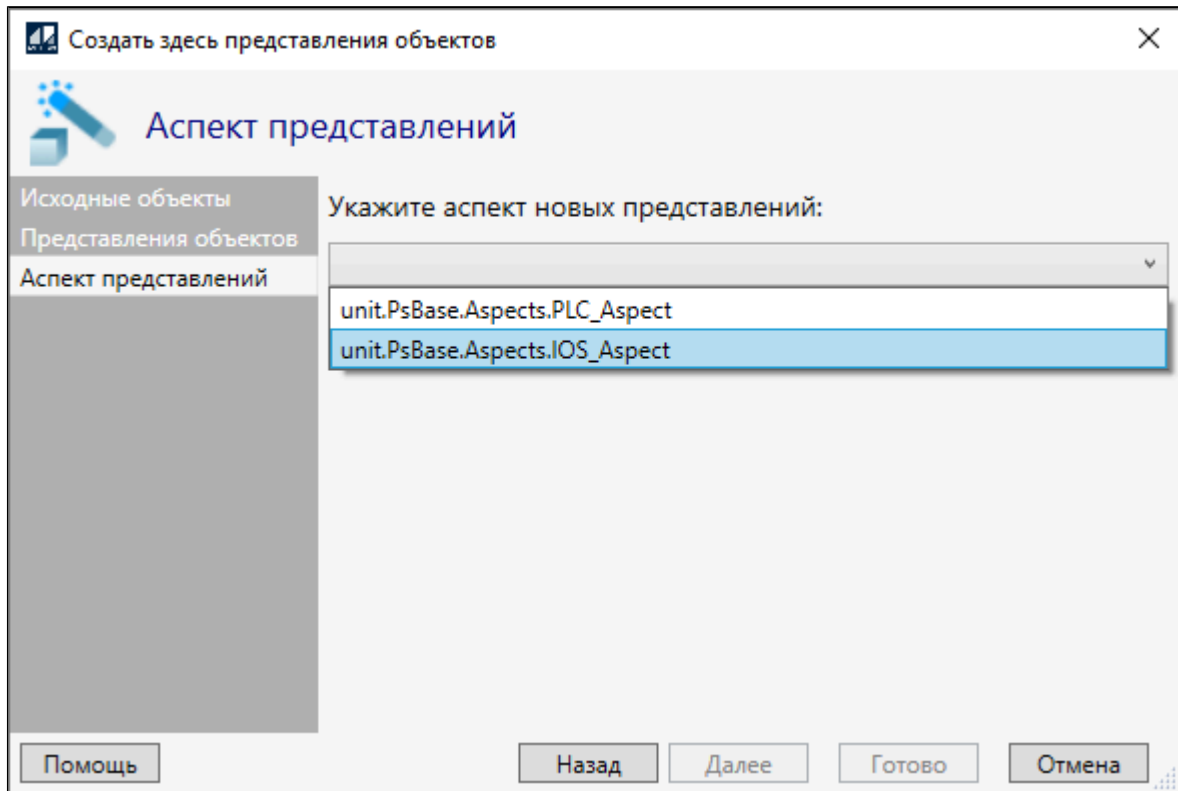
12. Отметьте логический объект диагностики "STATION" и нажмите кнопку "Далее".



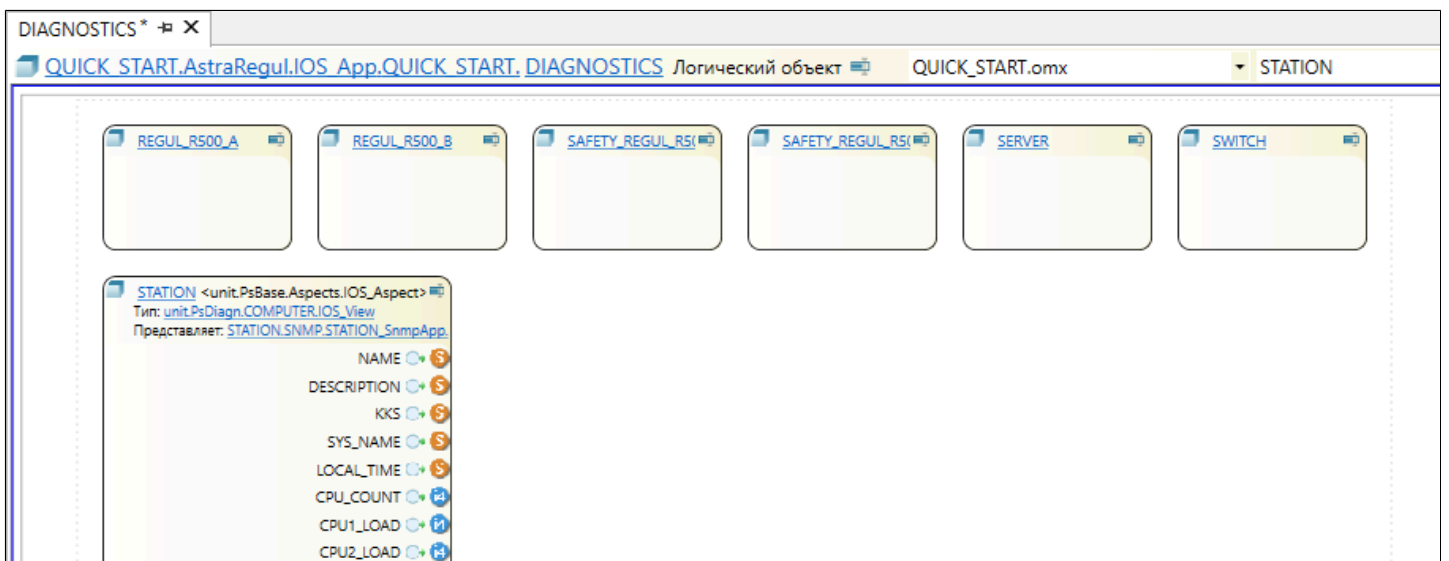
13. В столбце "Дочерние объекты" выберите значение "без дочерних" и нажмите кнопку "Далее".



14. Укажите аспект представлений "unit.PsBase.Aspects.IOS\_Aspect" и нажмите кнопку "Готово".



Добавление представления объектов для диагностики инженерной станции завершено.



15. Аналогичным образом необходимо добавить представление объекта "ARM1" в папку "DIAGNOSTICS".

REGUL\_R500\_A

REGUL\_R500\_B

SAFETY\_REGUL\_RS

SAFETY\_REGUL\_RS

SERVER

SWITCH

**STATION** <unit.PsBase.Aspects.IOS\_Aspect>  
Тип: unit.PsDiagn.COMPUTER.IOS\_View  
Представляет: STATION.SNMP.STATION\_SnmpApp

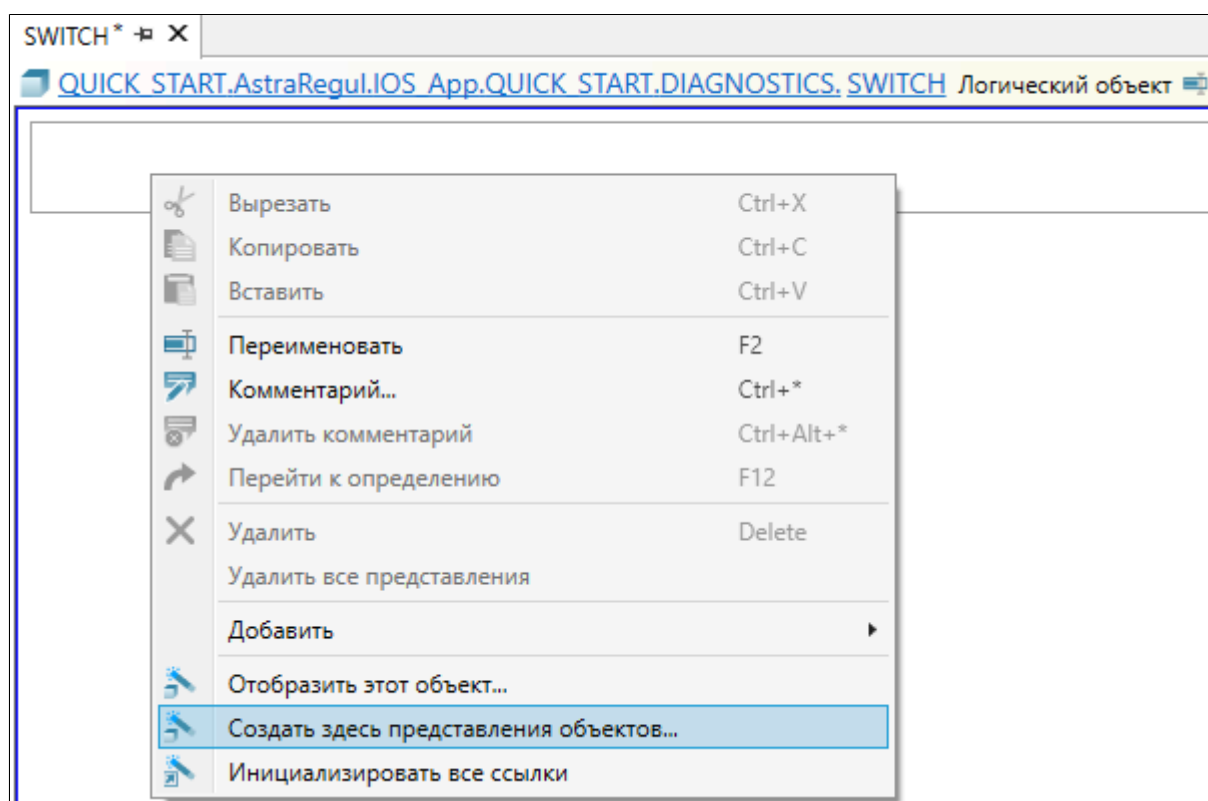
- NAME
- DESCRIPTION
- KKS
- SYS\_NAME
- LOCAL\_TIME
- CPU\_COUNT
- CPU1\_LOAD

**ARM1** <unit.PsBase.Aspects.IOS\_Aspect>  
Тип: unit.PsDiagn.COMPUTER.IOS\_View  
Представляет: ARM1.SNMP.ARM1\_SnmpApp.ARM1

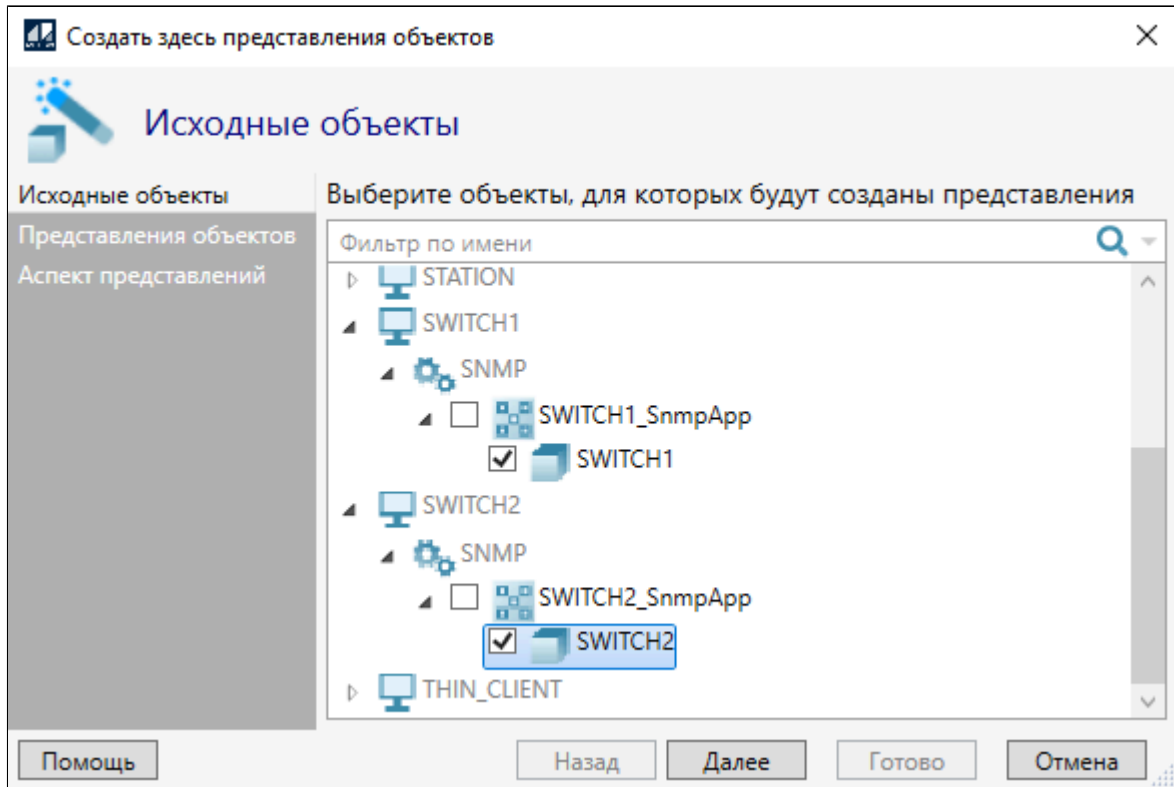
- NAME
- DESCRIPTION
- KKS
- SYS\_NAME
- LOCAL\_TIME
- CPU\_COUNT
- CPU1\_LOAD



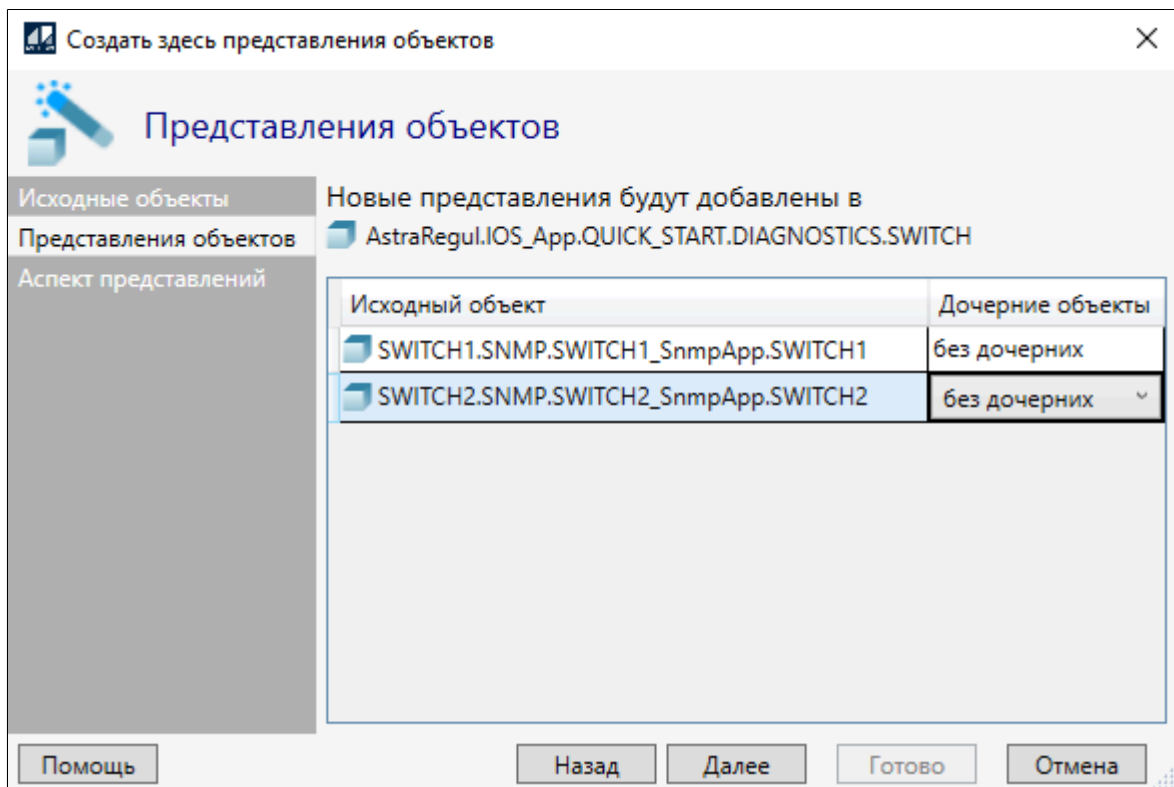
16. Перейдите в логический объект "SWITCH". Кликните правой кнопкой мыши по свободной рабочей области и в контекстном меню выберите команду "Создать здесь представления объектов".



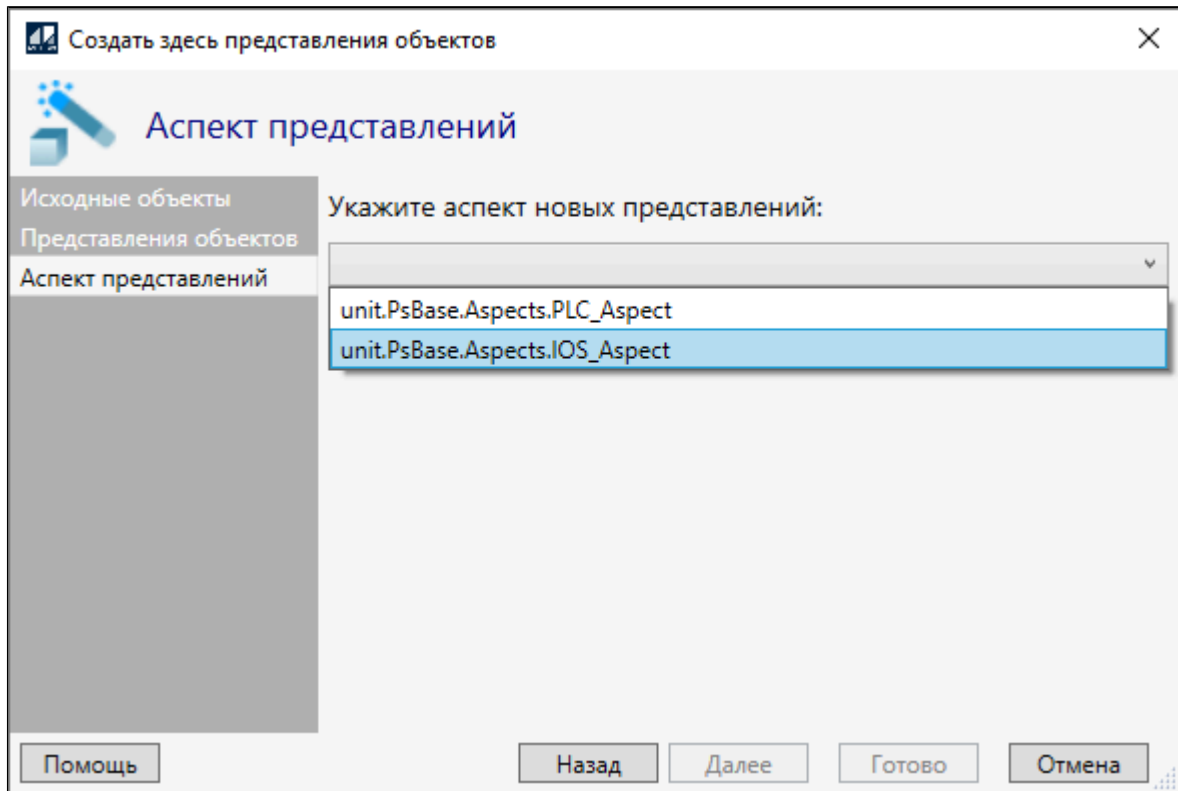
17. Отметьте логический объект диагностики "SWITCH1" и "SWITCH2".



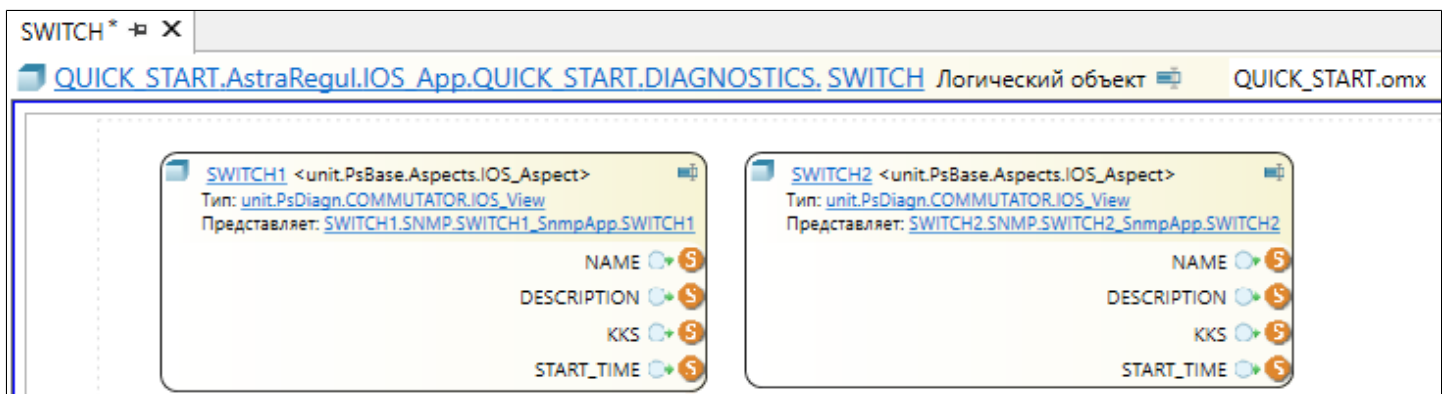
18. В столбце "Дочерние объекты" выберите значение "без дочерних".



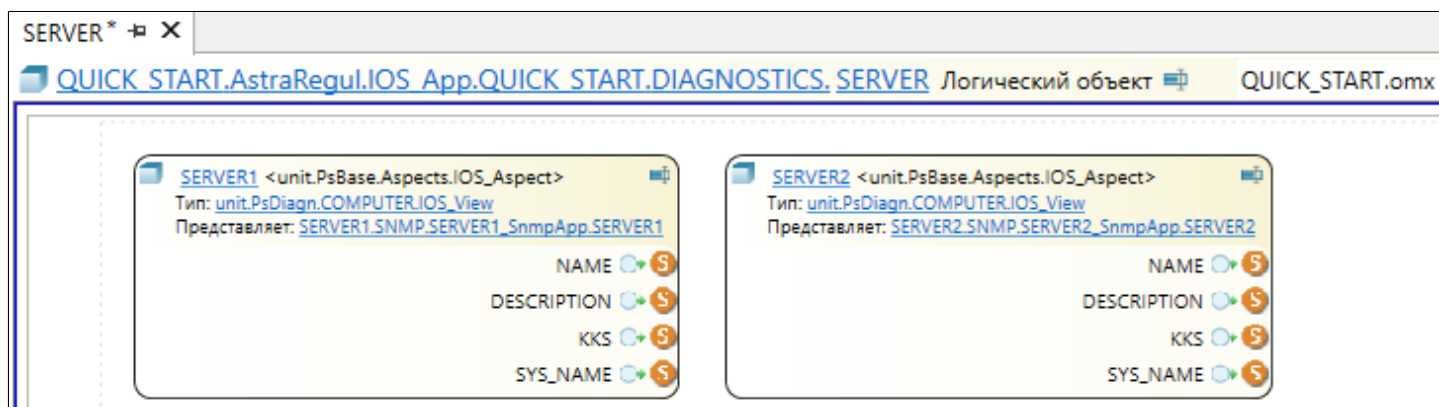
19. Укажите аспект представлений "unit.PsBase.Aspects.IOS\_Aspect" и нажмите кнопку "Готово".



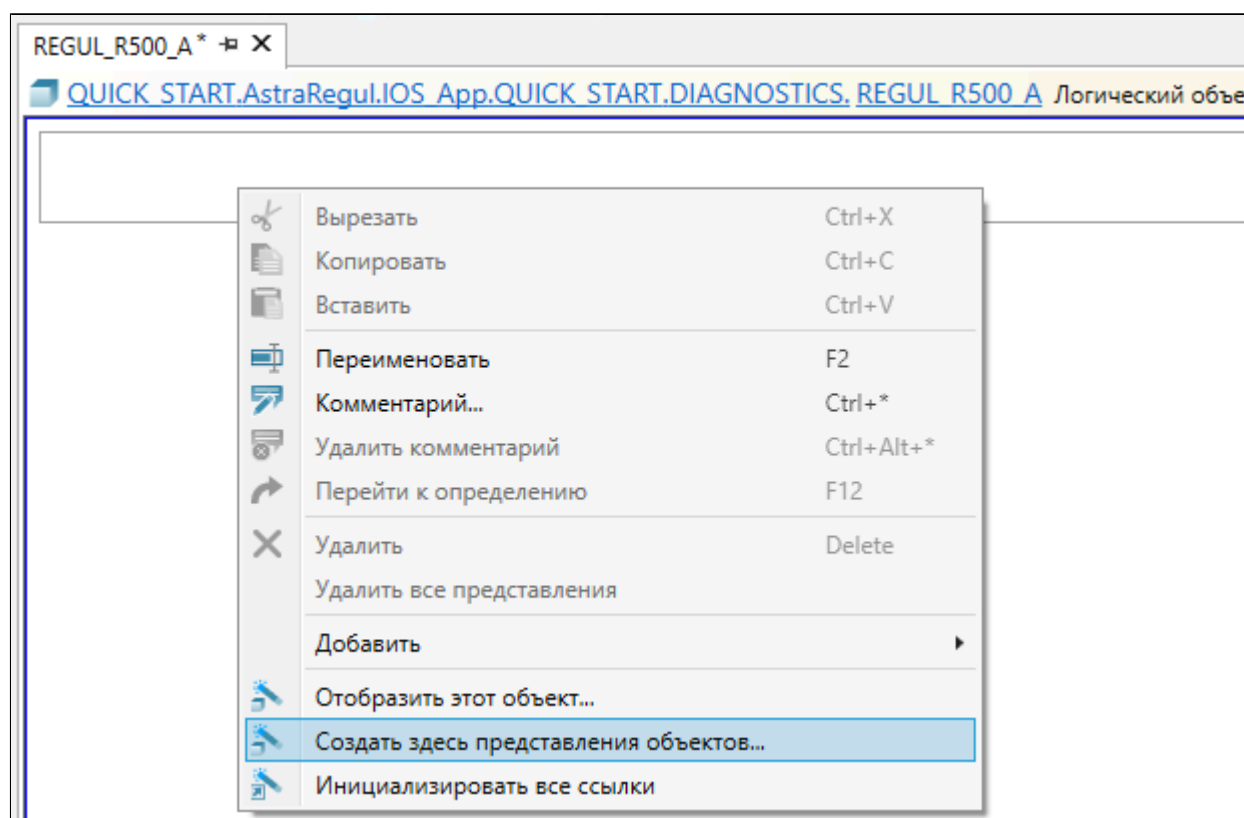
Добавление представления объектов для диагностики коммутаторов завершено.



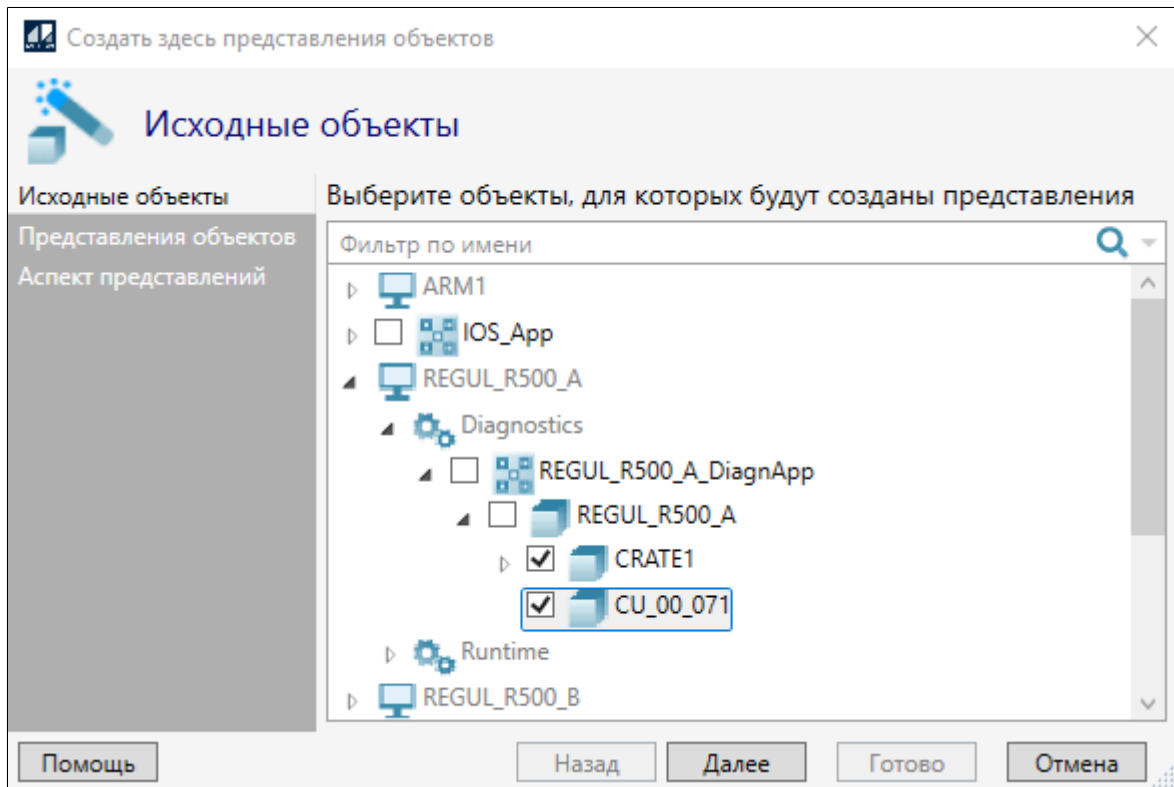
20. Аналогичным образом необходимо добавить представление объектов "SERVER1" и "SERVER2" в логический элемент "SERVER".



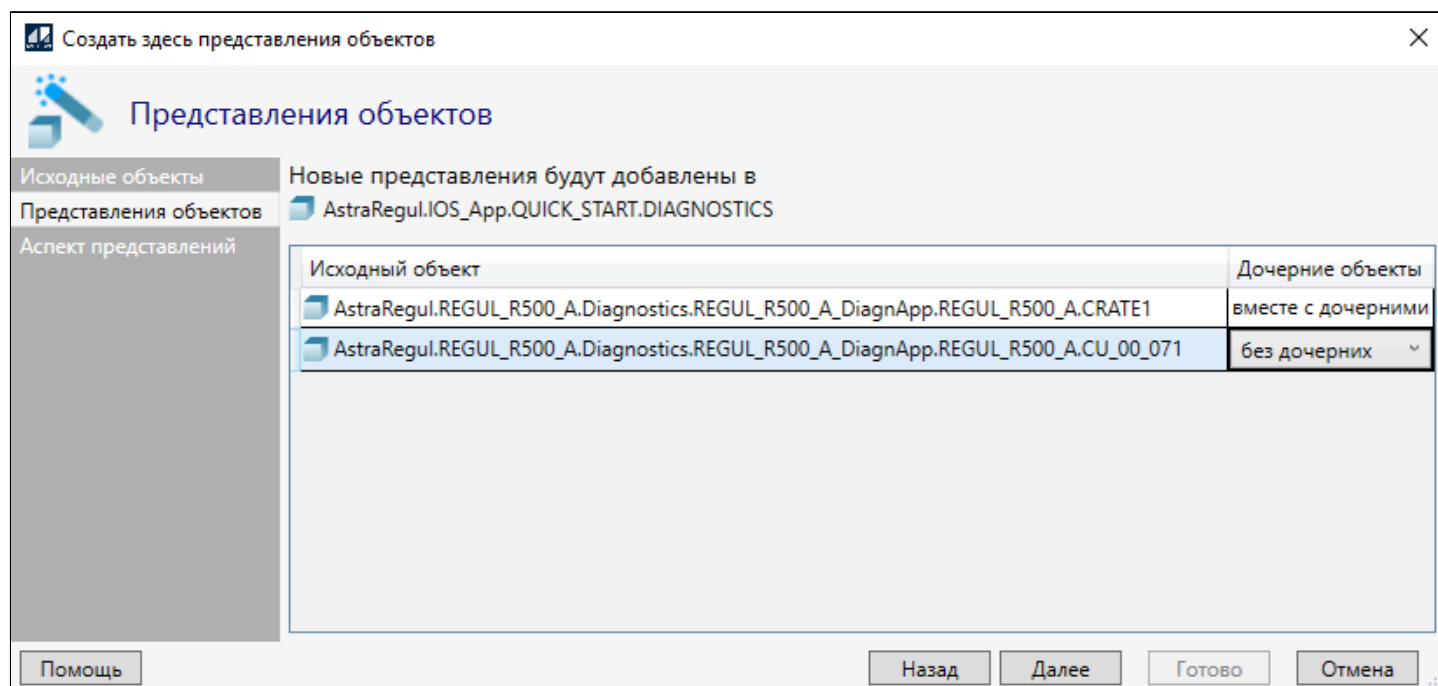
21. Перейдите в логический объект "REGUL\_R500\_A". Кликните правой кнопкой мыши по свободной рабочей области и в контекстном меню выберите команду "Создать здесь представления объектов".



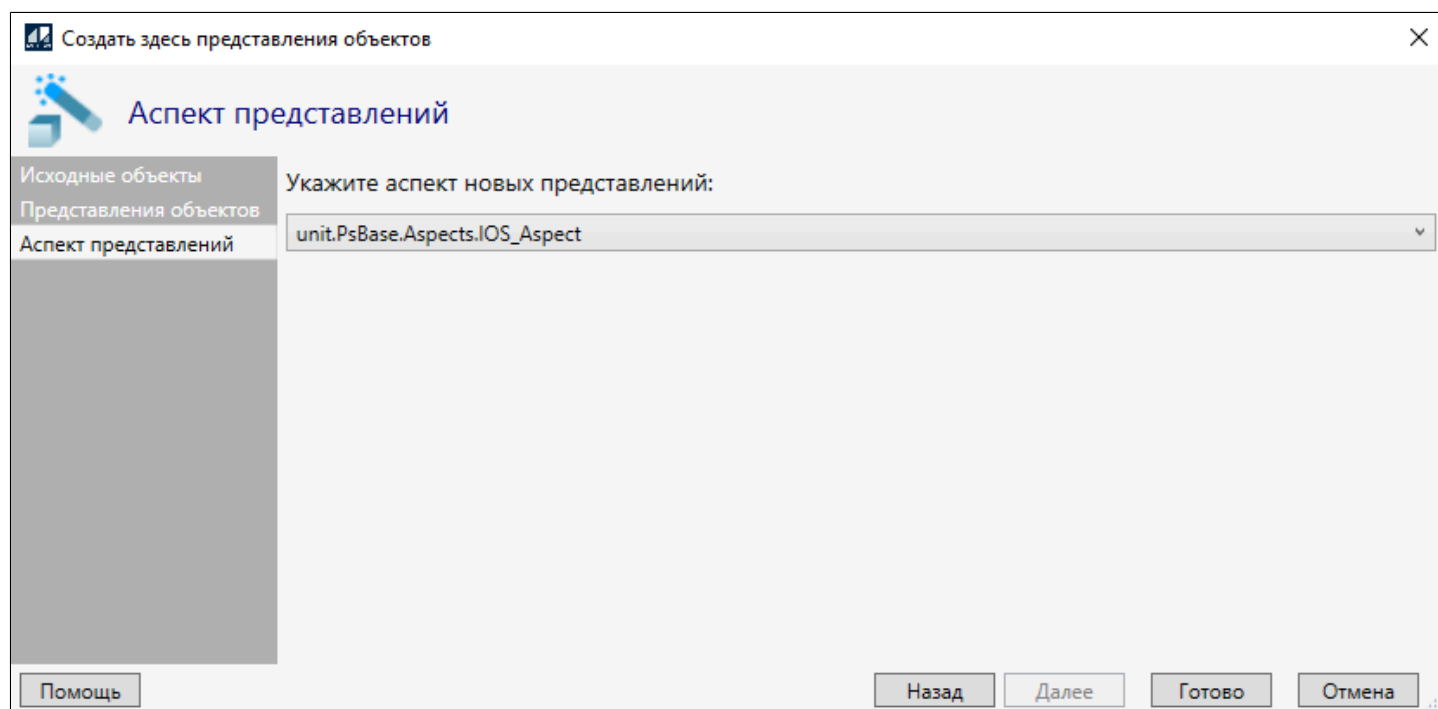
22. Отметьте необходимые модули ПЛК А.



23. В столбце "Дочерние объекты" выберите для ЦПУ значение "без дочерних", а для крейта - "вместе с дочерними".



24. Укажите аспект представлений "unit.PsBase.Aspects.IOS\_Aспект" и нажмите кнопку "Готово".



Добавление представления объектов для диагностики ПЛК А завершено.

**CRATE1** <unit.PsBase.Aspects.IOS\_Aspect>  
Представляет: AstraRegul.REGUL\_R500\_A.Diagnostics.REGUL\_R500\_A\_DiagnApp.REGUL\_R500\_A\_CRATE1

- ST\_02\_012
- ST\_02\_022
- PP\_00\_011
- AI\_08\_041
- AO\_08\_011
- DI\_32\_011
- DO\_32\_011

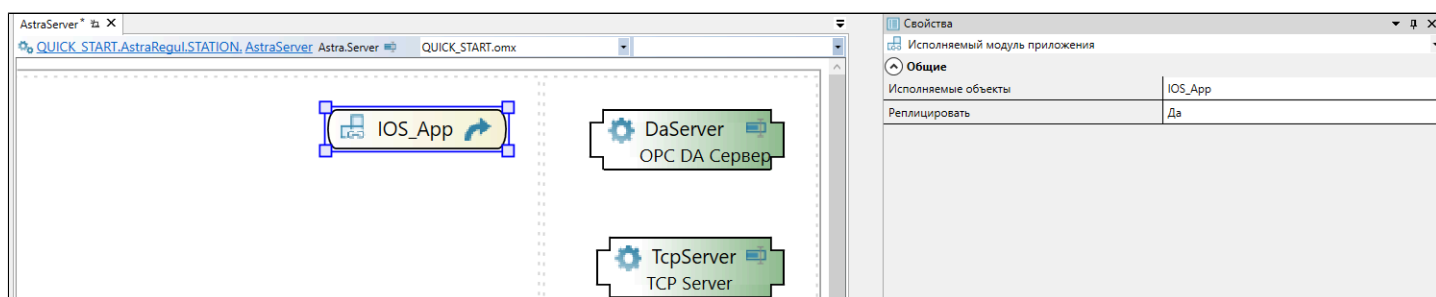
**CU\_00\_071** <unit.PsBase.Aspects.IOS\_Aspect>  
Тип: unit.PsDiagn.REGUL\_R500\_MODULES.CU.CU\_XX\_XXX\_IOS\_View  
Представляет: AstraRegul.REGUL\_R500\_A.Diagnostics.REGUL\_R500\_A\_DiagnApp.REGUL\_R500\_A\_CU\_00\_071

- NAME
- DESCRIPTION
- KKS
- VERSION
- CPU\_CNT
- CPU\_LOAD
- CPU\_LOAD

25. Аналогичным образом необходимо добавить представление объектов для диагностики ПЛК Б и контроллеров серии Safety.

Создание приложения сервера ввода/вывода завершено.

26. Перейдите в компонент "Astra.Server" инженерной станции и перетащите в рабочую область из панели элементов компонент "Исполняемый модуль приложения". Выделите объект и в окне "Свойства" укажите в качестве исполняемого объекта приложение "IOS\_App". Повторите те же действия для всех серверов, где установлен компонент "Astra.Server".





## 1.1.3.2.3. Построение решения

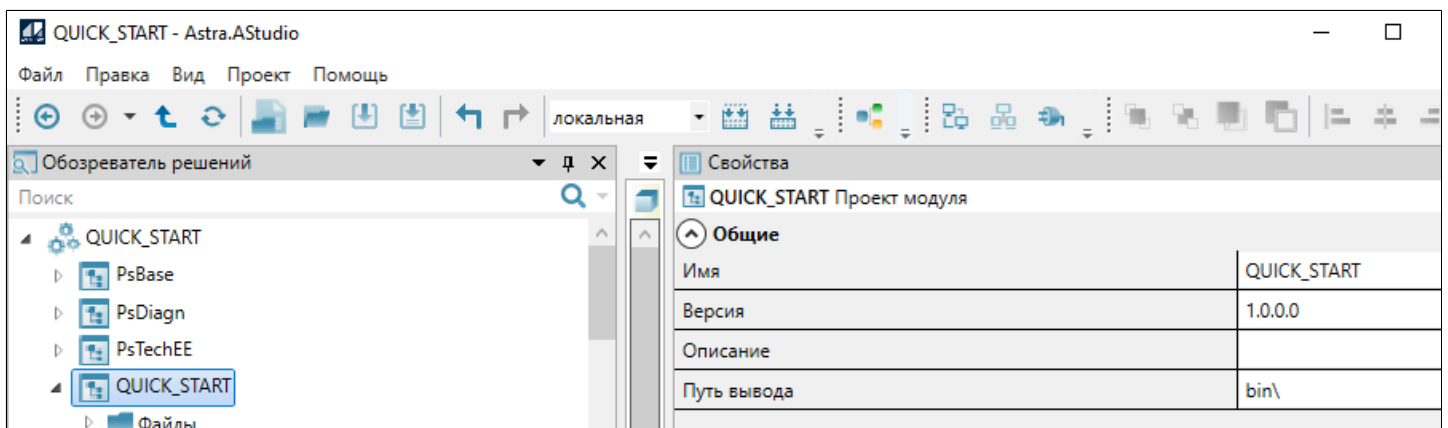
При построении решения будут созданы файлы конфигураций для серверов ввода/вывода Astra.Server и точек доступа Astra.AccessPoint.



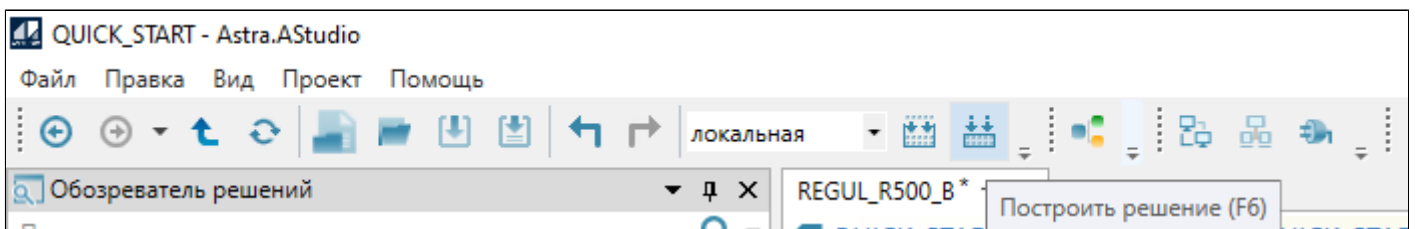
Для прочих программных компонентов, описанных в решении, конфигурации не создаются, однако их описание используется при создании конфигураций.

Чтобы построить решение, выполните следующие действия:

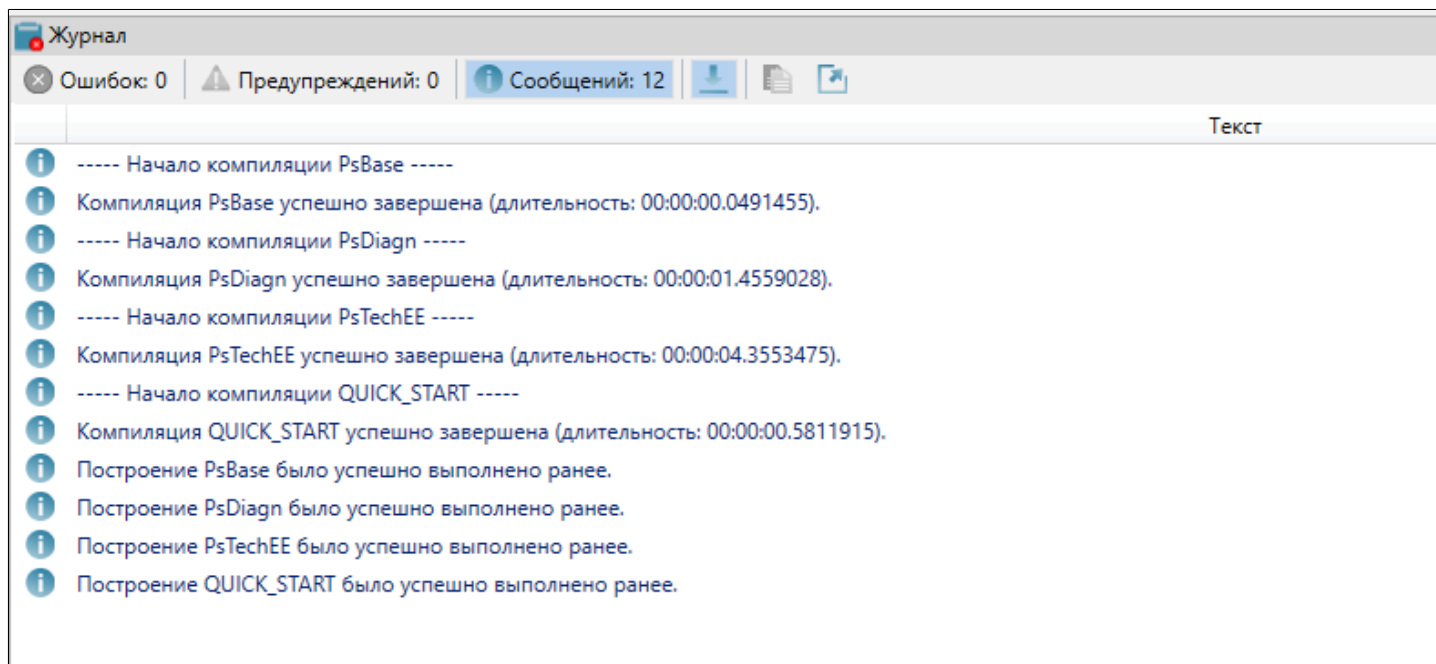
1. В свойствах проекта при необходимости укажите новую версию проекта.



2. В панели инструментов нажмите кнопку "Построить решение (F6)".



Перед созданием файлов конфигураций Astra.AStudio выполняет компиляцию решения. Результат построения проекта можно увидеть в журнале.



В результате компиляции проекта выполняется сборка проекта автоматизации и построение выходных конфигураций сервера Astra.Server.



Если при компиляции будут обнаружены ошибки, то в журнале сообщений появятся сообщения об ошибках и построение решения прервется.

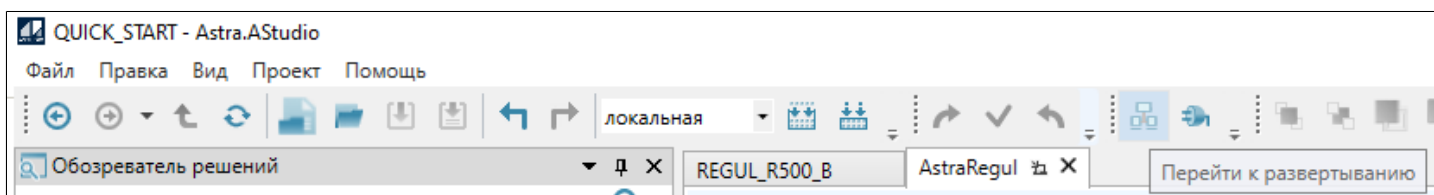
Файлы конфигураций будут созданы в папке, указанной в свойствах проекта.

## 1.1.3.2.4. Развертывание проекта

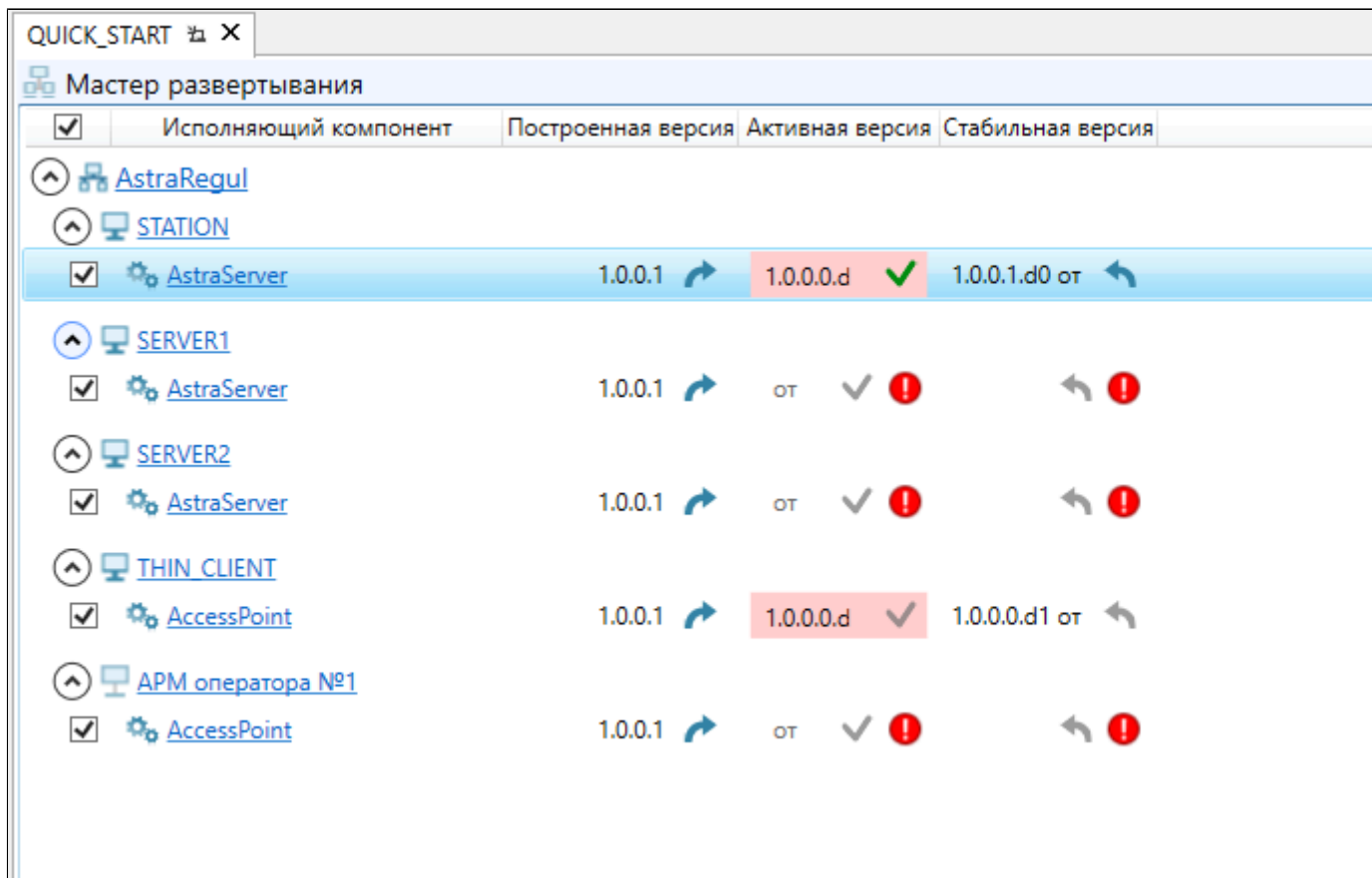
**Развертывание** - процесс применения новых конфигураций. Осуществляется с помощью [Мастера развертывания](#).

Чтобы развернуть конфигурацию на инженерную станцию, выполните следующие действия:

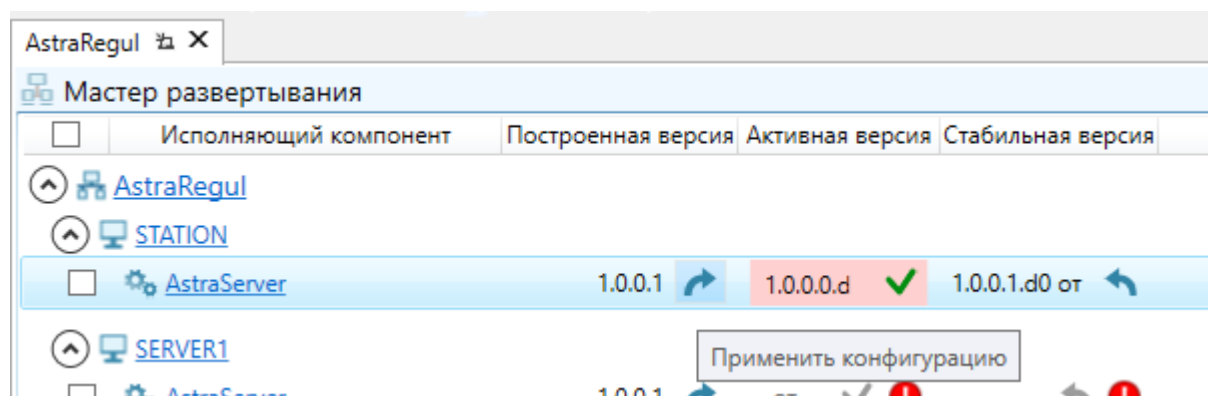
1. Нажмите кнопку "Перейти к развертыванию" на панели инструментов.



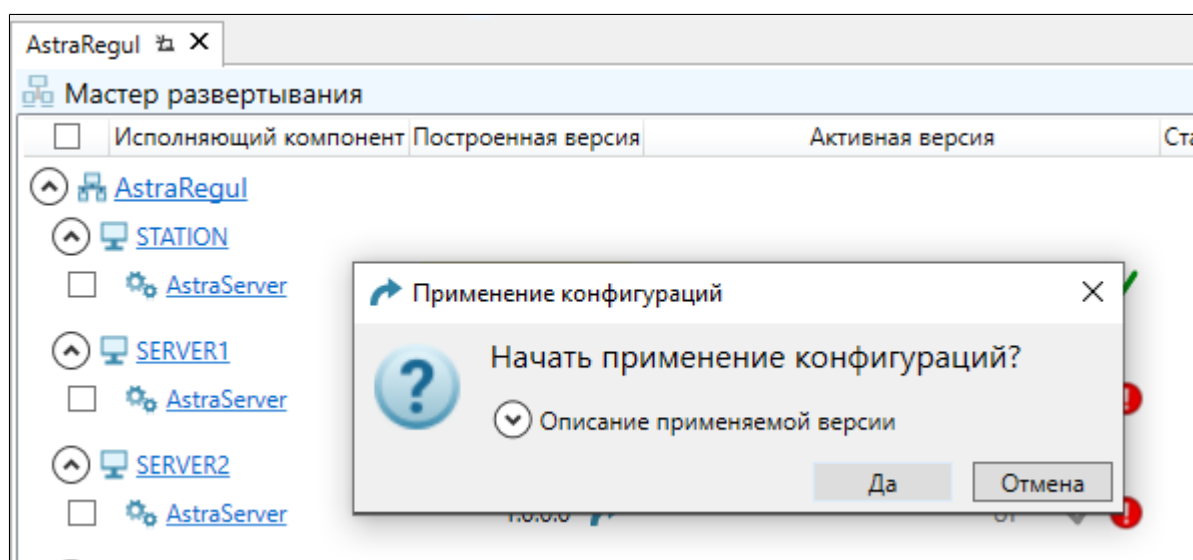
В открывшемся окне мастера развертывания отображена информация об исполняющем компоненте и последняя активная версия.



2. Чтобы применить текущую конфигурацию, нажмите кнопку "Применить конфигурацию".



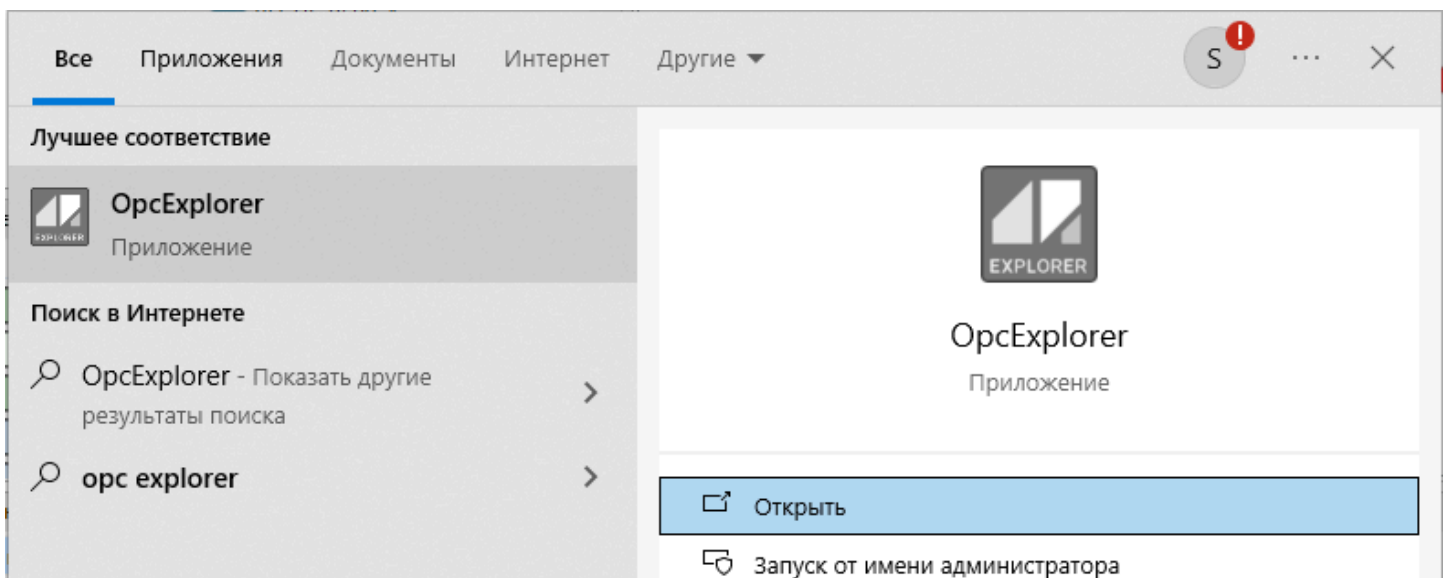
3. Подтвердите применение конфигурации, нажав кнопку "Да".



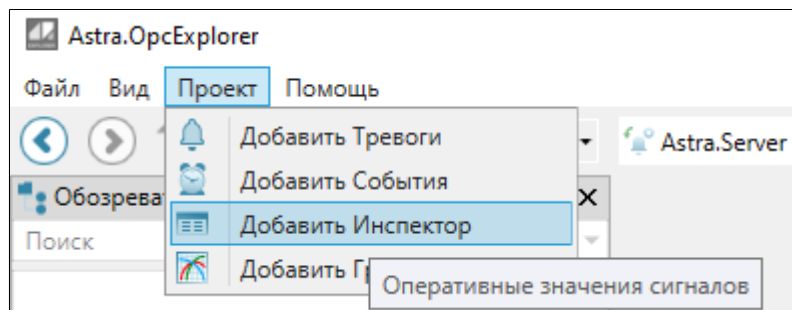
После успешного применения конфигурации к AstraServer в журнале появится соответствующее информационное сообщение.

- Компиляция PsDiagn была успешно выполнена ранее.
- Компиляция PsBase была успешно выполнена ранее.
- Выполнено применение к AstraServer по адресу AstraRegul конфигурации версии 1.0.0.1.d1
- Выполнено изменение конфигурации AstraServer по адресу AstraRegul с версии 1.0.0.0.d2 на версию 1.0.0.1.d1 от 21.11.2023 10:16:47
- Выполнено изменение конфигурации AstraServer по адресу AstraRegul с версии 1.0.0.0.d1 на версию 1.0.0.0.d1 от 22.11.2023 10:37:01

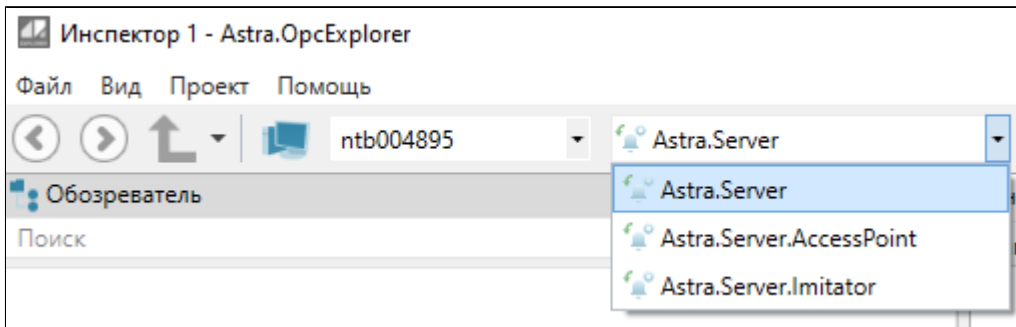
4. Запустите утилиту OpсExplorer из меню Пуск: "Пуск" → "AstraRegul" → "OpсExplorer".



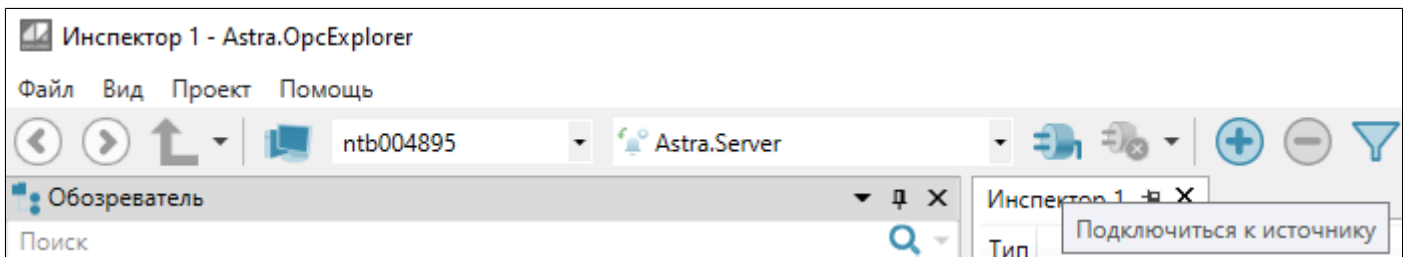
5. Чтобы посмотреть оперативные значения сигналов, выберите меню "Проект" → "Добавить инспектор".



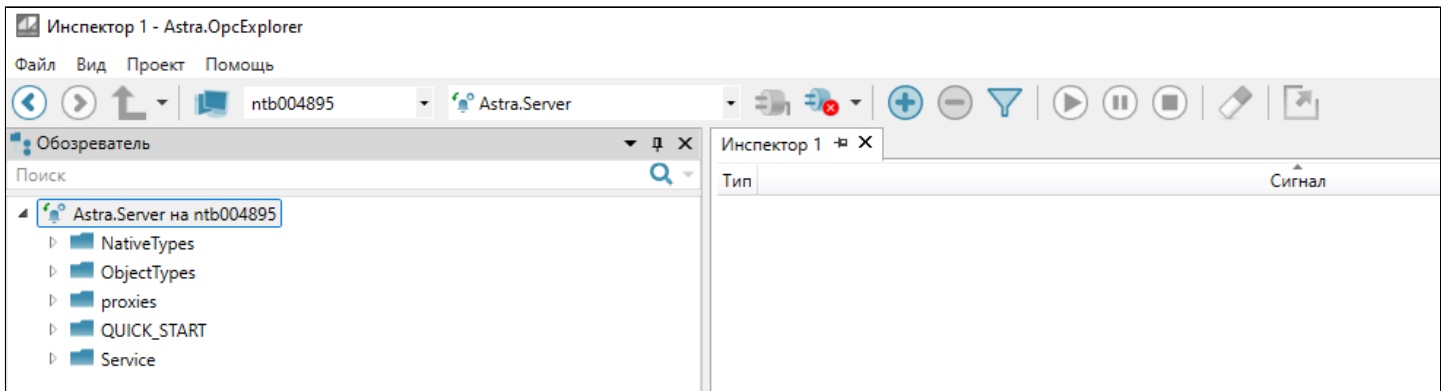
6. Выберите сетевое имя инженерной станции и источник данных "Astra.Server".



7. Нажмите кнопку "Подключиться к источнику" на панели инструментов.



8. Раскройте дерево сигналов сервера ввода/вывода.



9. Раскройте папку "QUICK\_START" и перейдите по следующему пути: "DIAGNOSTICS" → "REGUL\_R500\_A" → "CRATE1". Добавьте в рабочую область инспектора сигналы из папки "CRATE1".

Инспектор 1 - Astra.ОркExplorer

Астра.Сerver

Обозреватель

Астра.Сerver на mtb004895

NativeTypes

ObjectTypes

proxies

QUICK\_START

DIAGNOSTICS

ARM1

REGUL\_R500\_A

CRATE1

CU\_00\_071

REGUL\_R500\_B

SAFETY\_REGUL\_R500S\_A

SAFETY\_REGUL\_R500S\_B

SERVER

STATION

SWITCH

REGUL\_R500

SAFETY\_REGUL\_R500S

Service

Тип	Сигнал	Значение	Время	Источник
S	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.Agg_WarnAck.Comment		30.01.2024 14:43:15	Astra_Server
S	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.Agg_WarnAck.User		30.01.2024 14:43:15	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.DISCARDED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.HW_FAILURE	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.LOWER_ADC_VALID_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.LOWER_ELECTRIC_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.LOWER_TECH_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.UPPER_ADC_VALID_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.UPPER_ELECTRIC_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.B.UPPER_TECH_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
U	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH01.STATUS.VALUE	0	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.DISCARDED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.HW_FAILURE	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.LOWER_ADC_VALID_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.LOWER_ELECTRIC_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.LOWER_TECH_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.UPPER_ADC_VALID_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.UPPER_ELECTRIC_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.B.UPPER_TECH_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
U	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH02.STATUS.VALUE	0	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.DISCARDED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.HW_FAILURE	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.LOWER_ADC_VALID_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.LOWER_ELECTRIC_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.LOWER_TECH_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.UPPER_ADC_VALID_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.UPPER_ELECTRIC_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.B.UPPER_TECH_BOUND_EXCEED	False	31.01.2024 14:04:31	Astra_Server
U	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH03.STATUS.VALUE	0	31.01.2024 14:04:31	Astra_Server
B	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.AI.08_041.CH04.STATUS.B.DISCARDED	False	31.01.2024 14:04:31	Astra_Server

Всего 354 сигнала

10. Убедитесь, что с ПЛК приходят корректные данные.

## 1.1.3.2.5. Отладка развёрнутого решения



Для отладки конфигурации используйте инструмент OpsExplorer и сервисное приложение Статистика.

При отладке можно вручную проверить работу развёрнутого решения:

- › передачу данных между исполняющими компонентами;
- › правильную обработку данных при изменении значений;
- › генерацию событий.

Чтобы выполнить отладку, необходимо подключиться к исполняющим компонентам, работающим в среде исполнения, на которых запущено решение. После подключения к исполняющему компоненту в интерфейсе будут доступны:

- › просмотр значений параметров/переменных/событий у объектов, размещённых на исполняющем компоненте;
- › изменение их значений
- › возникающие события.



# Импорт/экспорт элементов

Элементы можно экспортировать в файл и импортировать из файла.



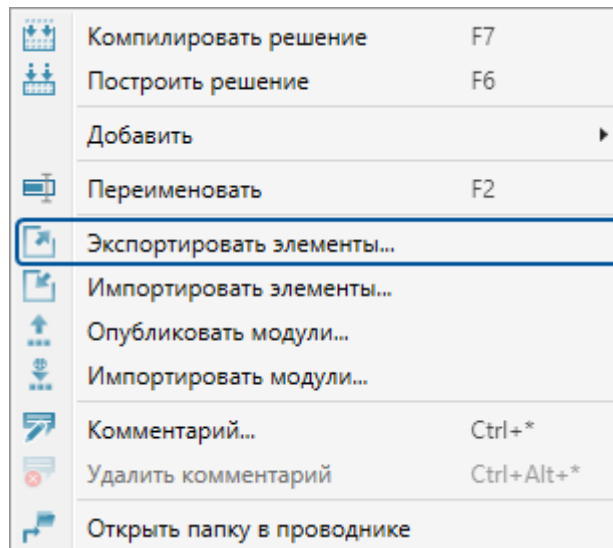
Файл \*.omx-export имеет xml-формат.

Импорт/экспорт элементов используется для добавления и редактирования в проект экземпляров типов, описанных в другой программе.

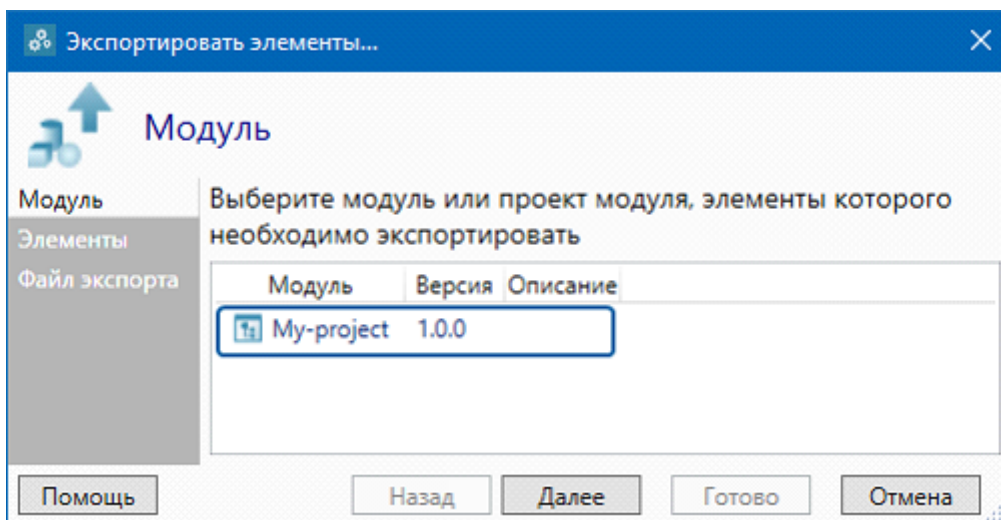
## 1.1.3.2.6.1. Экспорт элементов

Чтобы экспортировать элементы, выполните следующие действия:

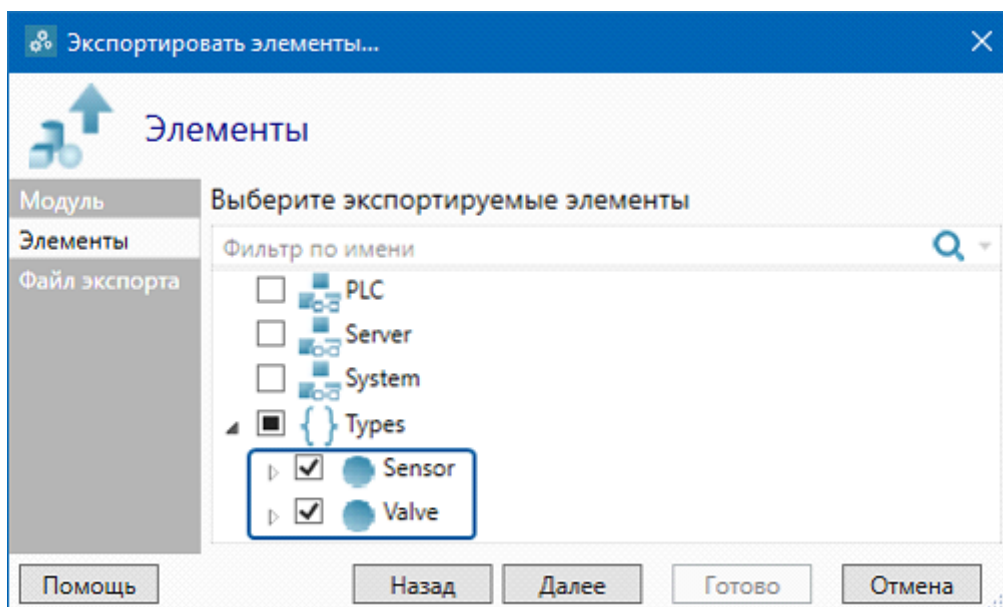
1. В обозревателе решений в контекстном меню решения, проекта или элемента выберите "Экспортировать элементы...".



Откроется мастер экспорта.

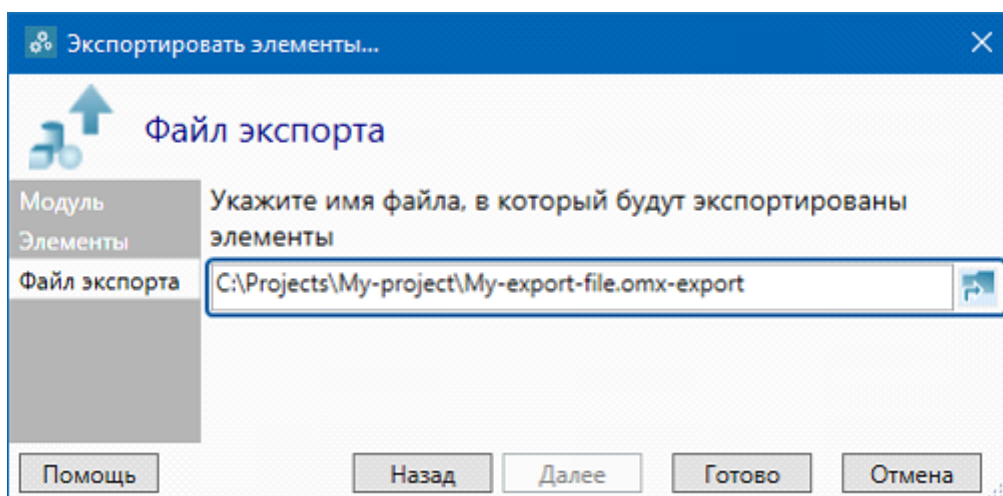


2. Выберите проект целиком или отдельные объекты.



3. В плане экспорта посмотрите, какие элементы будут экспортированы в файл.

4. Укажите название и месторасположение файла экспорта и нажмите кнопку "Готово".



В результате будет создан указанный файл. В файле будут сохранены выбранные элементы.



Если файл существует, он будет перезаписан.

Для каждого элемента сохраняются:

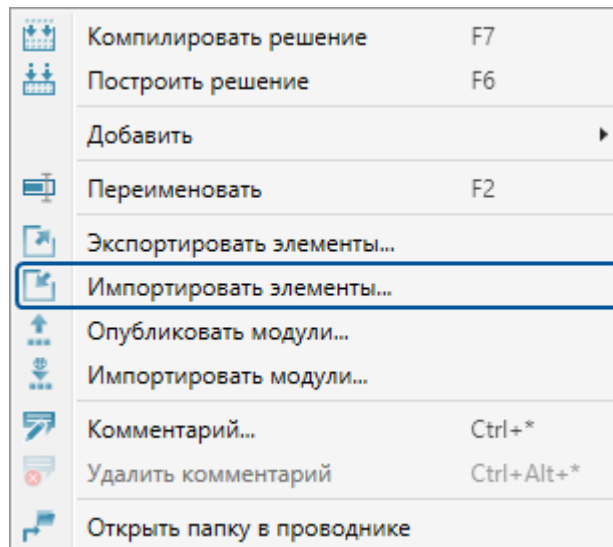
- свойства;

- › атрибуты;
- › вложенные элементы.

## 1.1.3.2.6.2. Импорт элементов

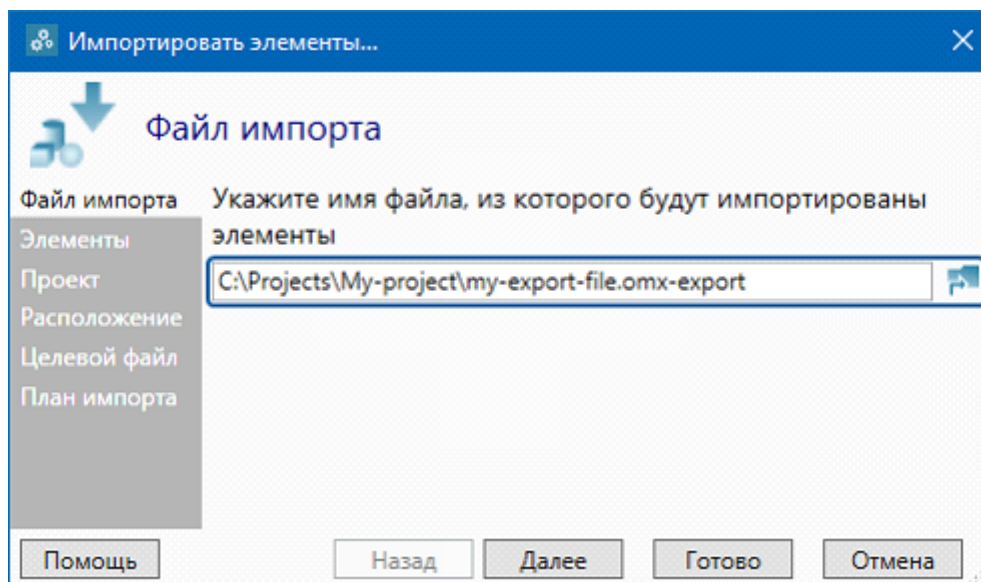
Чтобы импортировать элементы, выполните следующие действия:

1. В обозревателе решений в контекстном меню решения, проекта или элемента выберите "Импортировать элементы...".

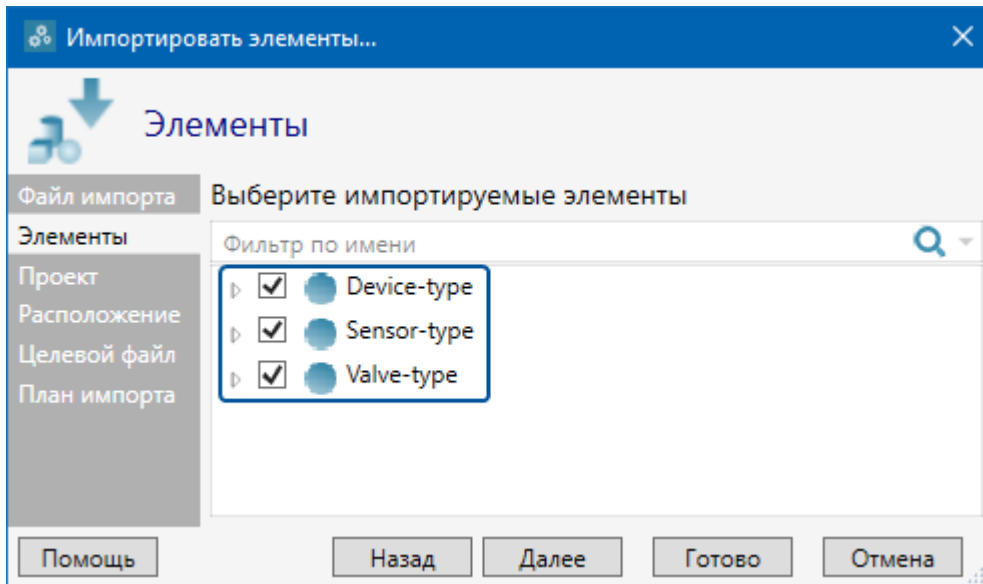


Откроется мастер импорта.

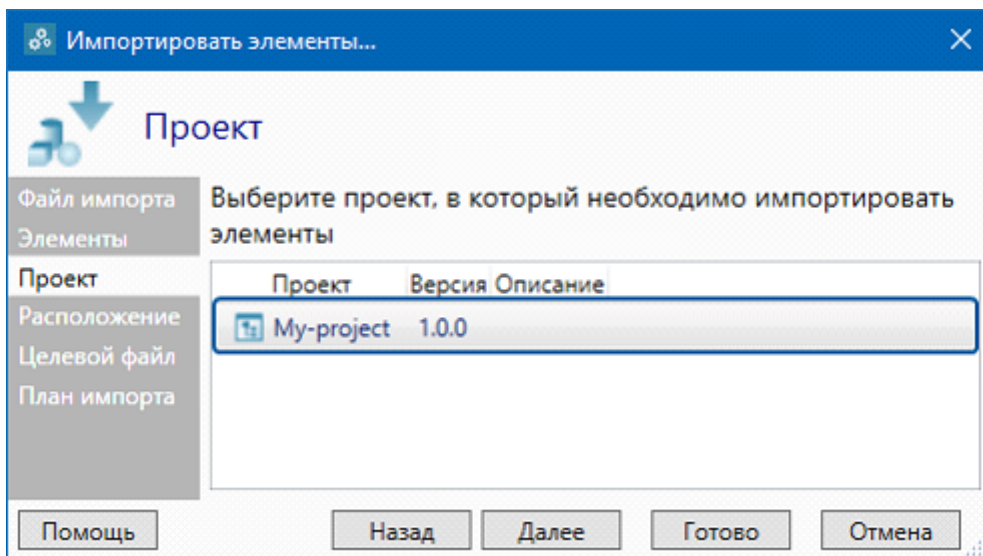
2. Выберите файл с расширением \*.omx-export.



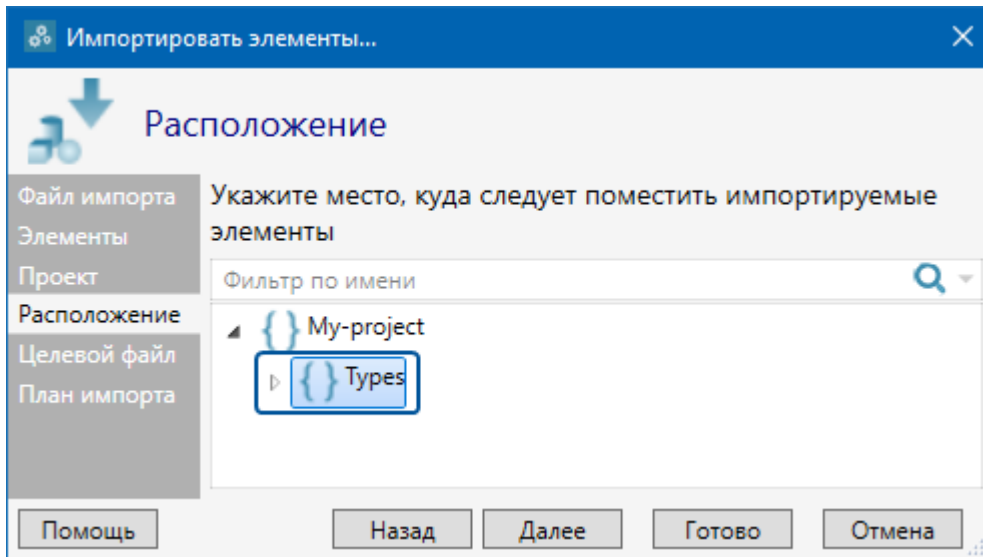
3. Выберите элементы, которые необходимо импортировать.



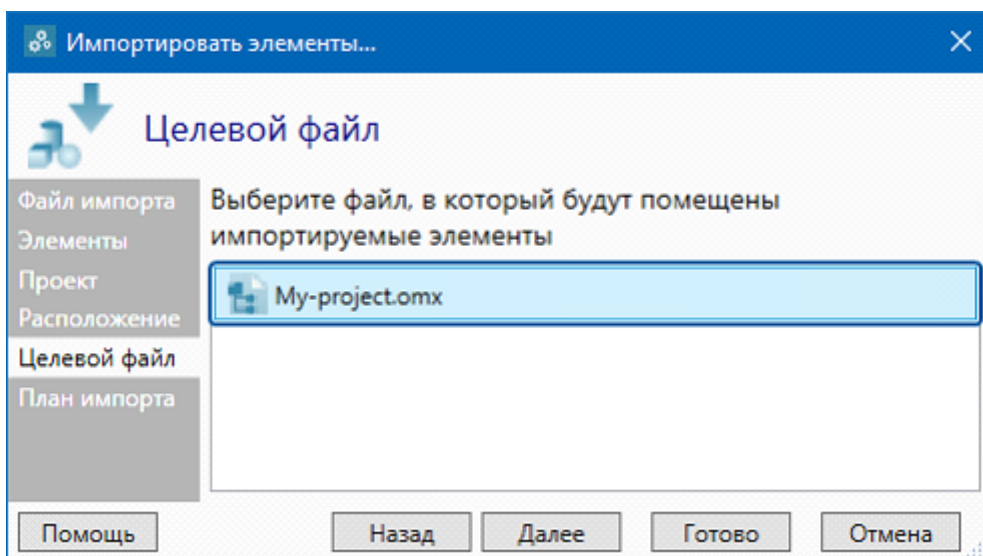
4. Если импорт вызван для решения, выберите проект.



5. Укажите место в проекте, куда следует импортировать элементы.



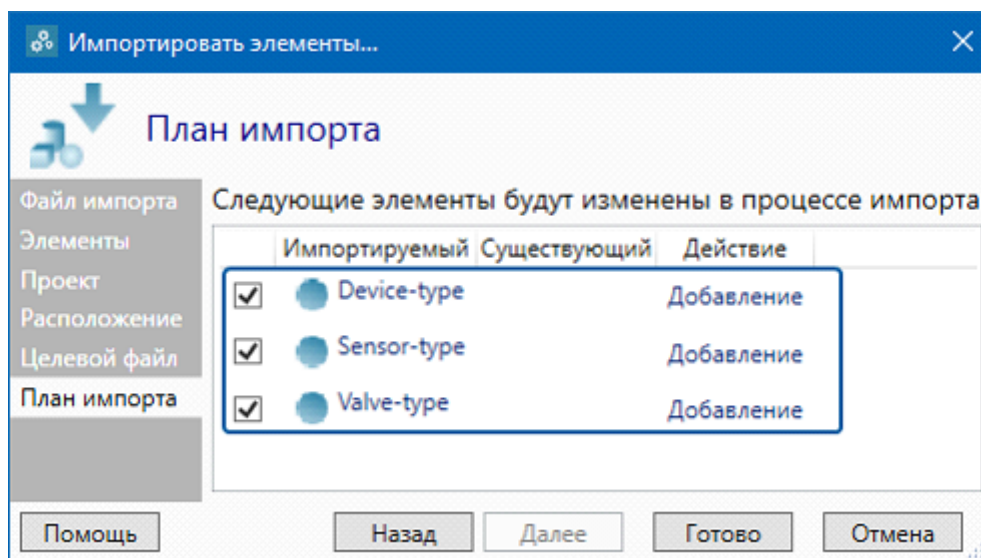
6. Выберите файл проекта, в который будут помещены импортируемые элементы.



Выберите файл проекта, в который будет записана информация об импортируемых элементах.

7. В плане импорта посмотрите изменения, которые будут внесены в проект при импорте.

При необходимости, снимите флаги с тех изменений, которые вносить не нужно.



8. Нажмите кнопку "Готово".

В результате выбранные элементы будут импортированы в выбранный проект и размещены в указанном месте. При импорте, если элемент отсутствует в проекте – он будет добавлен в указанное место.



Наличие элемента проверяется по его идентификатору **uuid** в файле. Если идентификатор не указан, он будет сгенерирован при добавлении элемента.

➤ Если элемент есть в проекте – он будет заменён на элемент, описанный в файле, и перемещён в указанное место.

При замене элемента будет заменено всё его описание, включая вложенные элементы.

➤ Если в файле элементу добавлен атрибут удаления «isDeleted="true"» – при импорте элемент будет удалён из проекта.



Если удаляется объект – будут также удалены объекты, представляющие его. Если удаляется тип – будут также удалены типы, представляющие его, и объекты всех удалённых типов.



## 1.1.4. Astra.HMI

**Astra.HMI** – среда разработки и исполнения визуальной части проектов автоматизации (мнемосимволов, окон управления, мнемосхем).

Astra.HMI позволяет представлять объекты технологического процесса в виде статических и анимированных объектов мнемосхемы для мониторинга за ходом процесса и управления им.

### Режимы работы:

- › Astra.HMI.Disigner. Редактор для графических объектов и мнемосхем.
- › Astra.HMI.Viewer. Среда исполнения.

### Основные возможности:

- › Визуальный редактор со стандартной библиотекой компонентов для построения мнемосхем.
- › Взаимодействие с источниками данных по проприетарному протоколу на базе TCP.
- › Объектно-ориентированный подход при разработке проектов.
- › Возможность создания собственных типов графических объектов и пользовательских библиотек.
- › Поддержка скриптовых языков Astra.Om и JavaScript.
- › Взаимодействие с подсистемой безопасности Astra.Security.
- › Взаимодействие с файловой системой, сетевым окружением и оборудованием компьютера.
- › Компоненты автоматической разметки элементов мнемосхемы.
- › Компоненты организации динамики на мнемосхеме.

## Объектно-ориентированный подход

В Astra.HMI применяется объектно-ориентированный подход.

Дизайнер Astra.HMI позволяет разрабатывать собственные типы графических объектов и пользовательские библиотеки.

Разработанные типы графических объектов можно многократно применять на разных экранных формах и в разных проектах.

## 1.1.4.1. Режим исполнения

Проекты автоматизации, созданные и скомпилированные с помощью дизайнера Astra.HMI, запускаются на исполнение с помощью среды исполнения Astra.HMI.Viewer.

- › [Запуск из режима разработки](#)
- › [Запуск из командной строки/терминала](#)
- › [Запуск из проводника](#)
- › [Запуск визуализатора с параметрами](#)

## 1.1.4.1.1. Запуск из режима разработки

Чтобы открыть отдельную форму проекта, перейдите в контекстное меню нужной формы в Библиотеке компонентов и выберите Показать в рантайме.

Чтобы запустить главную экранную форму, перейдите в меню Проект и выберите Показать главную форму в рантайме (F9).



При запуске выполняется компиляция главной формы и зависимых от нее форм, типов и объектов.

Чтобы запустить отдельную форму проекта, откройте нужную форму, перейдите в меню Проект и выберите Показать активную вкладку в рантайме (Ctrl+F9).



При запуске в рантайм отдельной формы, типа или объекта выполняется компиляция и проверка ошибок только выбранной формы, типа или объекта и зависимых форм, типов и объектов. При запуске главной формы компилируется и проверяется весь проект.

## 1.1.4.1.2. Запуск из командной строки/ терминала

### Запуск с компиляцией

Чтобы запустить в режим рантайма форму, тип или проект с автоматической компиляцией, следует использовать для запуска файл проекта в формате \*.hmi.



При запуске в рантайм отдельной формы или типа выполняется компиляция и проверка ошибок только выбранной формы или типа и зависимых форм и типов. При запуске главной формы компилируется и проверяется весь проект.

### ОС Windows

Перед запуском в рантайм отдельной формы, типа или всего проекта перейдите в папку установки Astra.HMI.



```
cd C:\Program Files\AstraRegul\Astra.HMI
```

Чтобы запустить в рантайм отдельную форму, тип или весь проект, выполните команду:



```
astra.hmi.viewer <полный путь к .hmi файлу> <имя формы> <имя  
типа\объекта>
```



Если в пути к .hmi файлу или в имени экранной формы присутствуют пробелы, используйте двойные кавычки.



Для запуска главной формы имя формы можно не указывать.



Запуск на исполнение типа:

```
astra.hmi.viewer C:\HMI\QUICK_START.hmi MainForm
```



Запуск на исполнение проекта (главной формы):

```
astra.hmi.viewer C:\HMI\QUICK_START.hmi
```



Запуск на исполнение формы с пробелами в названии (пробел в имени файла проекта QUICK START):

```
astra.hmi.viewer "C:\HMI\QUICK START.hmi"
```

## ОС Linux

Чтобы запустить в рантайм отдельную форму, тип или весь проект, находясь в директории с проектом, выполните команду:



```
astra.hmi.viewer <имя проекта>.hmi <имя формы> <имя типа/объекта>
```



Для запуска главной формы значение параметра имя формы можно не указывать.

Запуск на исполнение типа:



```
astra.hmi.viewer QUICK_START.hmi SW
```

Чтобы запустить в рантайм отдельную форму, тип или весь проект, находясь вне директории с проектом, используйте вышеописанные команды и укажите полный путь к проекту.

Запуск на исполнение проекта (главной формы):



astra.hmi.viewer HMI\QUICK\_START.hmi

## Запуск без компиляции

Чтобы запустить в режим рантайма форму, тип, или весь проект без компиляции, следует использовать для запуска файл скомпилированного проекта в формате \*.binom.



Файл \*.binom привязан к версии Astra.HMI.

Запускайте проект только при наличии той же версии, на которой был создан файл.

Команды запуска в рантайм формы, типа или проекта без компиляции аналогичны запуску с компиляцией. Вместо формата \*.hmi следует указывать формат \*.binom.

Запуск на исполнение проекта в ОС Windows:



```
astra.hmi.viewer C:\HMI\QUICK_START.binom
```

Запуск на исполнение проекта в ОС Linux:



```
astra.hmi.viewer QUICK_START.binom
```



### 1.1.4.1.3. Запуск из проводника

Чтобы запустить проект в рантайм, перейдите в папку проекта и два раза щелкните по файлу проекта с расширением \*.hmi. В результате будет запущена форма, указанная главной.



При запуске выполняется компиляция главной формы и зависимых от нее форм, типов и объектов.

## 1.1.4.1.4. Запуск визуализатора с параметрами

Чтобы отобразить справку, введите ключи `-?`, `-h` или `--help`.



```
astra.hmi.viewer.exe --help
```

Чтобы отобразить информацию о версии, используйте ключи `-v`, `--version`.



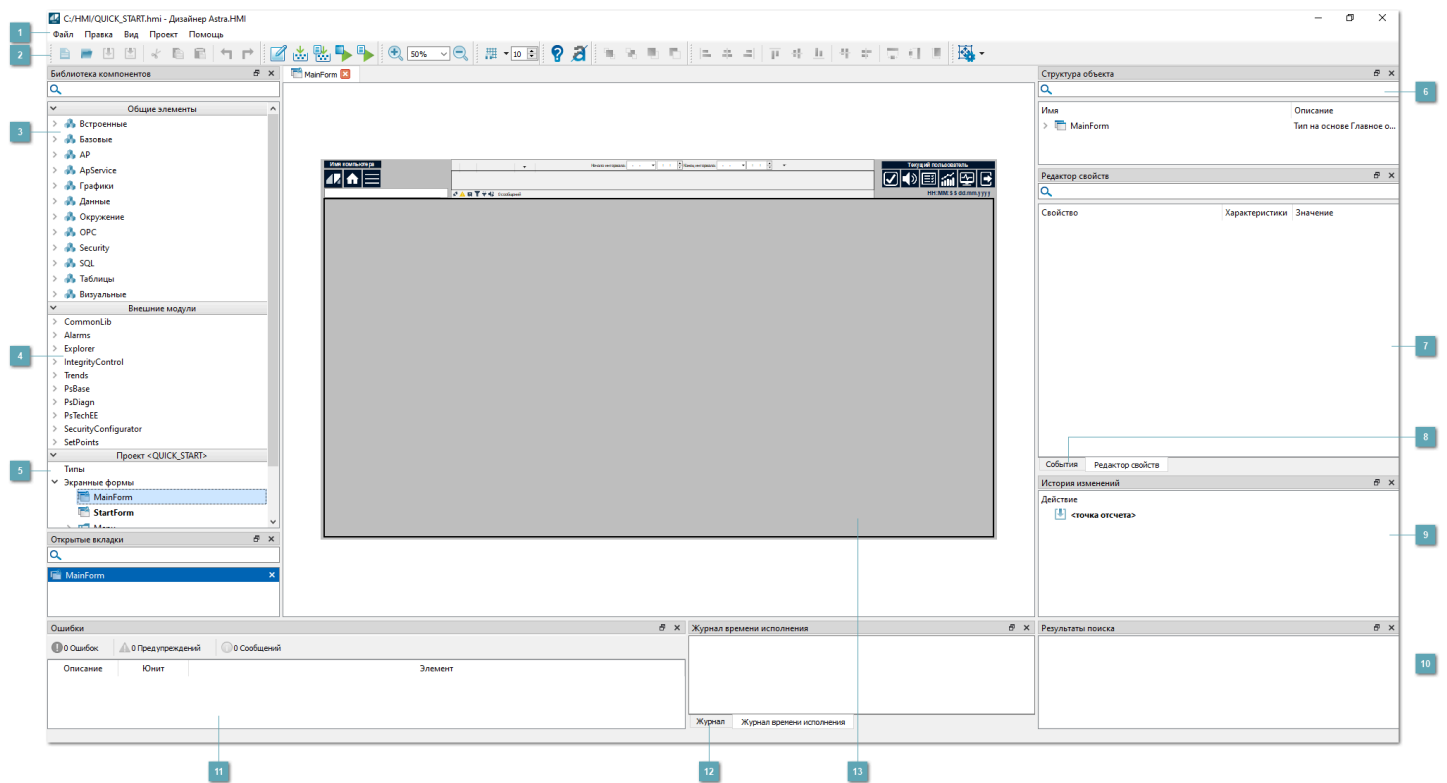
```
astra.hmi.viewer.exe --version
```

Чтобы отключить сглаживание изображений в визуализаторе, используйте ключи `--disable AA`, `--disable antialiasing`.



```
astra.hmi.viewer.exe C:\HMI\QUICK_START.hmi MainForm --disable AA
```

# 1.1.4.2. Интерфейс



## 1 Главное меню

Содержит:

- Файл – действия с проектами, открытие настроек приложения.
- Правка – отмена/повтор последних действий, копирование/вставка/удаление элементов, поиск и замена элементов.
- Вид – включение окон инструментов, сброс макета окон.
- Проект – создание и редактирование проекта, компиляция, запуск проекта в рантайме.
- Помощь – открытие встроенной справки, информации о программе.

## 2 Панель инструментов

Содержит:

- инструменты работы с проектом;
- инструменты масштабирования рабочей области;
- инструменты форматирования объектов;
- инструменты выравнивания и распределения;
- инструменты отмены/повтора действий.

### **3 Библиотека компонентов**

Содержит набор стандартных компонентов для разработки проекта.

### **4 Внешние модули**

Содержит подключаемые библиотеки.

### **5 Проект**

Содержит экранные формы, глобальные объекты и типы, созданные пользователем.

### **6 Структура объектов**

Содержит иерархию объектов рабочей области.

### **7 Редактор свойств**

Содержит перечень свойств выделенного объекта.

### **8 События**

Содержит перечень событий выделенного объекта.

### **9 История изменений**

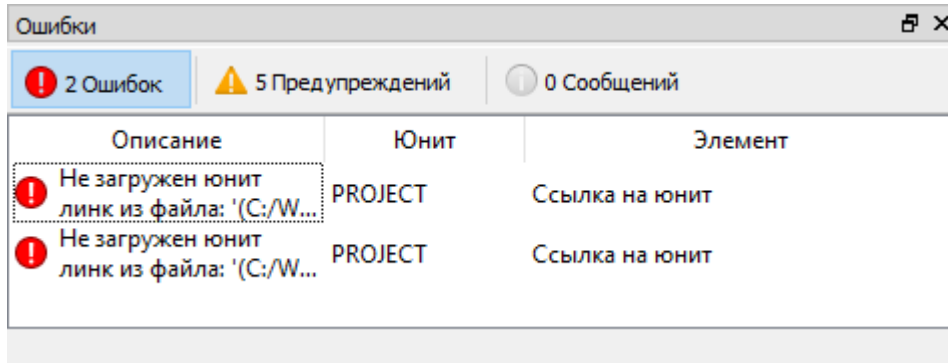
Содержит лог изменений, проводимых с объектами рабочей области.

### **10 Результаты поиска**

Содержит перечень найденных объектов по последнему запросу.

## 11 Ошибки

Содержит перечень критических ошибок, предупреждений и сообщений, которые возникли в ходе компиляции проекта. Чтобы перейти к месту ошибки в проекте, дважды щелкните по ошибке.



Описание	Юнит	Элемент
❗ Не загружен юнит линк из файла: '(C:/W...	PROJECT	Ссылка на юнит
❗ Не загружен юнит линк из файла: '(C:/W...	PROJECT	Ссылка на юнит

## 12 Журнал

Содержит лог событий, возникших в процессе компиляции проекта. Журнал времени исполнения содержит события происходящие в процессе исполнения скомпилированного проекта.

## 13 Рабочая область

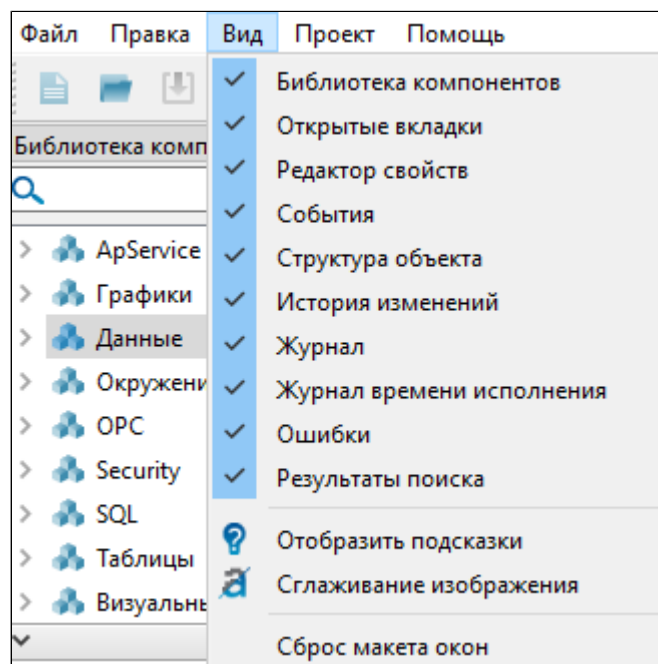
Рабочая область – холст для размещения экранных форм и объектов.

# Настройка среды разработки

Дизайнер Astra.HMI позволяет пользователю гибко настраивать области пользовательского интерфейса. Области можно скрывать, отображать, перемещать внутри области Дизайнера Astra.HMI, выносить за пределы редактора Astra.HMI, а также изменять размеры отображаемых областей. Настройки среды разработки сохраняются после перезапуска программы.

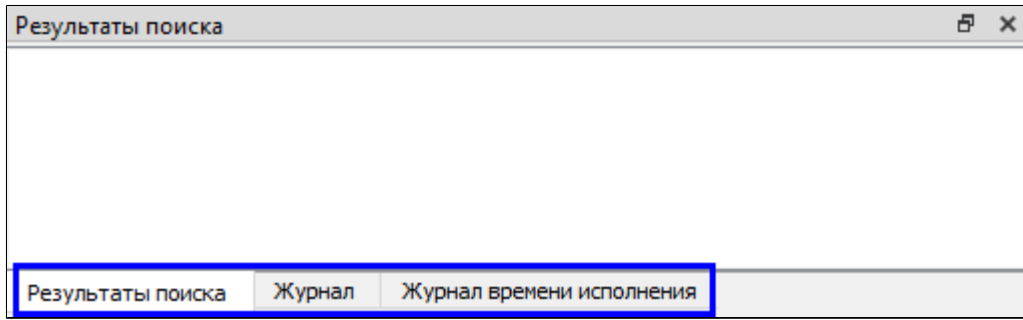
Чтобы скрыть/отобразить отдельные области пользовательского интерфейса, используйте флаги контекстного меню панели инструментов или опции меню Вид.

- ✓ Журнал
  - ✓ Журнал времени исполнения
  - ✓ События
  - ✓ Библиотека компонентов
  - ✓ Открытые вкладки
  - ✓ Структура объекта
  - ✓ Редактор свойств
  - ✓ Ошибки
  - ✓ История изменений
  - ✓ Результаты поиска
- 
- ✓ Основная панель инструментов
  - ✓ Панель проекта
  - ✓ Панель масштабирования
  - ✓ Панель инструментов
  - ✓ Панель настроек сцены
  - ✓ Панель порядка
  - ✓ Панель выравнивания
  - ✓ Настройки инструментов редактирования



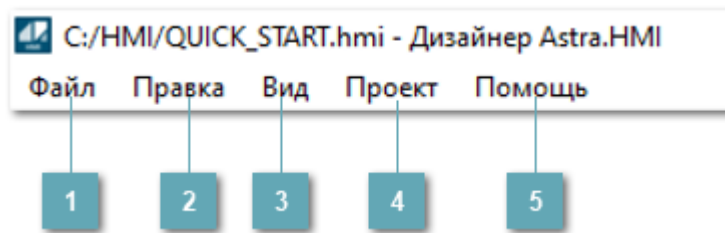
Отдельные составляющие панели инструментов можно перемещать в пределах и за пределами редактора Astra.HMI, а также менять их размер.

Чтобы создать области с вкладками, перетащите одну область на другую. Автоматически создадутся вкладки, соответствующие наложенным друг на друга областям интерфейса.





## 1.1.4.2.2. Главное меню



### 1 Файл

Содержит команды:

- › действий с проектами (создать, открыть, сохранить, закрыть);
- › действий с объектами (импорт, экспорт);
- › открыть настройки параметров проекта;
- › сброса данных приложения;
- › выхода из Дизайнера.

### 2 Правка

Содержит команды:

- › отменить/повторить действие;
- › копировать/вставить/вырезать;
- › найти/заменить.

### 3 Вид

Содержит команды:

- › включить/отключить окна инструментов;
- › отобразить подсказки;
- › сгладить изображения.

## 4 Проект

Содержит команды:

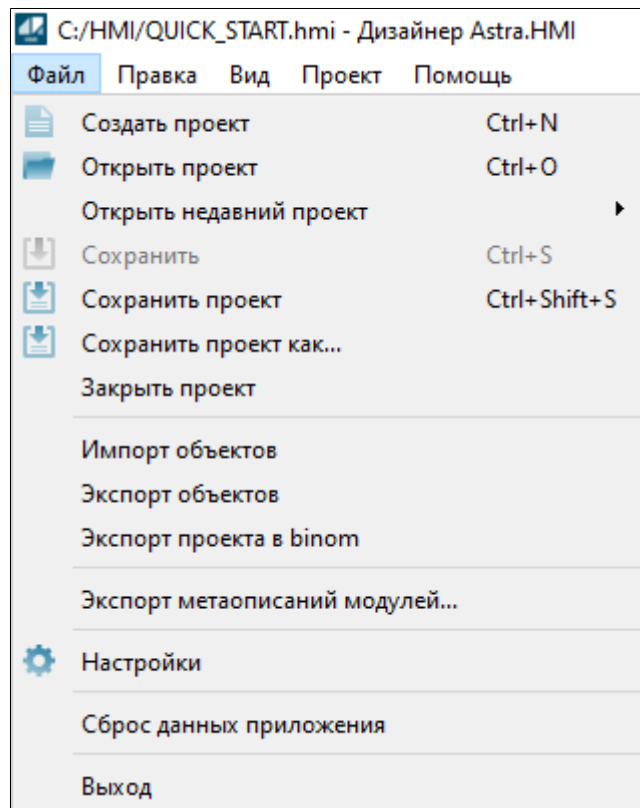
- › переименовать;
- › открыть мастер создания элементов;
- › добавить новую экранную форму;
- › запуск компиляции (всего проекта и текущего типа);
- › показать главную форму в рантайме;
- › показать активную вкладку в рантайме;
- › проверка целостности;
- › актуализировать версии элементов;
- › открыть редактор ссылок на юниты;
- › обновить ссылки на внешние юниты.

## 5 Помощь

Содержит команды:

- › открыть встроенную справку;
- › просмотреть информацию о программе;

## 1.1.4.2.2.1. Файл



### Создать проект

Чтобы создать новый проект в меню "Файл" выберите команду "Создать проект" (Ctrl+N).



Рекомендуем, для удобства, каждый проект создавать в отдельной папке.

Папка с проектом будет содержать в себе две папки – objects и resources, а также сам файл проекта с расширением \*.hmi.

### Открыть проект

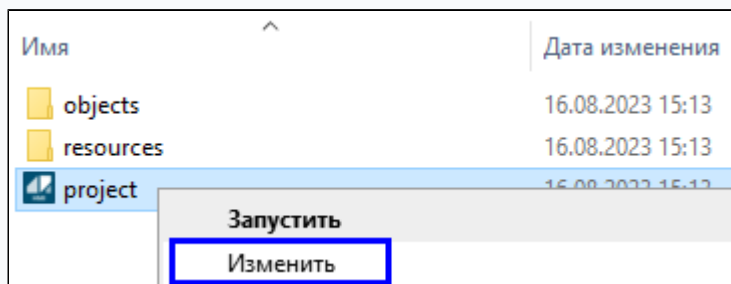
Для открытия готового проекта в меню "Файл" выберите команду "Открыть проект" (Ctrl+O).



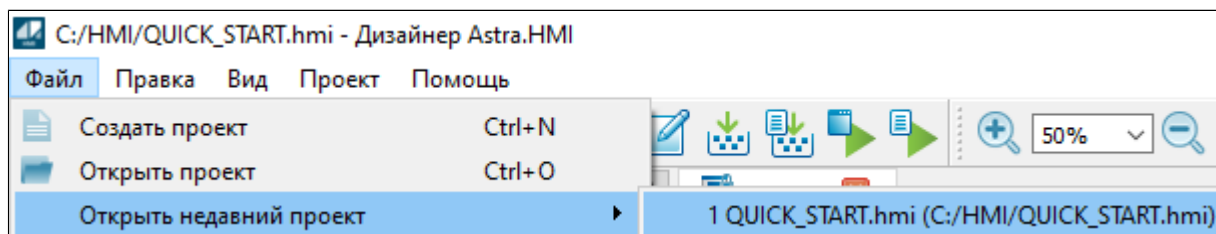
В заголовке окна дизайнера будет отображен путь открытого проекта.



Второй способ открыть проект: перейдите в папку проекта и выберите команду Изменить в контекстном меню файла проекта с расширением \*.hmi.



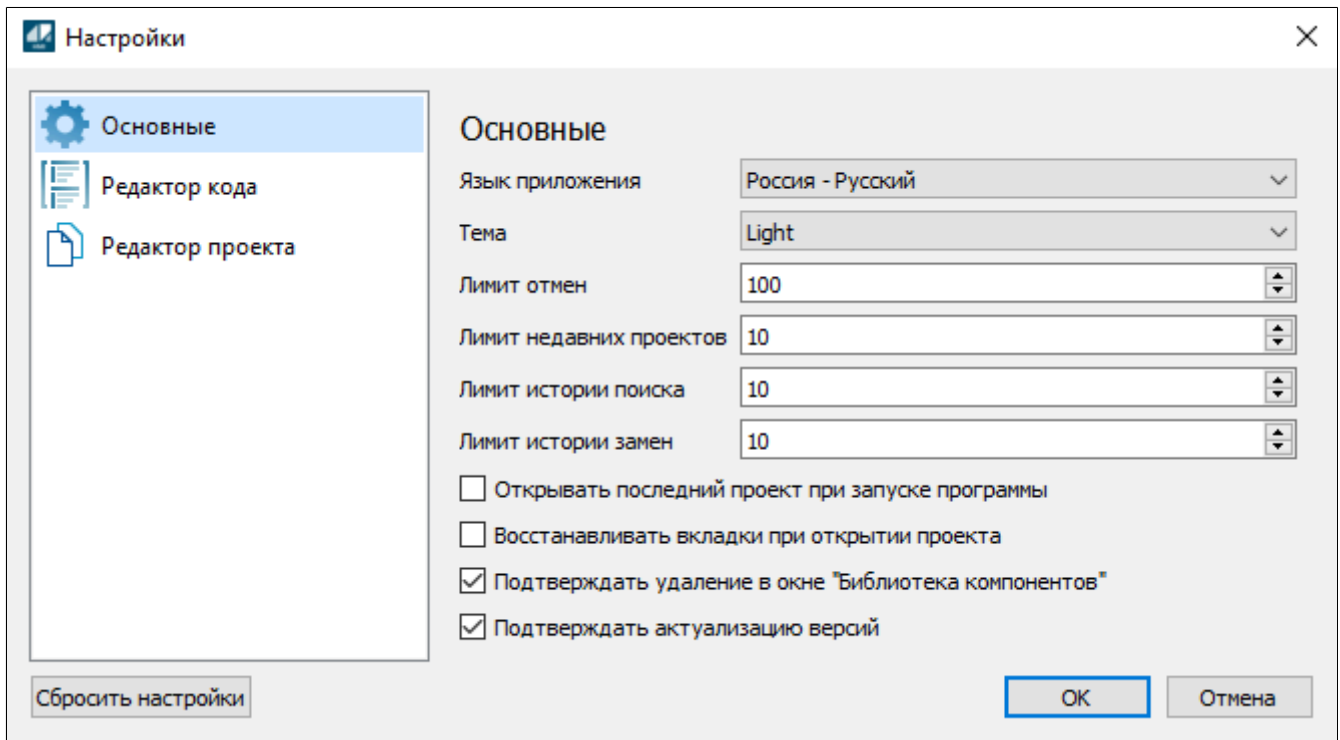
Вы можете открыть недавно закрытый проект, выбрав из выпадающего списка нужный проект в меню Файл → Открыть недавний проект.



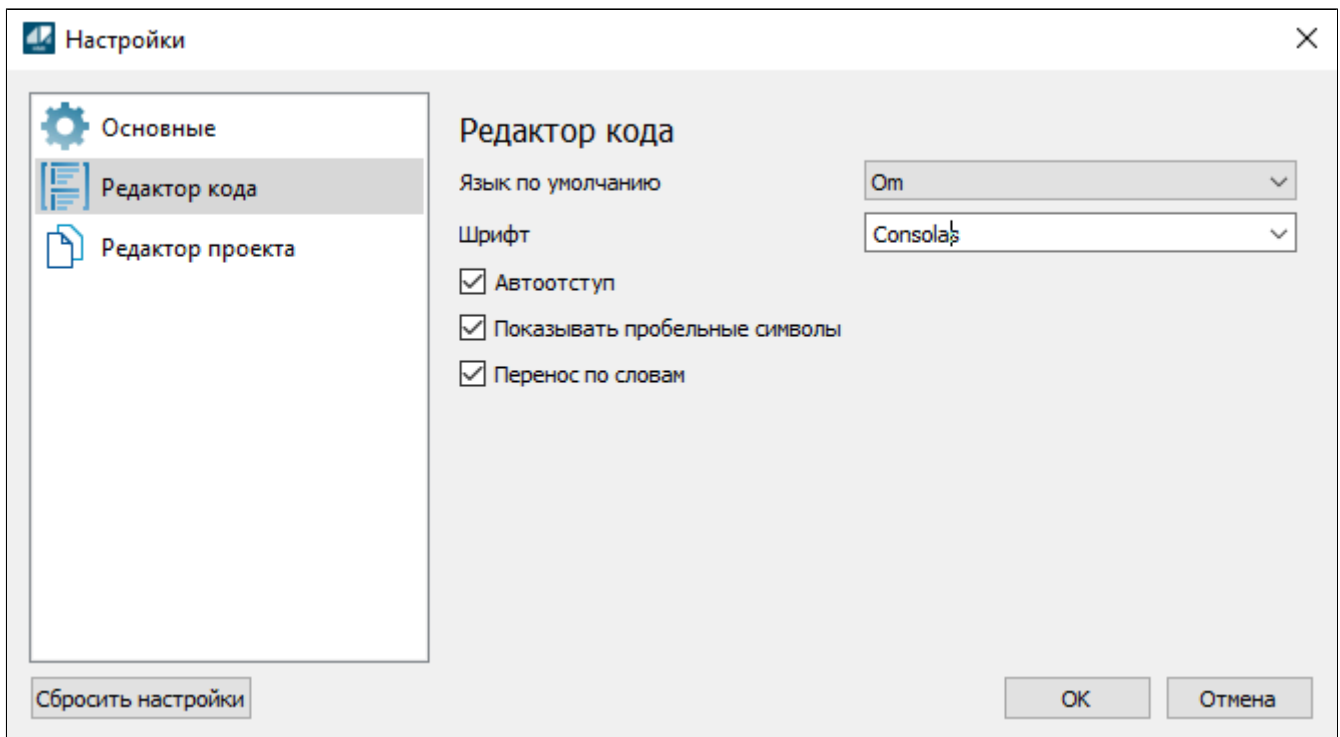
## Настройки

Чтобы открыть окно настроек Astra.HMI, перейдите в меню Файл и выберите Настройки.

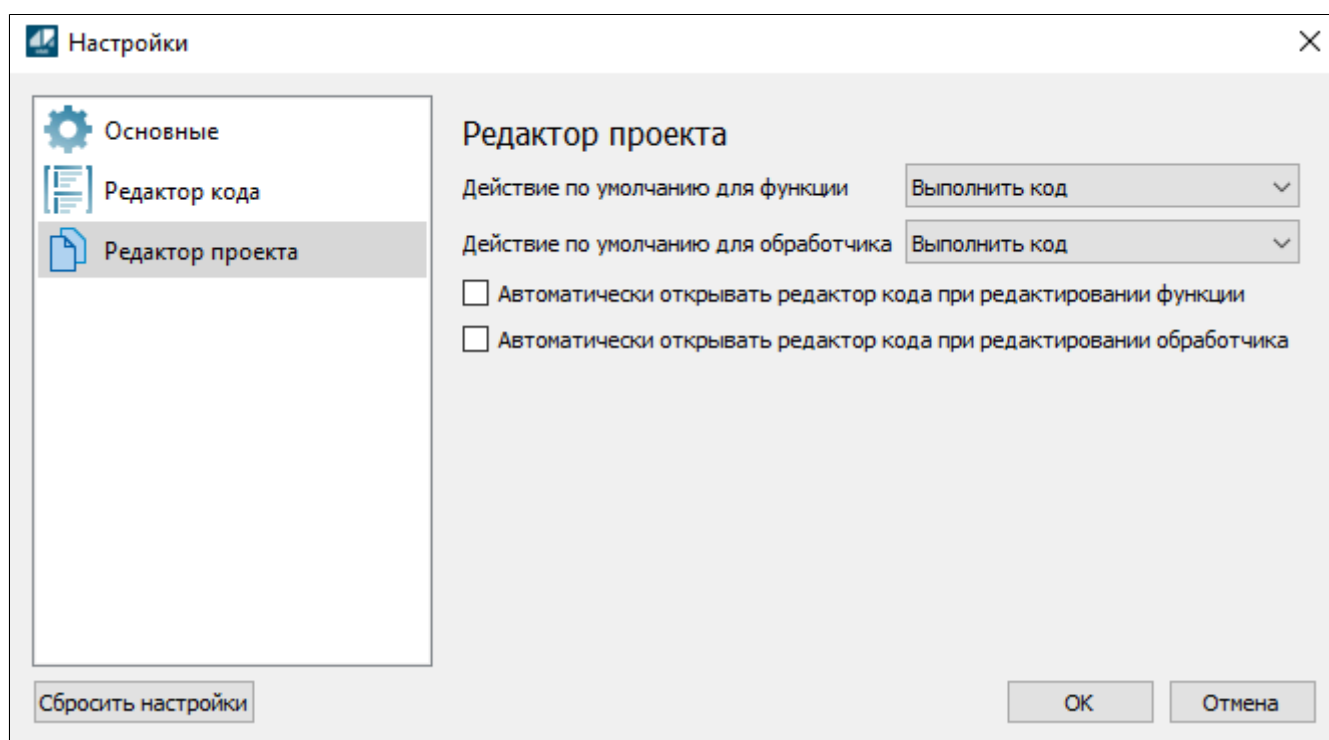
На вкладке "Основные" вы можете выбрать язык интерфейса, цветовую тему дизайнера, установить максимальное количество отменяемых действий в дизайнера и параметры открытия проекта, установить флаги для функций подтверждения актуализации версий и удаления компонентов в окне "Библиотека компонентов".



На вкладке "Редактор кода" вы можете выбрать язык для написания кода (Om или Javascript) и шрифт, установить флаги для добавления автоотступа, отображения пробельных символов и включения переноса слов.



На вкладке "Редактор проекта" вы можете выбрать действие, которое будет назначаться для функции или обработчика по умолчанию, установить флаги автоматического открытия редакторов кода для функции или обработчика.

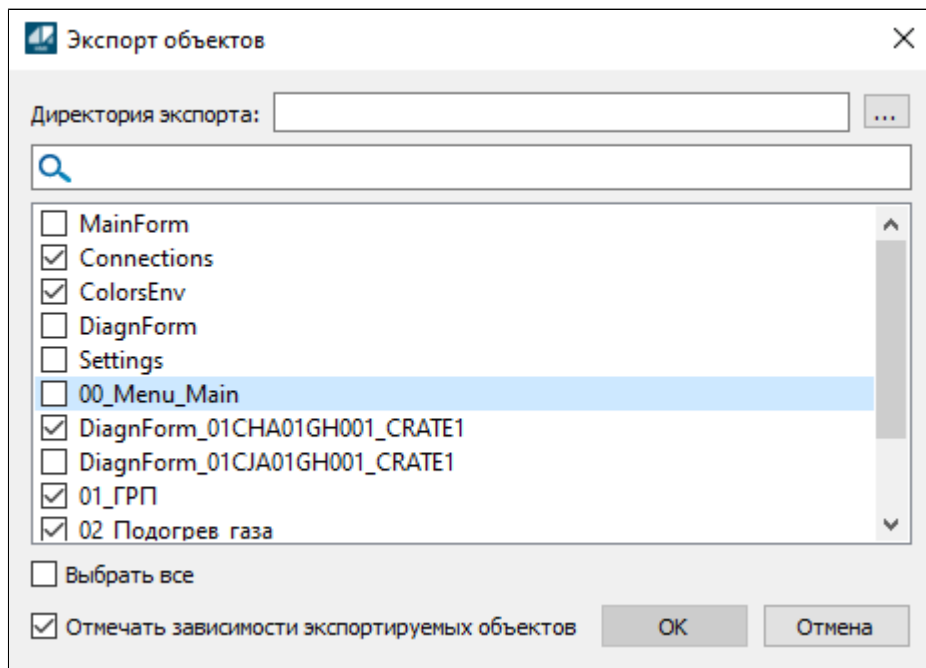


# Импорт и экспорт

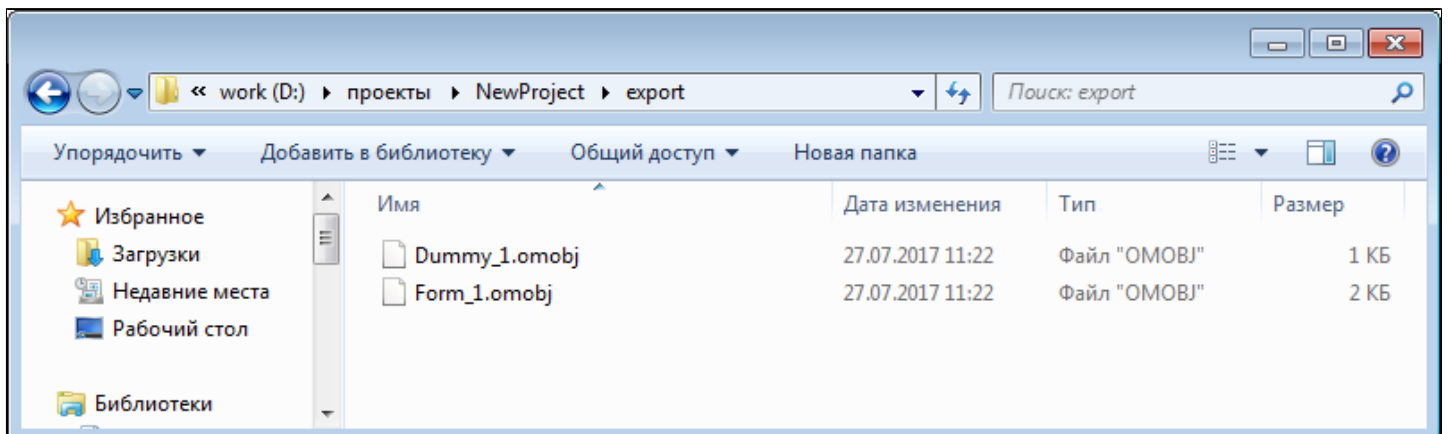
В дизайнере Astra.HMI вы можете экспортировать и импортировать экранные формы или типы графических объектов в формате \*.omobj, а также текстовое описание встроенных и нативных юнитов в формате \*.json.

Чтобы экспортировать типы графических объектов или формы:

1. Выберите команду Файл → Экспорт объектов.
2. В окне Экспорт объектов выберите директорию экспорта и типы, которые должны быть экспортированы.

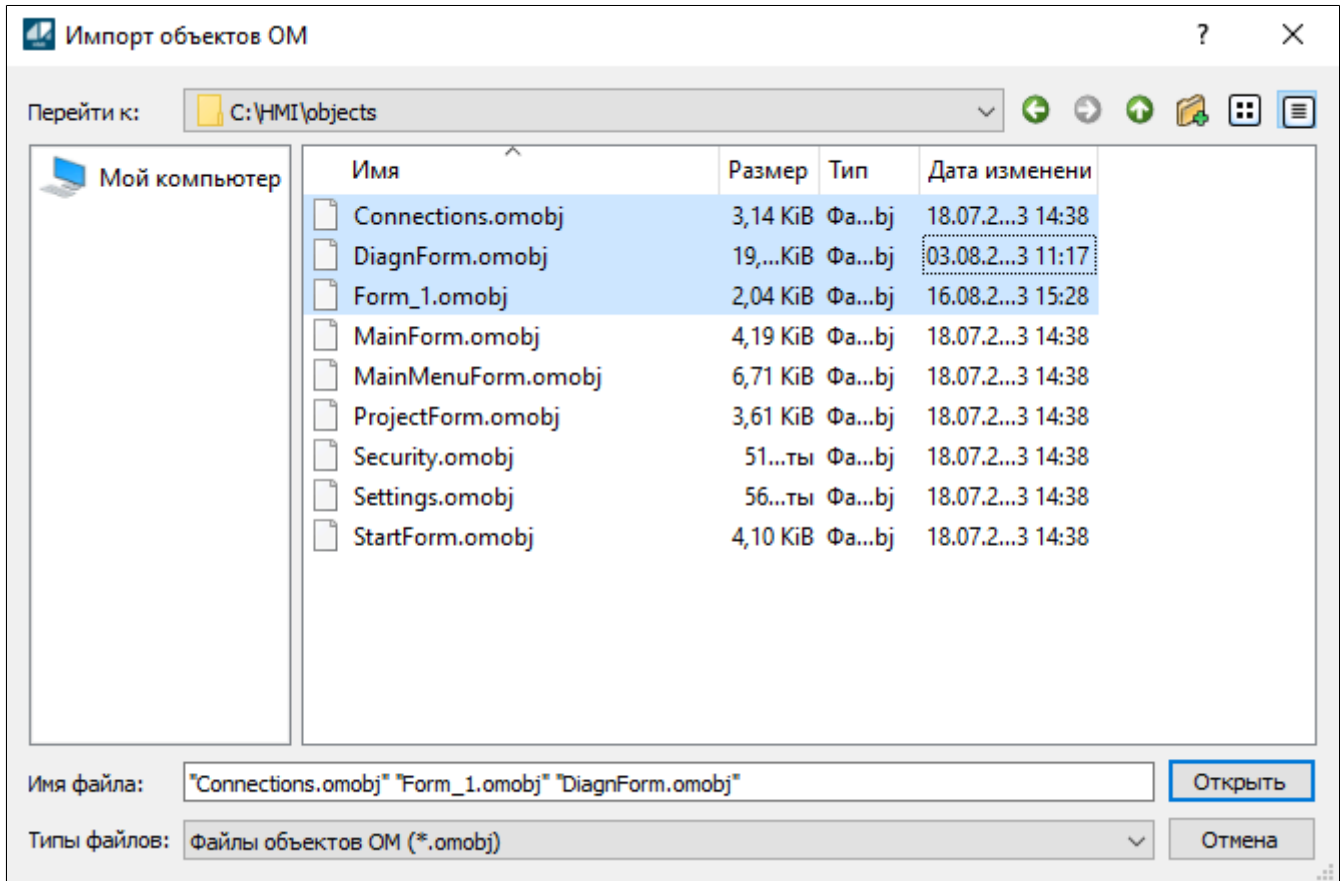


По указанной директории экспортируются файлы выбранных типов.



Чтобы импортировать типы графических объектов или формы:

1. Выберите команду Файл → Импорт объектов.
2. В окне Импорт объектов ОМ выберите типы и формы, которые необходимо добавить к уже существующим в проекте.



Чтобы выделить несколько объектов, используйте клавиши Ctrl и Shift.

Если импортированные типы и формы уже существуют в проекте, дизайнер Astra.HMI предложит сохранить их копии.

Чтобы экспортировать проект с расширением \*.binom:

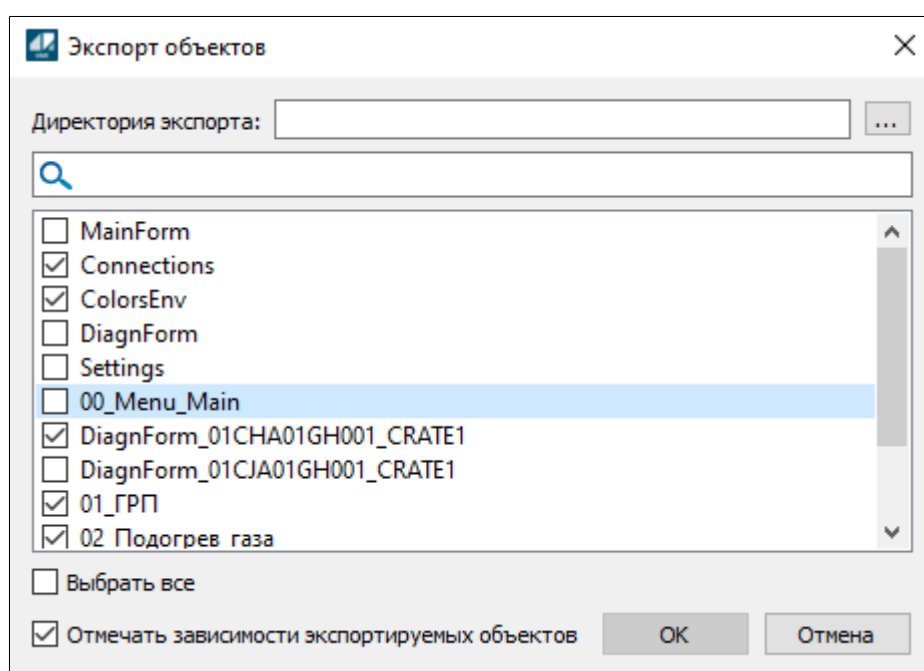
1. Выберите команду Файл → Экспорт проекта в binom.
2. В папку с проектом будет добавлена папка output с созданным файлом binom.



Локальный диск (C:) > HMI > output			
Имя	Дата изменения	Тип	Размер
resources	16.08.2023 15:36	Папка с файлами	
QUICK_START.binom	16.08.2023 15:36	Файл "BINOM"	47 КБ

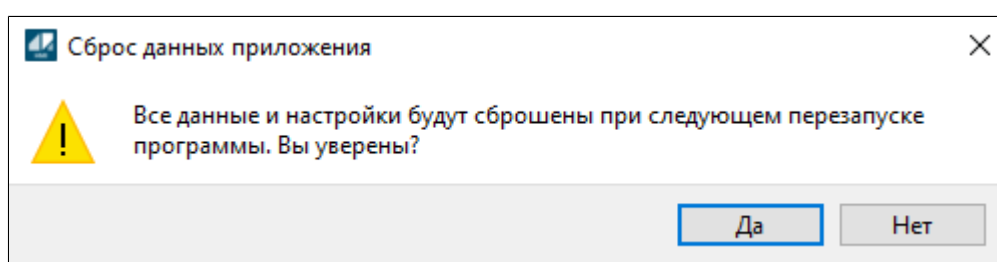
Чтобы экспортировать текстовое описание встроенных и нативных юнитов в формате \*.json:

1. Выберите команду **Файл → Экспорт метаописаний модулей...**
2. В окне **Экспорт метаописаний модулей в JSON-файл** выберите директорию экспорта и модули, которые должны быть экспортированы.

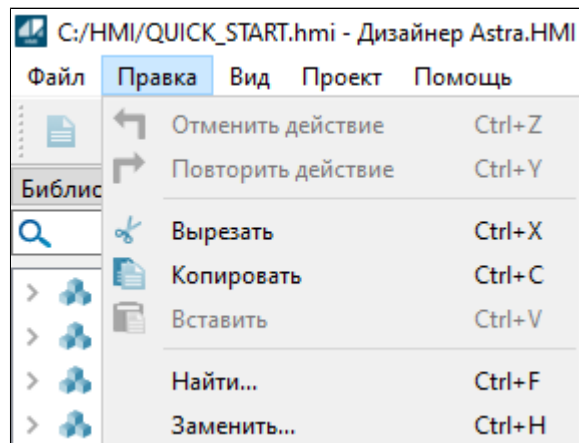


## Сброс данных приложения

Чтобы сбросить все данные и настройки в проекте, в меню **Файл** выберите команду "Сброс данных приложения". Сброс будет выполнен при следующем перезапуске дизайнера.



## 1.1.4.2.2. Правка



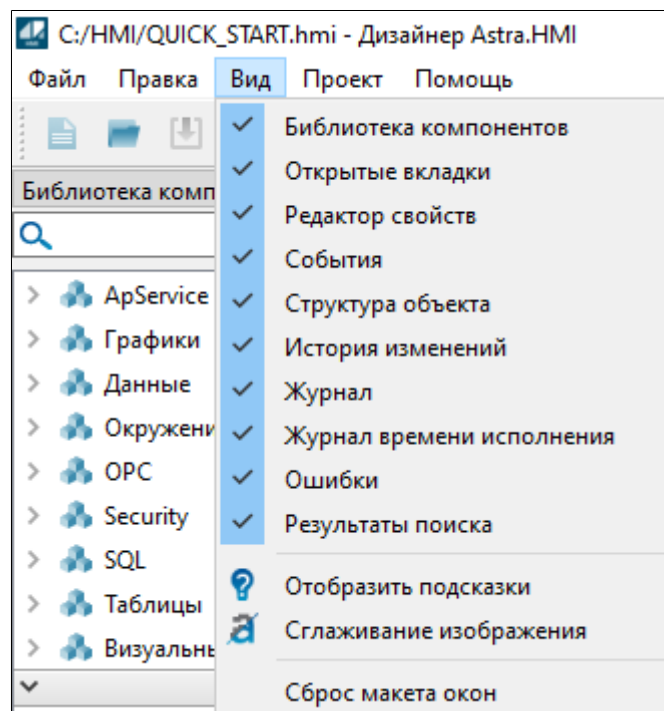
Для редактирования проекта в меню Правка доступны следующие команды:

- › отменить последнее действие (сочетание клавиш Ctrl+Z);
- › повторить последнее действие (сочетание клавиш Ctrl+Y);
- › копировать (сочетание клавиш Ctrl+C);
- › вырезать (сочетание клавиш Ctrl+X);
- › вставить (сочетание клавиш Ctrl+V);
- › найти (сочетание клавиш Ctrl+F);
- › заменить (сочетание клавиш Ctrl+H).

## 1.1.4.2.2.3. Вид

Дизайнер Astra.HMI позволяет пользователю гибко настраивать области пользовательского интерфейса. Области можно скрывать, отображать, перемещать внутри области Дизайнера Astra.HMI, выносить за пределы Дизайнера Astra.HMI, а также изменять размеры отображаемых областей. Настройки среды разработки сохраняются после перезапуска программы.

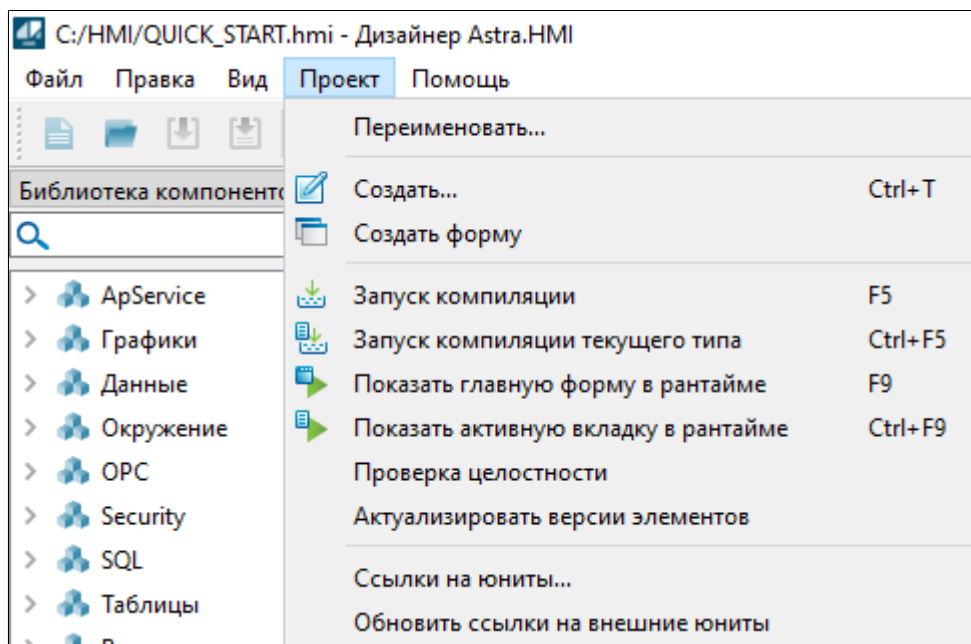
Чтобы скрыть/отобразить отдельные области пользовательского интерфейса, используйте флаги контекстного меню панели инструментов или опции меню Вид.



Отдельные составляющие панели инструментов можно перемещать в пределах/за пределами Дизайнера Astra.HMI и менять их размер.

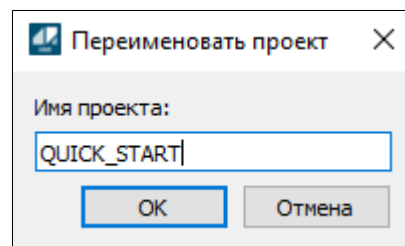
Отдельные области пользовательского интерфейса (кроме панели главного меню) можно перемещать в пределах/за пределами Дизайнера Astra.HMI и менять их размер.

## 1.1.4.2.2.4. Проект



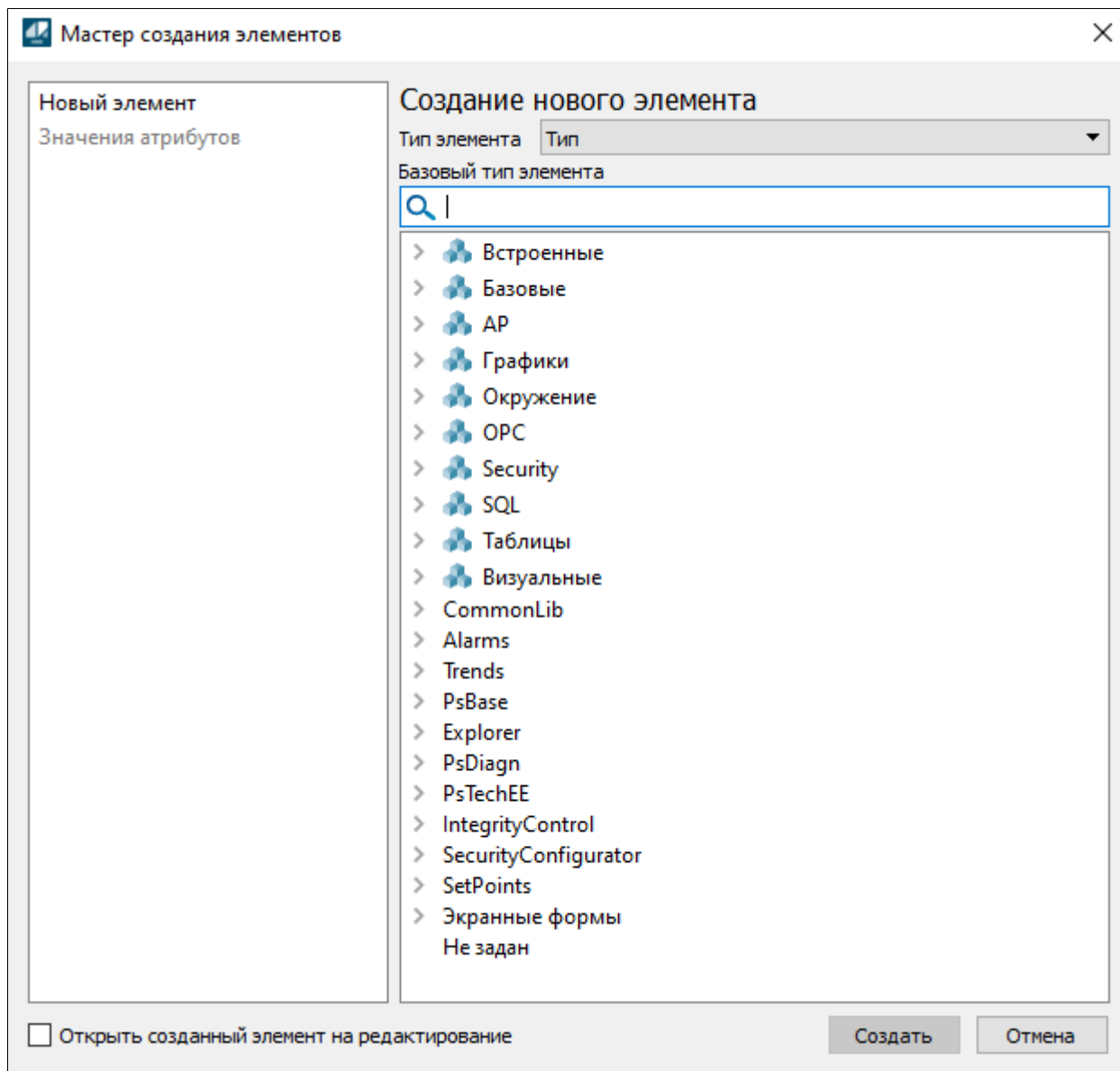
### Переименовать

Чтобы переименовать проект, в меню Проект выберите "Переименовать...". В открывшемся окне задайте новое имя проекта.



### Создать ...

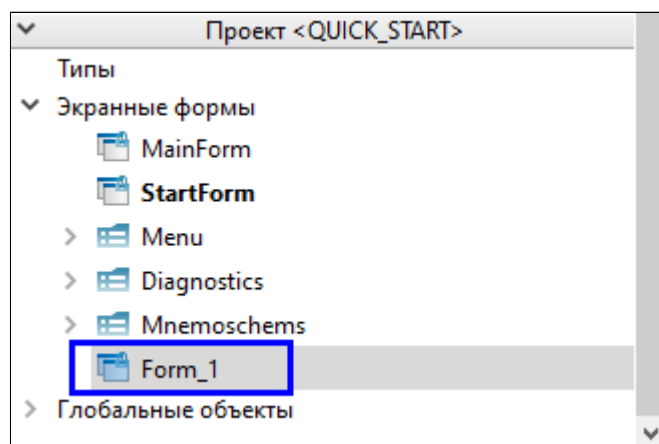
Чтобы создать новый элемент, в меню Проект выберите "Создать..." (Ctrl+T). Откроется мастер создания элементов.



В открывшемся мастере создания элементов задайте значения полей Тип элемента и Базовый тип элемента.

## Создать форму

Чтобы создать новую форму, выберите в меню Проект команду "Создать". Новая форма будет добавлена в раздел Экранные формы проекта.



# Компиляция

Компиляция формы, типа или всего проекта выполняется вручную либо автоматически при запуске в рантайм формы, типа или проекта.

Чтобы вручную скомпилировать отдельную форму или тип, откройте нужную вкладку, перейдите в меню Проект и выберите Запуск компиляции текущего типа (Ctrl+F5). При компиляции вкладки компилируются также зависимые от нее формы и типы.

Чтобы скомпилировать весь проект, в меню "Проект" выберите команду "Запуск компиляции" (F5).

Чтобы прервать компиляцию, выберите команду Проект → Остановить компиляцию (Ctrl+Break).



Команда отмены появляется в меню только во время компиляции.

Весь процесс, а затем результат компиляции отображаются в области Журнал.

```
Журнал
< Компиляция запущена >
< Компиляция юнита "UI" >
< Время компиляции: 0.009 секунд >
< Компиляция завершилась успешно >
< Компиляция завершена: 1 успешно, 0 неудачно, 0 актуально >
```

Скомпилированный проект сохраняется в папку проекта в файл формата \*.binom.



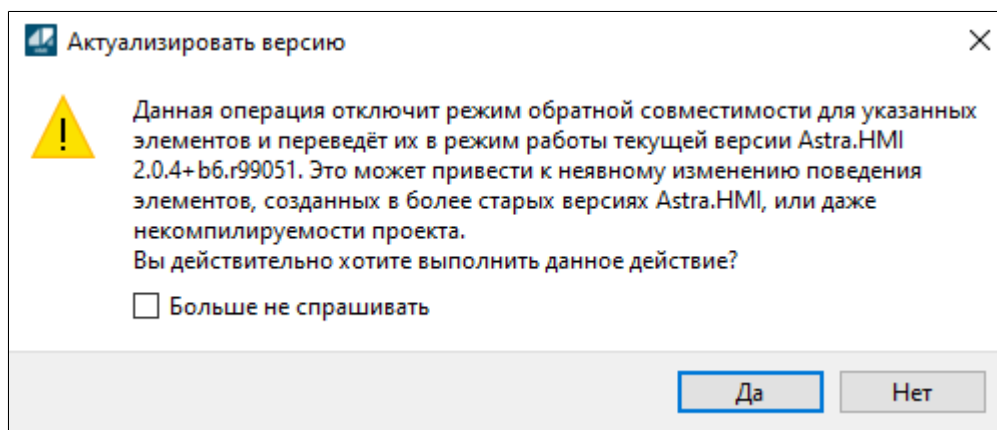
Файл с расширением \*.binom позволяет быстро запустить проект в режиме исполнения без компиляции.



Файл \*.binom привязан к компьютеру, на котором он был создан. После переноса проекта на другой компьютер, скомпилируйте проект.

## Актуализировать версии элементов

Чтобы принудительно актуализировать устаревшие версии элементов проекта выберите в меню Проект команду "актуализировать версии компонентов". Если в настройках проекта установлен флаг [Подтверждать актуализацию версий](#), то откроется окно с подтверждением действия.



## Ссылки на юниты

Команда используется для добавления сторонних .NET-компонентов (.NET Framework и .NET Core). Подробное описание процесса добавления сторонних .NET-компонентов приводится в разделе [Встраивание внешних библиотек .NET](#).

## Обновить ссылки на внешние юниты

Компоненты расширения добавляются в проект Astra.HMI в виде внешних модулей и располагаются в папке externals.



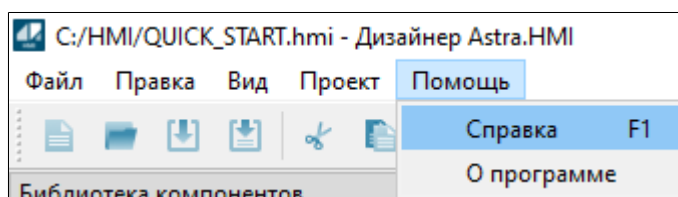
Локальный диск (C:) > HMI > externals				
Имя	Дата изменения	Тип	Размер	
Alarms	16.08.2023 15:28	Папка с файлами		
CommonLib	16.08.2023 15:28	Папка с файлами		
Explorer	16.08.2023 15:28	Папка с файлами		
IntegrityControl	16.08.2023 15:28	Папка с файлами		
PsBase	16.08.2023 14:17	Папка с файлами		
PsDiagn	16.08.2023 14:17	Папка с файлами		
PsTechEE	16.08.2023 14:17	Папка с файлами		
SecurityConfigurator	16.08.2023 15:28	Папка с файлами		
SetPoints	16.08.2023 15:28	Папка с файлами		
Trends	16.08.2023 15:28	Папка с файлами		

После добавления компонентов в данную папку в проекте необходимо выполнить обновление ссылок на внешние юниты, после чего эти компоненты отобразятся на вкладке "Внешние модули".

Внешние модули
> CommonLib
> Alarms
> Explorer
> IntegrityControl
> Trends
> PsBase
> PsDiagn
> PsTechEE
> SecurityConfigurator
> SetPoints

Чтобы обновить ссылки на внешние модули выберите в меню Проект команду Обновить ссылки на внешние юниты.

## 1.1.4.2.2.5. Помощь



Чтобы воспользоваться встроенной справкой в меню Помощь выберите Справка (F1).

Чтобы узнать версию компонента Astra.HMI, Qt и лицензии LGPL выберите вкладку "О программе" контекстного меню.

## 1.1.4.2.3. Панель инструментов



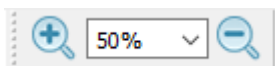
### 1 Основная панель инструментов



Содержит команды:

- › создать новый проект;
- › открыть проект;
- › сохранить;
- › сохранить проект;
- › вырезать;
- › копировать;
- › вставить;
- › отменить действие;
- › повторить действие.

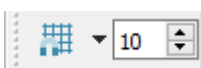
### 2 Панель масштабирования



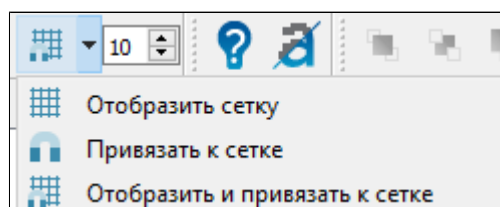
Содержит:

- › команду увеличить;
- › поле для ввода пользовательского значения масштаба;
- › команду уменьшить.

### 3 Панель инструментов



Содержит параметры отображения и привязки к сетке.



## 4 Панель порядка



Позволяет настроить порядок отображения объектов на панели оператора (отображение на переднем/заднем плане).

## 5 Панель выравнивания



Содержит команды:

- › выравнивать объекты (по левому краю, центру, правому краю, верху, вертикальному центру, низу);
- › распределить объекты (по вертикали, горизонтали);
- › установить равенство габаритов объектов (одинаковые ширина, высота или размер).

## 6 Панель настроек сцены



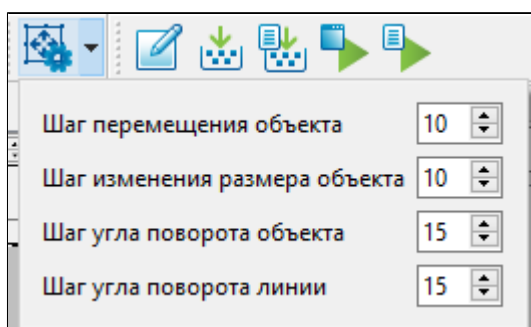
Содержит команды:

- › Отобразить подсказки;
- › Активировать сглаживание изображения.

## 7 Настройка инструментов редактирования



Позволяет настроить параметры работы с инструментами редактирования.



## 8 Панель проекта



Содержит команды:

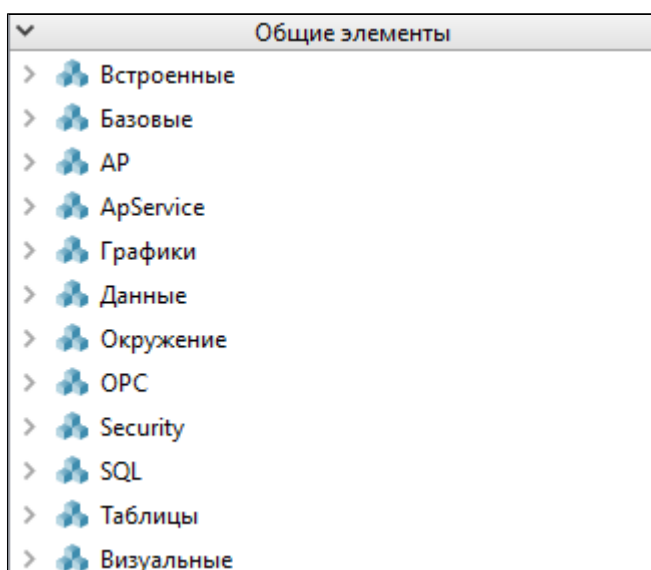
- › Создать тип или глобальный объект;
- › Запуск компиляции;
- › Запуск компиляции текущего типа;
- › Показать главную форму в рантайме;
- › Показать активную вкладку в рантайме.

## 1.1.4.2.4. Общие элементы

Содержит

компоненты:

- › [Встроенные](#)
- › [Базовые](#)
- › [AP](#)
- › [ApService](#)
- › [Графики](#)
- › [Данные](#)
- › [Окружение](#)
- › [OPC](#)
- › [Security](#)
- › [SQL](#)
- › [Таблицы](#)
- › [Визуальные](#)

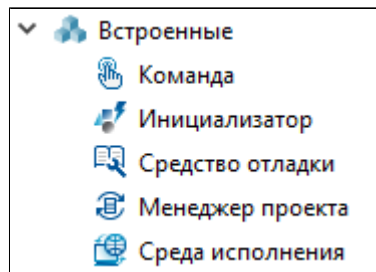


## 1.1.4.2.4.1. Встроенные

Содержит

компоненты:

- > [Команда](#)
- > [Инициализатор](#)
- > [Средство отладки](#)
- > [Менеджер проекта](#)
- > [Среда исполнения](#)



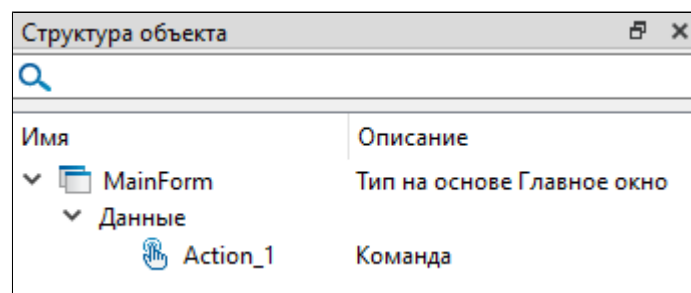
## 1.1.4.2.4.1.1. Команда

Невизуальный компонент позволяет определять пользовательские команды, которые можно многократно вызывать из скриптов проекта. Компонент невизуальный (не отображается на форме) и виден только в области Структура объекта.

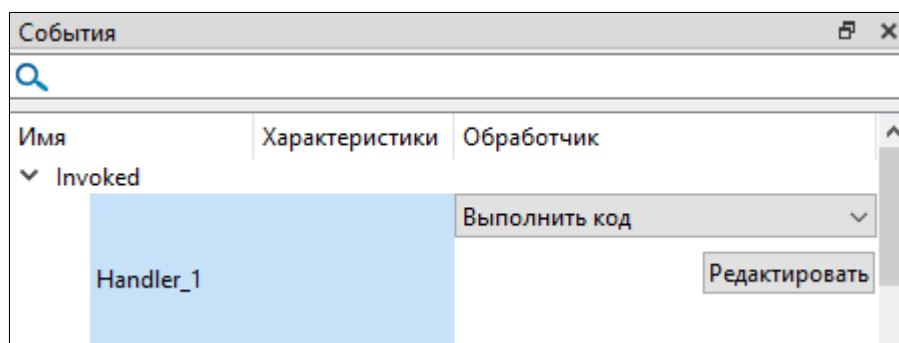
Логика команды определяется в обработчике события `Invoked`.

Чтобы выполнить команду, активируйте свойство Триггер или вызовите функцию `Invoke()`.

Чтобы определить новую команду, добавьте компонент Команда на экранную форму.



Чтобы определить команду, которая будет выполняться, настройте обработчик события `Invoked`.



Чтобы выполнить настроенную команду, вызовите функцию `Invoke()` из любого места проекта.





```
Action_1.Invoke();
```

Альтернативный способ выполнения команды – активация свойства Триггер. Свойство запускает процедуру в обработчике `Invoked`.



```
Action.InvokeTrigger = true;
```



`Invoked` – событие, которое наступает после активации свойства Триггер или вызова функции `Invoke()`.



Из-за того, что выполнение команд находится в общей очереди обработчиков событий, то команды вызываемые из скрипта, выполняются после выполнения основного кода скрипта. Чтобы вызывать исполнение неких процедур синхронно с кодом скрипта, применяйте функции.

## Примеры



```
if(Checkbox_1.State==true)
{
    Action_1.Invoke();
}
```

## 1.1.4.2.4.1.1.1. События

Событие	Описание
<a href="#">Invoked</a>	Активация свойства Триггер или вызова функции Invoke()
<a href="#">EnabledChanged</a>	Изменение свойства Включено

## 1.1.4.2.4.1.1.1.1. Invoked

Активация свойства Триггер или вызова функции Invoke().

## 1.1.4.2.4.1.1.1.2. EnabledChanged

Изменение свойства Включено.

### Параметры

Параметр	Тип	Описание
enabled	bool	Состояние команды: <ul style="list-style-type: none"><li>&gt; true – команда включена</li><li>&gt; false – команда выключена</li></ul>

## 1.1.4.2.4.1.1.2. Функции

Функция	Описание
<a href="#">Invoke</a>	Активация свойства Триггер или вызова функции Invoke()

## 1.1.4.2.4.1.1.2.1. Invoke

Запускает пользовательскую команду, указанную в обработчике события `Invoked`.



```
void Invoke()
```

### 1.1.4.2.4.1.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Включено</a>	Включает/выключает пользовательскую команду
<a href="#">Триггер</a>	Запустить процедуру в обработчике Invoked

## **1.1.4.2.4.1.1.3.1. Отображаемое имя**

Произвольное описание объекта (допускается русскоязычное).



## 1.1.4.2.4.1.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.1.1.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length

### Примеры



```
//Записать в текстовое поле длину массива  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.1.1.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.1.1.3.5. Включено

Включает/выключает пользовательскую команду.



bool Enabled

### Значение

Значение	Описание
true	Команда включена
false	Команда выключена

## 1.1.4.2.4.1.1.3.6. Триггер

Свойство, запускающее процедуру в обработчике `Invoked`.



`bool InvokeTrigger`

### Значение

Значение	Описание
<code>true</code>	Запуск команды
<code>false</code>	Остановка выполнения команды

## 1.1.4.2.4.1.2. Инициализатор

Компонент позволяет определить пользовательские команды, которые выполняются сразу при открытии экранной формы в рантайме. Компонент является невизуальным (не отображается на форме) и виден только в области Структура объекта. Компонент состоит из события `Initialized`. В обработчике события указываются команды, которые должны выполняться сразу после запуска экранной формы.

Отличие компонента Инициализатор от компонента Команда в том, что Команда выполняется уже после запуска экранной формы (если использовать событие `Opened`), а событие Инициализатора выполнится сразу же с открытием экранной формы.

## 1.1.4.2.4.1.2.1. События

Событие	Описание
<a href="#">Initialized</a>	Открытие экранной формы в рантайме

## 1.1.4.2.4.1.2.1.1. Initialized

Открытие экранной формы в режиме Runtime.

### Примеры



//Активировать пользовательскую команду с запуском экранной формы

```
Action.Enabled = true;
```



## 1.1.4.2.4.1.2.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.1.2.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.1.2.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.1.2.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.1.2.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

### **1.1.4.2.4.1.3. Средство отладки**

Компонент позволяет вести журнал времени исполнения кода на языке Astra.Om.

## 1.1.4.2.4.1.3.1. Функции

Функция	Описание
<a href="#">Log</a>	Выводит сообщение в Журнал времени исполнения

## 1.1.4.2.4.1.3.1.1. Log

Выводит сообщение в Журнал времени исполнения.



void Log(variant message)

### Параметры

Параметр	Тип	Описание
message	variant	Текст сообщения

### Примеры



```
//Вывести сообщение о завершении отладки:  
DebugTool.Log("Отладка завершена");
```



## 1.1.4.2.4.1.3.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## **1.1.4.2.4.1.3.2.1. Отображаемое имя**

Произвольное описание объекта (допускается русскоязычное).

## 1.1.4.2.4.1.3.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.1.3.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.1.3.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме исполнения.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.1.4. Менеджер проекта

Компонент позволяет без перезапуска обновлять проект Astra.HMI, работающий в режиме исполнения.

Благодаря типу Менеджер проекта вы можете запустить проект в режим исполнения на одном рабочем месте и дорабатывать проект на другом, обновляя запущенный проект по мере доработок.

Для работы типа Менеджер проекта необходимо подготовить две пустые папки:

- папка для обновлений – в эту папку нужно помещать файлы проекта с изменениями, если вы хотите подгрузить в запущенный проект эти изменения. Тип Менеджер проекта следит за содержимым этой папки и заменяет файлы текущего проекта файлами из нее;
- папка для автосохранения копии запущенного проекта – в эту папку при каждом обновлении проекта автоматически сохраняется копия текущего запущенного проекта для возможности отката изменений.



Чтобы проверить наличие файлов с изменениями, используйте функцию [CheckForUpdates](#).



Чтобы подгружать обновления в запущенный проект, используйте функцию [AsyncReload](#).



Чтобы вернуть проект к исходному виду, используйте функцию [AsyncReloadFromBackup](#).

## 1.1.4.2.4.1.4.1. События

Событие	Описание
<a href="#">ReloadStarted</a>	Запуск перезагрузки (обновления) проекта
<a href="#">Reloaded</a>	Завершение удачной попытки перезагрузки (обновления) проекта
<a href="#">ReloadFailed</a>	Завершение неудачной попытки перезагрузки (обновления) проекта
<a href="#">BackupStarted</a>	Событие начала создания резервной копии
<a href="#">BackupFailed</a>	Событие завершения неудачной попытки создания резервной копии
<a href="#">BackupSucceeded</a>	Событие завершения удачной попытки создания резервной копии

## 1.1.4.2.4.1.4.1.1. ReloadStarted

Запуск перезагрузки (обновления) проекта.



## 1.1.4.2.4.1.4.1.2. BackupFailed

Событие завершения неудачной попытки создания резервной копии.

## 1.1.4.2.4.1.4.1.3. BackupSucceeded

Событие завершения удачной попытки создания резервной копии.

## 1.1.4.2.4.1.4.1.4. BackupStarted

Событие начала создания резервной копии.

## 1.1.4.2.4.1.4.1.5. Reloaded

Завершение удачной попытки перезагрузки (обновления) проекта.

## 1.1.4.2.4.1.4.1.6. ReloadFailed

Завершение неудачной попытки перезагрузки (обновления) проекта.

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

### Примеры



```
//Вывести текст ошибки при неудачной перезагрузке проекта:  
Text_3 += "Failed Reload: " + errorMessage + "\r";
```

## 1.1.4.2.4.1.4.2. Функции

Функция	Описание
<a href="#">AsyncBackup</a>	Сделать резервную копию асинхронно
<a href="#">AsyncReload</a>	Заменяет файлы текущего запущенного проекта на файлы из указанной папки обновлений, компилирует и запускает обновленный проект
<a href="#">AsyncReloadFromBackup</a>	Загружает проект из резервной копии, компилирует и запускает восстановленный проект
<a href="#">CheckForUpdates</a>	Проверяет наличие обновлений проекта в указанной папке

## 1.1.4.2.4.1.4.2.1. AsyncBackup

Сделать резервную копию асинхронно.



```
void AsyncBackup(string PathToBackupDir)
```

### Параметры

Параметр	Тип	Описание
PathToBackupDir	string	Путь для резервного копирования

## 1.1.4.2.4.1.4.2.2. AsyncReload

Заменяет файлы текущего запущенного проекта на файлы из указанной папки обновлений, компилирует и запускает обновленный проект. При перезагрузке проекта также сохраняет копию текущего проекта.



```
void AsyncReload(string PathUpdate, string PathBackup)
```

### Параметры

Параметр	Тип	Описание
PathUpdate	string	Путь к проекту, изменения которого следует загрузить в запущенный проект
PathBackup	string	Путь к папке для автосохранения копии текущего проекта

### Примеры



```
//Подгрузка обновлений проекта:  
ProjectManager.AsyncReload("C:\\Backup\\Update\  
_BackupManager.hmi", "C:\\Backup\\Back");
```

### Подгрузка обновлений проекта

1. Запустите проект в режиме исполнения.
2. Откройте эпроект в Дизайнере Astra.HMI и внесите изменения.
3. Скопируйте все файлы проекта и подложите их в папку обновлений.



Папка может находиться на другом рабочем месте.

Подкладывать файлы в папку обновлений каждый раз после изменения проекта в дизайнере.



4. Для подгрузки обновлений вызывайте функцию `AsyncReload` в запущенном проекте.

## 1.1.4.2.4.1.4.2.3. AsyncReloadFromBackup

Загружает проект из резервной копии, компилирует и запускает восстановленный проект.



```
void AsyncReloadFromBackup(string PathBackup)
```

### Параметры

Параметр	Тип	Описание
PathBackup	string	Путь к папке, из которой следует загрузить резервную копию проекта

### Примеры



```
//Откатить изменения проекта:  
ProjectManager.AsyncReloadFromBackup("C:\\Backup\\Back");
```

## 1.1.4.2.4.1.4.2.4. CheckForUpdates

Проверяет наличие обновлений проекта в указанной папке.



bool CheckForUpdates(pathUpdate)

### Параметры

Параметр	Тип	Описание
PathUpdate	string	Путь к папке, в которой следует проверить обновления

### Возвращаемое значение

Значение	Описание
true	В указанной папке есть проект с расширением .hmi или файл *.binom
false	В указанной папке отсутствуют проект с расширением .hmi и файл *.binom

### Примеры



```
//Проверить изменения проекта  
isUpdated: bool = ProjectManager.CheckForUpdates("C:\\Backup\\  
Update\\_BackupManager.hmi") ? "true" : "false";
```

### 1.1.4.2.4.1.4.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.1.4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.1.4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.1.4.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.1.4.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме исполнения.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.1.5. Среда исполнения

Компонент позволяет получить сведения о среде исполнения Astra.HMI.

## 1.1.4.2.4.1.5.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.1.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.1.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.1.5.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.1.5.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме исполнения.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

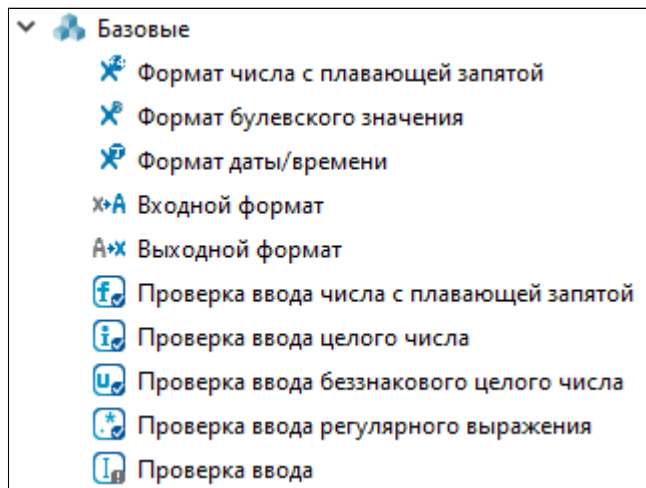


```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2. Базовые

Содержит компоненты:

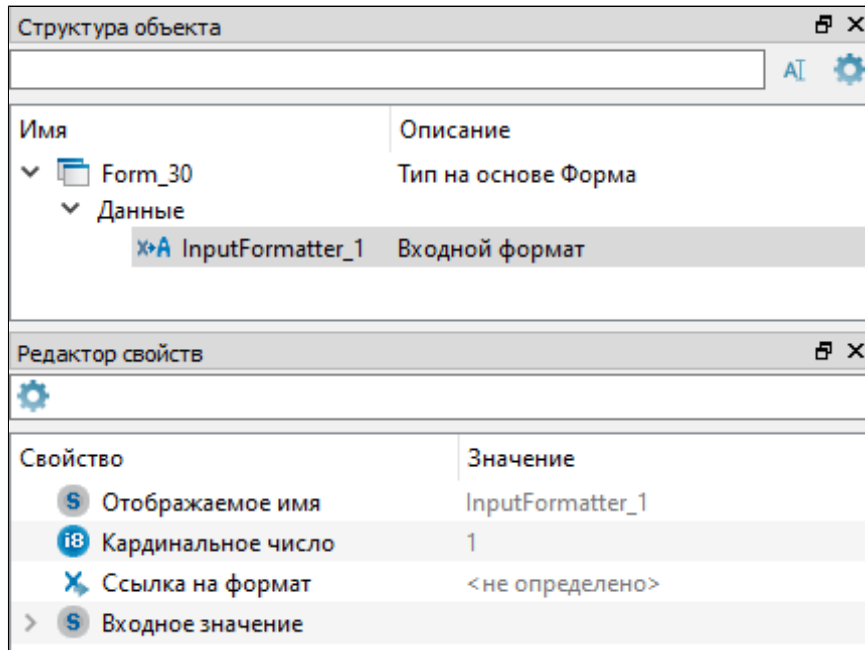
- > [Формат числа с плавающей запятой](#)
- > [Формат булевого значения](#)
- > [Формат даты/времени](#)
- > [Входной формат](#)
- > [Выходной формат](#)
- > [Проверка ввода числа с плавающей запятой](#)
- > [Проверка ввода целого числа](#)
- > [Проверка ввода беззнакового целого числа](#)
- > [Проверка ввода регулярного выражения](#)
- > [Проверка ввода](#)



Компоненты работают в связке с компонентами Входной формат и Выходной формат.

# Конвертация строк в универсальный тип Variant

Чтобы конвертировать строки в значения универсального типа variant, воспользуйтесь компонентом Входной формат. Добавьте на экранную форму невизуальный компонент Входной формат.



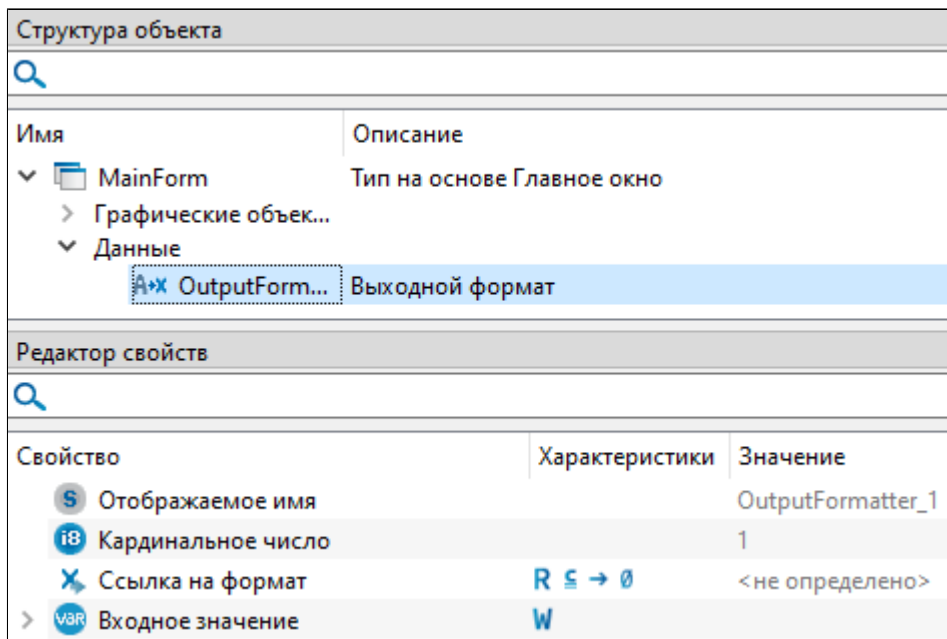
Преобразование выполняется в соответствии с форматом, указанным в свойстве Ссылка на формат.

Свойство	Тип	Описание
Входное значение (InValue)	string	Строка, которую нужно преобразовать к типу variant
Выходное значение (OutValue)	variant	Значение, преобразованное к типу variant



## Конвертация любых типов в строки

Чтобы конвертировать значения любых типов в строковый тип string, используйте компонент Выходной формат. Добавьте на экранную форму компонент Выходной формат. Компонент не визуальный и виден только в области Структура объекта.



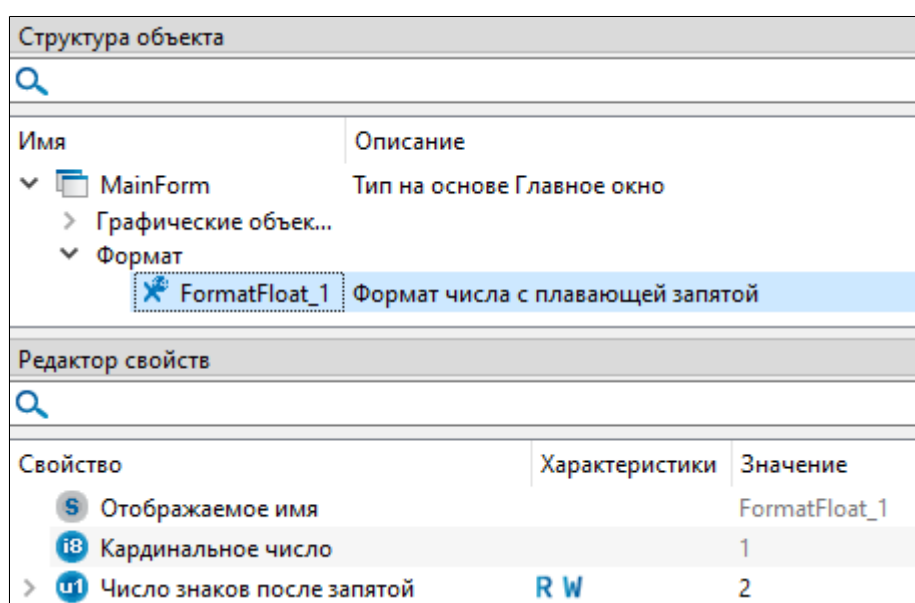
Преобразование выполняются в соответствии с форматом, указанным в свойстве Ссылка на формат.

Свойство	Тип	Описание
Входное значение (InValue)	variant	Значение, которое нужно преобразовать к типу string
Выходное значение (OutValue)	string	Значение, преобразованное к типу string

## 1.1.4.2.4.2.1. Формат числа с плавающей запятой

Формат представления чисел с плавающей точкой.

Компонент служит для представления чисел с плавающей точкой с нужной точностью. Компонент невидимый и виден только в области Структура объекта. На рисунке ниже показано, как на форму был добавлен компонент Формат числа с плавающей запятой.



## 1.1.4.2.4.2.1.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Число знаков после запятой</a>	Точность числа с плавающей запятой

## 1.1.4.2.4.2.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.1.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.1.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.1.1.5. Число знаков после запятой

Точность числа с плавающей запятой.



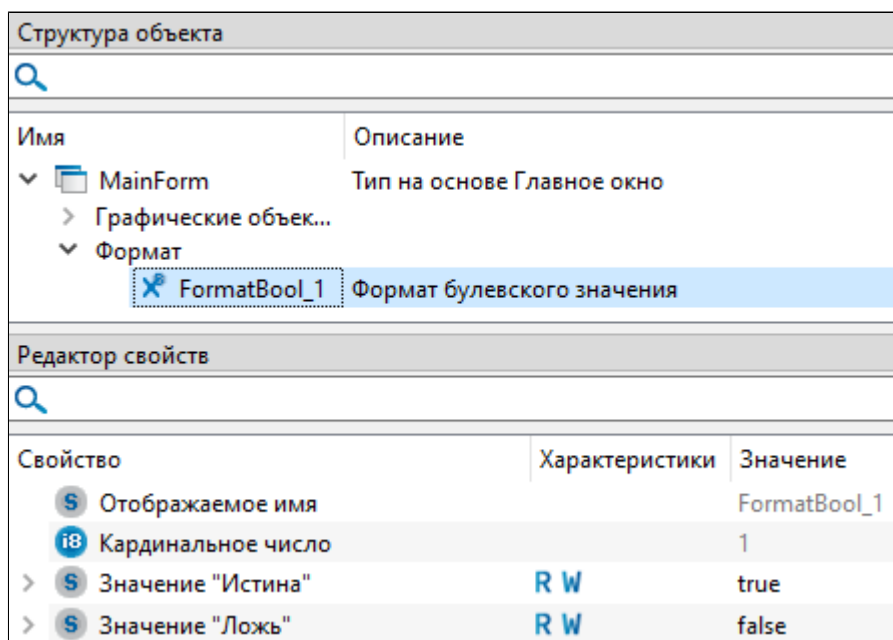
uint1 DigitsCount



## 1.1.4.2.4.2.2. Формат булевского значения

Формат представления булевых значений.

Компонент служит для представления логических значений (true/false) в альтернативном виде. Компонент невидимый и виден только в области Структура объекта. На рисунке ниже показано, как на форму был добавлен компонент Формат булевского значения.



Чтобы определить альтернативное строковое представление для логических значений, установите значения свойств.

Свойство	Тип	Описание
Значение "Истина"	string	Строка, которая соответствует значению true
Значение "Ложь"	string	Строка, которая соответствует значению false

## 1.1.4.2.4.2.2.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Значение "Истина"</a>	Строка, которая соответствует значению True
<a href="#">Значение "Ложь"</a>	Строка, которая соответствует значению False

## 1.1.4.2.4.2.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.2.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.2.1.5. Значение "Истина"

Строка, которая соответствует значению True.



string TrueString

## 1.1.4.2.4.2.2.1.6. Значение "Ложь"

Строка, которая соответствует значению False.



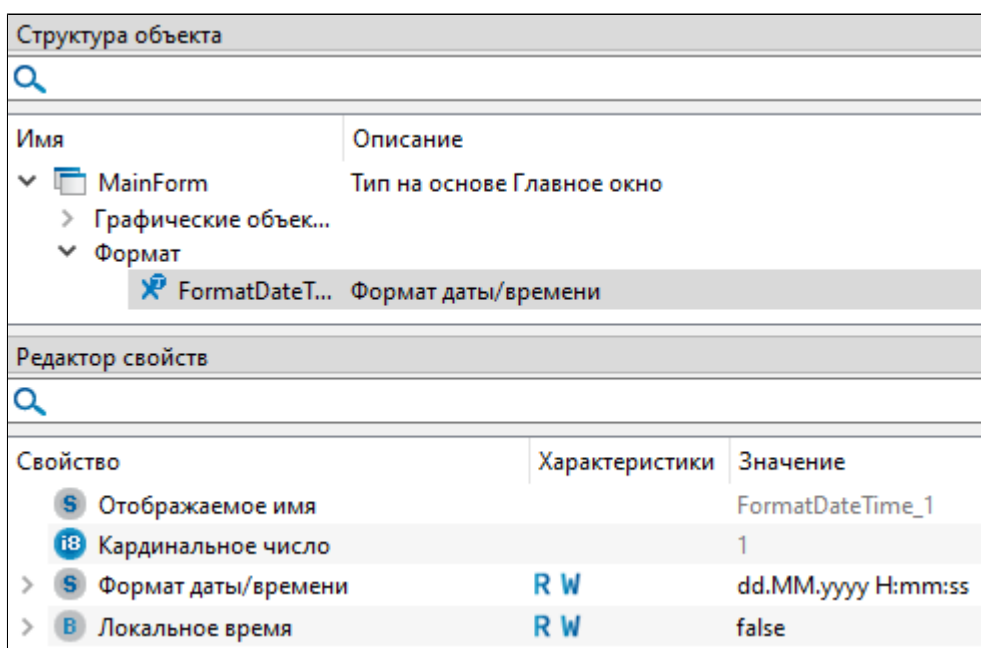
string FalseString



## 1.1.4.2.4.2.3. Формат даты/времени

Формат представления даты и времени.

Компонент служит для представления даты и времени в заданном формате. Компонент невизуальный и виден только в области Структура объекта. На рисунке ниже показано, как на форму был добавлен компонент Формат даты/времени.



Для настройки формата отображения даты и времени установите свойства компонента:

Свойство	Тип	Описание
Формат даты/времени	string	Формат представления метки времени. Доступны обозначения: dd – дни, MM – месяцы, yyyy – год, H – часы, mm – минуты, ss – секунды
Локальное время	bool	Если свойство активно, то оригинальная метка времени заменяется локальным временем

## 1.1.4.2.4.2.3.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Формат даты/времени</a>	Формат представления метки времени
<a href="#">Локальное время</a>	Использование локального времени в качестве оригинальной метки

## 1.1.4.2.4.2.3.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.3.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.3.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.3.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.3.1.5. Формат даты/времени

Формат представления метки времени.

Доступны обозначения: dd - дни, ММ-месяцы, уууу - год, Н - часы, mm - минуты, ss - секунды.



string DateTimeFormat

## 1.1.4.2.4.2.3.1.6. Локальное время

Использование локального времени в качестве оригинальной метки.



bool LocalTime

### Значение

Значение	Описание
true	Оригинальная метка времени заменяется локальным временем
false	Оригинальная метка времени



## 1.1.4.2.4.2.4. Входной формат

Преобразователь строковых значений (тип string) к универсальному типу variant. Преобразование выполняется в соответствии с форматом `om.automation.controls`.

The screenshot shows the IDE interface with two panels. The top panel, titled "Структура объекта" (Object Structure), displays a tree view of the project structure. The bottom panel, titled "Редактор свойств" (Properties Editor), shows the properties of the selected object, InputFormatter\_1.

**Структура объекта**

Имя	Описание
MainForm	Тип на основе Главного окна
Графические объекты	
Данные	
InputFormatter_1	Входной формат

**Редактор свойств**

Свойство	Характеристики	Значение
Отображаемое имя		InputFormatter_1
Кардинальное число		1
Ссылка на формат	R ≤ → 0	<не определено>
Входное значение	W	

## 1.1.4.2.4.2.4.1. События

Событие	Описание
<a href="#">ValueChanged</a>	Изменение входного значения преобразователя формата
<a href="#">FormatErrorChanged</a>	Ошибка преобразования к какому-либо формату

## 1.1.4.2.4.2.4.1.1. ValueChanged

Изменение входного значения преобразователя формата.

### Параметры

Параметр	Тип	Описание
value	variant	Значение, преобразованное к типу variant

## 1.1.4.2.4.2.4.1.2. FormatErrorChanged

Ошибка преобразования к какому-либо формату.

### Параметры

Параметр	Тип	Описание
hasError	bool	Наличие ошибки форматирования: <ul style="list-style-type: none"><li>› true – ошибка есть</li><li>› false – ошибки нет</li></ul>

## 1.1.4.2.4.2.4.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ссылка на формат</a>	Определяет формат для преобразования входного значения
<a href="#">Входное значение</a>	Строка, которую нужно преобразовать к типу variant
<a href="#">Ошибка форматирования</a>	Свойство содержит текстовое описание ошибки преобразования
<a href="#">Признак ошибки форматирования</a>	Индикатор возникновения ошибки в процессе преобразования формата
<a href="#">Выходное значение</a>	Значение, преобразованное к типу variant

## 1.1.4.2.4.2.4.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.4.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.4.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.4.2.5. Ссылка на формат

Ссылка на экземпляр одного из типов `om.automation.controls`. Определяет формат для преобразования входного значения.

## 1.1.4.2.4.2.4.2.6. Входное значение

Строка, которую нужно преобразовать к типу variant.



string InValue

### Примеры



```
//Подать на вход преобразователю значение сигнала (строку):  
InputFormatter.InValue = Opcltem.Value;
```

## 1.1.4.2.4.2.4.2.7. Ошибка форматирования

Свойство содержит текстовое описание ошибки преобразования.



string FormatErrorString



Доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать текст ошибки преобразования в текстовое поле:  
TextEdit.Text = InputFormatter.FormatErrorString;
```

## 1.1.4.2.4.2.4.2.8. форматирования

## Признак

## ошибки

Индикатор возникновения ошибки в процессе преобразования формата.



bool FormatError



Доступно только для чтения в режиме рантайма.

## Значение

Значение	Описание
true	Ошибка преобразования
false	Преобразование прошло без ошибок

## 1.1.4.2.4.2.4.2.9. Выходное значение

Значение, преобразованное к типу variant.



variant OutValue



Доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать преобразованное значение в поле ввода  
TextEdit.Text = InputFormatter.OutValue;
```

## 1.1.4.2.4.2.5. Выходной формат

Преобразователь входного значения универсального типа variant к строковому типу (string). Преобразование выполняются в соответствии с форматом om.automation.controls.

The screenshot displays two panels from a software development environment. The top panel, titled 'Структура объекта' (Object Structure), shows a tree view of the project hierarchy. The 'OutputFormatter\_1' object is selected and highlighted in blue. The bottom panel, titled 'Редактор свойств' (Property Editor), shows the properties of the selected object in a table format.

Имя	Описание
MainForm	Тип на основе Главного окна
Графические объекты	
Данные	
OutputFormatter_1	Выходной формат

Свойство	Характеристики	Значение
Отображаемое имя		OutputFormatter_1
Кардинальное число		1
Ссылка на формат	R ≤ → 0	<не определено>
Входное значение	W	

## 1.1.4.2.4.2.5.1. События

Событие	Описание
<a href="#">ValueChanged</a>	Изменение входного значения преобразователя формата
<a href="#">FormatErrorChanged</a>	Ошибка преобразования к какому-либо формату



## 1.1.4.2.4.2.5.1.1. ValueChanged

Изменение значения на выходе преобразователя формата.

### Параметры

Параметр	Тип	Описание
value	variant	Значение, преобразованное к типу string

## 1.1.4.2.4.2.5.1.2. FormatErrorChanged

Ошибка преобразования к какому-либо формату.

### Параметры

Параметр	Тип	Описание
hasError	bool	Наличие ошибки форматирования: <ul style="list-style-type: none"><li>› true – ошибка есть</li><li>› false – ошибки нет</li></ul>

## 1.1.4.2.4.2.5.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ссылка на формат</a>	Определяет формат для преобразования входного значения
<a href="#">Входное значение</a>	Строка, которую нужно преобразовать к типу variant
<a href="#">Ошибка форматирования</a>	Свойство содержит текстовое описание ошибки преобразования
<a href="#">Признак ошибки форматирования</a>	Индикатор возникновения ошибки в процессе преобразования формата
<a href="#">Выходное значение</a>	Значение, преобразованное к типу variant

## 1.1.4.2.4.2.5.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.5.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.5.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.5.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.5.2.5. Ссылка на формат

Ссылка на экземпляр одного из типов `om.automation.controls`. Определяет формат для преобразования входного значения.



## 1.1.4.2.4.2.5.2.6. Входное значение

Строка, которую нужно преобразовать к типу variant.



variant InValue

### Примеры



```
//Подать на вход преобразователю значение текстовой области:  
OutputFormatter.InValue = TextEdit.Text;
```

## 1.1.4.2.4.2.5.2.7. Ошибка форматирования

Свойство содержит текстовое описание ошибки преобразования.



string FormatErrorString

### Примеры



```
//Записать текст ошибки преобразования в текстовое поле:  
TextEdit.Text = OutputFormatter.FormatErrorString;
```

## 1.1.4.2.4.2.5.2.8. форматирования

## Признак

## ошибки

Индикатор возникновения ошибки в процессе преобразования формата.



bool FormatError

## Значение

Значение	Описание
true	Ошибка преобразования
false	Преобразование прошло без ошибок



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.5.2.9. Выходное значение

Значение, преобразованное к типу string.



variant OutValue



Доступно только для чтения в режиме рантайма.

### Примеры



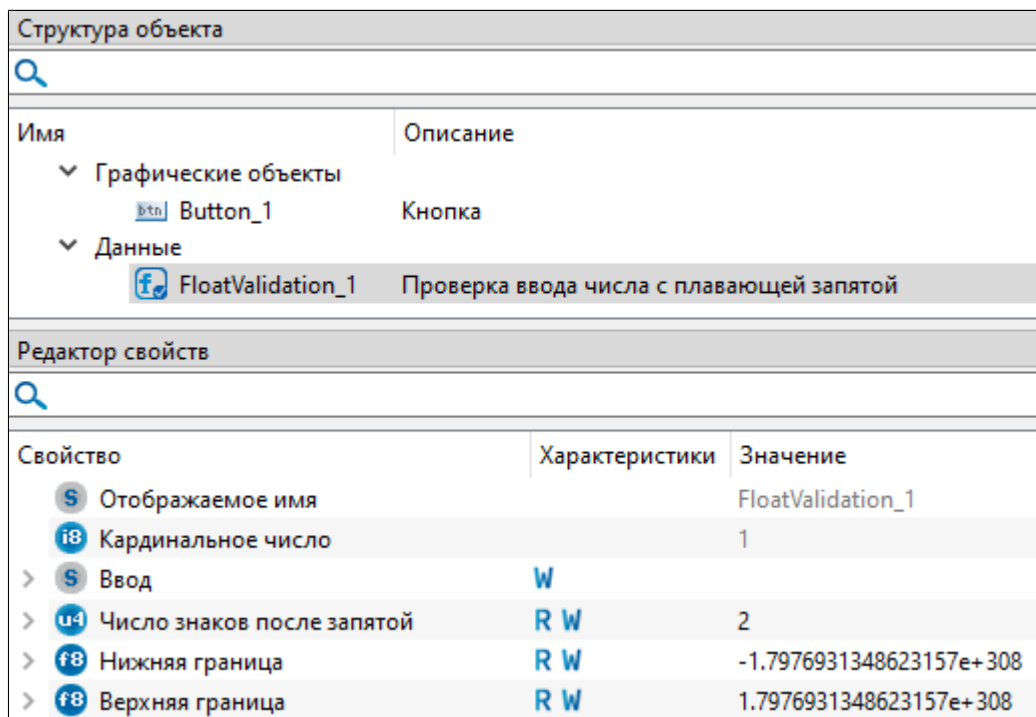
```
//Записать преобразованное значение в поле ввода:  
TextEdit.Text = OutputFormatter.OutValue;
```

## 1.1.4.2.4.2.6. Проверка ввода числа с плавающей запятой

Компонент для определения условий валидации чисел с плавающей запятой.

Для проверки чисел с плавающей запятой на соответствие заданным условиям выполните следующие действия:

1. Определите условия валидации: добавьте на рабочую область компонент библиотеки Проверка ввода числа с плавающей запятой (из группы `om.automation.controls`) и укажите значения свойств Число знаков после запятой, Нижняя граница, Верхняя граница.



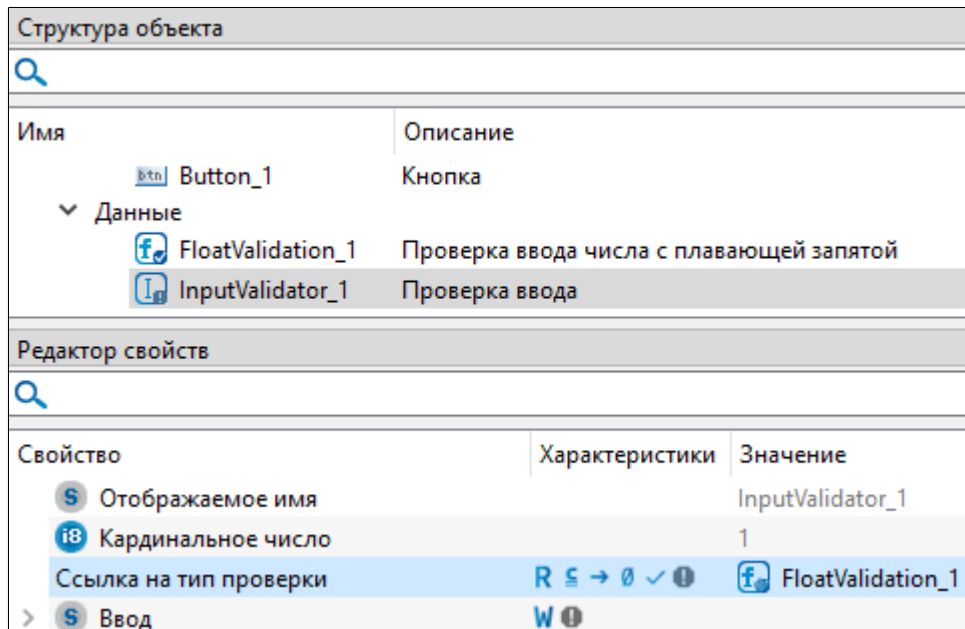
Структура объекта

Имя	Описание
Графические объекты	
Button_1	Кнопка
Данные	
FloatValidation_1	Проверка ввода числа с плавающей запятой


Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		FloatValidation_1
Кардинальное число		1
Ввод	W	
Число знаков после запятой	R W	2
Нижняя граница	R W	-1.7976931348623157e+308
Верхняя граница	R W	1.7976931348623157e+308


2. Добавьте на рабочую область компонент библиотеки Проверка ввода и сошлитесь в свойстве Ссылка на тип проверки на условие валидации, созданное в предыдущем шаге.



3. Запишите значение, которое следует валидировать, в свойство Input:


 InputValidator.Input = "1.33";

4. Получите результат валидации из свойства Result:

 Text\_1.Text = String.ToString(InputValidator.Result);

Свойство Result может принимать следующие условные коды:

- › 0 – значение не валидировано;
- › 1 – значение валидировано частично;
- › 2 – значение валидировано полностью.

 Если требуется реакция на каждую смену результата валидации, то воспользуйтесь специальным событием ResultChanged.

## 1.1.4.2.4.2.6.1. События

Событие	Описание
<a href="#">ResultChanged</a>	Изменение входного значения преобразователя формата

## 1.1.4.2.4.2.6.1.1. ResultChanged

Смена кода результата валидации.

### Параметры

Параметр	Тип	Описание
result	int4	Результат валидации в виде кода: <ul style="list-style-type: none"><li>› 0 – значение не валидировано</li><li>› 1 – значение валидировано частично</li><li>› 2 – значение валидировано полностью</li></ul>



## 1.1.4.2.4.2.6.2. Функции

Функция	Описание
<a href="#">Validate</a>	Проверка ввода

## 1.1.4.2.4.2.6.2.1. Validate

Выполняет проверку ввода.



void Validate()

## 1.1.4.2.4.2.6.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ввод</a>	Ввод значения, которое следует валидировать
<a href="#">Число знаков после запятой</a>	Точность числа с плавающей запятой
<a href="#">Нижняя граница</a>	Минимальное значение, при котором возможна валидация
<a href="#">Верхняя граница</a>	Максимальное значение, при котором возможна валидация
<a href="#">ValidDigitsCount</a>	Результат проверки на соответствие количеству цифр после запятой
<a href="#">ValidRange</a>	Результат проверки на вхождение числа в заданный диапазон
<a href="#">ValidTop</a>	Результат проверки на превышение указанного диапазона
<a href="#">ValidBottom</a>	Результат проверки на то, что число меньше указанного диапазона
<a href="#">Result</a>	Результат проверки
<a href="#">ResultString</a>	Текстовое описание результата проверки

## 1.1.4.2.4.2.6.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.6.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.6.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.6.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.6.3.5. Ввод

Строка, которую следует валидировать по указанным условиям.



string Input

### Примеры



```
//Проверить адрес электронной почты:  
InputValidator.Input = mymail@domain.com;
```



## 1.1.4.2.4.2.6.3.6. Число знаков после запятой

Точность числа с плавающей запятой.



uint4 DigitsCount

## 1.1.4.2.4.2.6.3.7. Нижняя граница

Минимальное значение, при котором возможна валидация.



float8 Bottom

## 1.1.4.2.4.2.6.3.8. Верхняя граница

Максимальное значение, при котором возможна валидация.



float8 Top

## 1.1.4.2.4.2.6.3.9. ValidDigitsCount

Результат проверки на соответствие количеству цифр после запятой.



bool ValidDigitsCount



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.6.3.10. ValidRange

Результат проверки на вхождение числа в заданный диапазон.



bool ValidRange



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.6.3.11. ValidTop

Результат проверки на превышение указанного диапазона.



bool ValidTop



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.6.3.12. ValidBottom

Результат проверки на то, что число меньше указанного диапазона.



bool ValidBottom



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.6.3.13. Result

Результат проверки.



int4 Result



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.2.6.3.14. ResultString

Текстовое описание результата проверки.



string ResultString



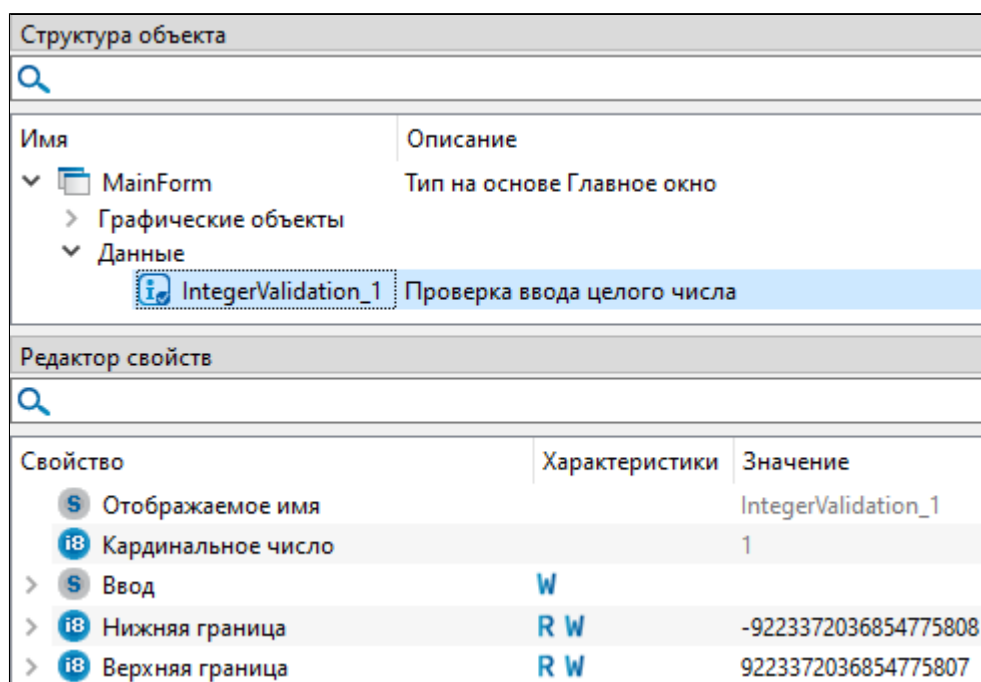
Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.7. Проверка ввода целого числа

Компонент для определения условий валидации целочисленных значений.

Для проверки целочисленных значений на соответствие заданным условиям выполните следующие действия:

1. Определите условия валидации: добавьте на рабочую область компонент библиотеки Проверка ввода целого числа (из группы `om.automation.controls`) и укажите значения свойств Нижняя граница и Верхняя граница.



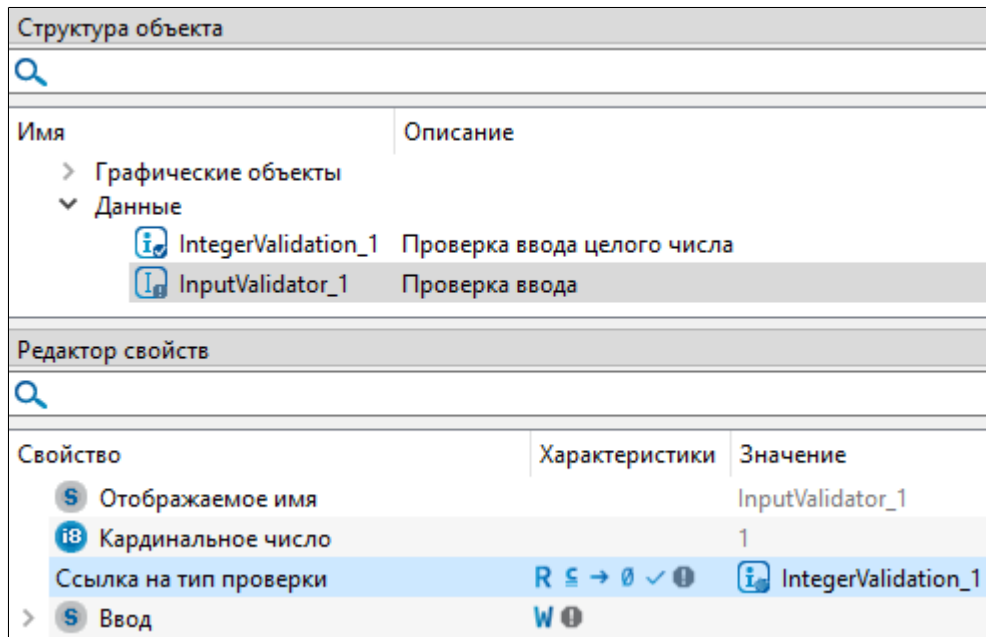
Структура объекта

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Данные	
IntegerValidation_1	Проверка ввода целого числа

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		IntegerValidation_1
Кардинальное число		1
Ввод	W	
Нижняя граница	R W	-9223372036854775808
Верхняя граница	R W	9223372036854775807

2. Добавьте на рабочую область компонент библиотеки Проверка ввода и сошлитесь в свойстве Ссылка на тип проверки на условие валидации, созданное в предыдущем шаге.



3. Запишите значение, которое следует валидировать, в свойство Input:



```
InputValidator.Input = "15";
```

4. Получите результат валидации из свойства Result:



```
Text_1.Text = String.ToString(InputValidator.Result);
```

Свойство Result может принимать следующие условные коды:

- > 0 – значение не валидировано;
- > 1 – значение валидировано частично;
- > 2 – значение валидировано полностью.



Если требуется реакция на каждую смену результата валидации, то воспользуйтесь специальным событием ResultChanged.

## 1.1.4.2.4.2.7.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ввод</a>	Ввод значения, которое следует валидировать
<a href="#">Нижняя граница</a>	Минимальное значение, при котором возможна валидация
<a href="#">Верхняя граница</a>	Максимальное значение, при котором возможна валидация
<a href="#">ValidRange</a>	Результат проверки на вхождение числа в заданный диапазон
<a href="#">ValidTop</a>	Результат проверки на превышение указанного диапазона
<a href="#">ValidBottom</a>	Результат проверки на то, что число меньше указанного диапазона
<a href="#">Result</a>	Результат проверки
<a href="#">ResultString</a>	Текстовое описание результата проверки

## 1.1.4.2.4.2.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.7.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.7.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.2.7.1.5. Ввод

Строка, которую следует валидировать по указанным условиям.



string Input

### Примеры



```
//Проверить адрес электронной почты:  
InputValidator.Input = mymail@domain.com;
```

## 1.1.4.2.4.2.7.1.6. Нижняя граница

Минимальное значение, при котором возможна валидация.



int8 Bottom

## 1.1.4.2.4.2.7.1.7. Верхняя граница

Максимальное значение, при котором возможна валидация.



int8 Top

## 1.1.4.2.4.2.7.1.8. ValidRange

Результат проверки на вхождение числа в заданный диапазон.



bool ValidRange



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.7.1.9. ValidTop

Результат проверки на превышение указанного диапазона



bool ValidTop



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.7.1.10. ValidBottom

Результат проверки на то, что число меньше указанного диапазона.



bool ValidBottom



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.7.1.11. Result

Результат проверки.



int4 Result



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.7.1.12. ResultString

Текстовое описание результата проверки.



string ResultString



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.2.7.2. События

Событие	Описание
<a href="#">ResultChanged</a>	Изменение входного значения преобразователя формата

## 1.1.4.2.4.2.7.2.1. ResultChanged

Смена кода результата валидации.

### Параметры

Параметр	Тип	Описание
result	int4	Результат валидации в виде кода: <ul style="list-style-type: none"><li>› 0 - значение не валидировано</li><li>› 1- значение валидировано частично</li><li>› 2 - значение валидировано полностью</li></ul>

### 1.1.4.2.4.2.7.3. Функции

Функция	Описание
<a href="#">Validate</a>	Проверка ввода

## 1.1.4.2.4.2.7.3.1. Validate

Выполняет проверку ввода.



void Validate()

## 1.1.4.2.4.2.8. Проверка ввода беззнакового целого числа

Компонент для определения условий валидации целочисленных беззнаковых значений. Свойства компонента аналогичны свойствам компонента Проверка ввода целого числа, но у свойств Нижняя граница и Верхняя граница тип uint4 вместо int4.

Для проверки целочисленных беззнаковых значений на соответствие заданным условиям выполните следующие действия:

1. Определите условия валидации: добавьте на рабочую область компонент библиотеки Проверка ввода беззнакового целого числа (из группы `om.automation.controls`) и укажите значения свойств Нижняя граница и Верхняя граница.

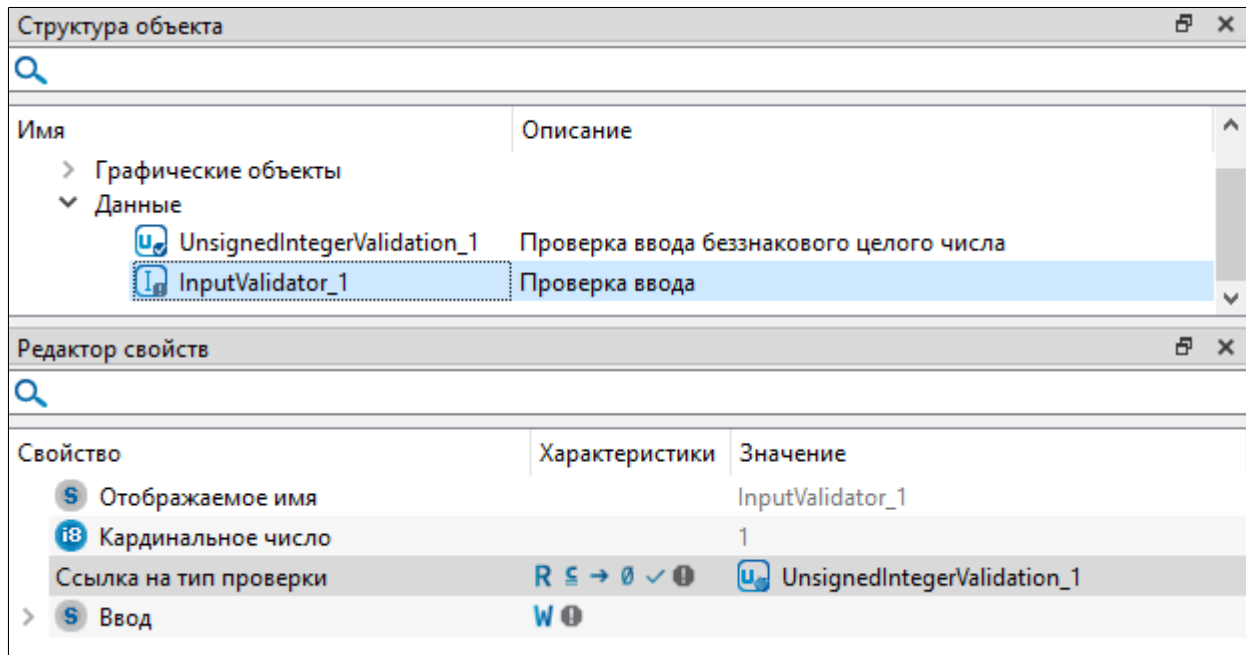
The screenshot shows two panels in a software development environment. The top panel, titled "Структура объекта" (Object Structure), contains a search bar and a tree view. The tree view is expanded to show "Данные" (Data) with a sub-item "UnsignedIntegerValidation\_1" selected. The bottom panel, titled "Редактор свойств" (Property Editor), shows a table of properties for the selected component.

Имя	Описание
Графические объекты	
btn Button_1	Кнопка
Данные	
UnsignedIntegerValidation_1	Проверка ввода беззнакового целого числа


  

Свойство	Характеристики	Значение
Отображаемое имя		UnsignedIntegerValidation_1
Кардинальное число		1
Ввод	W	
Нижняя граница	R W	0
Верхняя граница	R W	18446744073709551615

2. Добавьте на рабочую область компонент библиотеки Проверка ввода и сошлитесь в свойстве Ссылка на тип проверки на условие валидации, созданное в предыдущем шаге.



3. Запишите значение, которое следует валидировать, в свойство Input:


 InputValidator.Input = "15";

4. Получите результат валидации из свойства Result:

 Text\_1.Text = String.ToString(InputValidator.Result);

Свойство Result может принимать следующие условные коды:

- 0 – значение не валидировано;
- 1 – значение валидировано частично;
- 2 – значение валидировано полностью.

 Если требуется реакция на каждую смену результата валидации, то воспользуйтесь специальным событием ResultChanged.

## 1.1.4.2.4.2.8.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ввод</a>	Ввод значения, которое следует валидировать
<a href="#">Нижняя граница</a>	Минимальное значение, при котором возможна валидация
<a href="#">Верхняя граница</a>	Максимальное значение, при котором возможна валидация
<a href="#">ValidRange</a>	Результат проверки на вхождение числа в заданный диапазон
<a href="#">ValidTop</a>	Результат проверки на превышение указанного диапазона
<a href="#">ValidBottom</a>	Результат проверки на то, что число меньше указанного диапазона
<a href="#">Result</a>	Результат проверки
<a href="#">ResultString</a>	Текстовое описание результата проверки

## 1.1.4.2.4.2.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.2.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.8.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.8.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.8.1.5. Ввод

Строка, которую следует валидировать по указанным условиям.



string Input

### Примеры



```
//Проверить адрес электронной почты:  
InputValidator.Input = mymail@domain.com;
```

## 1.1.4.2.4.2.8.1.6. Нижняя граница

Минимальное значение, при котором возможна валидация.



uint8 Bottom

## 1.1.4.2.4.2.8.1.7. Верхняя граница

Максимальное значение, при котором возможна валидация.



uint8 Top

## 1.1.4.2.4.2.8.1.8. ValidRange

Результат проверки на вхождение числа в заданный диапазон.



bool ValidRange



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.8.1.9. ValidTop

Результат проверки на превышение указанного диапазона



bool ValidTop



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.2.8.1.10. ValidBottom

Результат проверки на то, что число меньше указанного диапазона.



bool ValidBottom



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.8.1.11. Result

Результат проверки.



int4 Result



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.8.1.12. ResultString

Текстовое описание результата проверки.



string ResultString



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.8.2. События

Событие	Описание
<a href="#">ResultChanged</a>	Изменение входного значения преобразователя формата

## 1.1.4.2.4.2.8.2.1. ResultChanged

Смена кода результата валидации.

### Параметры

Параметр	Тип	Описание
result	int4	Результат валидации в виде кода: <ul style="list-style-type: none"><li>› 0 – значение не валидировано</li><li>› 1 – значение валидировано частично</li><li>› 2 – значение валидировано полностью</li></ul>

### 1.1.4.2.4.2.8.3. Функции

Функция	Описание
<a href="#">Validate</a>	Проверка ввода

## 1.1.4.2.4.2.8.3.1. Validate

Выполняет проверку ввода.



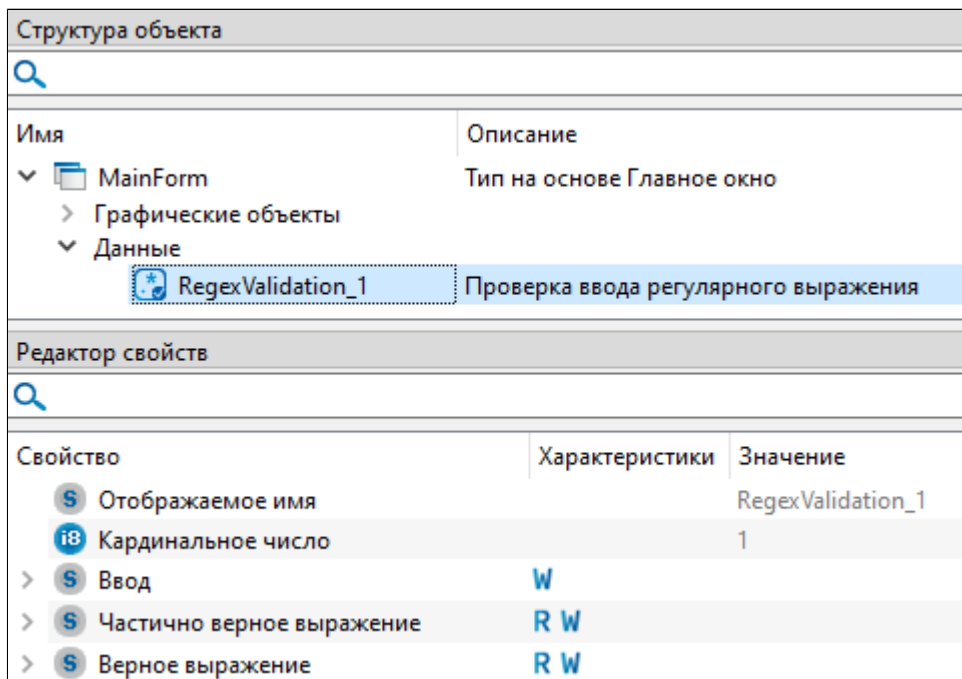
void Validate()

## 1.1.4.2.4.2.9. Проверка ввода регулярного выражения

Компонент для определения условий валидации по регулярному выражению.

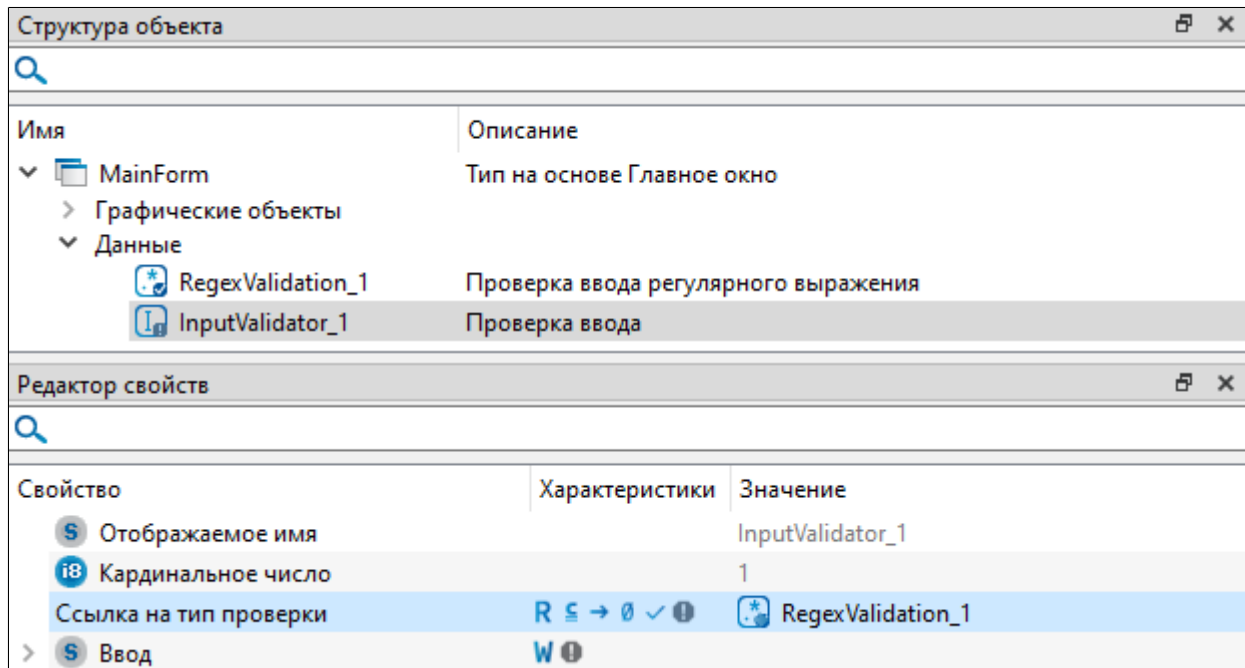
Для проверки строк на соответствие регулярному выражению выполните следующие действия:

1. Определите условия валидации: добавьте на рабочую область компонент библиотеки Проверка ввода регулярного выражения (из группы `om.automation.controls`) и укажите значения свойств Частично верное выражение и Верное выражение. На рисунке ниже показано регулярное выражение для проверки адреса электронной почты.



2. Добавьте на рабочую область компонент библиотеки Проверка ввода и сошлитесь в свойстве Ссылка на тип проверки на условие валидации, созданное в предыдущем шаге.





3. Запишите значение, которое следует валидировать, в свойство Input:



```
InputValidator.Input = "mymail@domain.com";
```

4. Получите результат валидации из свойства Result:



```
Text_1.Text = String.ToString(InputValidator.Result);
```

Свойство Result может принимать следующие условные коды:

- › 0 – значение не валидировано;
- › 1 – значение валидировано частично;
- › 2 – значение валидировано полностью.



Если требуется реакция на каждую смену результата валидации, то воспользуйтесь специальным событием ResultChanged.

## 1.1.4.2.4.2.9.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ввод</a>	Ввод значения, которое следует валидировать
<a href="#">Частично верное выражение</a>	Регулярное выражение, значение которого валидировано частично
<a href="#">Верное выражение</a>	Регулярное выражение, значение которого валидировано полностью
<a href="#">Result</a>	Результат проверки
<a href="#">ResultString</a>	Текстовое описание результата проверки

## 1.1.4.2.4.2.9.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.9.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.9.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.9.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.9.1.5. Ввод

Строка, которую следует валидировать по указанным условиям.



string Input

### Примеры



```
//Проверить адрес электронной почты:  
InputValidator.Input = mymail@domain.com;
```

## 1.1.4.2.4.2.9.1.6. Частично верное выражение

Регулярное выражение, при удовлетворении которого, результирующий код валидации (свойство Result компонента Проверка ввода) будет равен 1 (значение валидировано частично).



string PartialValidPattern

### Примеры



//Записать в текстовое поле регулярное выражение для частичной валидации:

```
TextField.Text = ValidationRegex.PartialValidPattern = "^[a-z0-9]+@[a-z0-9]+\.\w{2,6}$";
```



## 1.1.4.2.4.2.9.1.7. Верное выражение

Регулярное выражение, при удовлетворении которого, результирующий код валидации (свойство Result компонента Проверка ввода) будет равен 2 (значение валидировано полностью).



string ValidPattern

### Примеры



//Записать в текстовое поле регулярное выражение для полной валидации:

```
TextField.Text = ValidationRegex.ValidPattern = "[a-z0-9.-]+@[a-z0-9.-]+\\.\\w{2,6}$";
```

## 1.1.4.2.4.2.9.1.8. Result

Результат проверки.



int4 Result



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.9.1.9. ResultString

Текстовое описание результата проверки.



string ResultString



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.2.9.2. События

Событие	Описание
<a href="#">ResultChanged</a>	Изменение входного значения преобразователя формата

## 1.1.4.2.4.2.9.2.1. ResultChanged

Смена кода результата валидации.

### Параметры

Параметр	Тип	Описание
result	int4	Результат валидации в виде кода: <ul style="list-style-type: none"><li>› 0 – значение не валидировано</li><li>› 1 – значение валидировано частично</li><li>› 2 – значение валидировано полностью</li></ul>

## 1.1.4.2.4.2.9.3. Функции

Функция	Описание
<a href="#">Validate</a>	Проверка ввода

## 1.1.4.2.4.2.9.3.1. Validate

Выполняет проверку ввода.



void Validate()

## 1.1.4.2.4.2.10. Проверка ввода

Компонент для проверки различных типов данных на соответствие заданным условиям и ограничениям.



## 1.1.4.2.4.2.10.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ссылка на тип проверки</a>	Ссылка на экземпляр одного из типов-валидаторов <code>om.automation.controls</code>
<a href="#">Ввод</a>	Строка, которую следует валидировать по указанным условиям
<a href="#">Result</a>	Результат проверки

## 1.1.4.2.4.2.10.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.2.10.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.2.10.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.2.10.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.2.10.1.5. Ссылка на тип проверки

Ссылка на экземпляр одного из типов-валидаторов `om.automation.controls`.  
Определяет формат и условия валидации.



ValidationType

## 1.1.4.2.4.2.10.1.6. Ввод

Строка, которую следует валидировать по указанным условиям.



string Input

### Примеры



```
//Проверить адрес электронной почты:  
InputValidator.Input = mymail@domain.com;
```

## 1.1.4.2.4.2.10.1.7. Result

Результат проверки.



int4 Result



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.2.10.2. События

Событие	Описание
<a href="#">ResultChanged</a>	Изменение входного значения преобразователя формата

## 1.1.4.2.4.2.10.2.1. ResultChanged























Смена кода результата валидации.

### Параметры

Параметр	Тип	Описание
result	int4	Результат валидации в виде кода: <ul style="list-style-type: none"><li>➤ 0 – значение не валидировано</li><li>➤ 1 – значение валидировано частично</li><li>➤ 2 – значение валидировано полностью</li></ul>

## 1.1.4.2.4.3. AP

- › [Источник AP](#)
- › [Браузер источника AP](#)
- › [Запрос алармов](#)
- › [Запрос значений множества сигналов AP](#)
- › [Очередь активных алармов](#)
- › [Менеджер алармов](#)
- › [Учетные данные AP](#)
- › [Элемент AP <Type>](#)
- › [Запрос значений элемента AP](#)

- ▼  AP
  -  Источник AP
  -  Браузер источника AP
  -  Запрос алармов
  -  Запрос значений множества сигналов AP
  -  Очередь активных алармов
  - Менеджер алармов
  -  Учётные данные AP
  -  Элемент AP bool
  -  Элемент AP int1
  -  Элемент AP uint1
  -  Элемент AP int2
  -  Элемент AP uint2
  -  Элемент AP int4
  -  Элемент AP uint4
  -  Элемент AP int8
  -  Элемент AP uint8
  -  Элемент AP float
  -  Элемент AP double
  -  Элемент AP timestamp
  -  Элемент AP string
  -  Элемент AP
  -  Запрос значений элемента AP

## 1.1.4.2.4.3.1. Источник AP

Чтобы взаимодействовать с источником данных по протоколу TCP, добавьте на экранную форму один или несколько компонентов "Источник AP". Компонент невидимый и виден только в области "Структура объекта". На рисунке ниже показано, как на форму было добавлен компонент "Источник AP".

**Структура объекта**

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Данные	
ApSource_1	Источник AP

**Редактор свойств**

Свойство	Характеристики	Значение
Отображаемое имя		ApSource_1
Кардинальное число		1
Родительский источник	R W → 0	<не определено>
Источник учетных данных	R W → 0	<не определено>
Таймаут операций с БД	R ∞	<не определено>
Таймаут установления соединения с...	R ∞	<не определено>
Максимальный размер пула буферов	R ∞	<не определено>
Общее соединение	R ∞	<не определено>
Хост	R W	
Порт	R W	4388
Порт истории	R W	4950
Путь	R W	
Активность	R W	false
Период переподписки	R W	0
Имя клиента	R W	HMI
Идентификатор клиента	R W	HMI

Чтобы компонент мог взаимодействовать с источником данных, настройте минимальный набор его свойств.

Свойство	Описание
----------	----------

Хост	Сетевой адрес источника данных по протоколу ТСР. Для локального компьютера используется IP-адрес: 127.0.0.1
Порт	Номер порта для подключения к источнику данных. По умолчанию номер 4388
Путь	Путь до целевой ветки дерева сигналов источника данных по протоколу ТСР. Если путь не указан, то по умолчанию целевой веткой будет считаться корень дерева сигналов источника данных
Активность	Активность источника АР. С неактивными источниками не происходит обмен данными

## 1.1.4.2.4.3.1.1. События

Событие	Описание
<a href="#">OnErrorStateChanged</a>	Сигнал об изменении типа ошибки
<a href="#">OnConnectionStateChanged</a>	сигнал об изменении состояния соединения

## 1.1.4.2.4.3.1.1.1. OnErrorStateChanged

Данный сигнал уведомляет об изменении типа ошибки источника AP.

### Параметры

Параметры	Тип	Описание
errorState	uint1	Состояние ошибки

## 1.1.4.2.4.3.1.1.2. OnConnectionStateChanged

Данный сигнал уведомляет об изменении состояния соединения источника AP.

### Параметры

Параметр	Тип	Описание
connectionState	uint1	Состояние соединения



## 1.1.4.2.4.3.1.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Строковое описание ошибки</a>	Описание ошибки подключения к источнику данных в текстовом виде
<a href="#">Тип ошибки</a>	Код ошибки подключения к источнику данных
<a href="#">Состояние соединения</a>	Код состояния подключения к источнику данных
<a href="#">Родительский источник</a>	Свойство для создания иерархии источников данных
<a href="#">Источник учетных данных</a>	Ссылка на источник учетных данных
<a href="#">Таймаут операция с БД</a>	Таймаут операций с БД в мс
<a href="#">Таймаут установления соединения с БД</a>	Таймаут установления соединения с БД в мс
<a href="#">Максимальный размер пула буферов</a>	Максимальный размер пула буферов
<a href="#">Общее соединение</a>	Общее соединение для всех источников с одинаковым адресом подключения (Хост:Порт)
<a href="#">Хост</a>	Сетевой адрес источника данных по протоколу TCP
<a href="#">Порт</a>	Номер порта для подключения к источнику данных
<a href="#">Порт истории</a>	Порт для доступа к историческим данным

<a href="#">Путь</a>	Путь до целевой ветки дерева сигналов источника данных по протоколу TSP
<a href="#">Активность</a>	Активность источника AP. С неактивными источниками не происходит обмен данными
<a href="#">Период переподписки</a>	Интервал времени, через который будут предприниматься попытки переподписки на сигналы, отсутствующие в сервере
<a href="#">Имя клиента</a>	Отображаемое имя клиента
<a href="#">Идентификатор клиента</a>	Предназначен для идентификации клиента в рамках подключения к источнику AP
<a href="#">Полный тег</a>	Полный путь Источника AP

## 1.1.4.2.4.3.1.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.1.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.1.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.1.2.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index])
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.1.2.5. Строковое описание ошибки

Описание ошибки подключения к источнику данных в текстовом виде.



string ErrorString



Доступно только для чтения в режиме рантайма.

### Примеры

Записать описание ошибки подключения к источнику AP в поле редактирования:



```
TextEdit.Text = ApSource.ErrorString;
```

## 1.1.4.2.4.3.1.2.6. Тип ошибки

Код ошибки подключения к источнику данных.



uint1 ErrorState



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
0	Ошибок нет
1	Ошибка инициализации

### Примеры

Записать код ошибки подключения к источнику AP в поле редактирования:



```
TextEdit_1.Text = String.ToString(ApSource_1.ErrorState);
```



## 1.1.4.2.4.3.1.2.7. Состояние соединения

Код состояния подключения к источнику данных.



uint1 ConnectionState



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
0	Отключен
1	В процессе подключения
2	Подключен
3	В процессе отключения

### Примеры

Записать код состояния подключения к источнику AP в текстовое поле:



```
TextEdit_1.Text = String.ToString(ApSource_1.ConnectionState);
```

## 1.1.4.2.4.3.1.2.8. Родительский источник

Свойство для создания иерархии источников данных. Например, на экранной форме можно разместить три компонента "Источник AP", первый из которых будет содержать параметры подключения (свойства "[Хост](#)" и "[Порт](#)"). Второй и третий источники лишь будут определять конкретные ветки дерева сигналов (свойство "[Путь](#)"), но ссылаться при этом на первый источник как на **Родительский источник**.



ParentSource

### Примеры

Записать в поле ввода сетевой адрес родительского источника:



```
TextEdit.Text = ApSource.ParentSource.Location;
```

Записать в поле ввода код состояния подключения родительского источника:



```
TextEdit.Text = ApSource.ParentSource.ConnectionState;
```

## 1.1.4.2.4.3.1.2.9. Источник учетных данных

Источник учетных данных, используемый для подключения к защищенному источнику. Указывается в виде ссылки на объект типа "Учетные данные AP", существующий в проекте.



Credentials

## 1.1.4.2.4.3.1.2.10. Таймаут операций с БД

Максимальное время в миллисекундах для выполнения операций с базой данных. При превышении этого времени операция прерывается.



uint4 Timeout

### Примеры

Установить таймаут операций в 5 секунд.



```
ApSource_1.Timeout = 5000;
```

## 1.1.4.2.4.3.1.2.11. Таймаут установления соединения с БД

Время ожидания в миллисекундах для установления соединения с базой данных. Если соединение не устанавливается в течение этого времени, процесс подключения прерывается.



uint4 ConnectionEstablishTimeout

### Примеры

Установить таймаут соединения с БД в 10 секунд.



```
ApSource_1.ConnectionEstablishTimeout = 10000;
```

## 1.1.4.2.4.3.1.2.12. Максимальный размер пула буферов

Максимальный размер пула буферов в байтах, используемого для временного хранения данных перед их записью в Astra.Historian.



uint4 MaxBufferPoolSize

## 1.1.4.2.4.3.1.2.13. Общее соединение

Использовать ли общее соединение для всех источников данных, имеющих одинаковый адрес подключения (Хост, Порт). Если этот параметр не установлен (false), то для каждого источника данных будет устанавливаться отдельное соединение. По умолчанию используется общее соединение. Это свойство актуально только в случаях, когда не указан родительский источник данных.



bool ShareConnection



Актуально только, если не указан родительский источник.

## 1.1.4.2.4.3.1.2.14. Хост

Сетевой адрес источника данных по протоколу TCP. Для локального компьютера используется IP-адрес: "127.0.0.1".



string Location

### Примеры

Вывести сетевой адрес источника данных в поле ввода:



```
TextEdit.Text = ApSource.Location;
```



## 1.1.4.2.4.3.1.2.15. Порт

Номер порта для подключения к источнику данных. По умолчанию номер "4388".



uint2 Port

### Примеры

Вывести порт источника данных в поле ввода:



```
TextEdit.Text = ApSource.Port;
```

## 1.1.4.2.4.3.1.2.16. Порт истории

Порт для доступа к историческим данным.



uint2 HistoryPort

### Примеры

Вывести порт истории источника данных в поле ввода:



```
TextEdit.Text = ApSource.HistoryPort;
```

## 1.1.4.2.4.3.1.2.17. Путь

Путь до целевой ветки дерева сигналов источника данных по протоколу TCP. Если путь не указан, то по умолчанию целевой веткой будет считаться корень дерева сигналов источника данных.



string Path



Доступна возможность переопределения пути во время рантайма.

### Примеры

Вывести путь до целевой ветки дерева сигналов источника в поле ввода:



```
TextEdit.Text = ApSource.Path;
```

## 1.1.4.2.4.3.1.2.18. Активность

Активность источника AP. С неактивными источниками не происходит обмен данными.



bool Active

### Примеры

Записать текущее состояние источника AP в поле ввода:



```
TextEdit_1.Text = String.ToString(ApSource_1.Active);
```

## 1.1.4.2.4.3.1.2.19. Период переподписки

Интервал времени, через который будут предприниматься попытки переподписки на сигналы, отсутствующие в сервере, мс.

"0" - переподписка не выполняется.



uint8 ReAdvise

## 1.1.4.2.4.3.1.2.20. Имя клиента

Имя клиента, используемое для удобного отображения подключенного клиента к источнику AP.



string ClientDisplayName

### Примеры

Установить имя клиента "Оператор".



```
ApSource.ClientDisplayName = "Оператор";
```

## 1.1.4.2.4.3.1.2.21. Идентификатор клиента

Предназначен для идентификации клиента в рамках подключения к источнику AP.

Пустое значение – анонимный клиент.



string ClientId

### Примеры

Установить уникальный идентификатор клиента.



```
ApSource_1.ClientId = "client12345";
```

## 1.1.4.2.4.3.1.2.22. Полный тег

Полный путь Источника AP, формируемый путём конкатенации путей родительских источников.



string Tag



Доступно только для чтения в режиме рантайма.

### Примеры

Записать полный путь к сигналу источника AP в поле редактирования.



```
TextEdit.Text = APltem.Tag;
```



### 1.1.4.2.4.3.1.3. Функции

Функция	Описание
<a href="#">Ask</a>	Квитирование события

## 1.1.4.2.4.3.1.3.1. Ack

Квитирование события.



```
void Ack(string Source, string ConditionName, string Comment, string AcknowledgerId, uint8 ActiveTime, uint4 Cookie)
```

### Параметры

Параметр	Тип	Описание
Source	string	Источник
ConditionName	string	Имя условия
Comment	string	Комментарий
AcknowledgerId	string	Идентификатор подтверждения
ActiveTime	uint8	Активное время
Cookie	uint4	Данные Cookie

### Примеры

Квитировать событие

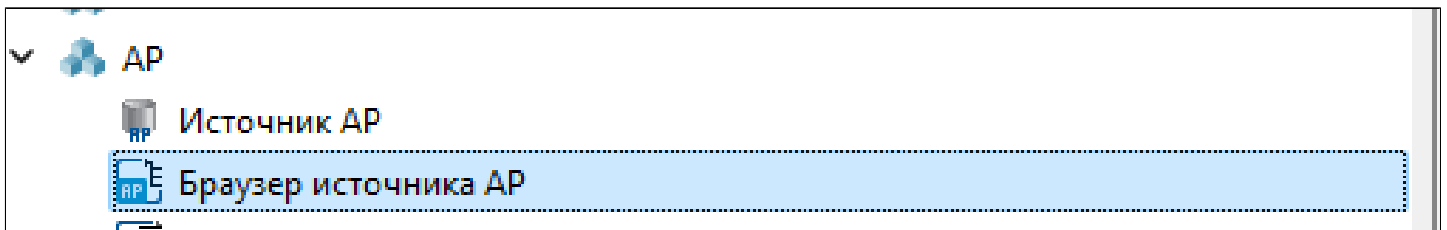


```
ApSource_1.Ack("Thermometer1", "HighTemperature", "Температура проверена и стабилизирована", "user123", 162030, 123456);
```

## 1.1.4.2.4.3.2. Браузер источника AP

Чтобы просматривать дерево сигналов источника данных (например, Astra.Server), а также осуществлять поиск узлов в этом источнике по заданному фильтру воспользуйтесь компонентом "Браузер источника AP".

Компонент "Браузер источника AP" расположен в юните AP Библиотеки компонентов. Экземпляр этого типа не отображается на форме и виден только в области "Структура объекта".



Структура объекта		
<input type="text"/>		
Имя	Описание	
▼ MainForm	Тип на основе Форма	
▼ Данные		
ApSourceBrowser_1	Браузер источника AP	

Свойства		
<input type="text"/>		
Свойство	Характеристики	Значение
S Отображаемое имя		ApSourceBrowser_1
i8 Кардинальное число		1
Источник AP	R ⊆ → ∅	<не определено>
> B Вывод описания предпочтительнее имени	R W	true

С помощью компонента можно получить информацию о структуре источника данных, включая имена узлов, их идентификаторы, описания, а также наличие дочерних элементов.

Полученные данные можно отобразить в виде древовидной структуры на форме с использованием компонентов "Источник данных дерева", "Дерево" и "Колонка дерева" из юнита "Визуальные".

## Возвращаемые данные

Компонент "Браузер источника AP" возвращает данные в табличном формате. Используя указанные ниже идентификаторы, вы можете обращаться к этим данным в проекте. Для просмотра результатов браузинга подключите "Браузер источника AP" к одному из компонентов: "Дерево" или "Таблица".

Идентификатор колонки	Тип колонки	Тип возвращаемого значения	Описание
name	1	string	Имя узла
id	2	string	Идентификатор
parent_id	3	string	Идентификатор родителя
display_name	4	string	Отображаемое имя
cdt	5	uint1	Идентификатор источника данных <ul style="list-style-type: none"> <li>&gt; 0 - e</li> <li>&gt; 1 - In</li> <li>&gt; 3 - U</li> <li>&gt; 5 - V</li> <li>&gt; 6 - U</li> <li>&gt; 7 - In</li> <li>&gt; 8 - U</li> <li>&gt; 9 - In</li> <li>&gt; 12 -</li> <li>&gt; 13 -</li> <li>&gt; 14 -</li> <li>&gt; 15 -</li> <li>&gt; 17 -</li> <li>&gt; 18 -</li> </ul>
historizing	5	bool	Флаг ведения истории

description	5	string	Описание
has_children	6	bool	Наличие

## 1.1.4.2.4.3.2.1. Функции

Функция	Описание
<a href="#">GetChildren</a>	Получение информации о дочерних объектах узла
<a href="#">GetChildrenWithRequestId</a>	Получение информации о дочерних объектах узла с указанием идентификатора запроса
<a href="#">GetSourceNodes</a>	Получение информации об узлах поддерева источника, имя которых соответствует фильтру

## 1.1.4.2.4.3.2.1.1. GetChildren

При вызове этой функции компонент Браузер источника AP выполняет браузинг узлов дерева сигналов источника данных, начиная с указанного родительского узла. Если параметр "parentNode" не задан (равен пустой строке), то браузинг начинается с корня дерева.



```
void GetChildren(string parentNode)
```

### Параметры

Параметр	Тип	Описание
parentNode	string	Строка, представляющая путь к родительскому узлу.

### Примеры

Выполнить браузинг узлов дерева сигналов, начиная с корня.



```
ApSourceBrowser_1.GetChildren(" ");
```

Выполнить браузинг узлов дерева сигналов, начиная с пути "папка1.общие\_параметры.сигналы".



```
ApSourceBrowser_1.GetChildren("папка1.общие_параметры.сигналы");
```

## 1.1.4.2.4.3.2.1.2. GetChildrenWithRequestId

При вызове этой функции компонент Браузер источника AP создает запрос к источнику данных для получения узлов дерева сигналов по указанному родительскому узлу.

Результаты запроса (дочерние узлы) автоматически прикрепляются к узлу с идентификатором запроса requestId в графическом компоненте Дерево (или другой структуре данных), который был предварительно создан с использованием метода AddItem().



```
void GetChildrenWithRequestId(string parentNode, string requestId)
```

### Параметры

Параметр	Тип	Описание
parentNode	string	Строка, представляющая родительский узел, для которого будет выполнен браузер.
requestId	string	Идентификатор запроса, который соответствует родительскому узлу, к которому должен быть прикреплен результат браузера.

### Примеры

Создать корневой элемент "Модули" в компоненте "Дерево" и загрузить дочерние элементы для узла "Модули".



```
x: string = Tree_1.AddItem("", "Модули");
```

Вызываем метод GetChildrenWithRequestId() для загрузки дочерних элементов для узла Модули с использованием идентификатора x:





```
ApSourceBrowser_1.GetChildrenWithRequestId("Модули", x);
```

## 1.1.4.2.4.3.2.1.3. GetSourceNodes

Получает информацию о поддереве узла источника данных, основываясь на заданном фильтре.

Параметр `filter` представляет собой строку с выражением в формате "wildcard filter", где символы "\*" и "?" используются для фильтрации узлов.



```
void GetSourceNodes(string filter)
```

### Параметры

Параметр	Описание
filter	Строка, содержащая выражение в формате wildcard filter для фильтрации узлов. Пример фильтрации: <ul style="list-style-type: none"><li>➤ <code>server_*</code> - этот фильтр выберет все узлы, начинающиеся с <code>server_</code>, например, <code>server_1</code>, <code>server_data</code>, <code>server_backup</code>, и т. д.</li><li>➤ <code>Param?</code> - этот фильтр выберет все узлы, где после слова <code>param</code> идет ровно один любой символ, например, <code>param1</code>, <code>param2</code>, <code>paramA</code>, но не <code>param</code>, так как в этом случае после <code>param</code> нет символа.</li></ul>

### Примеры

Отфильтровать узлы источника данных по имени, оставляя только те, которые начинаются с `sensor`.



```
// Фильтр для поиска узлов, начинающихся с "sensor"  
filter: string = "sensor*";  
// Вызов функции GetSourceNodes() с передачей фильтра  
ApSourceBrowser_1.GetSourceNodes(filter);
```

## 1.1.4.2.4.3.2.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Источник AP</a>	Ссылка на источник AP
<a href="#">Вывод описания предпочтительнее имени</a>	Выводить описание вместо имени, если это возможно

## 1.1.4.2.4.3.2.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.2.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.2.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.2.2.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.2.2.5. Источник AP

Ссылка на источник AP.




Source

Ссылка на компонент Источник AP, существующий в проекте.



# 1.1.4.2.4.3.2.2.6. Вывод описания предпочтительнее имени

Выводить описание вместо имени, если это возможно.

 bool DescriptionPreffered


Выводить описание узлов вместо их имен при отображении информации в дереве узлов.

## Значение

Значение	Описание
true	Вывод описания узлов вместо их имен.
false	Вывод имени узлов вместо их описания.

## Примеры

Выводить описание узлов вместо имени.

 ApSourceBrowser\_1.DescriptionPreffered = true;

## 1.1.4.2.4.3.3. Запрос алармов

Компонент выполняет роль провайдера данных и позволяет получать с сервера оперативные и исторические события по протоколу TCP. Компонент невизуальный (не отображается на форме) и виден только в области "Структура объекта".

Чтобы компонент Запрос алармов подключался к серверу и запрашивал все события (включая квитируемые и деактивированные):

1. Добавьте объект типа "Запрос алармов" на форму.
2. Укажите источник данных – сервер, с которого вы хотите запрашивать события, в свойстве Источник.



В проект уже должен быть добавлен объект типа "Источник AP".

3. Укажите какие именно данные и в каком виде вы хотите получать: оперативные события журналом, исторические события журналом или список активных условий. Используйте для этого свойство Режим работы.
4. Укажите хотите ли вы получать все существующие активные события или только те, которые возникли после подключения компонента к серверу. Используйте для этого свойство Запрашивать список активных событий при подключении к источнику.
5. Активируйте компонент в свойстве Активность.
6. Чтобы обновлять данные от источника, используйте функцию Reload.

Чтобы среди запрошенных событий не было квитируемых и деактивированных, установите true в свойстве "Удалять квитируемые и деактивированные события".

По каждому событию компонент Запрос алармов получает набор данных.

Идентификатор	Описание	Название столбца по умолчанию (если не заполнять свойство Заголовок для компонента Таблица: столбец)
source	Источник события (полный тег сигнала, изменение которого привело к генерации события)	Источник
time	Время генерации уведомления о событии	Время генерации
message	Сообщение	Сообщение
severity	Уровень важности события	Важность
condition_name	Имя условия генерации события	Условие
subcondition_name	Имя подусловия генерации события	Подусловие
quality	Текущее качество сигнала, изменение которого привело к генерации события	Качество
active_time	Время перехода состояния события в активное	Время срабатывания
actor_id	Имя пользователя, выполнившего квитирование сообщения о событии	Пользователь
ack	Признак квитирования события	Квитировано
active	Признак активности подусловия, по которому	Активность

	было сгенерировано событие	
cookie	Специальный идентификатор события, который имеет служебное назначение	Куки
ack_time	Время квитирования	Время квитирования
ack_required	Требование квитирования	Квитировать
value	Значение сигнала, изменение которого привело к генерации события	Значение
sound	Звуковой файл или группа звуковых файлов, воспроизводимых при выполнении подусловия генерации сообщения о событии	Звук
area_path	Тег родительского узла источника события	Зона
object_id	Идентификатор объекта на сервере, по которому было сгенерировано событие	ID объекта
object_name	Имя объекта на сервере, по которому было сгенерировано событие	Имя объекта
relative_tag	Относительный тег источника события. Состоит из имени объекта (object_name) и той части полного тега источника события, которая находится после имени объекта	Относительный тег

deactive_time	Время, когда событие перешло из активного состояния в неактивное	Время деактивации
---------------	--	-------------------

Используя указанные выше идентификаторы, вы можете обращаться к этим данным в проекте. Например, вывести их значения в визуальную таблицу на форме, используя компоненты "Таблица: модель данных" и "Таблица" из юнита "Таблицы".



Чтобы юнит "Таблицы" появился в библиотеке компонентов, установите Astra.HMI.Tables.

## 1.1.4.2.4.3.3.1. Функции

Функция	Описание
<a href="#">Reload</a>	Перезагрузить данные
<a href="#">Clear</a>	Очистить буфер данных

## 1.1.4.2.4.3.3.1.1. Reload

Применяет все заданные свойства объекта типа "Запрос алармов". По запросу потребителя (например, компонента после "Таблица: модель данных" из юнита "Таблицы") функция обновляет данные с источника - сервера, поставляющего оперативные и исторические данные.



```
void Reload()
```

### Примеры

Запросить данные с источника с учетом фильтра, заданного в текстовом поле:



```
ApAlarmQuery_1.JsonFilter = TextEdit_1.Text;  
ApAlarmQuery_1.Reload();
```

## 1.1.4.2.4.3.3.1.2. Clear

Очищает буфер данных.



```
void Clear(string_1)
```



## 1.1.4.2.4.3.3.2. События

Событие	Описание
<a href="#">OperativeBufferSizeExceeded</a>	Сигнал переполнения буфера оперативных событий
<a href="#">ClearFinished</a>	Сигнал завершения очистки буфера оперативных событий
<a href="#">StatusChanged</a>	Уведомляет об изменении статуса запроса

## 1.1.4.2.4.3.3.2.1. OperativeBufferSizeExceeded

Сигнал переполнения буфера оперативных событий ([Размер буфера оперативных событий](#)). Уведомляет, когда размер буфера превысил значение в свойстве "Размер буфера оперативных событий".

## 1.1.4.2.4.3.3.2.2. ClearFinished

Сигнал завершения очистки буфера оперативных событий ([Размер буфера оперативных событий](#)). Уведомляет о завершении ручной очистки буфера.

## 1.1.4.2.4.3.3.2.3. StatusChanged

Уведомляет об изменении статуса запроса.

## 1.1.4.2.4.3.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Источник</a>	Ссылка на компонент Источник AP
<a href="#">Режим работы</a>	Режим работы компонента в зависимости от вида запрашиваемых событий
<a href="#">Начало исторического интервала</a>	Начальное значение интервала запроса исторических данных от источника
<a href="#">Конец исторического интервала</a>	Конечное значение интервала запроса исторических данных от источника
<a href="#">Фильтр в JSON-формате</a>	Условия фильтрации запрашиваемых от источника оперативных данных
<a href="#">Фильтр исторических данных в JSON-формате</a>	Условия фильтрации запрашиваемых от источника исторических данных
<a href="#">Активность</a>	Активация компонента
<a href="#">Удалять квитируемые и деактивированные события</a>	Удаление из таблицы квитируемых событий при их деактивации и неактивных событий при их квитировании
<a href="#">Запрашивать список активных событий при подключении к источнику</a>	Включает запрос активных и неквитируемых событий с источника данных при каждом подключении
<a href="#">Интервал предзагрузки истории</a>	Интервал подгрузки исторических данных в таблицу оперативных данных
<a href="#">Расширяющий интервал</a>	Отрезок времени (в часах), на который сдвигается время

<a href="#">Хронология исторических событий</a>	Устанавливает хронологию, в которой будут запрашиваться исторические события
<a href="#">Режим сведения событий</a>	Режим сведения событий с соответствующими им активациями
<a href="#">Размер буфера оперативных событий</a>	Размер буфера оперативных событий
<a href="#">Параметры обработки данных</a>	Параметры обработки данных
<a href="#">Спецификация пользовательских атрибутов</a>	Спецификация пользовательских атрибутов

## 1.1.4.2.4.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.3.3.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.3.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.3.5. Источник

Сервер, поставляющий оперативные и исторические события. Указывается в виде ссылки на объект типа "Источник AP", существующий в проекте.

## 1.1.4.2.4.3.3.6. Режим работы

Устанавливает режим работы компонента в зависимости от вида запрашиваемых событий.



uint1 Mode

### Значение

Значение	Описание
1	Список активных событий
2	Журнал оперативных событий
3	Журнал исторических событий

## 1.1.4.2.4.3.3.3.7. Начало исторического интервала

Начальное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное значение переведите в формат UTC. Для перевода вы можете использовать функции "Parse" и "ToUtc" языка Astra.Om.



timestamp HistoricalRangeStart



Свойство используется только при запросе исторических данных (когда в свойстве "Режим работы" указано значение "3").



Начальной или конечной точкой интервала запроса может быть "0". Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

## Примеры

Запросить исторические события с 22.07.2022 20:00:25 по 25.07.2022 21:17:25.



```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");  
DateEnd: uint8 = DateTime.Pars ("25.07.2022 21:17:25");  
Events.TableModellvents.QuerySource1.HistoricalRangeStart =  
DateTime.ToUtc(DateStart);  
Events.TableModellvents.QuerySource1.HistoricalRangeEnd =  
DateTime.ToUtc(DateEnd);  
Events.TableModellvents.QuerySource1.Reload();  
Events.TableModellvents.BeginReadAsync();
```

## 1.1.4.2.4.3.3.3.8. Конец исторического интервала

Конечное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное значение переведите в формат UTC. Для перевода вы можете использовать функции "Parse" и "ToUtc" языка Astra.Om.



timestamp HistoricalRangeEnd



Свойство используется только при запросе исторических данных (когда в свойстве "Режим работы" указано значение "3").



Начальной или конечной точкой интервала запроса может быть "0". Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

## Примеры

Запросить исторические события с 22.07.2022 20:00:25 по 25.07.2022 21:17:25.



```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");  
DateEnd: uint8 = DateTime.Pars ("25.07.2022 21:17:25");  
Events.TableModellvents.QuerySource1.HistoricalRangeStart =  
DateTime.ToUtc(DateStart);  
Events.TableModellvents.QuerySource1.HistoricalRangeEnd =  
DateTime.ToUtc(DateEnd);  
Events.TableModellvents.QuerySource1.Reload();  
Events.TableModellvents.BeginReadAsync();
```

## 1.1.4.2.4.3.3.9. Фильтр в JSON-формате

Условия фильтрации запрашиваемых от источника оперативных данных. Задаются в виде строки - объекта JSON.



string Filter

Фильтрация возможна только для столбцов с идентификаторами (переменными): severity, message, source, area, acked.

Условие фильтрации представляет собой одиночный ({"ключ": "значение"}) или составной (состоящий из нескольких объектов) объект JSON.

Составной объект формируется с помощью:

- › логических операций:
  - › конъюнкция - "and" или "&"
  - › дизъюнкция - "or" или "|"
  - › инверсия - "not" или "!"
- › операций сравнения:
  - › больше - ">"
  - › больше либо равно - ">="
  - › меньше - "<"
  - › меньше либо равно - "<="
  - › соответствует - "=="
  - › не соответствует - "!="

В зависимости от используемой операции составной объект JSON может иметь вид:

- › {"and": [ARRAY]}
- › {"or": [ARRAY]}
- › {"not": { OBJECT }}
- › {"операция сравнения": { VARIABLE }} (операцию сравнения "==" можно не указывать, т.е. вид объекта - { VARIABLE })

где:

- › VARIABLE - переменная. Обозначается структурой <имя переменной>: <значение переменной>
- › OBJECT - переменная или составной объект JSON любого из указанных выше форматов
- › ARRAY - массив. Может состоять из переменных и составных объектов JSON (любого формата) в различных комбинациях

## Примеры

Запросить события с:

- › важностью (столбец severity) больше 10;
- › сообщением (столбец message), содержащим текст поставлен или с именем источника (столбец source), не соответствующим TU\_212:



```
{
  "AND": [
    { ">": { "severity": 10 } },
    {
      "OR": [
        { "message": "*поставлен*" },
        { "NOT": { "source": "TU_212" } }
      ]
    }
  ]
}
```

Запросить события с важностью 667:



```
{"=":{ "severity":667}}
```



## 1.1.4.2.4.3.3.10. Фильтр исторических данных в JSON-формате

Условия фильтрации запрашиваемых от источника исторических данных. Задаются в виде строки - объекта JSON. Правила формирования условий фильтрации аналогичны правилам при запросе оперативных данных.



string HistoricalFilter



Фильтр можно установить для всех данных, загружаемых в историческом режиме (когда в свойстве "Режим работы" указано значение "3"), а также для исторических данных, загружаемых в оперативном режиме (когда в свойстве "Режим работы" указано значение "2" или "1", а в свойстве "Интервал предзагрузки истории" задан интервал предзагрузки данных).

## 1.1.4.2.4.3.3.11. Активность

Активация компонента. С неактивным компонентом не происходит обмен данными.



bool Active

### Значение

Значение	Описание
true	Компонент активен и может запрашивать события
false	Компонент неактивен

## 1.1.4.2.4.3.3.12. Удалять квитированные и деактивированные события

Разрешает/запрещает удаление из списка событий, получаемых от источника (сервера, поставляющего оперативные события) квитированные и при этом деактивированные события.



bool RemoveAckedInactive

### Значение

Значение	Описание
true	Квитированные и деактивированные события удаляются из списка
false	Квитированные и деактивированные события остаются в списке



Свойство работает только в режиме журнала оперативных событий.

## 1.1.4.2.4.3.3.13. Запрашивать список активных событий при подключении к источнику

Включает запрос активных и неактивных событий с источника данных при каждом подключении.



bool RefreshAtStart

### Значение

Значение	Описание
true	Запрос активных и неактивных событий при подключении к серверу включен
false	Запрос активных и неактивных событий при подключении к серверу выключен

## 1.1.4.2.4.3.3.14. Интервал предзагрузки истории

Интервал (в секундах) подгрузки исторических данных в таблицу оперативных данных. Интервал начинается указанное количество секунд назад и заканчивается текущим моментом.



uint4 HistoryPreloadInterval

## 1.1.4.2.4.3.3.15. Расширяющий интервал

Отрезок времени (в часах), на который сдвигается время, указанное в свойстве "Конец исторического интервала". За счет такого сдвига можно расширить исходный запрашиваемый интервал, указанный в свойствах "Начало исторического интервала" и "Конец исторического интервала".

Если такое расширение установлено, с сервера запрашиваются также события, у которых время генерации больше времени, указанного в свойстве "Конец исторического интервала". Из полученных событий отбрасываются события, время срабатывания которых не входит в исходный запрашиваемый интервал.



uint4 HistoryExpandInterval



Свойство используется только при запросе исторических данных (когда в свойстве "Режим работы" указано значение "3").

## 1.1.4.2.4.3.3.3.16. Хронология исторических событий

Устанавливает хронологию, в которой будут запрашиваться исторические события за интервал, указанный в свойствах "Начало исторического интервала" и "Конец исторического интервала".



uint1 HistoricalEventsChronology



Свойство используется только при запросе исторических данных (когда в свойстве "Режим работы" указано значение "3").

### Значение

Значение	Описание
0	По времени генерации. Хронология выстраивается с учетом времени генерации уведомлений о событиях на сервере
1	По времени срабатывания. Хронология выстраивается с учетом времени активации условий на технологическом объекте

## 1.1.4.2.4.3.3.17. Режим сведения событий

Режим сведения событий с соответствующими им активациями.

Значение по умолчанию - "0" (сводить всё)



uint1 MergeMode



# 1.1.4.2.4.3.3.3.18. Размер буфера оперативных событий

Размер буфера оперативных событий.



uint4 OperativeBufferSize

# 1.1.4.2.4.3.3.3.19. Параметры обработки данных

Параметры обработки данных.



string DataProcessing

# 1.1.4.2.4.3.3.3.20. пользовательских атрибутов

## Спецификация

Спецификация пользовательских атрибутов.



string CustomAttributes

## **1.1.4.2.4.3.4. Запрос значений множества сигналов AP**

Запрос значений множества сигналов AP.

## 1.1.4.2.4.3.4.1. Функции

Функция	Описание
<a href="#">Reload</a>	Перезагрузить данные

## 1.1.4.2.4.3.4.1.1. Reload

Применяет все заданные свойства у объекта типа "Запрос значений множества сигналов AP". По запросу потребителя (например, компонента "Таблица: модель данных" из юнита "Таблицы") функция обновляет данные с источника данных, поставляющего исторические данные.



```
void Reload()
```

### Пример

Запросить значения множества сигналов с источника.



```
ApltemValueQuery_1.Mode=2;  
ApltemValueQuery_1.Reload();
```

## 1.1.4.2.4.3.4.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Источник</a>	Ссылка на компонент Источник AP
<a href="#">Начало исторического интервала</a>	Начало исторического интервала
<a href="#">Конец исторического интервала</a>	Конец исторического интервала
<a href="#">Список сигналов</a>	Список сигналов

## 1.1.4.2.4.3.4.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.3.4.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.4.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.4.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.4.2.5. Источник АР

Ссылка на компонент "Источник АР".

## 1.1.4.2.4.3.4.2.6. Начало исторического интервала

Начальное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное значение переведите в формат UTC. Для перевода вы можете использовать функции "Parse" и "ToUtc" языка Astra.Om.



timestamp HistoricalRangeStart



Начальной или конечной точкой интервала запроса может быть "0". Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

### Примеры

Запросить исторические события с 22.07.2022 20:00:25 по 25.07.2022 21:17:25.



```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");  
DateEnd: uint8 = DateTime.Pars ("25.07.2022 21:17:25");  
Events.TableModellvents.QuerySource1.HistoricalRangeStart =  
DateTime.ToUtc(DateStart);  
Events.TableModellvents.QuerySource1.HistoricalRangeEnd =  
DateTime.ToUtc(DateEnd);  
Events.TableModellvents.QuerySource1.Reload();  
Events.TableModellvents.BeginReadAsync();
```

## 1.1.4.2.4.3.4.2.7. Конец исторического интервала

Конечное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное значение переведите в формат UTC. Для перевода вы можете использовать функции "Parse" и "ToUtc" языка Astra.Om.



timestamp HistoricalRangeEnd



Начальной или конечной точкой интервала запроса может быть "0". Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

### Примеры

Запросить исторические события с 22.07.2022 20:00:25 по 25.07.2022 21:17:25.



```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");  
DateEnd: uint8 = DateTime.Pars ("25.07.2022 21:17:25");  
Events.TableModellvents.QuerySource1.HistoricalRangeStart =  
DateTime.ToUtc(DateStart);  
Events.TableModellvents.QuerySource1.HistoricalRangeEnd =  
DateTime.ToUtc(DateEnd);  
Events.TableModellvents.QuerySource1.Reload();  
Events.TableModellvents.BeginReadAsync();
```

## 1.1.4.2.4.3.4.2.8. Список сигналов

Список сигналов.



string Signals

Свойство "Список сигналов" предназначено для указания списка сигналов источника данных, по которым нужно получить данные. Формат задания – "JSON".

Передаваемая JSON-структура должна иметь следующий формат:



```
{
  "items": [ {
    "tag": "<TAG>",
    "value": "<COLUMN-NAME>",
    "quality": "<COLUMN-NAME>",
    "timestamp": "<COLUMN-NAME>",
  },
  ...
  ]
}
```

Где:

- «tag» – тег сигнала на сервере (полное имя сигнала, формирующееся из имен узлов дерева сигналов, последовательно включающих данный сигнал).
- «value» – идентификатор столбца таблицы, в который будет записываться значение сигнала.
- «quality» – идентификатор столбца таблицы, в который будет записываться качество сигнала (необязательный параметр).
- «timestamp»– идентификатор столбца таблицы, в который будет записываться метка времени сигнала (необязательный параметр).

## Пример

Необходимо настроить компонент "Запрос значений множества сигналов AP" так, чтобы значения сигналов с тегами «Deviation.s\_deviation4» и «Level.s\_level4» были записаны в столбцы с идентификаторами «Value\_1» и «Value\_2».

В свойство "Список сигналов" передадим JSON-структуру, которая описывает сигналы и их ассоциированные столбцы в выходном наборе данных:

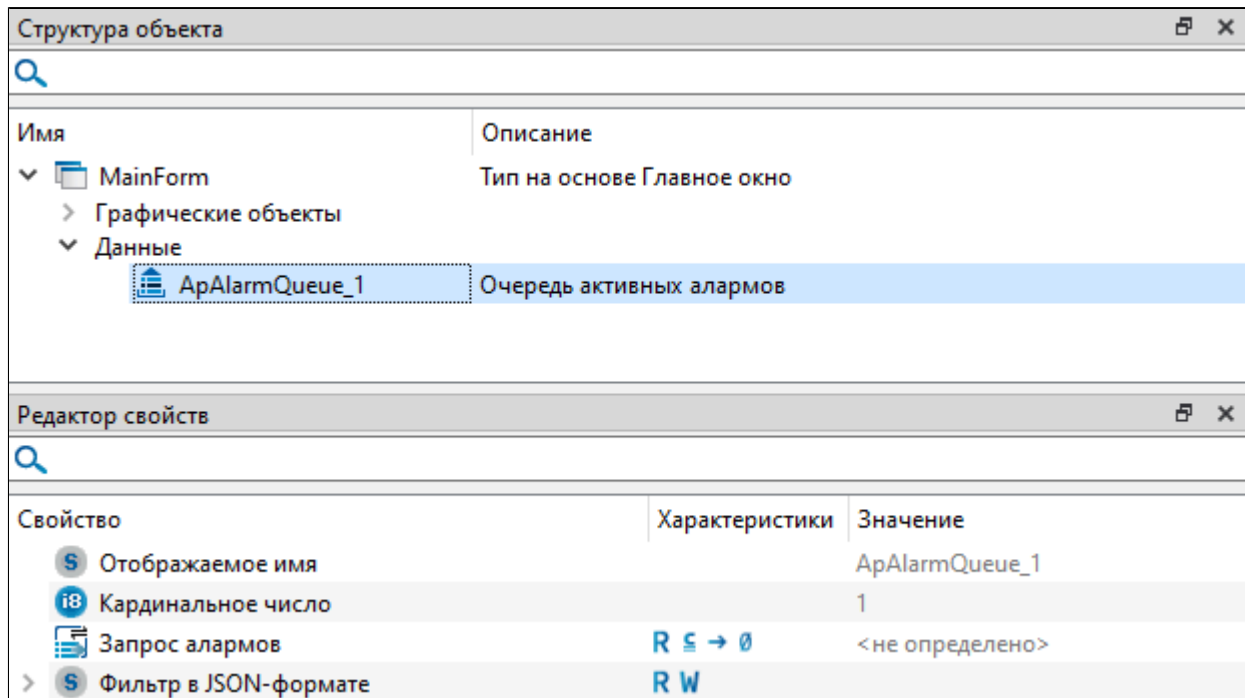


```
{
  "items": [
    {
      "tag": "Deviation.s_deviation4",
      "value": "Value_1"
    },
    {
      "tag": "Level.s_level4",
      "value": "Value_2"
    }
  ]
}
```



## 1.1.4.2.4.3.5. Очередь активных алармов

События, получаемые с помощью компонента Запрос алармов в оперативном режиме, можно помещать в очередь с помощью компонента Очередь активных алармов.



Очередь может содержать только неквитированные события. Квитированные события из очереди удаляются. Так вы можете использовать очередь, например, для построения очереди звуков неквитированных событий.

Чтобы работать с элементами очереди событий, используйте API компонента Очередь активных алармов.

## 1.1.4.2.4.3.5.1. События

Событие	Описание
<a href="#">OnActiveStateChanged</a>	Сигнал об изменении активности очереди
<a href="#">OnAlarmInserted</a>	Сигнал о добавлении аларма в очередь
<a href="#">OnAlarmRemoved</a>	Сигнал об удалении аларма из очереди

## 1.1.4.2.4.3.5.1.1. OnActiveStateChanged

Изменение значения свойства "Активность" объекта типа "Очередь активных алармов".

### Параметры

Параметр	Тип	Описание
activeState	bool	Состояние очереди. Соответствует значению свойства "Активность"

### Примеры

Выводить в журнал времени исполнения состояние очереди:



```
DebugTool_1.Log( "ACTIVE STATE CHSNGED: " +  
String.ToString( activeState ) );
```

## 1.1.4.2.4.3.5.1.2. OnAlarmInserted

Сигнал о добавлении аларма в очередь.

## 1.1.4.2.4.3.5.1.3. OnAlarmRemoved

Сигнал об удалении аларма из очереди.

## 1.1.4.2.4.3.5.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Запрос алармов</a>	Ссылка на запрос алармов
<a href="#">Фильтр в JSON-формате</a>	Фильтр в JSON-формате
<a href="#">Активность</a>	Наличие событий в очереди

## 1.1.4.2.4.3.5.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.5.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.3.5.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.5.2.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.5.2.5. Запрос алармов

Источник событий для очереди. Указывается в виде ссылки на объект типа "Запрос алармов", существующий в проекте.

## 1.1.4.2.4.3.5.2.6. Фильтр в JSON-формате

Условия фильтрации запрашиваемых от источника оперативных данных. Задаются в виде строки - объекта JSON.



string Filter

Фильтрация возможна только для столбцов с идентификаторами (переменными): severity, message, source, area, acked.

Условие фильтрации представляет собой одиночный ({"ключ": "значение"}) или составной (состоящий из нескольких объектов) объект JSON.

Составной объект формируется с помощью:

- логических операций:
  - конъюнкция - "and" или "&"
  - дизъюнкция - "or" или "|"
  - инверсия - "not" или "!"
- операций сравнения:
  - больше - ">"
  - больше либо равно - ">="
  - меньше - "<"
  - меньше либо равно - "<="
  - соответствует - "=="
  - не соответствует - "!="

В зависимости от используемой операции составной объект JSON может иметь вид:

- {"and": [ARRAY]}
- {"or": [ARRAY]}
- {"not": { OBJECT }}
- {"операция сравнения": { VARIABLE }} (операцию сравнения "==" можно не указывать, т.е. вид объекта - { VARIABLE })

где:

- › VARIABLE - переменная. Обозначается структурой <имя переменной>: <значение переменной>
- › OBJECT - переменная или составной объект JSON любого из указанных выше форматов
- › ARRAY - массив. Может состоять из переменных и составных объектов JSON (любого формата) в различных комбинациях

## Примеры

Запросить события с:

- › важностью (столбец severity) больше 10;
- › сообщением (столбец message), содержащим текст поставлен или с именем источника (столбец source), не соответствующим TU\_212:



```
{
  "AND": [
    { ">": { "severity": 10 } },
    {
      "OR": [
        { "message": "*поставлен*" },
        { "NOT": { "source": "TU_212" } }
      ]
    }
  ]
}
```

Запросить события с важностью 667:



```
{"=":{ "severity":667}}
```

## 1.1.4.2.4.3.5.2.7. АКТИВНОСТЬ

Свойство показывает, есть ли события в очереди.



bool Active



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	В очереди есть события
false	В очереди нет событий

## 1.1.4.2.4.3.5.3. Функции

Функция	Описание
<a href="#">MoveNext</a>	Перемещает указатель на следующее событие в очереди
<a href="#">MoveFirst</a>	Перемещает указатель на первое событие в очереди
<a href="#">MoveNextOrFirst</a>	Перемещает указатель на первое событие в очереди
<a href="#">CurrentAlarm</a>	Получает строку с данными текущего события (на котором сейчас находится указатель)
<a href="#">Reload</a>	Запускает обновление списка активных событий с источника (объекта типа Запрос алармов)
<a href="#">Clear</a>	Очищает очередь
<a href="#">RemoveCurrent</a>	Удаляет текущее событие (на котором сейчас находится указатель)
<a href="#">CancelRead</a>	Прекращает получение данных с источника (объекта типа Запрос алармов), выполняемое функцией Reload

## 1.1.4.2.4.3.5.3.1. MoveNext

Перемещает указатель на следующее событие в очереди.



bool MoveNext()

### Значение

Значение	Описание
true	Возвращается при успешном перемещении на следующий элемент очереди
false	Возвращается при отсутствии следующего элемента в очереди

### Примеры

Переместиться на следующее событие в очереди и при успешном перемещении вывести в журнал времени исполнения строку с данными текущего события:



```
moved: bool = ApAlarmQueue_1.MoveNext();
    DebugTool_1.Log(moved);
if (moved)
    DebugTool_1.Log(ApAlarmQueue_1.CurrentAlarm());
```



## 1.1.4.2.4.3.5.3.2. MoveFirst

Перемещает указатель на первое событие в очереди.



bool MoveFirst()

### Значение

Значение	Описание
true	Возвращается при успешном перемещении на первый элемент очереди
false	Возвращается при отсутствии элементов в очереди

### Примеры

Переместиться на первое событие в очереди и при успешном перемещении вывести в журнал времени исполнения строку с данными текущего события:



```
moved: bool = ApAlarmQueue_1.MoveFirst();
    DebugTool_1.Log(moved);
if (moved)
    DebugTool_1.Log(ApAlarmQueue_1.CurrentAlarm());
```

## 1.1.4.2.4.3.5.3.3. MoveNextOrFirst

Перемещает указатель на следующее событие в очереди, либо на первое, если указатель достиг конца очереди.



bool MoveNextOrFirst()

### Значение

Значение	Описание
true	Возвращается при успешном перемещении на следующий или первый элемент очереди
false	Возвращается при отсутствии следующего элемента в очереди или пустой очереди

### Примеры

Переместиться на следующее или первое событие в очереди и при успешном перемещении вывести в журнал времени исполнения строку с данными текущего события:



```
moved: bool = ApAlarmQueue_1.MoveNextOrFirst();
    DebugTool_1.Log(moved);
if (moved)
    DebugTool_1.Log(ApAlarmQueue_1.CurrentAlarm());
```

## 1.1.4.2.4.3.5.3.4. CurrentAlarm

Получает строку с данными текущего события (на котором сейчас находится указатель).



bool CurrentAlarm()

### Значение

Значение	Описание
source	Источник события (тег сигнала)
condition	Имя условия генерации события
activeTime	Время перехода состояния события в активное (в формате uint8).
sound	Звуковой файл или группа звуковых файлов, воспроизводимых при выполнении подусловия генерации сообщения о событии

### Примеры

Пример возвращаемой строки:



```
{"source":"Level.s_level", "condition":"LEVEL", "activeTime":  
133032079365115751, "sound":";sound"}
```

Вывести в журнал времени исполнения строку с данными текущего события:



```
DebugTool_1.Log("current: " = ApAlarmQueue_1.CurrentAlarm());
```

## 1.1.4.2.4.3.5.3.5. Reload

Запускает обновление списка активных событий с источника (объекта типа "Запрос алармов").



```
void Reload()
```

### Примеры

Запросить список активных событий с источника с учетом фильтра, заданного в текстовом поле:



```
ApAlarmQueue_1.JsonFilter = TextEdit_1.Text;  
ApAlarmQueue_1.Reload();
```

## 1.1.4.2.4.3.5.3.6. Clear

Очищает очередь.



```
void Clear()
```

### Примеры

Очистить очередь и вывести в журнал времени исполнения длину очереди:



```
len: var = ApAlarmQueue_1.Lenght;  
ApAlarmQueue_1.Clear();  
DebugTool_1.Log("clear: " + String.ToString(len));
```

## 1.1.4.2.4.3.5.3.7. RemoveCurrent

Удаляет текущее событие (на котором сейчас находится указатель). При удалении события указатель сбрасывается. Для удаления другого события принудительно переведите указатель на него с помощью функций "MoveNext", "MoveFirst" или "MoveNextOrFirst".



```
void RemoveCurrent()
```

### Примеры

Удалить текущее событие и вывести в журнал времени исполнения длину очереди:



```
len: var = ApAlarmQueue_1.Length;  
ApAlarmQueue_1.RemoveCurrent();  
DebugTool_1.Log("clear: " + String.ToString(len));
```

## 1.1.4.2.4.3.5.3.8. CancelRead

Прекращает получение данных с источника (объекта типа "Запрос алармов"), выполняемое функцией "Reload".



```
void CancelRead()
```

### Примеры

Завершить получение данных:



```
ApAlarmQueue_1.CancelRead();
```

## 1.1.4.2.4.3.6. Менеджер алармов

Менеджер алармов.



## 1.1.4.2.4.3.6.1. События

Событие	Описание
<a href="#">SetAreaDisabledFinished</a>	Сигнал о завершении SetAreaDisabled
<a href="#">SetSourceDisabledFinished</a>	Сигнал о завершении SetSourceDisabled
<a href="#">SetAreaSuppressedFinished</a>	Сигнал о завершении SetAreaSuppressed
<a href="#">SetSourceSuppressedFinished</a>	Сигнал о завершении SetSourceSuppressed
<a href="#">GetAreaStateFinished</a>	Сигнал о завершении GetAreaState
<a href="#">GetSourceStateFinished</a>	Сигнал о завершении GetSourceState
<a href="#">GetDisabledOrSuppressedAreasFinished</a>	Сигнал о завершении GetDisabledOrSuppressedAreas
<a href="#">GetDisabledOrSuppressedSourcesFinished</a>	Сигнал о завершении GetDisabledOrSuppressedSources

## **1.1.4.2.4.3.6.1.1. SetAreaDisabledFinished**

Сигнал о завершении SetAreaDisabled. Уведомляет о завершении установки состояния блокировки по зоне.

## 1.1.4.2.4.3.6.1.2. SetSourceDisabledFinished

Сигнал о завершении SetSourceDisabled. Уведомляет о завершении установки состояния блокировки по источнику.

## **1.1.4.2.4.3.6.1.3. SetAreaSuppressedFinished**

Сигнал о завершении SetAreaSuppressed. Уведомляет о завершении установки состояния подавления по зоне.

## 1.1.4.2.4.3.6.1.4. SetSourceSuppressedFinished

Сигнал о завершении SetSourceSuppressed. Уведомляет о завершении установки состояния подавления по источнику.

## 1.1.4.2.4.3.6.1.5. GetAreaStateFinished

Сигнал о завершении GetAreaState. Уведомляет о завершении запроса состояния зон.

## 1.1.4.2.4.3.6.1.6. GetSourceStateFinished

Сигнал о завершении GetSourceState. Уведомляет о завершении запроса состояния источников.

## 1.1.4.2.4.3.6.1.7.

### **GetDisabledOrSuppressedAreasFinished**

Сигнал о завершении GetDisabledOrSuppressedAreas. Уведомляет о завершении запроса состояния заблокированных или подавленных зон.



## 1.1.4.2.4.3.6.1.8.

### **GetDisabledOrSuppressedSourcesFinished**

Сигнал о завершении `GetDisabledOrSuppressedSources`. Уведомляет о завершении запроса состояния заблокированных или подавленных источников.

## 1.1.4.2.4.3.6.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Источник</a>	Ссылка на компонент Источник AP

## 1.1.4.2.4.3.6.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.6.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.6.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.6.2.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.6.2.5. Источник АР

Ссылка на компонент "Источник АР".

## 1.1.4.2.4.3.6.3. Функции

Функция	Описание
<a href="#">SetAreaDisabled</a>	Заблокировать зоны, переданные в формате массива JSON
<a href="#">SetSourceDisabled</a>	Заблокировать источники, переданные в формате массива JSON.
<a href="#">SetAreaSuppressed</a>	Подавить уведомления от зоны до указанного времени (UTC).
<a href="#">SetSourceSuppressed</a>	Подавить уведомления от источника до указанного времени (UTC).
<a href="#">GetAreaState</a>	Получить состояния зон, переданных в формате массива JSON.
<a href="#">GetSourceState</a>	Получить состояния источников, переданных в формате массива JSON.
<a href="#">GetDisabledOrSuppressedAreas</a>	Получить список заблокированных или подавленных зон.
<a href="#">GetDisabledOrSuppressedSources</a>	Получить список заблокированных или подавленных источников.



## 1.1.4.2.4.3.6.3.1. SetAreaDisabled

Заблокировать зоны, переданные в формате массива JSON.



```
void SetAreaDisabled(string _1, bool _2)
```

## 1.1.4.2.4.3.6.3.2. SetSourceDisabled

Заблокировать источники, переданные в формате массива JSON.



```
void SetSourceDisabled(string _1, bool _2)
```

## 1.1.4.2.4.3.6.3.3. SetAreaSuppressed

Подавить уведомления от зоны до указанного времени (UTC).



```
void SetAreaSuppressed(string _1, bool _2, timestamp _3)
```

## 1.1.4.2.4.3.6.3.4. SetSourceSuppressed

Подавить уведомления от источника до указанного времени (UTC).



```
void SetSourceSuppressed(string _1, bool _2, timestamp _3)
```

## 1.1.4.2.4.3.6.3.5. GetAreaState

Получить состояния зон, переданных в формате массива JSON.



```
void GetAreaState(string _1)
```

## 1.1.4.2.4.3.6.3.6. GetSourceState

Получить состояния источников, переданных в формате массива JSON.



```
void GetSourceState(string _1)
```

## 1.1.4.2.4.3.6.3.7. GetDisabledOrSuppressedAreas

Получить список заблокированных или подавленных зон.



```
void GetDisabledOrSuppressedAreas(string _1)
```

## 1.1.4.2.4.3.6.3.8.

# GetDisabledOrSuppressedSources

Получить список заблокированных или подавленных источников.



```
void GetDisabledOrSuppressedSources(string _1)
```



## 1.1.4.2.4.3.7. Учетные данные AP

Компонент используется для подключения к защищенному паролем источнику данных в проекте Astra.HMI.

В качестве источника может использоваться, к примеру, Astra.Server.

## 1.1.4.2.4.3.7.1. Функции

Функция	Описание
<a href="#">SetPassword</a>	Позволяет указать пароль для подключения к защищенному источнику данных
<a href="#">Save</a>	Сохранить учетные данные
<a href="#">Load</a>	Загрузить учетные данные
<a href="#">GetLastError</a>	Получить последнюю ошибку

## 1.1.4.2.4.3.7.1.1. SetPassword

Позволяет указать пароль для подключения к защищенному источнику данных



```
void SetPassword(string Password)
```

### Параметры

Параметр	Тип	Описание
Password	string	Строковый параметр, содержащий пароль.

### Примеры

Указать пароль доступа «Qwerty1234-» для подключения к защищенному источнику данных:



```
ApCredentials_1.SetPassword("Qwerty1234-");
```

## 1.1.4.2.4.3.7.1.2. Save

Сохраняет пароль доступа в зашифрованном виде (хэш пароля) в файл, который указывается в свойстве "Путь к файлу с учетными данными".



bool Save()

### Значение

Значение	Описание
true	Сохранение учетных данных прошло успешно
false	Свойство "Путь к файлу с учетными данными" не заполнено или сохранение не удалось (например, файла по указанному пути не существует)

### Примеры

Сохранить пароль для подключения к защищенному источнику данных и вывести результат в текстовое поле:



```
Text_1 = "Save: " + (String.ToString(ApCredentials_1.Save()));
```

## 1.1.4.2.4.3.7.1.3. Load

Загружает пароль доступа из файла, указанного в свойстве "Путь к файлу с учетными данными".



bool Load()

### Значение

Значение	Описание
true	Загрузка пароля прошла успешно
false	Свойство "Путь к файлу с учетными данными" не заполнено или сохранение не удалось (например, файла по указанному пути не существует)

### Примеры

Загрузить пароль к защищенному источнику данных из файла и вывести результат в текстовое поле:



```
Text_1 = "Load: " + (String.ToString(ApCredentials_1.Load()));
```

## 1.1.4.2.4.3.7.1.4. GetLastError

Возвращает текстовое описание последней возникшей ошибки при выполнении операций с учетными данными. Если ошибок не было, функция вернет пустую строку.



string GetLastError()

## 1.1.4.2.4.3.7.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Путь к файлу с учетными данными</a>	Путь к файлу с учётными данными
<a href="#">Присвоены ли учетные данные</a>	Проверка присвоения учетных данных

## 1.1.4.2.4.3.7.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.3.7.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.7.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.7.2.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.7.2.5. Путь к файлу с учётными данными

Путь к файлу, в котором сохраняются с помощью функции "Save" учетные данные для подключения к защищенному источнику.



string FilePath



Пользователь самостоятельно создает файл и выбирает расширение файла на свое усмотрение (например, .txt, .doc, .tex, .text и другие). Также допустимо использование файла без расширения.

Если путь к файлу указан, пароль будет автоматически подгружен при запуске проекта в рантайме, позволяя осуществить автоматическую аутентификацию. Если путь не указан, пользователю придется вводить пароль вручную через интерфейсный элемент, например, текстовое поле, при каждом запуске проекта в рантайме.

### Примеры

Вывести в текстовое поле заданный путь к файлу с учетными данными.



```
Text_1.Text = ApCredentials_1.FilePath;
```

## 1.1.4.2.4.3.7.2.6. Присвоены ли учетные данные

Указывает на то, были ли предоставлены учетные данные компоненту "Учетные данные AP" в проекте Astra.HMI для подключения к защищенному источнику данных.



bool Assigned



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Учетные данные предоставлены
false	Учетные данные не предоставлены

### Примеры

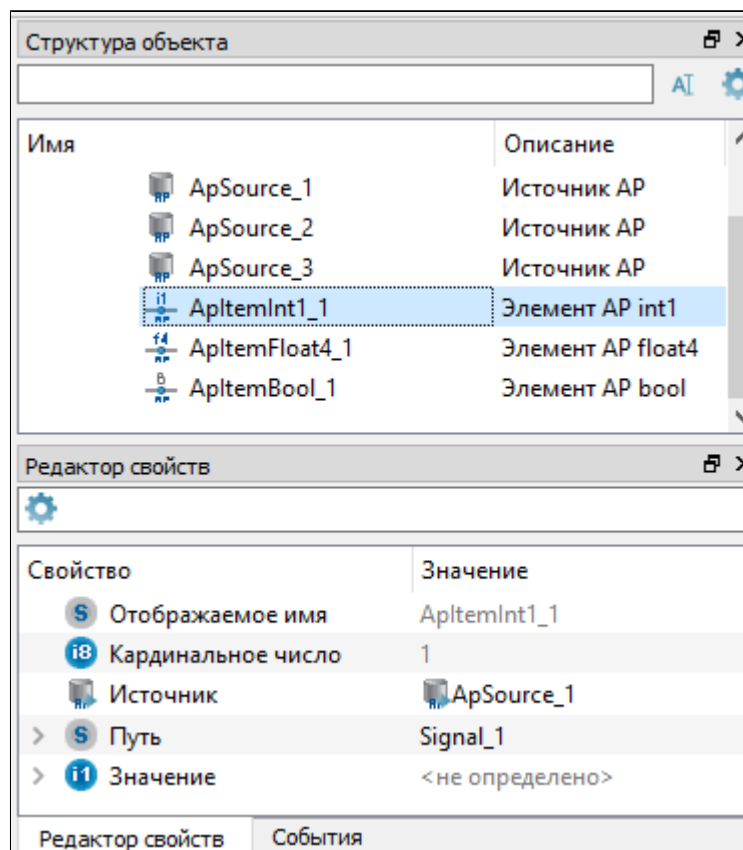
Проверить, предоставлены ли учетные данные компоненту "Учетные данные AP" для подключения к защищенному источнику. Результат вывести в текстовое поле.



```
Text_1.Text = String.ToString(ApCredentials_1.Assigned);
```

## 1.1.4.2.4.3.8. Элемент AP <type>

Для обеспечения совместимости типов и оптимального распределения ресурсов при взаимодействии с сигналами источника данных, необходимо учитывать тип сигнала на сервере. Каждому типу сигнала на сервере соответствует отдельный компонент "Элемент AP <type>", где <type> - тип сигнала на сервере. Если тип сигнала на сервере не известен, используйте универсальный компонент "Элемент AP", который оперирует универсальным типом данных "variant". После определения типов сигналов добавьте нужное количество компонентов "Элемент AP <type>" на экранную форму из расчета один сигнал – один элемент AP. Компонент невизуальный и виден только в области "Структура объекта".



Чтобы компонент "Элемент AP <type>" мог взаимодействовать с конкретным сигналом источника данных, настройте минимальный набор его свойств: Источник и Путь.

## 1.1.4.2.4.3.8.1. События

Событие	Описание
<a href="#">OnDescriptionChanged</a>	Изменение значения свойства сигнала Описание
<a href="#">OnUnitsChanged</a>	Изменение сигнального свойства 100 (EUnit)
<a href="#">OnTimeStampChanged</a>	Изменение метки времени сигнала
<a href="#">OnQualityChanged</a>	Изменение качества сигнала
<a href="#">OnValueChangedAsString</a>	Изменение значения сигнала строкового типа
<a href="#">OnValueChanged</a>	Изменение значения сигнала
<a href="#">OnAdviseStateChanged</a>	Изменение значения свойства Состояние подписки
<a href="#">OnVqtChanged</a>	Изменение значения, качества или метки
<a href="#">OnHistorizingChanged</a>	Изменение флага ведения истории

## 1.1.4.2.4.3.8.1.1. OnDescriptionChanged

Изменение значения свойства сигнала "Описание".

### Параметры

Параметры	Тип	Описание
description	string	Новое описание сигнала



## 1.1.4.2.4.3.8.1.2. OnAdviseStateChanged

Изменение значения свойства "Состояние подписки".

### Параметры

Параметр	Тип	Описание
adviseState	uint1	Новое состояние подписки. Соответствует свойству "Состояние подписки"

## 1.1.4.2.4.3.8.1.3. OnValueChanged

Изменение значения сигнала.

### Параметры

Параметр	Описание
value	Новое значение элемента

## 1.1.4.2.4.3.8.1.4. OnValueChangedAsString

Изменение значения сигнала строкового типа.

### Параметры

Параметр	Тип	Описание
value	string	Новое значение в виде строки

## 1.1.4.2.4.3.8.1.5. OnQualityChanged

Изменение качества сигнала.

### Параметры

Параметр	Тип	Описание
value	uint2	Новое значение качества

## 1.1.4.2.4.3.8.1.6. OnTimeStampChanged

Изменение метки времени сигнала.

### Параметры

Параметр	Описание
time	Новое значение времени

## 1.1.4.2.4.3.8.1.7. OnVqtChanged

Изменение значения, качества или метки времени.

### Параметры

Параметр	Тип	Описание
value	bool	Новое значение элемента.
quality	uint4	Новое значение качества.
time	timestamp	Новое значение метки времени.

## 1.1.4.2.4.3.8.1.8. OnUnitsChanged

Изменение сигнального свойства 100 (EUnit).

### Параметры

Параметр	Тип	Описание
units	string	Новое значение единиц измерения

## 1.1.4.2.4.3.8.1.9. OnHistorizingChanged

Изменение значения свойства "Флаг ведения истории".

### Параметры

Параметр	Тип	Описание
historizing	bool	Новое состояние флага ведения истории.



## 1.1.4.2.4.3.8.2. Функции

Функция	Описание
<a href="#">TrySetValue</a>	Записывает значение сигнала на сервер

## 1.1.4.2.4.3.8.2.1. TrySetValue

Записывает значение сигнала на сервер.



bool TrySetValue (Value)

### Параметры

Параметр	Тип	Описание
value	variant	Значение сигнала

### Значение

Значение	Описание
true	Значение успешно записано на сервер
false	Не удалось записать значение

### Примеры

Попытаться присвоить значение сигналу и вывести результат в текстовое поле:



```
Result: var = Apltem_1.TrySetValue("Строка 1");  
TextEdit_1.Text = String.ToString(Result);
```

## 1.1.4.2.4.3.8.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Источник</a>	Ссылка на компонент Источник AP
<a href="#">Путь</a>	Путь до сигнала относительно корня Источника
<a href="#">Значение</a>	Значение сигнала
<a href="#">Значение в виде строки</a>	Значение сигнала в строковом формате
<a href="#">Качество</a>	Качество сигнала
<a href="#">Метка времени</a>	Метка времени сигнала
<a href="#">Полный тег</a>	Тег сигнала
<a href="#">Оформлена ли подписка?</a>	Текущее состояние подписки на сигнал
<a href="#">Единицы измерения</a>	Единицы измерения сигнала
<a href="#">Описание</a>	Описание сигнала
<a href="#">Состояние подписки</a>	Код состояния подписки на сигнал
<a href="#">Строковое описание состояния подписки</a>	Описание состояния подписки на сигнал в текстовом виде
<a href="#">Флаг ведения истории</a>	Флаг ведения истории

<a href="#">Канонический тип данных</a>	Канонический тип данных
---	-------------------------

## 1.1.4.2.4.3.8.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.8.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.8.3.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.8.3.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.3.8.3.5. Источник

Ссылка на компонент "Источник AP", к дереву которого относится данный компонент "Элемент AP".

### Примеры

Записать в поле ввода сетевой адрес источника, к которому относится сигнал:



```
TextEdit.Text = ApItem.Source.Location;
```

## 1.1.4.2.4.3.8.3.6. Путь

Путь до сигнала относительно корня "Источника".



string Path

### Примеры

Записать путь к сигналу источника в поле ввода:



```
TextEdit.Text = AItem.Path;
```

## 1.1.4.2.4.3.8.3.7. Значение

Значение сигнала (сигнальное свойство 2 – Value). Тип значения зависит от типа компонента "Элемент AP <type>". Если используется нетипизированный компонент "Элемент AP", то для работы с данными применяется тип "variant".



<type> Value

## Примеры

Увеличить значение сигнала при выполнении условия:



```
if(SWStatus == 1){  
    ApItem.Value += 1;  
}
```

Записать значение сигнала в поле редактирования при выполнении условия:



```
if(SWStatus == 0){  
    TextEdit.Text = ApItem.Value;  
}
```

## 1.1.4.2.4.3.8.3.8. Значение в виде строки

Значение сигнала в строковом формате.



string ValueAsString



Доступно только для чтения в режиме рантайма.

### Примеры

Записать значение сигнала источника AP в поле ввода:



```
var string = ApItem.ValueAsString;  
Object.Text = string;
```

## 1.1.4.2.4.3.8.3.9. Качество

Качество сигнала (сигнальное свойство 3 - Quality).



uint2 Quality



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
COMM_FAILURE (24)	Отсутствие соединения с источником данных
UNCERTAIN (64)	Связь с источником данных установлена, но данные еще не получены
BAD_CONFIG (4)	Неверно заданные параметры сигнала

### Примеры

Записать качество сигнала источника AP в текстовое поле:



```
TextEdit_1.Text = String.ToString(ApItem_1.Quality);
```

## 1.1.4.2.4.3.8.3.10. Метка времени

Метка времени сигнала.



timestamp TimeStamp



Доступно только для чтения в режиме рантайма.

### Примеры

Записать метку времени сигнала источника AP в текстовое поле:



```
TextEdit_1.Text = String.ToString(ApItem_1.TimeStamp);
```

## 1.1.4.2.4.3.8.3.11. Полный тег

Тег сигнала - полный путь к сигналу, начиная от корня дерева сигналов.



string Tag



Доступно только для чтения в режиме рантайма.

### Примеры

Записать полный путь к сигналу источника AP в поле редактирования:



```
TextEdit.Text = Apltem.Tag;
```

## 1.1.4.2.4.3.8.3.12. Оформлена ли подписка?

Текущее состояние подписки на сигнал.



bool Advised



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Есть подписка на сигнал
false	Нет подписки на сигнал

### Примеры

Записать текущее состояние подписки на сигнал источника AP в поле ввода:



```
TextEdit_1.Text = String.ToString(ApItem_1.Advised);
```



## 1.1.4.2.4.3.8.3.13. Единицы измерения

Единицы измерения сигнала (сигнальное свойство 100 - EUnit).



string Units



Доступно только для чтения в режиме рантайма.

### Примеры

Записать единицы измерения сигнала источника AP в текстовое поле:



```
TextField.Text = ApItem.Units;
```

## 1.1.4.2.4.3.8.3.14. Описание

Описание сигнала (сигнальное свойство 101 - Description).



string Description



Доступно только для чтения в режиме рантайма.

### Примеры

Записать описание сигнала источника AP в текстовое поле:



```
TextField.Text = ApItem.Description;
```

## 1.1.4.2.4.3.8.3.15. Состояние подписки

Код состояния подписки на сигнал.



uint1 AdviseState



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
0	Не подписано (not advised)
1	Выполняется процесс подписки (advising)
2	Подписано (advised)
3	Выполняется процесс отписки (unadvising)
4	Ошибка подписки (advice error)
5	Ошибка отписки (unadvice error)

### Примеры

Записать код состояния подписки на сигнал источника AP в текстовое поле:



```
TextEdit_1.Text = String.ToString(Apltem_1.AdviseState);
```

## 1.1.4.2.4.3.8.3.16. Строковое описание состояния подписки

Описание состояния подписки на сигнал в текстовом виде.



string AdviseStateString



Доступно только для чтения в режиме рантайма.

### Примеры

Записать текст описания состояния подписки на сигнал источника AP в текстовое поле:



```
TextField.Text = APltem.AdviseStateString;
```

## 1.1.4.2.4.3.8.3.17. Флаг ведения истории

Активирована ли функция записи истории для сигнала на сервере.



bool Historizing



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	История сигнала записывается.
false	История сигнала не записывается.

## 1.1.4.2.4.3.8.3.18. Канонический тип данных

Тип сигнала на сервере.



uint1 CanonicalType



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.3.9. Запрос значений элемента AP

Тип запроса значений элемента AP.

## 1.1.4.2.4.3.9.1. Функции

Функция	Описание
<a href="#">Reload</a>	Перезагрузить данные



## 1.1.4.2.4.3.9.1.1. Reload

Применяет все заданные свойства у объекта типа "Запрос значений элемента AP". По запросу потребителя (например, компонента "Таблица: модель данных" из юнита "Таблицы") функция обновляет данные с источника данных, поставляющего оперативные и исторические данные.



```
void Reload()
```

### Примеры

Запросить оперативные данные с источника.



```
ApItemValueQuery_1.Mode = 2;  
ApItemValueQuery_1.Reload();
```

## 1.1.4.2.4.3.9.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Длина</a>	Размер массива (количество элементов в массиве)
<a href="#">Индекс</a>	Индекс элемента в массиве
<a href="#">Элемент AP</a>	Ссылка на элемент AP
<a href="#">Режим работы</a>	Режим работы
<a href="#">Начало исторического интервала</a>	Начало исторического интервала
<a href="#">Конец исторического интервала</a>	Конец исторического интервала
<a href="#">Активность</a>	Активность компонента
<a href="#">Оперативный интервал</a>	Оперативный интервал
<a href="#">Размер оперативного буфера</a>	Размер оперативного буфера
<a href="#">Предзагрузка исторических данных</a>	Предзагрузка исторических данных

## 1.1.4.2.4.3.9.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.3.9.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.3.9.2.3. Длина

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры

Записать в текстовое поле длину массива:



```
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.3.9.2.4. Индекс

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры

Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:



```
TextEdit_2.Text = String.ToString(Index);
```

Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:



```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:



```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.3.9.2.5. Элемент AP

Ссылка на "Элемент AP".

## 1.1.4.2.4.3.9.2.6. Режим работы

Режим работы компонента в зависимости от вида запрашиваемых значений.



uint4 Mode

### Значение

Значение	Описание
1	Исторические значения
2	Оперативные значения



## 1.1.4.2.4.3.9.2.7. Начало исторического интервала

Начальное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное значение переведите в формат UTC. Для перевода вы можете использовать функции "Parse" и "ToUtc" языка Astra.Om.



timestamp HistoricalRangeStart



Свойство используется только при запросе исторических данных (когда в свойстве "Режим работы" указано значение "1").



Начальной или конечной точкой интервала запроса может быть "0". Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

## Примеры

Запросить исторические события с 22.07.2022 20:00:25 по 25.07.2022 21:17:25.



```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");  
DateEnd: uint8 = DateTime.Pars ("25.07.2022 21:17:25");  
Events.TableModellvents.QuerySource1.HistoricalRangeStart =  
DateTime.ToUtc(DateStart);  
Events.TableModellvents.QuerySource1.HistoricalRangeEnd =  
DateTime.ToUtc(DateEnd);  
Events.TableModellvents.QuerySource1.Reload();  
Events.TableModellvents.BeginReadAsync();
```

## 1.1.4.2.4.3.9.2.8. Конец исторического интервала

Конечное значение интервала запроса исторических данных от источника. Перед указанием значения переведите ваше строковое значение даты в метку времени формата uint8, а затем полученное значение переведите в формат UTC. Для перевода вы можете использовать функции "Parse" и "ToUtc" языка Astra.Om.



timestamp HistoricalRangeEnd



Свойство используется только при запросе исторических данных (когда в свойстве "Режим работы" указано значение "1").



Начальной или конечной точкой интервала запроса может быть "0". Это значит, что данные будут запрошены за бесконечный интервал в прошлом.

## Примеры

Запросить исторические события с 22.07.2022 20:00:25 по 25.07.2022 21:17:25.



```
DateStart: uint8 = DateTime.Pars ("22.07.2022 20:00:25");  
DateEnd: uint8 = DateTime.Pars ("25.07.2022 21:17:25");  
Events.TableModellvents.QuerySource1.HistoricalRangeStart =  
DateTime.ToUtc(DateStart);  
Events.TableModellvents.QuerySource1.HistoricalRangeEnd =  
DateTime.ToUtc(DateEnd);  
Events.TableModellvents.QuerySource1.Reload();  
Events.TableModellvents.BeginReadAsync();
```

## 1.1.4.2.4.3.9.2.9. АКТИВНОСТЬ

Активация компонента. С неактивным компонентом не происходит обмен данными.



bool Active

## 1.1.4.2.4.3.9.2.10. Оперативный интервал

Общий период времени (в секундах), за который предварительно читаются и подготавливаются к отображению данные.



uint4 OperativeInterval

## 1.1.4.2.4.3.9.2.11. Размер оперативного буфера

Период времени (в секундах), отображаемый на трендовом поле.



uint4 OperativeBufferSize

## 1.1.4.2.4.3.9.2.12. Предзагрузка исторических данных

Включение/отключение подгрузки исторических данных на графики перед отображением новых значений.



bool PreloadHistoricalData

## 1.1.4.2.4.4. Графики

Чтобы получить возможность представлять данные, принятые от источника, в виде графиков, следует подключить внешнюю библиотеку `astra.hmi.charts` (для ОС Windows) и `libastra.hmi.charts` (для ОС Linux).

Чтобы подключить библиотеки:

- › ОС Windows

Разместите файл библиотеки в папке установки Astra.HMI.

- › ОС Linux

Скопируйте библиотеку в папку Astra.HMI командой:



```
cp <имя библиотеки>.so /opt/AstraRegul/Astra.HMI
```

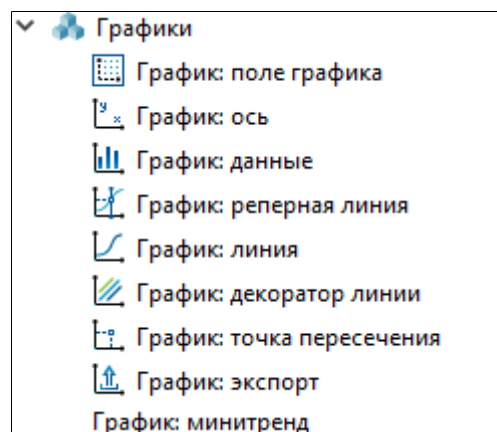


Версия SDK, на которой выполнялась сборка библиотеки, должна быть такой же, какая используется в Astra.HMI.

После подключения библиотеки в библиотеку компонентов Astra.HMI добавится юнит Графики с компонентами:

Компоненты:

- › [График: поле графика](#)
- › [График: ось](#)
- › [График: данные](#)
- › [График: реперная линия](#)
- › [График: линия](#)
- › [График: декоратор линии](#)
- › [График: точка пересечения](#)



- > [График: экспорт](#)
- > [График: минитренд](#)

## Настройка получения данных от источника

В качестве источника, к примеру, используйте Astra.Server, с которого можно запросить оперативные и исторические значения по TCP.

Чтобы настроить получение данных от источника, выполните:

1. Добавьте компонент Источник AP. Задайте значения в свойствах Хост, Порт, Порт истории (если планируете запрашивать исторические значения), Активность.

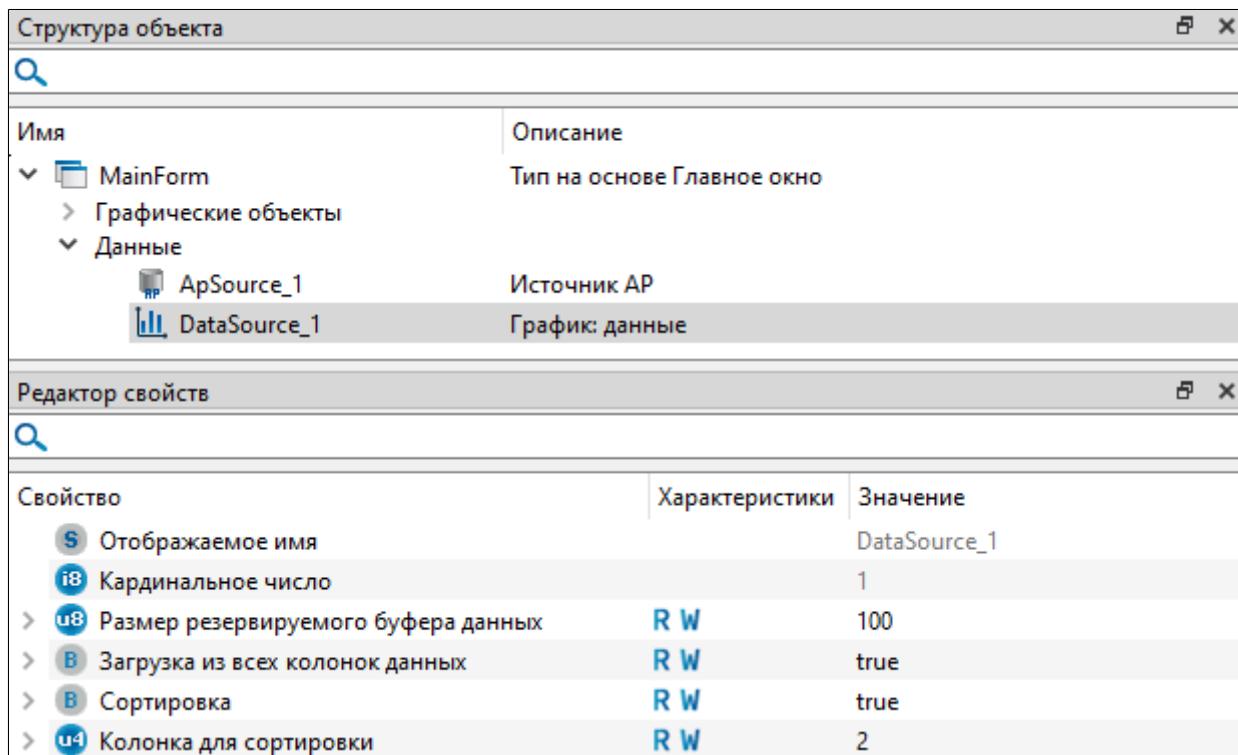
The screenshot shows two windows from a software application:

- Структура объекта (Object Structure):** A tree view showing the hierarchy of components. Under 'MainForm' > 'Графические объекты' > 'Данные', the component 'ApSource\_1' is selected, with the description 'Источник AP'.
- Редактор свойств (Properties Editor):** A table listing the properties of the selected component. The properties are:
 

Свойство	Характеристики	Значение
Отображаемое имя		ApSource_1
Кардинальное число		1
Родительский источник	R W → 0	<не определено>
Источник учетных данных	R W → 0	<не определено>
Таймаут операций с БД	R ∞	<не определено>
Таймаут установления соединения с БД	R ∞	<не определено>
Максимальный размер пула буферов	R ∞	<не определено>
Общее соединение	R ∞	<не определено>
> Хост	R W	localhost
> Порт	R W	4388
> Порт истории	R W	4950
> Путь	R W	
> Активность	R W	true
> Период переподписки	R W	0
> Имя клиента	R W	HMI
> Идентификатор клиента	R W	HMI



2. Добавьте компонент График: данные. С помощью компонента можно оперировать значениями только одного сигнала. Вы можете использовать компонент в качестве контейнера для объединения компонентов, работающих с одной линией.



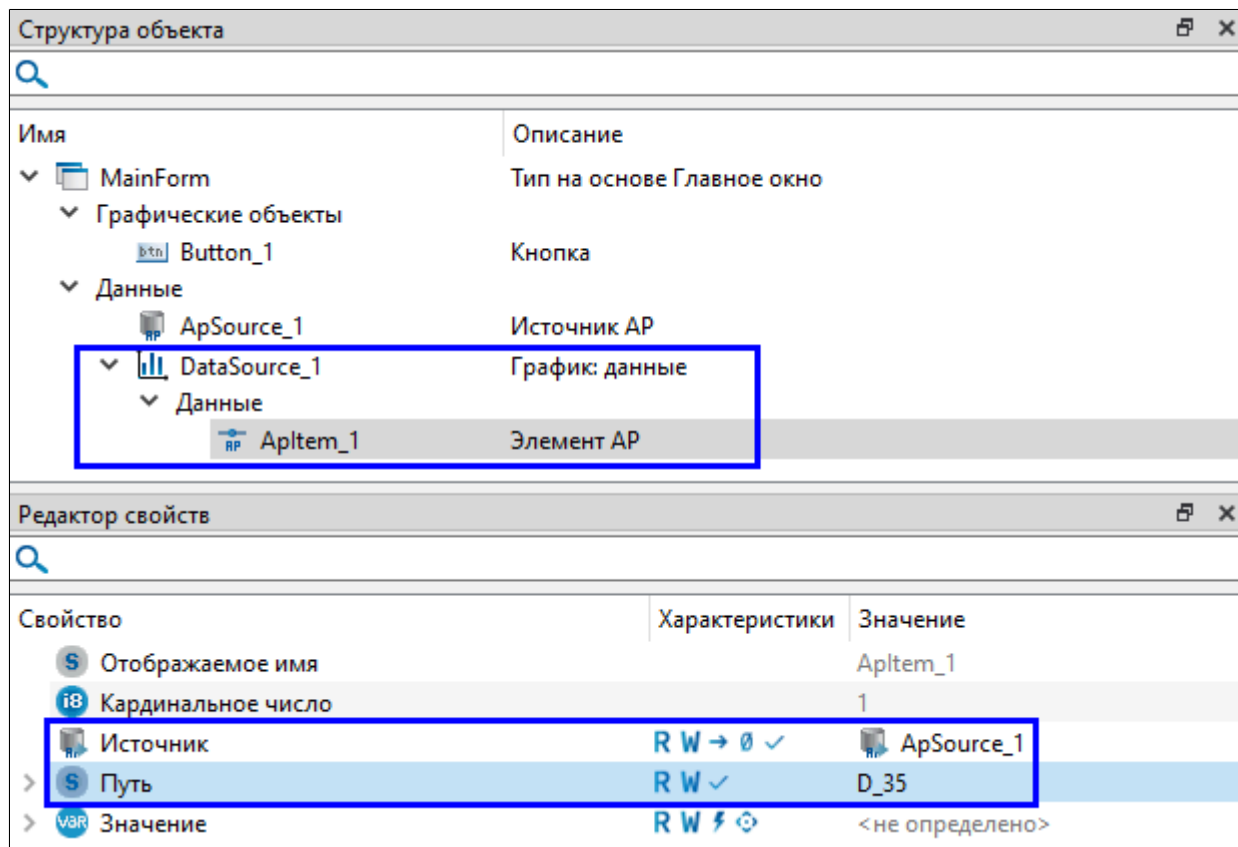
Структура объекта

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Данные	
ApSource_1	Источник AP
DataSource_1	График: данные

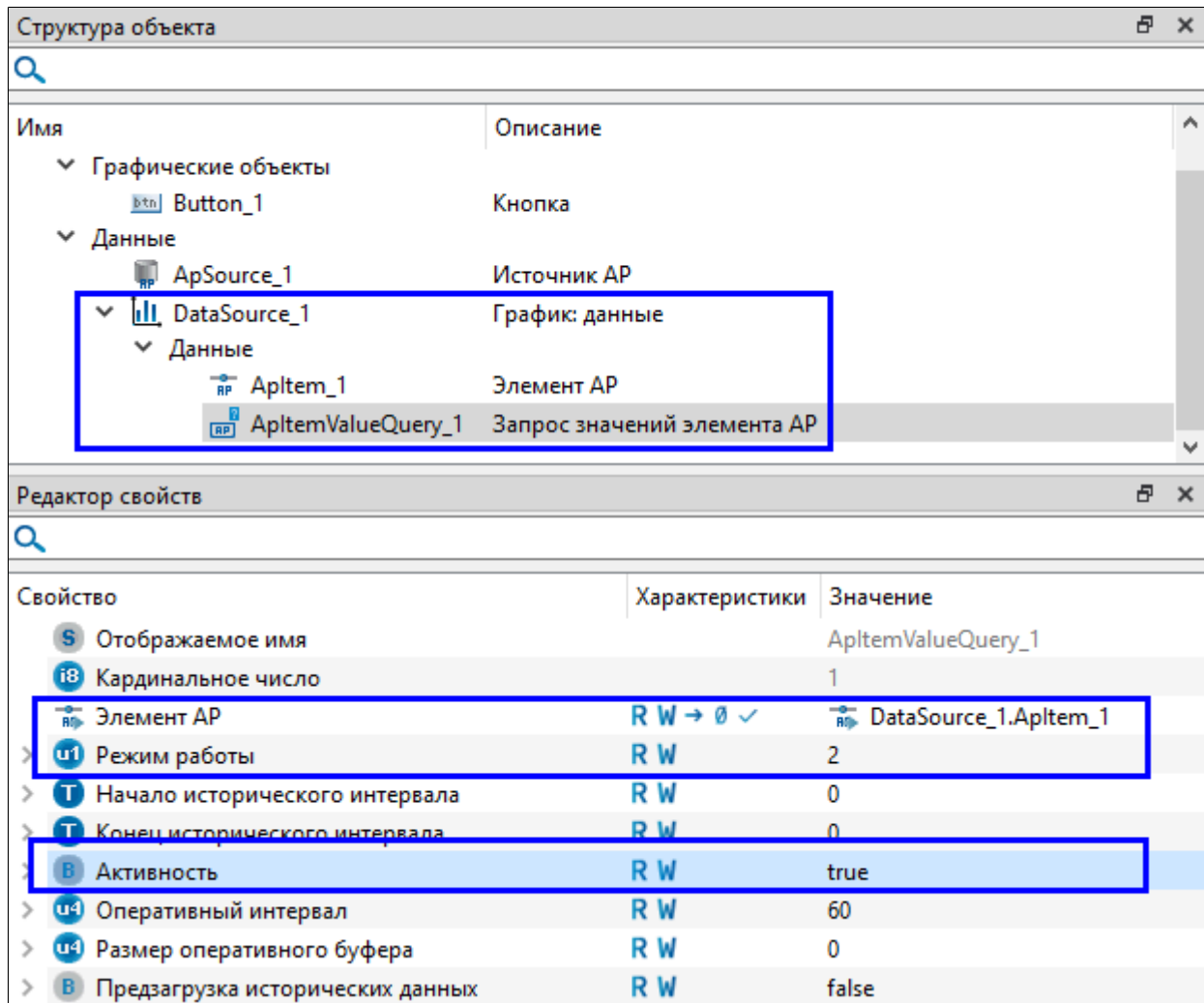
Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		DataSource_1
Кардинальное число		1
Размер резервируемого буфера данных	R W	100
Загрузка из всех колонок данных	R W	true
Сортировка	R W	true
Колонка для сортировки	R W	2

3. Добавьте компонент Элемент AP дочерним компоненту График: данные. Укажите источник данных в свойстве Источник и привяжите компонент к сигналу в свойстве Путь.



4. Добавьте компонент Запрос значений элемента AP дочерним компоненту График: данные. Укажите элемент (сигнал), с которого компонент будет принимать данные, в свойстве Элемент AP и активируйте компонент в свойстве Активность. Задайте режим работы компонента в зависимости от вида запрашиваемых значений (1 – исторические значения, 2 – оперативные значения).



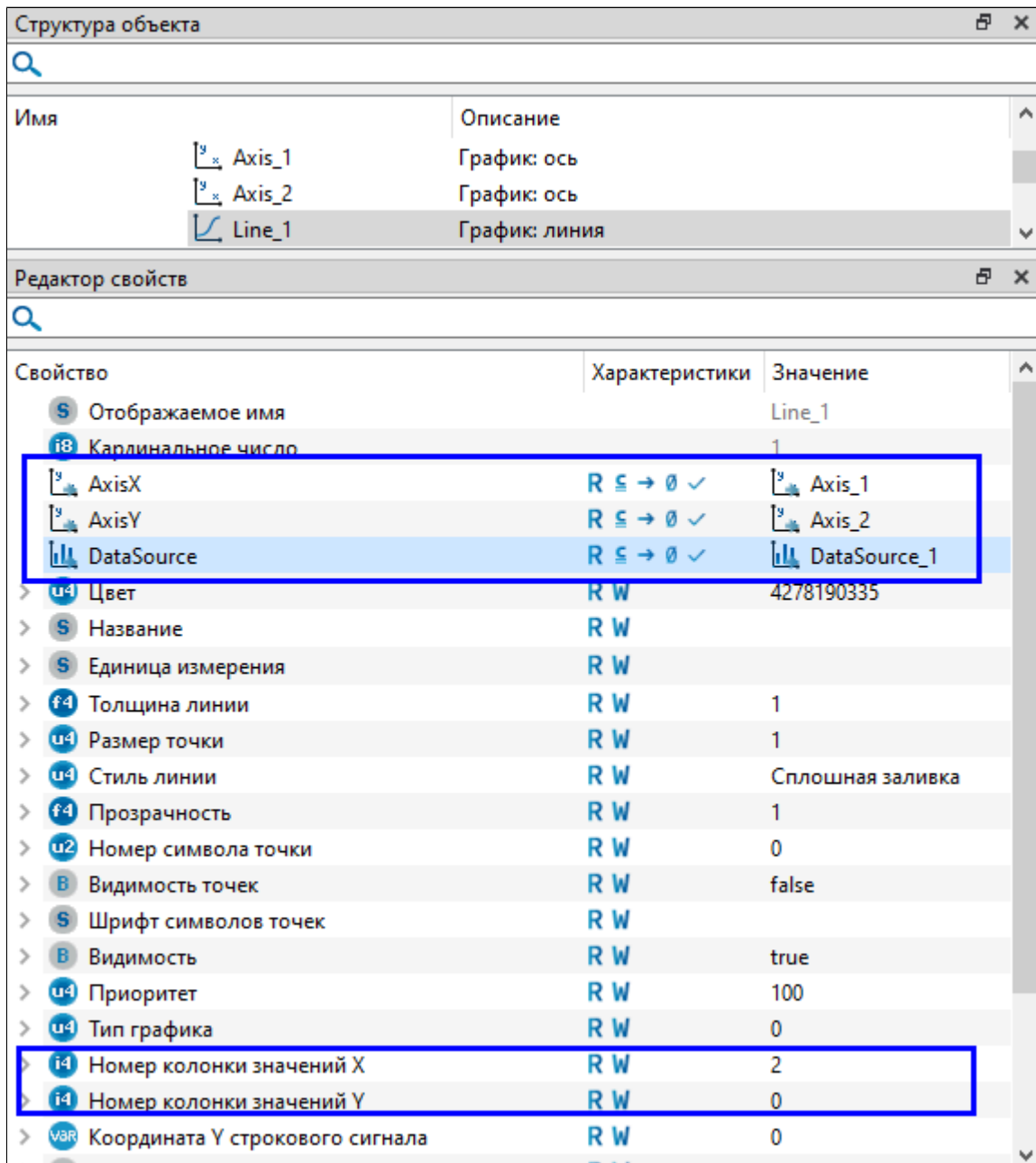
## Отображение одного графика на экранной форме

Чтобы вывести значения, полученные от источника, в трендовом виде, выполните:

1. Добавьте компонент График: поле графика. Настройте его внешний вид с помощью свойств.
2. Добавьте компоненты График: ось по количеству нужных осей (минимальное количество - 2) дочерними компоненту График: поле графика. Задайте значения свойства Ориентация оси для каждой оси.
3. Добавьте компонент График: линия. Поскольку компонент отвечает за вывод графика только по одному сигналу, для удобства можно расположить компонент дочерним для компонента График: данные.

Для линии:

1. Привяжите линию к сигналу с помощью свойства DataSource;
2. Укажите какие именно данные сигнала будут выводиться на графике в свойствах Номер колонки значений X и Номер колонки значений Y. Указывайте номера колонок, используемых в источнике: 0 – колонка значений сигнала (value), 2 – значения времени (timestamp), 2 – значения качества (quality);
3. Привяжите линию к осям с помощью свойств AxisX и AxisY.



4. Чтобы отобразить график оперативных данных, пропишите скрипт, который будет выполнять запрос оперативных данных от источника (функция Reload() компонента Запрос значений элемента AP) и

чение полученных данных компонентом График: данные (функция `BeginReadAsync()` компонента График: данные).

Чтобы отобразить график исторических данных, укажите интервал запроса в свойствах Начало исторического интервала и Конец исторического интервала компонента Запрос значений элемента AP.

## Отображение нескольких графиков на экранной форме

Вложите подготовленный компонент График: данные в компонент График: поле графика. Добавьте компоненту График: поле графика столько дочерних компонентов График: данные, сколько графиков вам необходимо отобразить на трендовом поле.

Для получения данных по одному сигналу и отображения их в виде графика один компонент График: данные должен быть связанным или содержать:

- Один компонент Запрос значений элемента AP.
- Один компонент График: линия (может не быть дочерним).
- Один компонент Элемент AP (может не быть дочерним).

## 1.1.4.2.4.4.1. График: поле графика

Компонент позволяет подготовить область для отображения графиков.

## 1.1.4.2.4.4.1.1. События

Событие	Описание
<a href="#">OnWheelMouse</a>	Вращение колеса мыши
<a href="#">MouseDown</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок мыши по объекту

## 1.1.4.2.4.4.1.1.1. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.4.1.1.2. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.4.1.1.3. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.4.1.1.4. MouseDoubleClick

Двойной щелчок мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.4.1.1.5. OnWheelMouse

Вращение колеса мыши.

### Параметры

Параметр	Тип	Описание
steps	float	Шаги поворота (знак указывает направление)
left	float	Координата курсора по оси X
top	float	Координата курсора по оси Y
keyboardmodifier	int	Коды клавиш клавиатуры, которые были зажаты при возникновении события

## 1.1.4.2.4.4.1.1.6. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.4.1.1.7. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.4.1.2. Функции

Функция	Описание
<a href="#">Update</a>	Обновляет (перерисовывает) график
<a href="#">ResetZoom</a>	Восстанавливает исходный масштаб графика



## 1.1.4.2.4.4.1.2.1. Update

Обновляет (перерисовывает) график.



Update()

## 1.1.4.2.4.4.1.2.2. ResetZoom

Восстанавливает исходный масштаб графика.



ResetZoom()

## 1.1.4.2.4.4.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Размер точки курсора</a>	Размер точки курсора на графике (в пикселях)
<a href="#">Толщина линии курсора</a>	Толщина линии курсора на графике
<a href="#">Режим курсора</a>	Режим курсора
<a href="#">Цвет фона</a>	Цвет фона графика
<a href="#">Цвет курсора</a>	Цвет курсора на графике
<a href="#">Цвет активной реперной линии</a>	Цвет активной реперной линии графика
<a href="#">Толщина активной реперной линии</a>	Толщина активной реперной линии графика
<a href="#">Стиль линии курсора</a>	Стиль линии курсора на графике
<a href="#">Таймлайн</a>	Скрытие/отображение панели таймлайна
<a href="#">Легенда</a>	Скрытие/отображение легенды на графике

<a href="#">Выравнивание легенды</a>	Положение поля легенды на графике
<a href="#">Экстраполяция</a>	Разрешение/запрет продолжения линии до конца выбранного интервала
<a href="#">Перемещать выноски репера</a>	Разрешение/запрет перемещения выносок репера
<a href="#">Шрифт</a>	Шрифт элемента
<a href="#">Коэффициент масштабирования по X, %</a>	Коэффициент масштабирования графика по оси X в процентах
<a href="#">Коэффициент масштабирования по Y, %</a>	Коэффициент масштабирования графика по оси Y в процентах
<a href="#">Управление выравниванием</a>	Включение/отключение автоматического выравнивания по оси X при приближении ползунков таймлайна к предельным положениям
<a href="#">Автообновление</a>	Разрешение/запрет автоматического обновления графика при изменении данных
<a href="#">Z-значение</a>	Z-значение
<a href="#">Способ размещения метки репера</a>	Способ размещения метки репера
<a href="#">Правая граница экстраполяции</a>	Правая граница экстраполяции
<a href="#">Минимум минитренда</a>	Минимум минитренда
<a href="#">Максимум минитренда</a>	Максимум минитренда
<a href="#">Отступ</a>	Отступ от границы компонента
<a href="#">Окно настроек</a>	Показывать окно настроек
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали

## 1.1.4.2.4.4.1.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.4.1.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.1.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.4.1.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

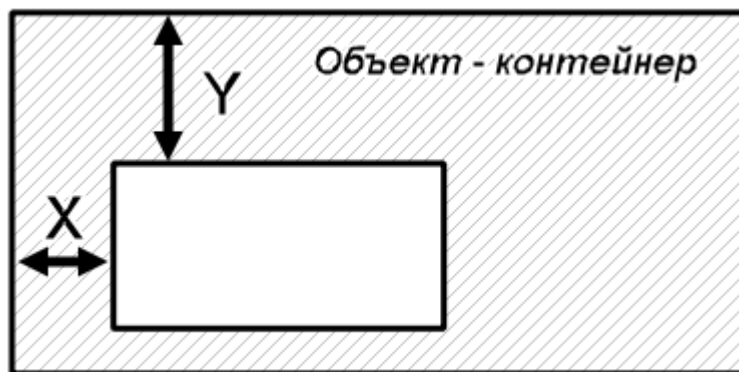
## 1.1.4.2.4.4.1.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.4.1.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.4.1.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.4.1.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.4.1.3.10. Видимость

Видимость объекта на форме.




bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.4.1.3.11. Включено

Активность объекта.


 bool Enabled



### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры

 //Активировать объект  
Object.Enabled = true;

## 1.1.4.2.4.4.1.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

### Примеры




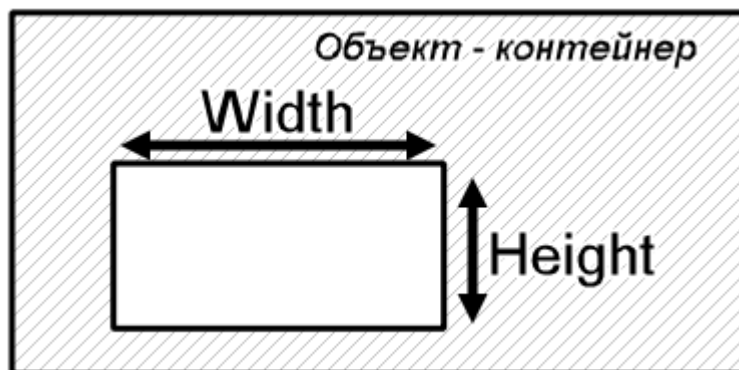
```
//Задать всплывающую подсказку для объекта  
Object.Tooltip = "Задвижка на всасе насоса";
```



# 1.1.4.2.4.4.1.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

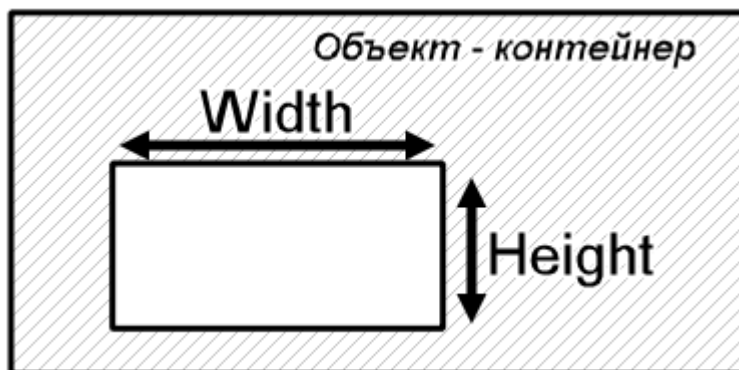


## 1.1.4.2.4.4.1.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.4.1.3.15. Размер точки курсора

Размер точки курсора на графике (в пикселях).



uint4 CursorPointSize

## 1.1.4.2.4.4.1.3.16. Толщина линии курсора

Толщина линии курсора на графике.



uint4 CursorLineWidth

## 1.1.4.2.4.4.1.3.17. Режим курсора

Режим курсора.



uint4 CursorMode

### Значение

Значение	Описание
0	Перемещение оси зажатой правой кнопкой мыши (значение по умолчанию)
1	Масштабирование выделенной области
2	Перемещение графика в нужном направлении
3	Добавление, выделение, перемещение реперов

## 1.1.4.2.4.4.1.3.18. Цвет фона

Цвет фона графика.



uint4 BrushColor

## 1.1.4.2.4.4.1.3.19. Цвет курсора

Цвет курсора на графике.



uint4 CursorColor

## 1.1.4.2.4.4.1.3.20. Цвет активной реперной линии

Цвет активной реперной линии графика.



uint4 ActiveControllLineColor



## 1.1.4.2.4.4.1.3.21. Толщина активной реперной линии

Толщина активной реперной линии графика.



uint4 ActiveControllLineWidth

## 1.1.4.2.4.4.1.3.22. Способ размещения метки репера

Способ размещения метки репера.



uint4 ControlLineAxisLablesLocation

## 1.1.4.2.4.4.1.3.23. Стиль линии курсора

Стиль линии курсора на графике.



uint4 CursorLineStyle

## 1.1.4.2.4.4.1.3.24. Таймлайн

Скрытие/отображение панели таймлайна.



bool ShowTimeLine

### Значение

Значение	Описание
true	Панель таймлайна отображается
false	Панель таймлайна скрыта

## 1.1.4.2.4.4.1.3.25. Легенда

Скрытие/отображение легенды на графике.



bool ShowLegend

### Значение

Значение	Описание
true	Легенда отображается
false	Легенда скрыта

## 1.1.4.2.4.4.1.3.26. Выравнивание легенды

Положение поля легенды на графике.



uint4 AlignLegend

## 1.1.4.2.4.4.1.3.27. Экстраполяция

Разрешение/запрет продолжения линии до конца выбранного интервала.



bool Extrapolation

### Значение

Значение	Описание
true	Экстраполяция включена
false	Экстраполяция выключена

## 1.1.4.2.4.4.1.3.28. экстраполяции

Правая

граница

Правая граница экстраполяции.



variant RightExtrapolationLimit



## 1.1.4.2.4.4.1.3.29. Минимум минитренда

Минимум минитренда.



variant MinTimeLine

## 1.1.4.2.4.4.1.3.30. Максимум минитренда

Максимум минитренда.



variant MaxTimeLine

## 1.1.4.2.4.4.1.3.31. Перемещать выноски репера

Разрешение/запрет перемещения выносок репера.



bool MovePointLables

### Значение

Значение	Описание
true	Разрешено перемещать выноски
false	Запрещено перемещать выноски

# 1.1.4.2.4.4.1.3.32. Шрифт

Шрифт элемента.



string Font

# 1.1.4.2.4.4.1.3.33. масштабирования по X, %

## Коэффициент

Коэффициент масштабирования графика по оси X в процентах.



uint4 ZoomX

## 1.1.4.2.4.4.1.3.34. масштабирования по Y, %

## Коэффициент

Коэффициент масштабирования графика по оси Y в процентах.



uint4 ZoomY

## 1.1.4.2.4.4.1.3.35. Управление выравниванием

Включение/отключение автоматического выравнивания по оси X при приближении ползунков таймлайна к предельным положениям.



bool EnableScaleAxisX

### Значение

Значение	Описание
true	Автоматическое выравнивание включено
false	Автоматическое выравнивание выключено

## 1.1.4.2.4.4.1.3.36. Автообновление

Разрешение/запрет автоматического обновления графика при изменении данных.



bool AutoUpdate

### Значение

Значение	Описание
true	Автоматическое обновление графика разрешено
false	Автоматическое обновление графика запрещено



## 1.1.4.2.4.4.1.3.37. Отступ

Отступ от границы компонента.



string Padding

## 1.1.4.2.4.4.1.3.38. Окно настроек

Показывать окно настроек.



bool ShowWinOptions

## 1.1.4.2.4.4.2. График: ось

Компонент позволяет добавить на трендовое поле оси для отображения графиков в определенном масштабе.

## 1.1.4.2.4.4.2.1. События

Событие	Описание
<a href="#">OnRangeChange</a>	Изменение диапазона выводимых данных
<a href="#">OnClickValue</a>	Выбор значения репера щелчком мыши
<a href="#">OnWheelMouse</a>	Вращение колеса мыши

## 1.1.4.2.4.4.2.1.1. OnRangeChange

Изменение диапазона выводимых данных.

### Параметры

Параметр	Тип	Описание
Min	float	Минимальное значение диапазона
Max	float	Максимальное значение диапазона

## 1.1.4.2.4.4.2.1.2. OnClickValue

Выбор значения репера щелчком мыши.

### Параметры

Параметр	Тип	Описание
clickValue	variant	Координата курсора по оси, соответствующая месту щелчка мыши

## 1.1.4.2.4.4.2.1.3. OnWheelMouse

Вращение колеса мыши.

### Параметры

Параметр	Тип	Описание
steps	float	Шаги поворота (знак указывает направление)
left	float	Координата курсора по оси X
top	float	Координата курсора по оси Y
keyboardmodifier	int	Коды клавиш клавиатуры, которые были зажаты при возникновении события

## 1.1.4.2.4.4.2.2. Функции

Функция	Описание
<a href="#">GetID</a>	Возвращает ID оси
<a href="#">GetAxisWidthCalc</a>	Вычисленная ширина оси
<a href="#">SetInterval</a>	Устанавливает интервал оси
<a href="#">SetAxisType</a>	Устанавливает ориентацию оси



## 1.1.4.2.4.4.2.2.1. GetID

Возвращает ID оси.



GetID()

## 1.1.4.2.4.4.2.2.2. GetAxisWidthCalc

Вычисленная ширина оси.



```
int4 GetAxisWidthCalc()
```

## 1.1.4.2.4.4.2.2.3. SetInterval

Устанавливает интервал оси.



```
void SetInterval (variant MinValue, variant MaxValue)
```

### Параметры

Параметр	Тип	Описание
MinValue	variant	Минимальное значение
MaxValue	variant	Максимальное значение

## 1.1.4.2.4.4.2.2.4. SetAxisType

Устанавливает ориентацию оси.



```
void setAxisType(uint4 Type)
```

### Параметры

Параметр	Тип	Описание
Type	uint4	Значения: <ul style="list-style-type: none"><li>› 1 – горизонтальная</li><li>› 2 – вертикальная</li></ul>

## 1.1.4.2.4.4.2.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Коэффициент масштабирования, %</a>	Коэффициент масштабирования оси в процентах
<a href="#">Видимость оси</a>	Скрытие/отображение оси
<a href="#">Положение названия оси</a>	Положение названия оси
<a href="#">Показывать единицы измерения</a>	Скрытие/отображение единиц изменения
<a href="#">Название оси</a>	Название, отображаемое рядом с осью
<a href="#">Максимум</a>	Максимальное значение отображаемых данных
<a href="#">Минимум</a>	Минимальное значение отображаемых данных
<a href="#">Формат метки времени</a>	Формат меток времени на оси
<a href="#">Формат меток</a>	Формат меток на оси
<a href="#">Положение единиц измерения</a>	Положение единиц измерения на оси
<a href="#">Единицы измерения</a>	Единицы измерения отображаемых данных
<a href="#">Логарифмическая шкала</a>	Скрытие/отображение логарифмической шкалы

<a href="#">Тип масштабирования</a>	Тип масштабирования оси
<a href="#">Количество делений</a>	Количество делений на оси
<a href="#">Тип данных</a>	Тип данных оси
<a href="#">Выравнивание</a>	Положение оси относительно поля графика
<a href="#">Ориентация оси</a>	Ориентация оси
<a href="#">Цвет</a>	Цвет элемента
<a href="#">Шрифт</a>	Шрифт элемента
<a href="#">Показывать граничные значения</a>	Показывать граничные значения
<a href="#">Автоматическое назначение ширины оси</a>	Вычисление ширины оси в зависимости от диапазона
<a href="#">Ширина оси</a>	Ширина оси при выключении AutoWidth

## 1.1.4.2.4.4.2.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.2.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.4.2.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.2.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.4.2.3.5. Цвет

Цвет элемента.



uint4 Color

## 1.1.4.2.4.4.2.3.6. Ориентация оси

Ориентация оси.



uint4 Type

### Значение

Значение	Описание
1	Горизонтальная
2	Вертикальная

## 1.1.4.2.4.4.2.3.7. Выравнивание

Положение оси относительно поля графика.



uint4 Align

### Значение

Значение	Описание
1	Слева для вертикальной оси, снизу для горизонтальной
2	Справа для вертикальной оси, сверху для горизонтальной

## 1.1.4.2.4.4.2.3.8. Тип данных

Тип данных оси.



uint4 DataType

### Значение

Значение	Описание
0	Числовые значения
2	Время

## 1.1.4.2.4.4.2.3.9. Количество делений

Количество делений на оси.



uint4 Ticks

## 1.1.4.2.4.4.2.3.10. Показывать граничные значения

Показывать граничные значения.



bool ShowBoundaryValues



## 1.1.4.2.4.4.2.3.11. Тип масштабирования

Тип масштабирования оси.



uint4 Scale

### Значение

Значение	Описание
0	Автоматическое масштабирование отключено
1	Полное автоматическое масштабирование. Размер оси может изменяться в зависимости от отображаемых данных: как в сторону расширения, так и в сторону уменьшения
2	Автоматическое расширение масштаба. Размер оси может изменяться только в сторону увеличения

## 1.1.4.2.4.4.2.3.12. Логарифмическая шкала

Скрытие/отображение логарифмической шкалы (деления на шкале проставлены в соответствии со степенями числа 10).



bool ScaleLog

### Значение

Значение	Описание
true	Шкала отображается
false	Шкала скрыта

## 1.1.4.2.4.4.2.3.13. Единицы измерения

Единицы измерения отображаемых данных.



string Unit

## 1.1.4.2.4.4.2.3.14. измерения

## Положение

## единиц

Положение единиц измерения на оси.



uint4 UnitPos

## Значение

Значение	Описание
0	Слева для горизонтальной оси, снизу для вертикальной
1	По центру
2	Справа для горизонтальной оси, сверху для вертикальной

## 1.1.4.2.4.4.2.3.15. Формат меток

Формат меток на оси.



string Format

## 1.1.4.2.4.4.2.3.16. Формат метки времени

Формат меток времени на оси.



string FormatT

## 1.1.4.2.4.4.2.3.17. Минимум

Минимальное значение отображаемых данных.



variant Min

## 1.1.4.2.4.4.2.3.18. Максимум

Максимальное значение отображаемых данных.



variant Max



## 1.1.4.2.4.4.2.3.19. Название оси

Название, отображаемое рядом с осью.



string AxisName

## 1.1.4.2.4.4.2.3.20. Показывать единицы измерения

Скрытие/отображение единиц измерения.



bool ShowUnit

### Значение

Значение	Описание
true	Единицы измерения видны
false	Единицы измерения скрыты

## 1.1.4.2.4.4.2.3.21. Положение названия оси

Положение названия оси.



uint4 AxisNamePos

### Значение

Значение	Описание
0	Слева для горизонтальной оси, снизу для вертикальной
1	По центру
2	Справа для горизонтальной оси, сверху для вертикальной

## 1.1.4.2.4.4.2.3.22. Видимость оси

Скрытие/отображение оси.



bool ShowUnit

### Значение

Значение	Описание
true	Ось видна
false	Ось скрыта

## 1.1.4.2.4.4.2.3.23. Шрифт

Шрифт элемента.



string Font

## 1.1.4.2.4.4.2.3.24. масштабирования, %

## Коэффициент

Коэффициент масштабирования оси в процентах.



uint4 Zoom

## 1.1.4.2.4.4.2.3.25. Автоматическое назначение ширины оси

Вычисление ширины оси в зависимости от диапазона.



bool AutoWidth

## 1.1.4.2.4.4.2.3.26. Ширина оси

Ширина оси при выключении AutoWidth.



uint4 AxisWidth



## 1.1.4.2.4.4.3. График: данные

Компонент, выполняющий чтение оперативных или исторических данных с источника.

## 1.1.4.2.4.4.3.1. События

Событие	Описание
<a href="#">SaveFinished</a>	Завершение сохранения данных
<a href="#">ReadFinished</a>	Завершение чтения данных
<a href="#">LoadFinished</a>	Завершение загрузки данных

## 1.1.4.2.4.4.3.1.1. SaveFinished

Завершение сохранения данных.

### Параметры

Параметр	Тип	Описание
errorCode	uint1	Код ошибки, возникшей при сохранении: <ul style="list-style-type: none"><li>› 0 – запись завершена успешно</li><li>› 1 – недостаточно прав для записи файла</li></ul>

## 1.1.4.2.4.4.3.1.2. ReadFinished

Завершение чтения данных.

### Параметры

Параметр	Тип	Описание
rowsRead	uint8	Количество прочитанных строк
elapsedMs	float	Время чтения данных (мсек)
errorCode	uint1	Код ошибки, возникшей при сохранении: <ul style="list-style-type: none"><li>› 0 – загрузка завершена успешно</li><li>› 1 – отсутствует указанный файл</li><li>› 1 – недостаточно прав для чтения файла</li></ul>
message	string	Сообщение

## 1.1.4.2.4.4.3.1.3. LoadFinished

Завершение загрузки данных.

## 1.1.4.2.4.4.3.2. Функции

Функция	Описание
<a href="#">Load</a>	Загружает данные из файла в формате CSV или JSON
<a href="#">Save</a>	Сохраняет данные в файл в формате CSV или JSON
<a href="#">Clear</a>	Очищает содержимое
<a href="#">EnableRT</a>	Разрешает/запрещает чтение данных в режиме рантайма
<a href="#">CancelReadAsync</a>	Отменяет обновление данных через IRowSet
<a href="#">BeginReadAsync</a>	Обновляет данных через IRowSet
<a href="#">LoadExtended</a>	Загружает данные из файла с параметрами
<a href="#">GetDescriptionByCode</a>	Возвращает текстовое описание кода ошибки

## 1.1.4.2.4.4.3.2.1. BeginReadAsync

Обновляет данных через IRowSet.



```
void BeginReadAsync()
```

## 1.1.4.2.4.4.3.2.2. CancelReadAsync

Отменяет обновление данных через IRowSet.



```
void CancelReadAsync()
```



## 1.1.4.2.4.4.3.2.3. Clear

Очищает содержимое.



```
void Clear()
```

## 1.1.4.2.4.4.3.2.4. EnableRT

Разрешает/запрещает чтение данных в режиме рантайма.



```
void EnableRT(bool EnableFlag)
```

### Параметры

Параметр	Тип	Описание
EnableFlag	bool	Флаг разрешения/запрещения чтения данных

## 1.1.4.2.4.4.3.2.5. Save

Сохраняет данные в файл в формате CSV или JSON.



```
void Save(string FilePath, string Format)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Полный путь к файлу
Format	string	Формат данных

## 1.1.4.2.4.4.3.2.6. Load

Загружает данные из файла в формате CSV или JSON.



```
void Load(string FilePath, string Format)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Полный путь к файлу
Format	string	Формат данных

## 1.1.4.2.4.4.3.2.7. LoadExtended

Загружает данные из файла с параметрами.



```
void LoadExtended(string FilePath, string Format, string Parameters)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Полный путь к файлу
Format	string	Формат данных
Parameters	string	Параметры загрузки

## 1.1.4.2.4.4.3.2.8. GetDescriptionByCode

Возвращает текстовое описание кода ошибки.



```
string GetDescriptionByCode(uint1 Code)
```

### Параметры

Параметр	Тип	Описание
Code	uint1	Код ошибки

## 1.1.4.2.4.4.3.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Size</a>	Максимальное количество точек, одновременно отображаемых на графике
<a href="#">Колонка для сортировки</a>	Номер колонки для сортировки
<a href="#">Сортировка</a>	Разрешение/запрет сортировки данных
<a href="#">Загрузка из всех колонок данных</a>	Загрузка данных из всех колонок либо только из указанных
<a href="#">Размер резервируемого буфера данных</a>	Размер буфера данных (в количестве точек)

## 1.1.4.2.4.4.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.4.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.4.3.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.3.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.4.3.3.5.      Размер      резервируемого буфера данных

Размер буфера данных (в количестве точек).



uint8 LengthReserved

## 1.1.4.2.4.4.3.3.6. Загрузка из всех колонок данных

Загрузка данных из всех колонок либо только из указанных.



bool LoadColumns

### Значение

Значение	Описание
true	Загружать данные из всех колонок
false	Загружать данные из колонок, указанных в компоненте <a href="#">График: линия</a>

## 1.1.4.2.4.4.3.3.7. Сортировка

Разрешение/запрет сортировки данных.



uint4 Sort

### Значение

Значение	Описание
true	Сортировка разрешена
false	Сортировка запрещена

## 1.1.4.2.4.4.3.3.8. Колонка для сортировки

Номер колонки для сортировки.



uint4 SortColumnNumber

## 1.1.4.2.4.4.3.3.9. Размер данных

Максимальное количество точек, одновременно отображаемых на графике.



uint4 Size



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.4.4. График: реперная линия

Компонент позволяет просматривать значения графика в точках пересечения линии с графиком сигнала.

## 1.1.4.2.4.4.1. События

Событие	Описание
<a href="#">ChangeValue</a>	Изменение значения
<a href="#">DataChanged</a>	Сигнал об изменении данных, доступных из CrossPoint
<a href="#">OnActiveStateChanged</a>	Сигнал об изменении активности репера

# 1.1.4.2.4.4.4.1.1. ChangeValue

Изменение значения.

## 1.1.4.2.4.4.1.2. DataChanged

Сигнал об изменении данных, доступных из CrossPoint.

## 1.1.4.2.4.4.4.1.3. OnActiveStateChanged

Сигнал об изменении активности репера.

## 1.1.4.2.4.4.2. Функции

Функция	Описание
<a href="#">setActive</a>	Устанавливает активность репера
<a href="#">getActive</a>	Возвращает активность репера
<a href="#">GetLine</a>	Установить Id линии для движения репера
<a href="#">SetLine</a>	Запрос Id линии для движения репера
<a href="#">GetAxisId</a>	Запрос ID оси репера
<a href="#">SetAxisById</a>	Назначить ось по ID

## 1.1.4.2.4.4.2.1. SetActive

Устанавливает активность репера.



```
void SetActive(bool ActiveFlag)
```

### Параметры

Параметр	Тип	Описание
ActiveFlag	bool	Возможные значения: <ul style="list-style-type: none"><li>› true – репер активен</li><li>› false – репер неактивен</li></ul>

## 1.1.4.2.4.4.2.2. getActive

Возвращает активность репера.



bool getActive()



## 1.1.4.2.4.4.2.3. GetLine

Установить Id линии для движения репера.



int4 GetLine()

## 1.1.4.2.4.4.2.4. SetLine

Запрос ID линии для движения репера.



int4 SetLine()

## 1.1.4.2.4.4.2.5. GetAxisId

Запрос ID оси репера.



int4 GetAxisId()

## 1.1.4.2.4.4.2.6. SetAxisById

Назначить ось по ID.



```
void SetAxisById(int4 AxisId)
```

### Параметры

Параметр	Тип	Описание
AxisId	int4	ID оси

## 1.1.4.2.4.4.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет</a>	Цвет элемента
<a href="#">Шрифт</a>	Шрифт элемента
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Режим</a>	Вид отображаемых значений
<a href="#">Формат меток на пересечении с линиями</a>	Формат меток на пересечении с линиями
<a href="#">Цвет фона значений</a>	Цвет фона значений
<a href="#">Прозрачность фона значений</a>	Прозрачность фона значений
<a href="#">Отображение значений</a>	Скрытие/отображение точек пересечения реперной линии с графиками
<a href="#">Прозрачность</a>	Прозрачность элемента
<a href="#">Стиль</a>	Стиль элемента
<a href="#">Значение</a>	Значение графика в точке пересечения
<a href="#">Толщина линии</a>	Толщина линии
<a href="#">Видимость метки на оси</a>	Видимость метки на оси
<a href="#">Axis</a>	Ссылка на элемент График: ось
<a href="#">AxisID</a>	Идентификатор оси

LineID

Идентификатор линии

## 1.1.4.2.4.4.4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.4.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.4.4.3.5. Axis

Ссылка на элемент График: ось.

## 1.1.4.2.4.4.4.3.6. Цвет

Цвет элемента.



uint4 Color

## 1.1.4.2.4.4.4.3.7. Толщина линии

Толщина линии.



float4 LineWidth

## 1.1.4.2.4.4.4.3.8. Значение

Значение графика в точке пересечения.



variant Value

## 1.1.4.2.4.4.4.3.9. Стил

Стил элемента.



uint4 BrushStyle

## 1.1.4.2.4.4.4.3.10. Прозрачность

Прозрачность элемента.



float4 BrushOpacity



## 1.1.4.2.4.4.4.3.11. Отображение значений

Скрытие/отображение точек пересечения реперной линии с графиками.



bool ShowValues

### Значение

Значение	Описание
true	Точки пересечения отображаются
false	Точки пересечения скрыты

## 1.1.4.2.4.4.4.3.12. Цвет фона значений

Цвет фона значений.



uint4 BgValuesColor

# 1.1.4.2.4.4.4.3.13. Прозрачность фона значений

Прозрачность фона значений.



float4 BgValuesOpacity

## 1.1.4.2.4.4.4.3.14. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.4.4.3.15.      **Формат      меток      на** **пересечении с линиями**

Формат меток на пересечении с линиями.



string Format

# 1.1.4.2.4.4.4.3.16. Шрифт

Шрифт элемента.



string Font

## 1.1.4.2.4.4.4.3.17. Режим

Вид отображаемых значений.



uint4 Mode

### Значение

Значение	Описание
0	Точное значение
1	Интервальные значения

## 1.1.4.2.4.4.4.3.18. Видимость метки на оси

Видимость метки на оси.



bool ShowAxisValues



## 1.1.4.2.4.4.4.3.19. AxisID

Идентификатор оси.



int4 AxisID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.4.3.20. LineID

Идентификатор линии.



int4 LineID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.5. График: линия

Компонент позволяет отобразить линию на основе данных, полученных от источника.

## 1.1.4.2.4.4.5.1. События

Событие	Описание
<a href="#">SaveFinished</a>	Завершение сохранения данных
<a href="#">ReadFinished</a>	Завершение чтения данных

## 1.1.4.2.4.4.5.1.1. SaveFinished

Завершение сохранения данных.

### Параметры

Параметр	Тип	Описание
errorCode	uint1	Код ошибки, возникшей при сохранении: <ul style="list-style-type: none"><li>› 0 – запись завершена успешно</li><li>› 1 – недостаточно прав для записи файла</li></ul>

## 1.1.4.2.4.4.5.1.2. ReadFinished

Завершение чтения данных.

### Параметры

Параметр	Тип	Описание
rowsRead	uint8	Количество прочитанных строк
elapsedMs	float	Время чтения данных (мс)
errorCode	uint1	Код ошибки, возникшей при сохранении: <ul style="list-style-type: none"><li>➤ 0 – загрузка завершена успешно</li><li>➤ 1 – отсутствует указанный файл</li><li>➤ 2 – недостаточно прав для чтения файла</li></ul>
message	string	Сообщение

## 1.1.4.2.4.4.5.2. Функции

Функция	Описание
<a href="#">Load</a>	Загружает данные из файла в формате CSV или JSON
<a href="#">Save</a>	Сохраняет данные в файл в формате CSV или JSON
<a href="#">SetAxisByIds</a>	Назначает оси по ID
<a href="#">SetAxisById</a>	Назначает ось по ID
<a href="#">GetGlobalMaxY</a>	Возвращает максимум загруженных данных по оси Y (тип variant)
<a href="#">GetGlobalMinY</a>	Возвращает минимум загруженных данных по оси Y (тип variant)
<a href="#">GetGlobalMaxX</a>	Возвращает максимум загруженных данных по оси X (тип variant)
<a href="#">GetGlobalMinX</a>	Возвращает минимум загруженных данных по оси X (тип variant)
<a href="#">GetMaxY</a>	Возвращает максимум по оси Y (тип variant)
<a href="#">GetMinY</a>	Возвращает минимум по оси Y (тип variant)
<a href="#">GetMaxX</a>	Возвращает максимум по оси X (тип variant)
<a href="#">GetMinX</a>	Возвращает минимум по оси X (тип variant)
<a href="#">GetActiveControlLineLables</a>	Возвращает значения пересечений с активной реперной линией в виде строки
<a href="#">SetColor</a>	Задаёт цвет линии





## 1.1.4.2.4.4.5.2.1. Load

Загружает данные из файла в формате CSV или JSON.



```
void Load(string FilePath, string Format)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Полный путь к файлу
Format	string	Формат данных

## 1.1.4.2.4.4.5.2.2. Save

Сохраняет данные в файл в формате CSV или JSON.



```
void Save(string FilePath, string Format)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Полный путь к файлу
Format	string	Формат данных

## 1.1.4.2.4.4.5.2.3. SetAxisByIds

Назначает оси по ID.



```
void SetAxisByIds(int4 xAxisId, int4 yAxisId)
```

### Параметры

Параметр	Тип	Описание
xAxisId	int4	Идентификатор (ID) оси X
yAxisId	int4	Идентификатор (ID) оси Y

## 1.1.4.2.4.4.5.2.4. SetAxisByld

Назначает ось по ID.



```
void SetAxisByld(int4 AxisId)
```

### Параметры

Параметр	Тип	Описание
AxisId	int4	Идентификатор (ID) оси

## 1.1.4.2.4.4.5.2.5. GetGlobalMaxY

Возвращает максимум загруженных данных по оси Y (тип variant).



variant GetGlobalMaxY()

## 1.1.4.2.4.4.5.2.6. GetGlobalMinY

Возвращает минимум загруженных данных по оси Y (тип variant).



variant GetGlobalMinY()

## 1.1.4.2.4.4.5.2.7. GetGlobalMaxX

Возвращает максимум загруженных данных по оси X (тип variant).



variant GetGlobalMaxX()

## 1.1.4.2.4.4.5.2.8. GetGlobalMinX

Возвращает минимум загруженных данных по оси X (тип variant).



variant GetGlobalMinX()



## 1.1.4.2.4.4.5.2.9. GetMaxY

Возвращает максимум по оси Y (тип variant).



variant GetMaxY()

## 1.1.4.2.4.4.5.2.10. GetMinY

Возвращает минимум по оси Y (тип variant).



variant GetMinY()

## 1.1.4.2.4.4.5.2.11. GetMaxX

Возвращает максимум по оси X (тип variant).



variant GetMaxX()

## 1.1.4.2.4.4.5.2.12. GetMinX

Возвращает минимум по оси X (тип variant).



variant GetMinX()

## 1.1.4.2.4.4.5.2.13. GetActiveControlLineLables

Возвращает значения пересечений с активной реперной линией в виде строки.



string GetActiveControlLineLables()

## 1.1.4.2.4.4.5.2.14. SetColor

Задает цвет линии.



```
void SetColor(uint4 Color)
```

### Параметры

Параметр	Тип	Описание
color	uint4	Код цвета линии

## 1.1.4.2.4.4.5.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет</a>	Цвет элемента
<a href="#">Видимость точек</a>	Видимость объекта на форме
<a href="#">Единица измерения</a>	Единицы измерения отображаемых данных
<a href="#">Прозрачность</a>	Прозрачность элемента
<a href="#">Толщина линии</a>	Толщина линии
<a href="#">ID</a>	Идентификатор (ID) линии
<a href="#">Вывод значений пересечений с реперной линией</a>	Скрывает/отображает значения пересечений с реперной линией
<a href="#">Номер колонки значений Y</a>	Задает номер колонки для числовых значений
<a href="#">Номер колонки значений X</a>	Номер колонки для временных значений
<a href="#">Тип графика</a>	Тип графика
<a href="#">Приоритет</a>	Приоритет элемента
<a href="#">Стиль линии</a>	Стиль линии элемента
<a href="#">Размер точки</a>	Размер точки пересечения с элементом
<a href="#">Название</a>	Название элемента
<a href="#">Номер символа точки</a>	Номер символа точки
<a href="#">Координата Y строкового сигнала</a>	Координата Y строкового сигнала

<a href="#">Диаграмма взаимозависимостей</a>	Сортировка данных по колонке времени сервера
<a href="#">Tag</a>	Тег сигнала
<a href="#">AxisX</a>	Горизонтальная ось. Ссылка на элемент График: ось
<a href="#">AxisY</a>	Горизонтальная ось. Ссылка на элемент График: ось
<a href="#">DataSource</a>	Ссылка на элемент График: данные
<a href="#">Шрифт символов точек</a>	Шрифт символов точек
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Множитель</a>	Множитель
<a href="#">Смещение</a>	Смещение
<a href="#">AxisXID</a>	Идентификатор (ID) оси X
<a href="#">AxisYID</a>	Идентификатор (ID) оси Y
<a href="#">DataSourceID</a>	Идентификатор (ID) источника данных



## 1.1.4.2.4.4.5.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.5.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.4.5.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.5.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
else  
    {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.4.5.3.5. AxisX

Горизонтальная ось. Ссылка на элемент График: ось.

## 1.1.4.2.4.4.5.3.6. AxisY

Горизонтальная ось. Ссылка на элемент График: ось.

## 1.1.4.2.4.4.5.3.7. DataSource

Ссылка на элемент График: данные.

## 1.1.4.2.4.4.5.3.8. Цвет

Цвет элемента.



uint4 Color



## 1.1.4.2.4.4.5.3.9. Название

Название элемента.



string Title

## 1.1.4.2.4.4.5.3.10. Единица измерения

Единицы измерения отображаемых данных.



string Unit

## 1.1.4.2.4.4.5.3.11. Толщина линии

Толщина линии.



float4 LineWidth

## 1.1.4.2.4.4.5.3.12. Размер точки

Размер точки пересечения с элементом.



uint4 PointSize

## 1.1.4.2.4.4.5.3.13. Стиль линии

Стиль линии элемента.



uint4 LineBrushStyle

## 1.1.4.2.4.4.5.3.14. Прозрачность

Прозрачность элемента.



float4 BrushOpacity

## 1.1.4.2.4.4.5.3.15. Номер символа точки

Номер символа точки.



uint2 PointsymbolCode

## 1.1.4.2.4.4.5.3.16. Видимость точек

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт



## 1.1.4.2.4.4.5.3.17. Шрифт символов точек

Шрифт символов точек.



string PointsymbolFont

## 1.1.4.2.4.4.5.3.18. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.4.5.3.19. Приоритет

Приоритет элемента.



uint4 zOrder

## 1.1.4.2.4.4.5.3.20. Тип графика

Тип графика.



uint4 DrawingType

### Значение

Значение	Описание
0	Линейный график
1	Ступенчатый график
2	Гистограмма

## 1.1.4.2.4.4.5.3.21. Номер колонки значений X

Номер колонки для временных значений.



uint4 ColumnNumberTime

## 1.1.4.2.4.4.5.3.22. Номер колонки значений Y

Задает номер колонки для числовых значений.



uint4 ColumnNumberValue

## 1.1.4.2.4.4.5.3.23. Координата Y строкового сигнала

Координата Y строкового сигнала.



variant StringTagsYValue

# 1.1.4.2.4.4.5.3.24. ВЗАИМОЗАВИСИМОСТИ

## Диаграмма

Сортировка данных по колонке времени сервера.



bool SortByServerTimeColumn



## 1.1.4.2.4.4.5.3.25. Вывод значений пересечений с реперной линией

Скрывает/отображает значения пересечений с реперной линией.



bool ShowActiveControlLineLables

### Значение

Значение	Описание
true	Отображает значения пересечений
false	Скрывает значения пересечений

## 1.1.4.2.4.4.5.3.26. Множитель

Множитель.



float8 Ratio

## 1.1.4.2.4.4.5.3.27. Смещение

Смещение.



float8 Offset

## 1.1.4.2.4.4.5.3.28. ID

Идентификатор (ID) линии.



int4 ID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.5.3.29. AxisXID

Идентификатор (ID) оси X.



int4 AxisXID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.5.3.30. AxisYID

Идентификатор (ID) оси Y.



int4 AxisYID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.5.3.31. DataSourceID

Идентификатор (ID) источника данных.



int4 DataSourceID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.5.3.32. Tag

Тег сигнала.



string Tag



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.4.6. График: декоратор линии

Компонент позволяет декорировать линию.

## 1.1.4.2.4.4.6.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет</a>	Цвет элемента
<a href="#">Название</a>	Название элемента
<a href="#">Толщина линии</a>	Толщина линии
<a href="#">Размер точки</a>	Размер точки пересечения с элементом
<a href="#">Стиль линии</a>	Стиль линии элемента
<a href="#">Приоритет</a>	Приоритет элемента
<a href="#">Номер колонки декоратора</a>	Номер колонки данных в источнике, предоставляющем данные для определения условий декорирования
<a href="#">Значение максимума декоратора</a>	Значение максимума декоратора
<a href="#">Активность</a>	Активирует/деактивирует декоратор
<a href="#">Значение минимума декоратора</a>	Значение минимума декоратора
<a href="#">ID</a>	Идентификатор (ID) декоратора линии
<a href="#">LineID</a>	Идентификатор (ID) линии

## 1.1.4.2.4.4.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.4.6.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.6.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.4.6.1.5. Цвет

Цвет элемента.



uint4 Color

## 1.1.4.2.4.4.6.1.6. Название

Название элемента.



string Title



## 1.1.4.2.4.4.6.1.7. Толщина линии

Толщина линии.



float4 LineWidth

## 1.1.4.2.4.4.6.1.8. Размер точки

Размер точки пересечения с элементом.



uint4 PointSize

## 1.1.4.2.4.4.6.1.9. Стиль линии

Стиль линии элемента.



uint4 LineBrushStyle

## 1.1.4.2.4.4.6.1.10. Активность

Активирует/деактивирует декоратор.



bool Active

### Значение

Значение	Описание
true	Декоратор активен
false	Декоратор неактивен

# 1.1.4.2.4.4.6.1.11. декоратора

Значение

минимума

Значение минимума декоратора.



float8 MinValue

# 1.1.4.2.4.4.6.1.12.      Значение      максимума декоратора

Значение максимума декоратора.



float8 MaxValue

## 1.1.4.2.4.4.6.1.13. Номер колонки декоратора

Номер колонки данных в источнике, предоставляющем данные для определения условий декорирования.



int4 ColumnNumber

## 1.1.4.2.4.4.6.1.14. Приоритет

Приоритет элемента.



int4 zOrder



## 1.1.4.2.4.4.6.1.15. ID

Идентификатор (ID) декоратора линии.



int4 ID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.6.1.16. LineID

Идентификатор (ID) линии.



int4 LineID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.7. График: точка пересечения

Компонент позволяет настроить отображение данных в точке пересечения реперной линии с линией графика.

## 1.1.4.2.4.4.7.1. Функции

Функция	Описание
<a href="#">GetValue</a>	Возвращает значение из указанной колонки
<a href="#">Prev</a>	Переходит к предыдущей точке
<a href="#">Next</a>	Переходит к следующей точке
<a href="#">First</a>	Выбирает первую точку
<a href="#">PrevLine</a>	Переходит к предыдущей линии
<a href="#">NextLine</a>	Переходит к следующей линии
<a href="#">FirstLine</a>	Выбирает первую линию
<a href="#">GetLineID</a>	Возвращает ID линии
<a href="#">SetLineID</a>	Задаёт значение ID линии

## 1.1.4.2.4.4.7.1.1. GetValue

Возвращает значение из указанной колонки.



GetValue(int4 ColumnId)

### Параметры

Параметр	Тип	Описание
ColumnId	int4	Идентификатор (ID) колонки

## 1.1.4.2.4.4.7.1.2. Prev

Переходит к предыдущей точке.



bool Prev()

## 1.1.4.2.4.4.7.1.3. Next

Переходит к следующей точке.



bool Next()

## 1.1.4.2.4.4.7.1.4. First

Выбирает первую точку.



bool First()



## 1.1.4.2.4.4.7.1.5. PrevLine

Переходит к предыдущей линии.



bool PrevLine()

## 1.1.4.2.4.4.7.1.6. NextLine

Переходит к следующей линии.



bool NextLine()

## 1.1.4.2.4.4.7.1.7. FirstLine

Выбирает первую линию.



bool FirstLine()

## 1.1.4.2.4.4.7.1.8. GetLineID

Возвращает ID линии.



uint4 GetLineID()

## 1.1.4.2.4.4.7.1.9. SetLineID

Задает значение ID линии.



void SetLineID (ID)

### Параметры

Параметр	Тип	Описание
ID	uint4	Идентификатор (ID) линии

## 1.1.4.2.4.4.7.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет</a>	Цвет элемента
<a href="#">Стиль</a>	Стиль элемента
<a href="#">Прозрачность</a>	Прозрачность элемента
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Шрифт</a>	Шрифт элемента
<a href="#">Y</a>	Значение точки пересечения по оси Y
<a href="#">X</a>	Значение точки пересечения по оси X
<a href="#">Exists</a>	Наличие пересечения с линией
<a href="#">Текст</a>	Текст всплывающей подсказки
<a href="#">Прозрачность фона значений</a>	Прозрачность фона значений
<a href="#">Цвет фона значений</a>	Цвет фона значений
<a href="#">Размер точки</a>	Размер точки пересечения
<a href="#">ID</a>	Идентификатор (ID) элемента
<a href="#">LineID</a>	Идентификатор (ID) линии

## 1.1.4.2.4.4.7.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.7.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.4.7.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.7.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.4.7.2.5. Цвет

Цвет элемента.



uint4 Color

## 1.1.4.2.4.4.7.2.6. Стил

Стил элемента.



uint4 BrushStyle

## 1.1.4.2.4.4.7.2.7. Прозрачность

Прозрачность элемента.



float4 BrushOpacity

## 1.1.4.2.4.4.7.2.8. Цвет фона значений

Цвет фона значений.



uint4 BgValuesColor

## 1.1.4.2.4.4.7.2.9. Прозрачность фона значений

Прозрачность фона значений.



float4 BgValuesOpacity

## 1.1.4.2.4.4.7.2.10. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт



## 1.1.4.2.4.4.7.2.11. Шрифт

Шрифт элемента.



string Font

## 1.1.4.2.4.4.7.2.12. Размер точки

Размер точки пересечения.



uint4 PointSize

## 1.1.4.2.4.4.7.2.13. Текст

Текст всплывающей подсказки.



string ToolTipText

## 1.1.4.2.4.4.7.2.14. Y

Значение точки пересечения по оси Y.



variant Y



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.7.2.15. X

Значение точки пересечения по оси X.



variant X



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.7.2.16. Exists

Наличие пересечения с линией.



bool Exists



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Пересекает линию
false	Не пересекает линию

## 1.1.4.2.4.4.7.2.17. ID

Идентификатор (ID) элемента.



int4 ID



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.4.7.2.18. LineID

Идентификатор (ID) линии.



int4 LineID



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.4.8. График: экспорт

Компонент позволяет сохранять в файл данные, отображаемые на трендовом поле.

## 1.1.4.2.4.4.8.1. События

Событие	Описание
<a href="#">DataExportFailed</a>	Сигнал об ошибке экспорта

## 1.1.4.2.4.4.8.1.1. DataExportFailed

Сигнал об ошибке экспорта.

## 1.1.4.2.4.4.8.2. Функции

Функция	Описание
<a href="#">Reload</a>	Запрашивает данные графика

## 1.1.4.2.4.4.8.2.1. Reload

Запрашивает данные графика.



void Reload()

## 1.1.4.2.4.4.8.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Колонка для сортировки</a>	Номер колонки для сортировки
<a href="#">Сортировка</a>	Разрешение/запрет сортировки данных
<a href="#">Параметры экспорта</a>	Параметры экспорта
<a href="#">Источник данных</a>	Источник данных

## 1.1.4.2.4.4.8.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.4.8.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.4.8.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.4.8.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.

int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.4.8.3.5. Источник данных

Источник данных.

## 1.1.4.2.4.4.8.3.6. Сортировка

Разрешение/запрет сортировки данных.



bool Sort

## 1.1.4.2.4.4.8.3.7. Колонка для сортировки

Номер колонки для сортировки.



uint4 SortColumnNumber

## 1.1.4.2.4.4.8.3.8. Параметры экспорта

Параметры экспорта.



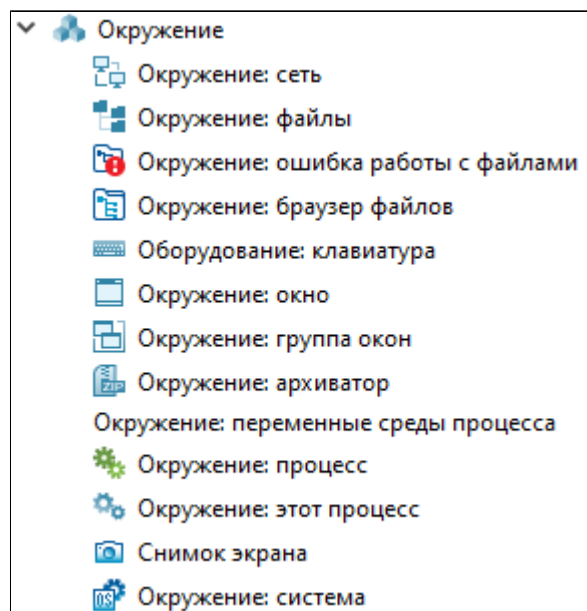
string Parameters

## 1.1.4.2.4.5. Окружение

Набор компонентов Astra.HMI для взаимодействия с сетью, файловой системой и оборудованием компьютера.

Содержит компоненты:

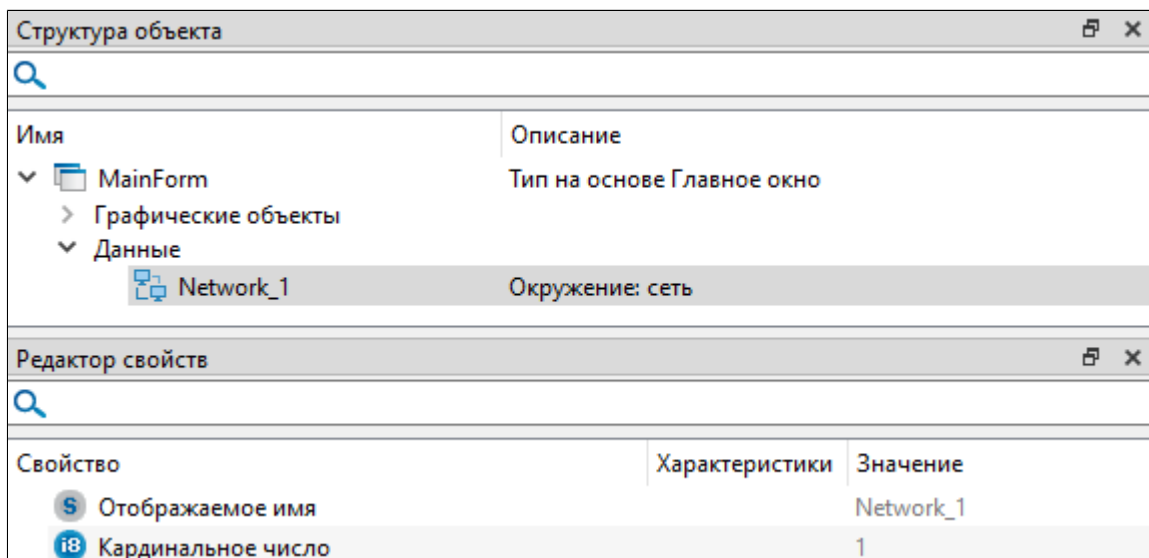
- > [Окружение: сеть](#)
- > [Окружение: файлы](#)
- > [Окружение: ошибка работы с файлами](#)
- > [Окружение: браузер файлов](#)
- > [Оборудование: клавиатура](#)
- > [Окружение: окно](#)
- > [Окружение: группа окон](#)
- > [Окружение: этот процесс](#)
- > [Окружение: архиватор](#)
- > [Окружение: переменные среды процесса](#)
- > [Окружение: процесс](#)
- > [Окружение: этот процесс](#)
- > [Снимок экрана](#)
- > [Окружение: система](#)



## 1.1.4.2.4.5.1. Окружение: сеть

Позволяет работать с сетевым окружением.

Чтобы взаимодействовать с сетевым окружением, добавьте на экранную форму компонент Окружение: сеть.



Компонент позволяет получить имя сетевого компьютера через свойство ComputerName.



## 1.1.4.2.4.5.1.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">ComputerName</a>	Имя сетевого компьютера
<a href="#">FullComputerName</a>	Полное имя сетевого компьютера в домене

## 1.1.4.2.4.5.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.1.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.1.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.1.1.5. ComputerName

Имя сетевого компьютера.



string ComputerName



Доступно только для чтения в режиме рантайма.

### Примеры



```
//Вывести имя сетевого компьютера в поле ввода:  
TextEdit.Text = Network.ComputerName;
```

## 1.1.4.2.4.5.1.1.6. FullComputerName

Полное имя сетевого компьютера в домене.



string FullComputerName



Доступно только для чтения в режиме рантайма.

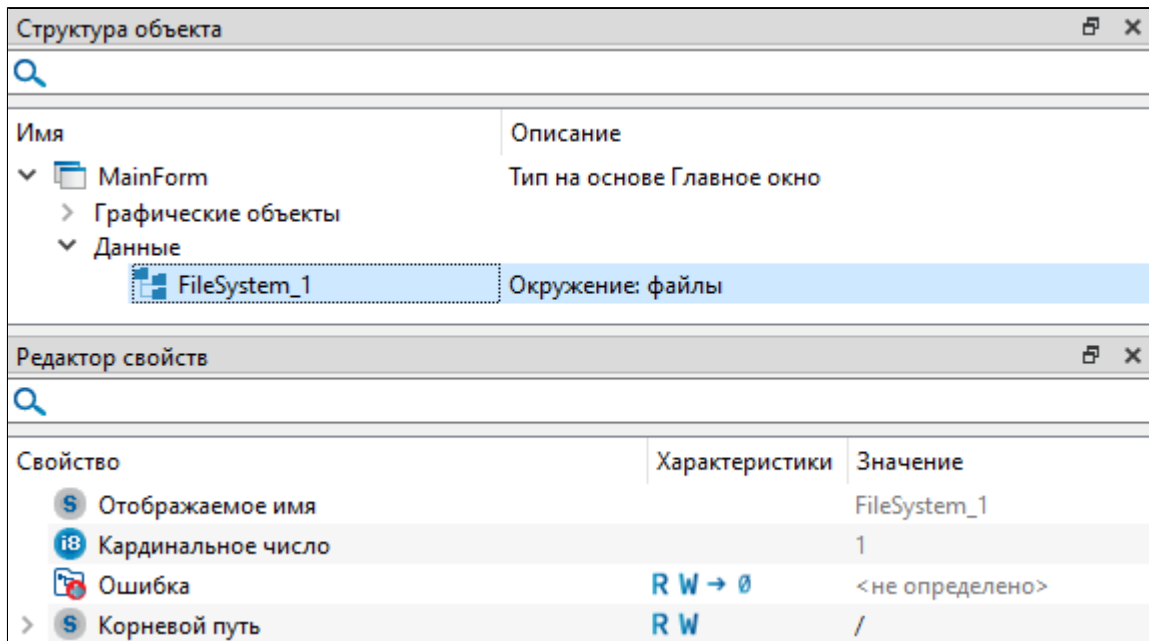
### Примеры



```
//Вывести имя сетевого компьютера в поле ввода:  
TextEdit.Text = Network.FullComputerName;
```

## 1.1.4.2.4.5.2. Окружение: файлы

Компонент позволяет работать с каталогами файловой системы, читать и записывать содержимое текстовых файлов. Компонент не визуальный и виден только в области Структура объекта.





## 1.1.4.2.4.5.2.1. Функции

Функция	Описание
<a href="#">WriteTextFile</a>	Создает текстовый файл *.txt и записывает в него указанную строку
<a href="#">ReadBinaryAsJSON</a>	Предоставляет содержимое бинарного файла для чтения
<a href="#">ReadTextFile</a>	Предоставляет содержимое текстового файла для чтения. Содержимое файла должно быть в кодировке UTF-8
<a href="#">CreateRecursiveFolder</a>	Создает папку по указанному пути
<a href="#">CombinePath</a>	Объединяет указанные текстовые пути в единый путь
<a href="#">Exists</a>	Определяет существование файла или папки.
<a href="#">CanonicalPath</a>	Канонический путь
<a href="#">ParentPath</a>	Родительский путь
<a href="#">AbsolutePath</a>	Абсолютный путь
<a href="#">RelativePath</a>	Относительный путь
<a href="#">FileName</a>	Имя файла
<a href="#">Stem</a>	Имя файла без расширения
<a href="#">Extension</a>	Расширение файла
<a href="#">HasFileName</a>	Существует ли имя файла по указанному пути
<a href="#">HasParentPath</a>	Существует ли родительский путь
<a href="#">IsFile</a>	Является ли путь файлом
<a href="#">IsDirectory</a>	Является ли путь директорией
<a href="#">IsSymLink</a>	Является ли путь символической ссылкой
<a href="#">IsParent</a>	Является ли один путь родителем другого
<a href="#">FileSize</a>	Размер файла
<a href="#">CreatedTime</a>	Время создания элемента
<a href="#">ModifiedTime</a>	Время изменения элемента

<a href="#">ToGeneric</a>	Обобщенное представление пути
<a href="#">ToNative</a>	Нативное представление пути
<a href="#">PreferredSeparator</a>	Предпочитаемый разделитель
<a href="#">RemoveRecursive</a>	Удалить рекурсивно
<a href="#">AppendTextFile</a>	Записать в конец текстового файла

## 1.1.4.2.4.5.2.1.1. CanonicalPath

Канонический путь.



string CanonicalPath(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.2. ParentPath

Родительский путь.



string ParentPath(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.3. AbsolutePath

Абсолютный путь.



string AbsolutePath(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.4. RelativePath

Относительный путь.



string RelativePath(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.5. FileName

Возвращаемое имя файла.



string FileName(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.6. Stem

Имя файла без расширения.



string Stem(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь



## 1.1.4.2.4.5.2.1.7. Extension

Расширение файла.



string Extension(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.8. HasParentPath

Существует ли родительский путь.



bool HasParentPath(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.9. HasFileName

Существует ли имя файла по указанному пути.



`bool HasFileName(string path)`

### Параметры

Параметр	Тип	Описание
<code>path</code>	<code>string</code>	Текстовый путь

## 1.1.4.2.4.5.2.1.10. IsFile

Является ли путь файлом.



`bool IsFile(string path)`

### Параметры

Параметр	Тип	Описание
<code>path</code>	<code>string</code>	Текстовый путь

## 1.1.4.2.4.5.2.1.11. IsDirectory

Является ли путь директорией.



bool IsDirectory(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.12. IsSymLink

Является ли путь символической ссылкой.



`bool IsSymLink(string path)`

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.13. IsParent

Является ли один путь родителем другого.



```
bool IsParent(string parentPath, string childPath)
```

### Параметры

Параметр	Тип	Описание
parentPath	string	Текстовый путь 1 (родительский)
childPath	string	Текстовый путь 2 (проверяемый)

## 1.1.4.2.4.5.2.1.14. FileSize

Размер файла.



uint8 FileSize(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь



## 1.1.4.2.4.5.2.1.15. CreatedTime

Время создания элемента.



timestamp CreatedTime(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.16. ModifiedTime

Время изменения элемента.



timestamp ModifiedTime(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.17. ToGeneric

Обобщенное представление пути.



string ToGeneric(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.18. ToNative

Нативное представление пути.



```
string ToNative(string path)
```

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

# PreferredSeparator

Предпочитаемый разделитель.



string PreferredSeparator()

## 1.1.4.2.4.5.2.1.19. Exists

Определяет существование файла или папки.



bool Exists(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

### Примеры



```
//Если по указанному пути файл существует, вывести в поле ввода  
"Файл существует"  
if (FileSystem.Exists(filePath))  
{  
    TextEdit.Text = "Файл существует";  
};
```

## 1.1.4.2.4.5.2.1.20. CombinePath

Объединяет текстовые пути FilePath1 и FilePath2 в единый путь. Позволяет облегчить формирование составных путей. В двойных кавычках указывается дополнительный путь до папки или текстового файла.



```
void CombinePath(FilePath left, FilePath right)
```

### Параметры

Параметр	Тип	Описание
left	string	Текстовый путь 1
right	string	Текстовый путь 2

### Примеры



```
//Объединить пути и записать в поле ввода. Результат слияния -  
строка: C:\Users\пользователь\TestFolder\TestFile.txt  
TextEdit.Text = FileSystem.CombinePath(FileSystem.CurrentUserFolder, "\  
\TestFolder\TestFile.txt");
```

## 1.1.4.2.4.5.2.1.21. CreateRecursiveFolder

Создает папку по указанному пути.



void CreateRecursiveFolder (FilePath)

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь к создаваемой папке

### Примеры



```
//Создать папку:  
FileSystem.CreateRecursiveFolder(FilePath);
```



## 1.1.4.2.4.5.2.1.22. RemoveRecursive

Удалить рекурсивно.



```
void RemoveRecursive(string path)
```

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.2.1.23. ReadTextFile

Предоставляет содержимое текстового файла для чтения. Содержимое файла должно быть в кодировке UTF-8.



void ReadTextFile (FilePath)

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь до текстового файла

### Примеры



```
//Содержимое файла поместить в текстовое поле. В переменной  
FilePath записать путь до текстового файла:  
TextEdit.Text = FileSystem.ReadTextFile(FilePath);
```

## 1.1.4.2.4.5.2.1.24. WriteTextFile

Создает текстовый файл \*.txt и записывает в него указанную строку. Файл будет сохранен по пути, который вы указали в переменной FilePath.



```
void WriteTextFile(FilePath, Text)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь к текстовому файлу
Text	string	Текстовая строка

### Примеры



```
//Записать строку текста в файл по указанному пути:  
Files.WriteTextFile(FilePath, "Содержимое для текстового файла");
```

## 1.1.4.2.4.5.2.1.25. AppendTextFile

Записать в конец текстового файла.



```
void AppendTextFile(string filePath, string content)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к текстовому файлу
content	string	Текст для записи в конце текстового файла

## 1.1.4.2.4.5.2.1.26. ReadBinaryAsJSON

Предоставляет содержимое бинарного файла для чтения.



void ReadBinaryAsJSON (string FilePath, string Type, uint8 Position, uint2 Length)

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь к бинарному файлу
Type	string	Тип элементов бинарного файла
Position	uint8	Позиция, с которой нужно начать чтение
Length	uint2	Length отрезка данных

### Примеры



```
//Считать из бинарного файла bin.dat в текстовое поле 20 элементов  
типа int2. Считывание начинать с 100 позиции:  
TextEdit.Text = FileSystem.ReadBinaryAsJSON("bin.dat", "int2", 100, 20);
```

## 1.1.4.2.4.5.2.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">CurrentUserFolder</a>	Полный путь к папке текущего пользователя операционной системы
<a href="#">AllUsersFolder</a>	Полный путь к общей папке пользователей операционной системы
<a href="#">ProjectFolder</a>	Полный путь к папке текущего проекта
<a href="#">Ошибка</a>	Объект, хранящий ошибку с ее кодом и описанием
<a href="#">Корневой путь</a>	Путь, в пределах которого работает тип

## 1.1.4.2.4.5.2.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.2.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.5.2.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.2.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.2.2.5. Ошибка

Объект, хранящий ошибку с ее кодом и описанием.

## 1.1.4.2.4.5.2.2.6. Корневой путь

Путь, в пределах которого работает тип.



string RootPath

## 1.1.4.2.4.5.2.2.7. CurrentUserFolder

Полный путь к папке текущего пользователя операционной системы.



string CurrentUserFolder



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести полный путь к папке текущего пользователя ОС в текстовое поле:

```
Text_1 = FileSystem.CurrentUserFolder;
```

## 1.1.4.2.4.5.2.2.8. AllUsersFolder

Полный путь к общей папке пользователей операционной системы.



string AllUsersFolder



В ОС Linux общая папка пользователей отсутствует, поэтому в данных системах свойство возвращает путь к папке текущего пользователя.



Доступно только для чтения в режиме рантайма.

### Примеры



```
//Вывести полный путь к общей папке пользователей ОС в текстовое поле:  
Text_1 = FileSystem.AllUsersFolder;
```

## 1.1.4.2.4.5.2.2.9. ProjectFolder

Полный путь к папке проекта HMI.



string ProjectFolder



Доступно только для чтения в режиме рантайма.

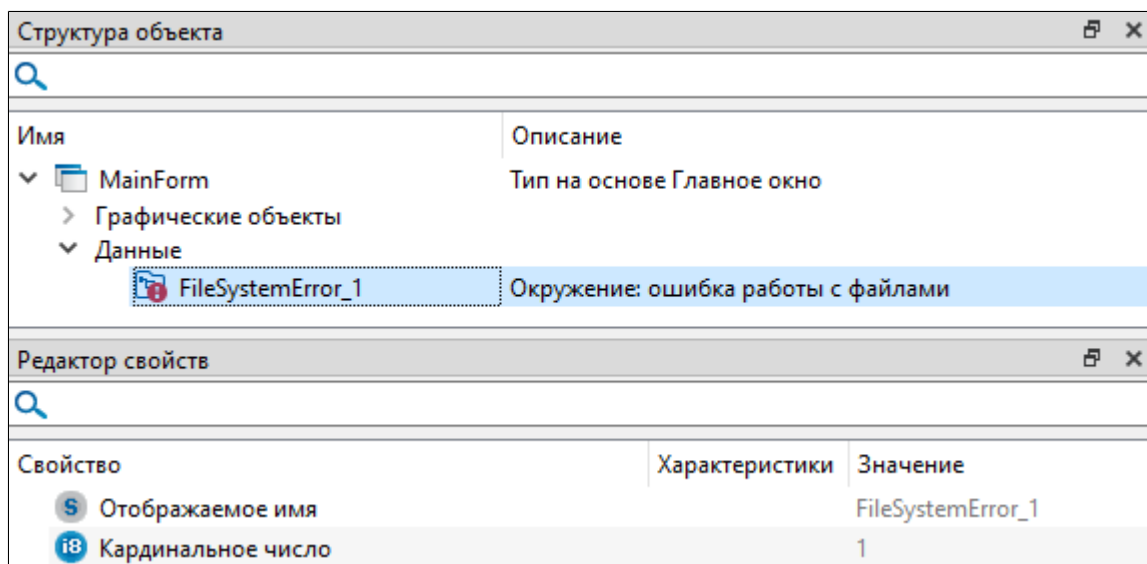
### Примеры



```
//Вывести полный путь к проекту в текстовое поле:  
Text_1 = FileSystem.ProjectFolder;
```

## 1.1.4.2.4.5.3. Окружение: ошибка работы с файлами

Тип, хранящий ошибку при работе с файлами в других типах. Компонент не визуальный и виден только в области Структура объекта.





## 1.1.4.2.4.5.3.1. События

Событие	Описание
<a href="#">Occured</a>	Произошла ошибка

## 1.1.4.2.4.5.3.1.1. Occured

Произошла ошибка.

## 1.1.4.2.4.5.3.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">State</a>	Произошла ли ошибка при выполнении последней операции
<a href="#">Text</a>	Текст ошибки последней операции
<a href="#">Code</a>	Код ошибки последней операции

## 1.1.4.2.4.5.3.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.3.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.3.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.3.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.3.2.5. State

Произошла ли ошибка при выполнении последней операции.



bool State



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.5.3.2.6. Text

Текст ошибки последней операции.



bool Text



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.3.2.7. Code

Код ошибки последней операции.



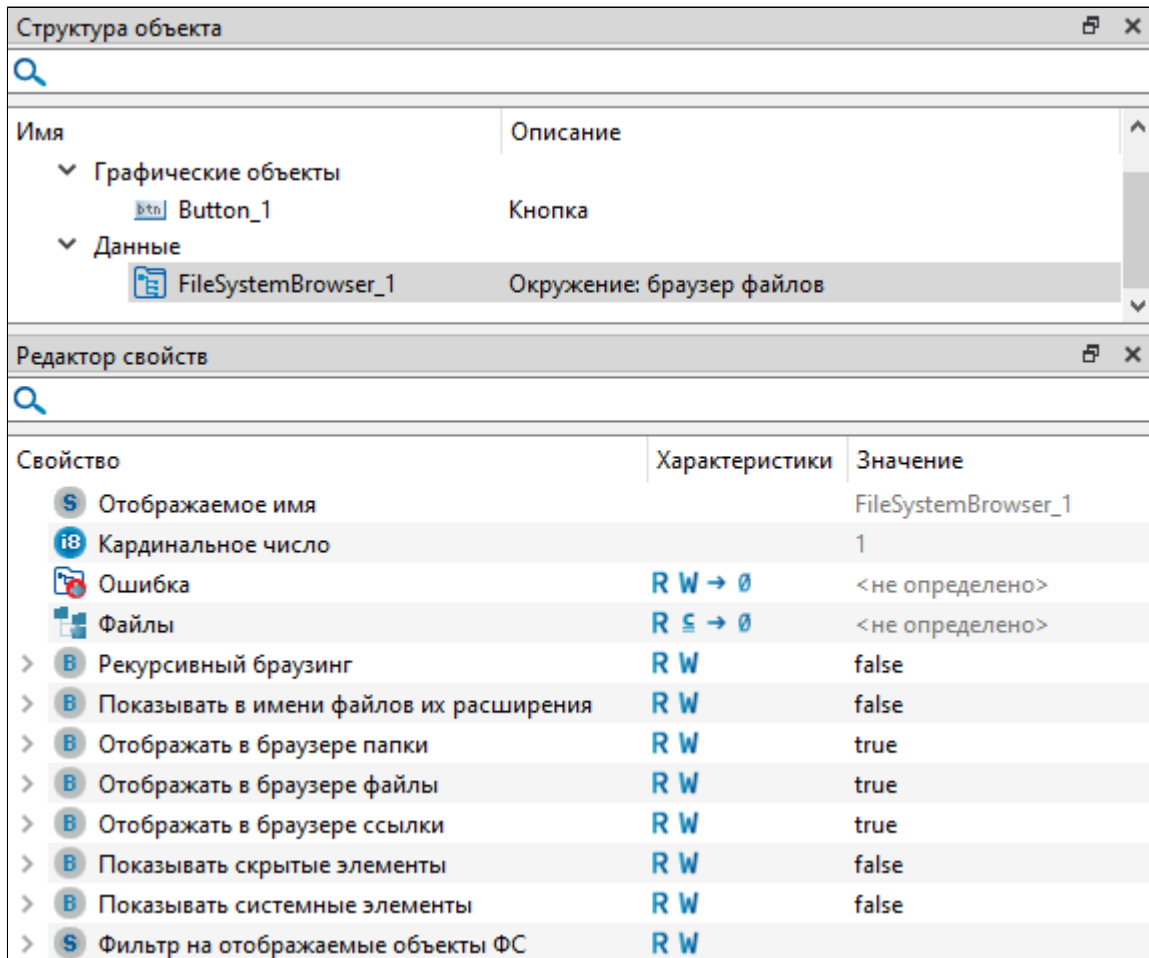
uint4 Code



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.4. Окружение: браузер файлов

Компонент предназначен для отображения файловой системы.



**Структура объекта**

Имя	Описание
Графические объекты	
Button_1	Кнопка
Данные	
FileSystemBrowser_1	Окружение: браузер файлов

**Редактор свойств**

Свойство	Характеристики	Значение
Отображаемое имя		FileSystemBrowser_1
Кардинальное число		1
Ошибка	R W → 0	<не определено>
Файлы	R ↯ → 0	<не определено>
> Рекурсивный браунинг	R W	false
> Показывать в имени файлов их расширения	R W	false
> Отображать в браузере папки	R W	true
> Отображать в браузере файлы	R W	true
> Отображать в браузере ссылки	R W	true
> Показывать скрытые элементы	R W	false
> Показывать системные элементы	R W	false
> Фильтр на отображаемые объекты ФС	R W	

## 1.1.4.2.4.5.4.1. Функции

Функция	Описание
<a href="#">Browse</a>	Браузинг

## 1.1.4.2.4.5.4.1.1. Browse

Начать браузеринг.



void Browse(string path)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь

## 1.1.4.2.4.5.4.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ошибка</a>	Объект, хранящий ошибку с ее кодом и описанием
<a href="#">Файлы</a>	Объект для работы с файловой системой
<a href="#">Рекурсивный браузеринг</a>	Погружать все дерево заданного пути или только потомков
<a href="#">Показывать в имени файлов их расширения</a>	Показывать в имени файлов их расширения
<a href="#">Отображать в браузере папки</a>	Отображать в браузере папки
<a href="#">Отображать в браузере файлы</a>	Отображать в браузере файлы
<a href="#">Отображать в браузере ссылки</a>	Отображать в браузере ссылки
<a href="#">Показывать скрытые элементы</a>	Показывать скрытые элементы
<a href="#">Показывать системные элементы</a>	Показывать системные элементы
<a href="#">Фильтр на отображаемые объекты</a>	Фильтр на отображение файлов, папок, ссылок



## 1.1.4.2.4.5.4.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.5.4.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.4.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.4.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.4.2.5. Ошибка

Объект, хранящий ошибку с ее кодом и описанием.

## 1.1.4.2.4.5.4.2.6. Файлы

Объект для работы с файлами.

## 1.1.4.2.4.5.4.2.7. Рекурсивный браузинг

Погружать все дерево заданного пути или только потомков.



bool Recursive

## 1.1.4.2.4.5.4.2.8. Показывать в имени файлов их расширения

Показывать в имени файлов их расширения.



bool WithExtensions

## 1.1.4.2.4.5.4.2.9. Отображать в браузере папки

Отображать в браузере папки.



bool IncludeFolders



## 1.1.4.2.4.5.4.2.10. Отображать в браузере файлы

Отображать в браузере файлы.



bool IncludeFiles

## 1.1.4.2.4.5.4.2.11. Отображать в браузере ССЫЛКИ

Отображать в браузере ссылки.



bool IncludeSymLinks

## 1.1.4.2.4.5.4.2.12. элементы

## Показывать скрытые

Показывать скрытые элементы.



bool ShowHiddenObjects

## 1.1.4.2.4.5.4.2.13. Показывать системные элементы

Показывать системные элементы.



bool ShowSystemObjects

## 1.1.4.2.4.5.4.2.14. Фильтр на отображаемые объекты ФС

Фильтр на отображение файлов, папок, ссылок.

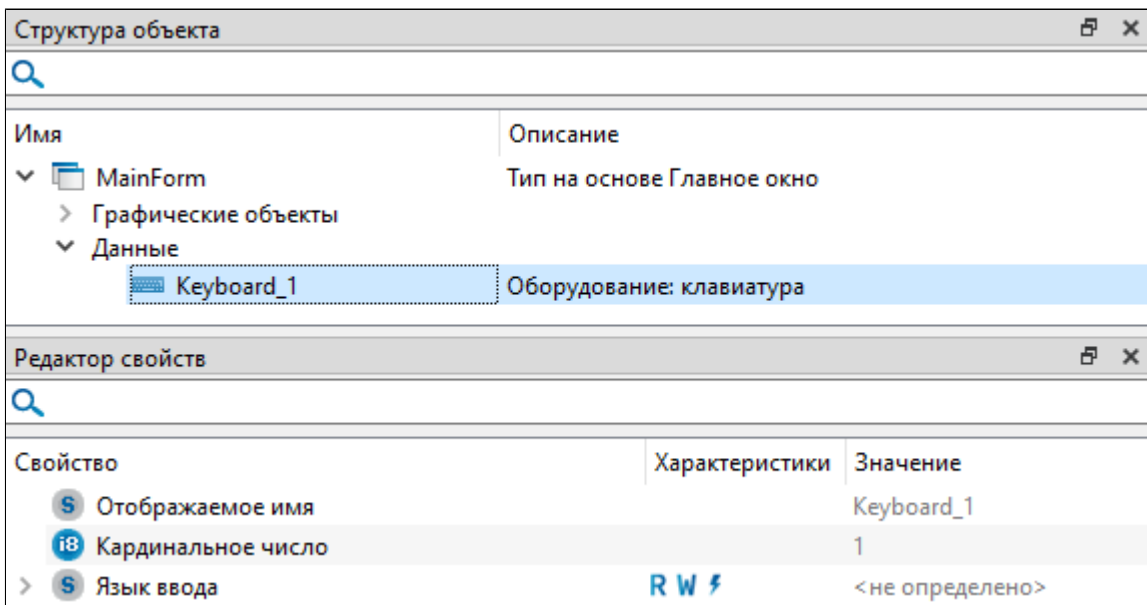


string Filter

## 1.1.4.2.4.5.5. Оборудование: клавиатура

Компонент позволяет получить текущую информацию о раскладке клавиатуры, состояниях клавиш и обрабатывать события смены раскладки.

Чтобы взаимодействовать с клавиатурой, добавьте на экранную форму компонент Оборудование: клавиатура. Компонент не визуальный и виден только в области Структура объекта.



The screenshot displays two windows from a development environment:

- Структура объекта (Object Structure):** A tree view showing the hierarchy of components. The selected component is `Keyboard_1`, which is a data component under `MainForm`. Its description is "Оборудование: клавиатура".
- Редактор свойств (Properties Editor):** A table showing the properties of the selected component.

Свойство	Характеристики	Значение
Отображаемое имя		Keyboard_1
Кардинальное число		1
Язык ввода	RW ⚡	<не определено>

## 1.1.4.2.4.5.5.1. Функции

Функция	Описание
<a href="#">LocalizedLanguageName</a>	Получает имя языка ввода
<a href="#">KeyState</a>	Отслеживает текущее состояние клавиши клавиатуры
<a href="#">NextLocale</a>	Переключиться на следующий язык
<a href="#">PrevLocale</a>	Переключиться на предыдущий язык

## 1.1.4.2.4.5.5.1.1. LocalizedLanguageName

Получает имя языка ввода.



```
string LocalizedLanguageName(string Locale)
```

### Параметры

Параметр	Тип	Описание
Locale	string	Текущая раскладка

### Возвращаемое значение

Тип	Описание
string	Языка ввода

### Примеры



```
//Вывести имя языка текущей раскладки клавиатуры в поле ввода:  
TextEdit.Text = Keyboard.LocalizedLanguageName("en-US");
```



## 1.1.4.2.4.5.5.1.2. KeyState

Отслеживает текущее состояние клавиши клавиатуры.



uint4 KeyState(Code)

### Параметры

Параметр	Тип	Описание
Code	uint4	Код клавиши

### Возвращаемое значение

Значение	Описание
3	Нажата и включена
2	Отпущена и включена
1	Нажата и выключена
0	Отпущена и выключена

### Коды клавиш

Клавиша	Код	Клавиша	Код
BackSpace	83	T	29
Tab	85	U	30
Enter	82	V	31
Shift	48	W	32
Ctrl	47	X	33
Alt	49	Y	34

Pause	80	Z	35
CapsLock	78	клавиша Windows	50
Esc	75	клавиша Applications	81
Пробел	84	NumPad 0	110
PageUp	103	NumPad 1	102
PageDown	104	NumPad 2	107
End	102	NumPad 3	104
Home	101	NumPad 4	108
←	108	NumPad 5	
↑	105	NumPad 6	106
→	106	NumPad 7	101
↓	107	NumPad 8	105
Insert	110	NumPad 9	103
Delete	109	NumPad *	97
0	0	NumPad +	99
1	1	NumPad -	98
2	2	NumPad .	109
3	3	NumPad /	96
4	4	F1	51
5	5	F2	52
6	6	F3	53
7	7	F4	54
8	8	F5	55
9	9	F6	56

A	10	F7	57
B	11	F8	58
C	12	F9	59
D	13	F10	60
E	14	F11	61
F	15	F12	62
G	16	NumLock	79
H	17	ScrollLock	77
I	18	;	42
J	19	=	38
K	20	~ `	36
L	21	- _	37
M	22	. >	44
N	23	/	46
O	24	[	40
P	25	\	45
Q	26	]	41
R	27	' "	39
S	28	, <	43

## Примеры



//Установить режим работы в зависимости от состояния клавиши Shift:

```
if (Keyboard_1.KeyState(48) == 2)
{
```

```
    AlarmsControl.ActiveMode = 0;
}
else
{
    if (Keyboard_1.KeyState(48) == 0)
        AlarmsControl.ActiveMode = 1;
}
```

## 1.1.4.2.4.5.5.1.3. NextLocale

Переключиться на следующий язык.



void NextLocale()

## 1.1.4.2.4.5.5.1.4. PrevLocale

Переключиться на предыдущий язык.



void PrevLocale()

## 1.1.4.2.4.5.5.2. События

Событие	Описание
<a href="#">LocaleChanged</a>	Сигнал смены языка

## 1.1.4.2.4.5.5.2.1. LocaleChanged

Сигнал смены языка.



## 1.1.4.2.4.5.5.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Язык ввода</a>	Текущая раскладка клавиатуры

## 1.1.4.2.4.5.5.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.5.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.5.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.5.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.5.5.3.5. Язык ввода

Текущая раскладка клавиатуры.



string Locale

### Примеры

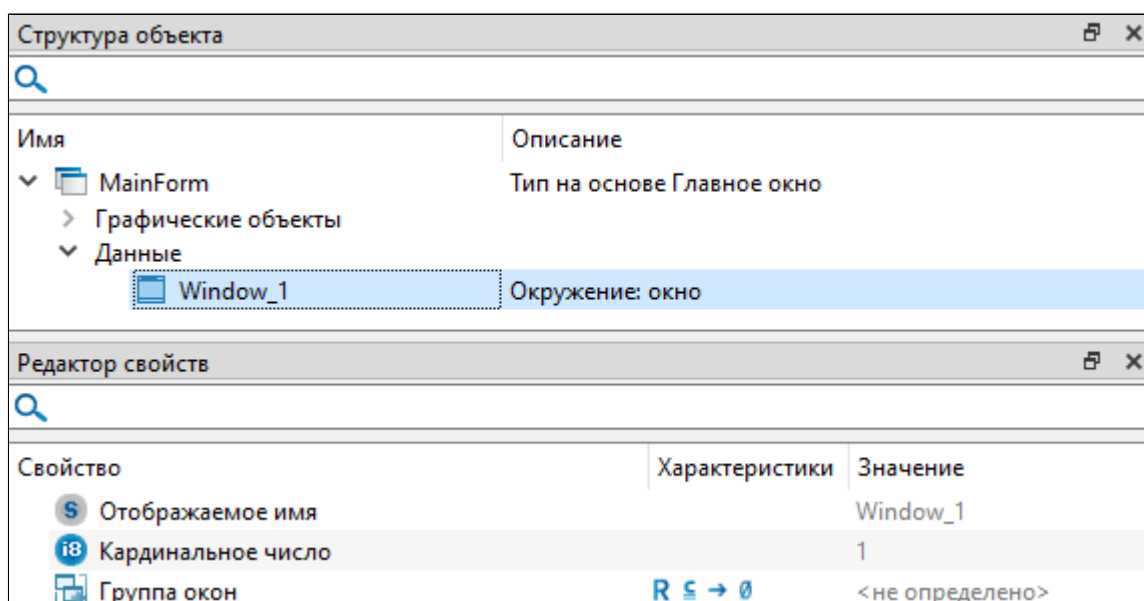


//Вывести информацию о текущей раскладке клавиатуры в поле ввода:

```
TextEdit.Text = Keyboard.Locale;
```

## 1.1.4.2.4.5.6. Окружение: окно

Позволяет работать с оконным окружением.



## 1.1.4.2.4.5.6.1. События

Событие	Описание
<a href="#">MessageReceived</a>	Получение сообщения



## 1.1.4.2.4.5.6.1.1. MessageReceived

Получение сообщения.

### Параметры

Параметр	Тип	Описание
messageType	string	Тип полученного сообщения (определяется пользователем). Значение параметра должно совпадать с типом сообщения в функции, отправляющей сообщение окну
messageData	string	Тело сообщения (определяется пользователем). Значение параметра может использоваться для коммуникаций между окнами

## 1.1.4.2.4.5.6.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Группа окон</a>	Ссылка на компонент Группа окон, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">WindowID</a>	Идентификатор окна

## 1.1.4.2.4.5.6.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.6.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.6.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.6.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.6.2.5. Группа окон

Ссылка на компонент Группа окон, относительно которого будет происходить дальнейшая работа с компонентом. Указывается на вкладке Редактор свойств.



Group

## 1.1.4.2.4.5.6.2.6. WindowID

Идентификатор (ID) окна.



string WindowID



Доступно только для чтения в режиме рантайма.

### Примеры



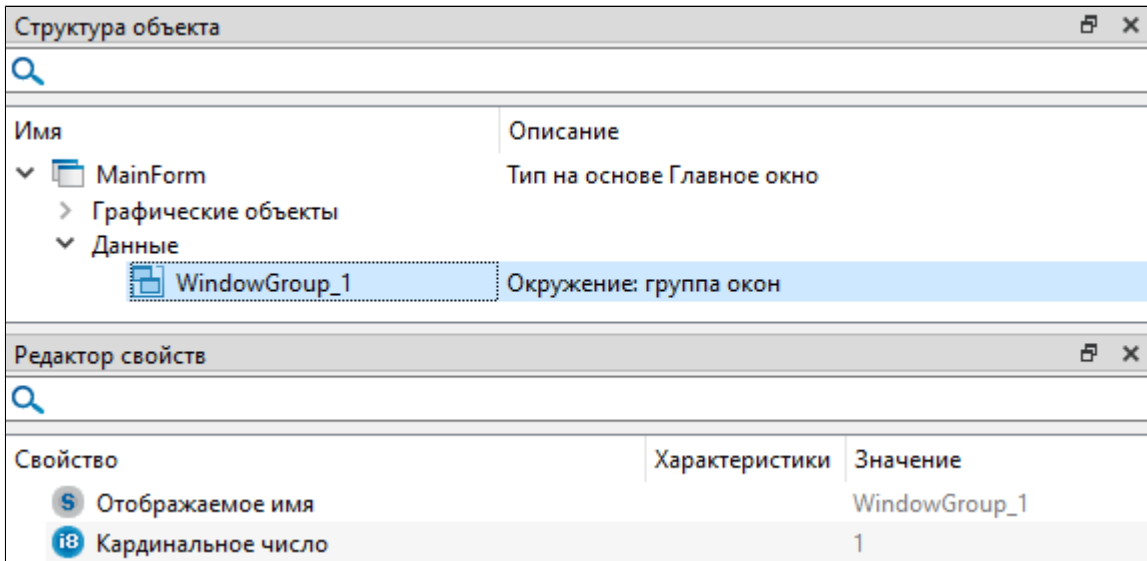
```
//Вывести идентификатор окна в поле ввода:  
TextEdit.Text = Window.WindowID;
```



## 1.1.4.2.4.5.7. Окружение: группа окон

Позволяет работать с окнами в группе.

Чтобы взаимодействовать с окнами в группе, добавьте на экранную форму компонент Окружение: группа окон. Компонент невидимый и виден только в области Структура объекта.



Компонент позволяет взаимодействовать с окнами внутри группы через свойства, функции и события.

## 1.1.4.2.4.5.7.1. События

Событие	Описание
<a href="#">WindowClosed</a>	Заккрытие окна в группе
<a href="#">WindowOpened</a>	Открытие нового окна в группе

## 1.1.4.2.4.5.7.1.1. WindowClosed

Закрытие окна в группе.

### Параметры

Параметр	Тип	Описание
windowID	string	Идентификатор закрываемого окна

## 1.1.4.2.4.5.7.1.2. WindowOpened

Открытие нового окна в группе.

### Параметры

Параметр	Тип	Описание
windowID	string	Идентификатор нового окна

## 1.1.4.2.4.5.7.2. Функции

Функция	Описание
<a href="#">SendToAll</a>	Отправляет сообщение всем окнам, находящимся в группе
<a href="#">SendMessageByID</a>	Отправляет сообщение конкретному окну, окно определяется по идентификатору
<a href="#">GetWindowIDbyIndex</a>	Определяет идентификатор окна по его индексу. Index первому элементу присваивается 0

## 1.1.4.2.4.5.7.2.1. SendToAll

Отправляет сообщение всем окнам, находящимся в группе.



```
void SendToAll(string MessageType, string MessageData)
```

### Параметры

Параметр	Тип	Описание
MessageType	string	Строковый параметр, содержащий тип сообщения. Значение должно совпадать со значением типа в событии MessageReceived (компонент Окружение: окно)
MessageData	string	Строковый параметр, содержащий тело сообщения. Значение должно совпадать со значением тела в событии MessageReceived (компонент Окружение: окно)

### Примеры



```
//Отправить сообщение на закрытие окна всем окнам:  
SendToAll("close", "Закрыть окно");
```

## 1.1.4.2.4.5.7.2.2. SendMessageByID

Отправляет сообщение конкретному окну, окно определяется по идентификатору.



```
void SendMessageByID(string WindowID, string MessageType, string  
MessageData)
```

### Параметры

Параметр	Тип	Описание
WindowID	string	Идентификатор окна, которому будет отправлено сообщение
MessageType	string	Строковый параметр, содержащий тип сообщения. Значение должно совпадать со значением типа в событии MessageReceived (компонент Окружение: окно)
MessageData	string	Строковый параметр, содержащий тело сообщения. Значение должно совпадать со значением тела в событии MessageReceived (компонент Окружение: окно)

### Примеры



```
//Отправить сообщение на закрытие окна первому открытому окну:  
SendMessageByID(GetWindowIDbyIndex(0), "close", "Закрыть окно");
```

## 1.1.4.2.4.5.7.2.3. GetWindowIDbyIndex

Определяет идентификатор окна по его индексу. Index первому элементу присваивается 0.



```
string GetWindowIDbyIndex(Index)
```

### Параметры

Параметр	Тип	Описание
Index	uint4	Идентификатор окна

### Примеры



```
//Вывести идентификатор первого окна группы в поле ввода:  
TextEdit.Text = GetWindowIDbyIndex(0);
```



## 1.1.4.2.4.5.7.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">WindowCount</a>	Количество окон в группе

## 1.1.4.2.4.5.7.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.7.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.7.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.7.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.7.3.5. WindowCount

Возвращает количество окон в группе.



uint8 WindowCount



Доступно только для чтения в режиме рантайма.

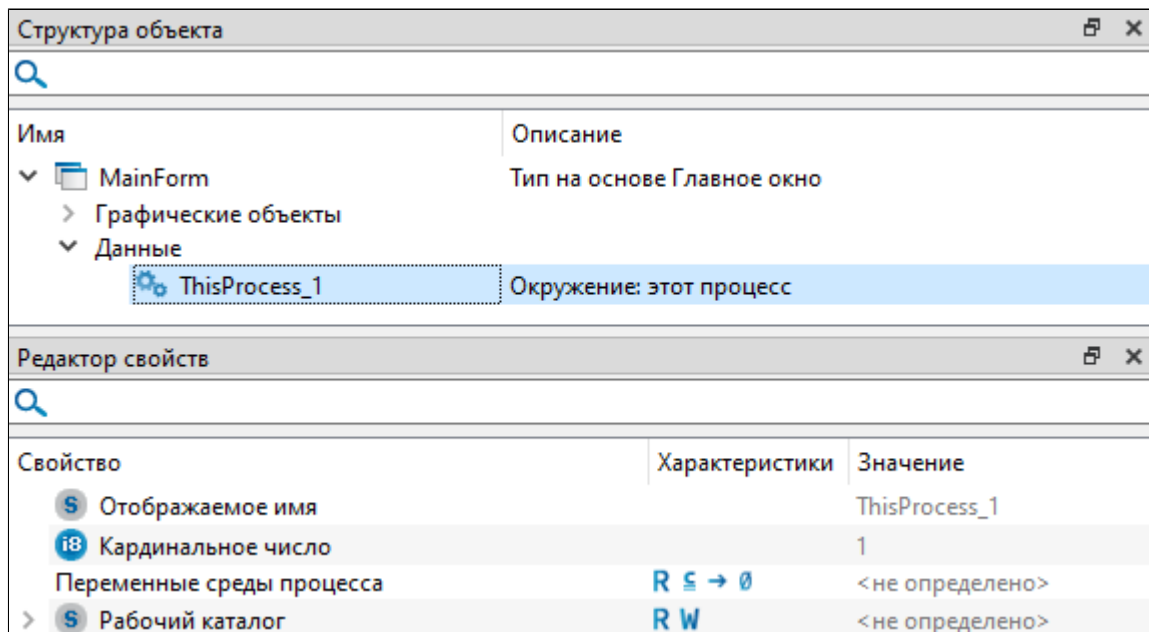
### Примеры



```
//Вывести количество окон в группе в поле ввода:  
TextEdit.Text = String.ToString(WindowGroup.WindowCount);
```

## 1.1.4.2.4.5.8. Окружение: этот процесс

Функции для работы с этим процессом HMI.



The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project's structure. The 'MainForm' folder is expanded, showing 'Графические объекты' (Visual Objects) and 'Данные' (Data). Under 'Данные', the 'ThisProcess\_1' object is selected and highlighted in blue. The description for 'ThisProcess\_1' is 'Окружение: этот процесс' (Environment: this process). The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties of the selected 'ThisProcess\_1' object. The properties are listed in a table with columns for 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		ThisProcess_1
Кардинальное число		1
Переменные среды процесса	R ⊆ → ∅	<не определено>
Рабочий каталог	R W	<не определено>

## 1.1.4.2.4.5.8.1. События

Событие	Описание
<a href="#">OnRunError</a>	Ошибка запуска



## 1.1.4.2.4.5.8.1.1. OnRunError

Ошибка запуска приложения.

## 1.1.4.2.4.5.8.2. Функции

Функция	Описание
<a href="#">Run</a>	Запуск процесса

## 1.1.4.2.4.5.8.2.1. Run

Запустить дочерний процесс по имени или пути с аргументами.



uint4 Run(string path, string arguments)

### Параметры

Параметр	Тип	Описание
path	string	Текстовый путь
arguments	string	Аргументы для запуска



Метод устарел, для запуска процесса используйте компонент Окружение: процесс.

## 1.1.4.2.4.5.8.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Переменные среды процесса</a>	Ссылка на переменные среды процесса
<a href="#">Рабочий каталог</a>	Рабочий каталог текущего процесса HMI
<a href="#">ID</a>	Идентификатор (ID) элемента
<a href="#">ExecutablePath</a>	Путь к исполняемому файлу процесса

## 1.1.4.2.4.5.8.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.8.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.8.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.8.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`



## 1.1.4.2.4.5.8.3.5. Переменные среды процесса

Ссылка на переменные среды процесса.



Environment

## 1.1.4.2.4.5.8.3.6. Рабочий каталог

Рабочий каталог текущего процесса HMI.



string WorkingDirectory

## 1.1.4.2.4.5.8.3.7. ID

Идентификатор (ID) элемента.



string Environment



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.8.3.8. ExecutablePath

Путь к исполняемому файлу процесса.



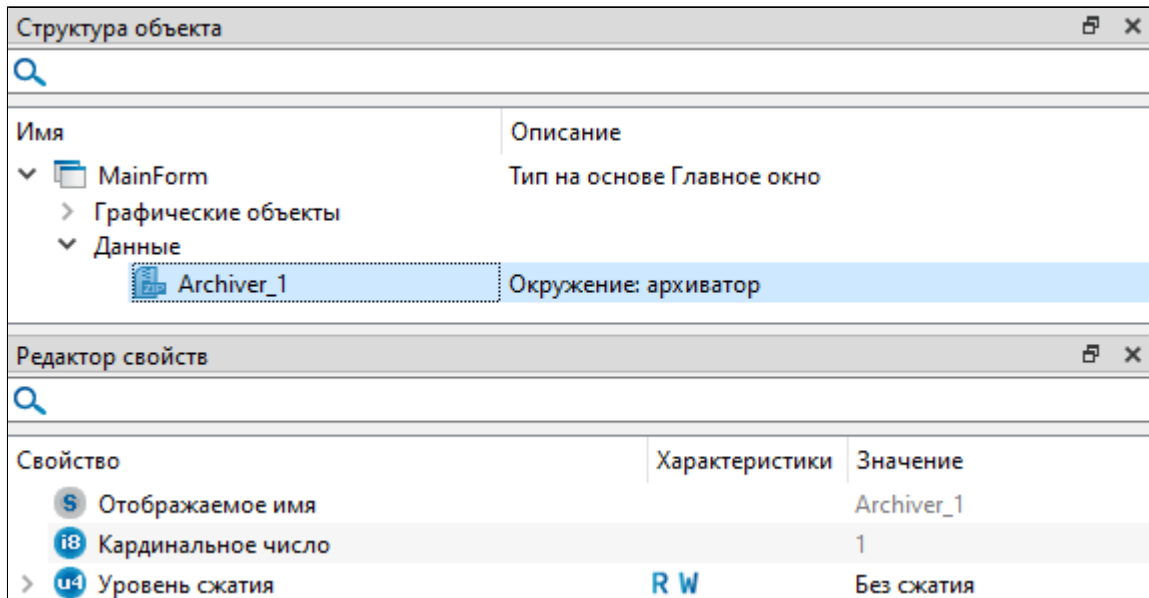
string ExecutablePath



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.9. Окружение: архиватор

Чтобы запаковать файлы в архив формата 7z и сохранить получившийся архив в любую папку, используйте элемент Окружение: архиватор.



Для упаковки используйте функцию `AsyncPack`. Перед упаковкой укажите степень сжатия в свойстве `Уровень сжатия`.

Для распаковки используйте функцию `AsyncUnpack`. Чтобы просмотреть список файлов и папок в архиве без распаковки, используйте функцию `BrowseAsJSON`.

## 1.1.4.2.4.5.9.1. События

Событие	Описание
<a href="#">UnpackFinished</a>	Завершение распаковки архива
<a href="#">PackFinished</a>	Завершение упаковки в архив

## 1.1.4.2.4.5.9.1.1. UnpackFinished

Завершение распаковки архива.

### Параметры

Параметр	Тип	Описание
error	string	Текст ошибки, возникшей при распаковке

## 1.1.4.2.4.5.9.1.2. PackFinished

Завершение упаковки в архив.

### Параметры

Параметр	Тип	Описание
error	string	Текст ошибки, возникшей при упаковке



## 1.1.4.2.4.5.9.2. Функции

Функция	Описание
<a href="#">AsyncUnpack</a>	Распаковывает указанный архив
<a href="#">AsyncPack</a>	Упаковывает файлы в архив формата 7z
<a href="#">Interrupt</a>	Останавливает текущую операцию (упаковку файлов или распаковку архива)
<a href="#">BrowseAsJSON</a>	Возвращает список файлов и папок в указанном архиве

## 1.1.4.2.4.5.9.2.1. AsyncUnpack

Распаковывает указанный архив.



```
bool AsyncUnpack(string FilePath, string DestPath)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь к архиву
DestPath	string	Путь к папке, в которую следует распаковать архив

### Возвращаемое значение

Значение	Описание
true	Архив успешно распакован
false	Распаковка архива не удалась

### Примеры



```
//Распаковать архив:
```

```
result: bool = Archiver_1.AsyncUnpack("C:/Tags/KP_086.7z", "S:/  
WORK_GROUP/Tags/KP_086");  
TextEdit_3 = Str.ToString(result);
```

## 1.1.4.2.4.5.9.2.2. AsyncPack

Упаковывает файлы в архив формата 7z.



```
bool AsyncPack(string SourcePath, string DestPath)
```

### Параметры

Параметр	Тип	Описание
SourcePath	string	Путь к папке с файлами, которые должны быть упакованы в архив
DestPath	string	Путь к папке, в которую следует сохранить результат упаковки

### Возвращаемое значение

Значение	Описание
true	Архив успешно создан
false	Архивирование не удалось

### Примеры



```
//Архивировать теги КП 086:  
result: bool = Archiver_1.AsyncPack("C:/DEMO_PROJECT/КП_086", "C:/  
Tags/КП_086.7z");  
TextEdit_3 = Str.ToString(result);
```

## 1.1.4.2.4.5.9.2.3. Interrupt

Останавливает текущую операцию- упаковку файлов или распаковку архива.



bool Interrupt()

### Примеры



```
//Остановить архивирование:  
Archiver_1.Interrupt();
```

## 1.1.4.2.4.5.9.2.4. BrowseAsJSON

Возвращает список файлов и папок в указанном архиве в JSON-формате.



string BrowseAsJSON(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь к архиву

### Значение

Значение	Описание
JSON-строка	Строка в JSON-формате. Содержит список файлов и папок в архиве

### Примеры



```
//Показать содержимое архива:  
content: string = Archiver_1.BrowseAsJSON("C:/Tags/KP_086.7z");  
TextEdit_3 = Str.ToString(content);
```

## 1.1.4.2.4.5.9.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">busy</a>	Позволяет проверить занят ли архиватор упаковкой или распаковкой
<a href="#">Уровень сжатия</a>	Степень сжатия архивируемых данных

## 1.1.4.2.4.5.9.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.9.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.5.9.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.9.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.9.3.5. Уровень сжатия

Степень сжатия архивируемых данных.



uint4 CompressionLevel

### Значение

Значение	Описание
0	Без сжатия
1	Быстрый
2	Нормальный
3	Максимальный

## 1.1.4.2.4.5.9.3.6. busy

Позволяет проверить занят ли архиватор упаковкой или распаковкой.



bool busy



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Архиватор занят
false	Архиватор свободен

## 1.1.4.2.4.5.10. Окружение: переменные среды процесса

Позволяет работать с переменными среды процесса в Astra.HMI.

The screenshot displays two windows from the Astra.HMI development environment. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project. Under 'MainForm', the 'Данные' (Data) folder is expanded, and 'ProcessEnvironment\_1' is selected. The description for 'ProcessEnvironment\_1' is 'Окружение: переменные среды процесса'. The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties for the selected 'ProcessEnvironment\_1' object. It has two properties: 'Отображаемое имя' (Visible Name) with a value of 'ProcessEnvironment\_1' and 'Кардинальное число' (Cardinality) with a value of '1'.

Имя	Описание
MainForm	Тип на основе Главное окно
Данные	
ProcessEnvironment_1	Окружение: переменные среды процесса

Свойство	Характеристики	Значение
Отображаемое имя		ProcessEnvironment_1
Кардинальное число		1

## 1.1.4.2.4.5.10.1. Функции

Функция	Описание
<a href="#">Get</a>	Позволяет получить значение переменной среды по ее имени
<a href="#">Set</a>	Позволяет задать значение переменной среды по её имени
<a href="#">Append</a>	Используется для добавления значения в конец переменной среды-списка
<a href="#">Prepend</a>	Используется для добавления значения в начало переменной среды-списка
<a href="#">Remove</a>	Используется для удаления указанной переменной среды
<a href="#">Clear</a>	Используется для удаления всех переменных среды
<a href="#">Reset</a>	Используется для сброса компонента к его первоначальному состоянию

## 1.1.4.2.4.5.10.1.1. Get

Позволяет получить значение переменной среды по ее имени.



string Get(string Name)

### Параметры

Параметр	Тип	Описание
Name	string	Имя переменной среды, значение которой необходимо получить

### Возвращаемое значение

Возвращает значение переменной среды с указанным именем. Если переменная с таким именем не найдена, функция вернет пустую строку.

### Примеры



```
//Вывести в текстовое поле значение переменной «TEMP»:  
Text_1.Text = ProcessEnvironment_1.Get("TEMP");
```

## 1.1.4.2.4.5.10.1.2. Set

Позволяет задать значение переменной среды по её имени. С помощью этой функции вы можете устанавливать значения как системных переменных среды, так и создавать или изменять пользовательские переменные, которые будут использоваться при запуске процессов с помощью компонента Окружение: процесс.



```
void Set(string Name, string Value)
```

### Параметры

Параметр	Тип	Описание
Name	string	Имя переменной среды, значение которой необходимо получить
Value	string	Значение, которое будет установлено для указанной переменной среды

### Примеры



```
//Задать значение "Hello,World!" для пользовательской переменной  
"CUSTOM_VAR":  
ProcessEnvironment_1.Get("CUSTOM_VAR" , "Hello,World!");
```



## 1.1.4.2.4.5.10.1.3. Append

Используется для добавления значения в конец переменной среды-списка. Если переменная среды уже содержит значения, то новое значение будет добавлено после последнего существующего значения с разделителем «;». Если переменная среды пуста, то новое значение будет установлено как первое значение переменной.



```
void Append(string Name, string Value)
```

### Параметры

Параметр	Тип	Описание
Name	string	Имя переменной среды, к которой нужно добавить значение в конец переменной среды
Value	string	Значение, которое нужно добавить в конец переменной среды

### Примеры



```
//Для переменной среды «PATH» со значением C:\Program Files  
\SomeApp добавить в конец списка C:\NewFolder:  
ProcessEnvironment_1.Append("PATH", "C:\\NewFolder"); // Результат  
выполнения: переменная "PATH" станет "C:\Program Files\SomeApp;  
\NewFolder";
```



Если у нас нет переменной «PATH», и мы вызываем функцию `Append("PATH", "C:\\NewFolder")`, то после выполнения функции значение переменной «PATH» будет установлено как `C:\NewFolder`.

## 1.1.4.2.4.5.10.1.4. Prepend

Используется для добавления значения в начало переменной среды-списка. Если переменная среды уже содержит значения, то новое значение будет добавлено перед существующим значением с разделителем «;» . Если переменная среды пуста, то новое значение будет установлено как первое значение переменной.



```
void Prepend(string Name, string Value)
```

### Параметры

Параметр	Тип	Описание
Name	string	Имя переменной среды, к которой нужно добавить значение в начало переменной среды
Value	string	Значение, которое нужно добавить в начало переменной среды

### Примеры

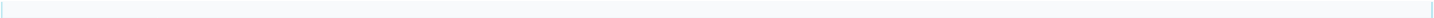


```
//Для переменной среды «PATH» со значением C:\Program Files\Java\jdk1.8.0\bin;C:\Python27;C:\Windows\System32 добавить в начало списка C:\MyApp.:
```

```
ProcessEnvironment_1.Prepend("PATH", "C:\\MyApp"); // Результат выполнения: переменная "PATH" станет "C:\MyApp;C:\Program Files\Java\jdk1.8.0\bin;C:\Python27;C:\Windows\System32";
```



Если у нас нет переменной «PATH», и мы вызываем функцию `Prepend("PATH", "C:\\MyApp")`, то после выполнения функции значение переменной «PATH» будет установлено как `C:\MyApp`.



## 1.1.4.2.4.5.10.1.5. Remove

Используется для удаления указанной переменной среды.



string Get(string Name)

### Параметры

Параметр	Тип	Описание
Name	string	Имя переменной среды, которую нужно удалить

### Примеры



```
//Удалить переменную среды «MY_VARIABLE»:  
ProcessEnvironment_1.Remove("MY_VARIABLE");
```

## 1.1.4.2.4.5.10.1.6. Clear

Используется для удаления всех переменных среды, которые были установлены в компоненте Окружение: переменные среды процесса.



void Clear()

### Примеры



```
//Удалить все переменные среды, которые были установлены в  
компоненте Окружение: переменные среды процесса.:  
ProcessEnvironment_1.Clear();
```

## 1.1.4.2.4.5.10.1.7. Reset

Используется для сброса компонента Окружение: переменные среды процесса к его первоначальному состоянию, удаляя все установленные переменные среды и возвращая компонент в исходное состояние.



```
void Reset()
```

### Примеры



```
//Сбросить компонент Окружение: переменные среды процесса к его  
первоначальному состоянию:  
ProcessEnvironment_1.Reset();
```

## 1.1.4.2.4.5.10.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.5.10.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.5.10.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.10.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.10.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



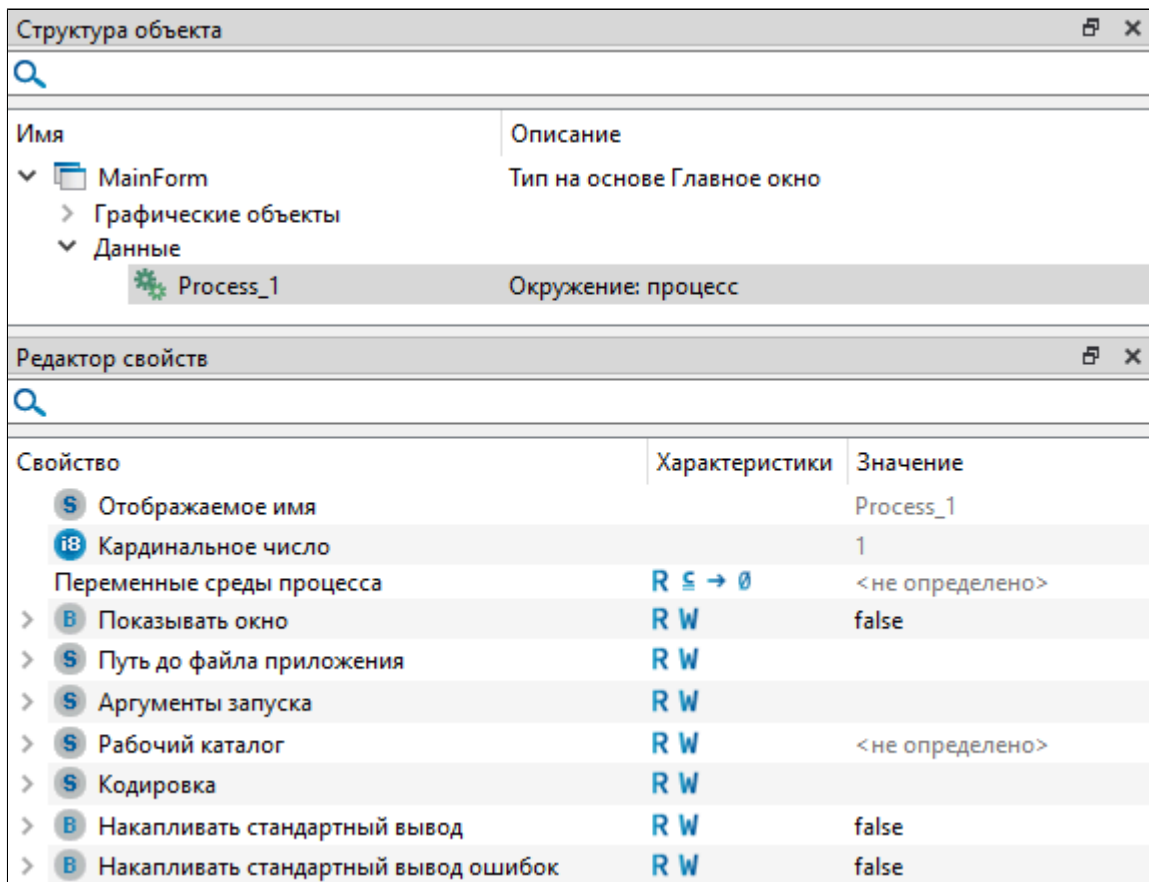
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.11. Окружение: процесс

Позволяет получить ID процесса Astra.HMI.



The screenshot displays two windows from an IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project. Under 'MainForm', there are 'Графические объекты' (Graphic Objects) and 'Данные' (Data). Under 'Данные', there is a folder named 'Process\_1' with a gear icon, described as 'Окружение: процесс' (Environment: process). The bottom window, titled 'Редактор свойств' (Properties Editor), shows a table of properties for the selected 'Process\_1' object.

Свойство	Характеристики	Значение
Отображаемое имя		Process_1
Кардинальное число		1
Переменные среды процесса	R $\leq$ $\rightarrow$ $\emptyset$	<не определено>
Показывать окно	R W	false
Путь до файла приложения	R W	
Аргументы запуска	R W	
Рабочий каталог	R W	<не определено>
Кодировка	R W	
Накапливать стандартный вывод	R W	false
Накапливать стандартный вывод ошибок	R W	false

## 1.1.4.2.4.5.11.1. События

Событие	Описание
<a href="#">StateChanged</a>	Сигнал об изменении состояния процесса
<a href="#">Started</a>	Процесс запущен
<a href="#">Finished</a>	Процесс завершен
<a href="#">StartFailed</a>	Ошибка запуска процесса
<a href="#">OnStdout</a>	Появились данные в stdout
<a href="#">OnStderr</a>	Появились данные в stderr

## 1.1.4.2.4.5.11.1.1. StateChanged

Сигнал об изменении состояния процесса.

### Параметры

Параметр	Тип	Описание
state	uint1	Состояние процесса

### Значение

Значение	Описание
0	Процесс выключен
1	Процесс запускается
1	Процесс запущен

## 1.1.4.2.4.5.11.1.2. Started

Процесс был запущен и начал свою работу.

### Параметры

Параметр	Тип	Описание
ID	uint4	Идентификатор запущенного процесса

## 1.1.4.2.4.5.11.1.3. Finished

Процесс завершил свою работу.

### Параметры

Параметр	Тип	Описание
exitCode	uint4	Код завершившегося процесса



## 1.1.4.2.4.5.11.1.4. StartFailed

Процесс не запущен из-за возникшей ошибки.

### Параметры

Параметр	Тип	Описание
error	string	Описание ошибки, которая привела к неудачному запуску процесса

## 1.1.4.2.4.5.11.1.5. OnStdout

В стандартный поток вывода «stdout» поступили новые данные от процесса, запущенного с помощью Окружение: процесс.

### Параметры

Параметр	Тип	Описание
value	string	Данные из стандартного потока вывода «stdout» процесса

## 1.1.4.2.4.5.11.1.6. OnStderr

В стандартный поток вывода ошибок «stderr» поступили новые данные, от процесса, запущенного с помощью Окружение: процесс.

### Параметры

Параметр	Тип	Описание
value	string	Данные из стандартного потока вывода ошибок «stderr»

## 1.1.4.2.4.5.11.2. Функции

Функция	Описание
<a href="#">Run</a>	Запуск процесса
<a href="#">Terminate</a>	Завершение процесса
<a href="#">Write</a>	Записать данные в поток stdin
<a href="#">ClearAccumulatedStdout</a>	Очистить накопленный вывод потока stdout
<a href="#">ClearAccumulatedStderr</a>	Очистить накопленный вывод потока stderr

## 1.1.4.2.4.5.11.2.1. Run

Запускает процесс с использованием текущих настроек компонента Окружение: процесс. При вызове этой функции процесс начинает свою работу, выполняя указанное исполняемое приложение с переданными аргументами командной строки и в указанном рабочем каталоге.

Функцию Run() следует вызывать после того, как вы настроили необходимые параметры процесса, такие как [Путь до файла приложения](#) и [Аргументы запуска](#).



void Run()

### Примеры



//Запустить «example.txt» с помощью приложения «notepad.exe» из системной папки C:\Windows\System32\. Окно запущенного приложения сделать видимым:

```
Process_1.Path = "C:\\Windows\\System32\\notepad.exe";
```

```
Process_1.Arguments = "C:\\Work\\example.txt";
```

```
Process_1.ShowWindow = true;
```

```
Process_1.Run();
```

## 1.1.4.2.4.5.11.2.2. Terminate

Завершает процесс, который был запущен с помощью компонента Окружение: процесс. При вызове этой функции процесс прекращает свою работу, и его выполнение завершается.



`void Terminate()`

## 1.1.4.2.4.5.11.2.3. Write

Используется для передачи данных в запущенный процесс и взаимодействия с ним через ввод данных.



```
void Write(string Message)
```

### Параметры

Параметр	Тип	Описание
Message	string	Данные для записи

### Примеры



```
//Передать в качестве аргумента имя «UserName»:  
Process_1.Write("UserName");
```

## 1.1.4.2.4.5.11.2.4. ClearAccumulatedStdout

Используется для очистки накопленного стандартного вывода «stdout». После вызова этой функции свойство Накопленный стандартный вывод будет пустым.



```
void ClearAccumulatedStdout()
```



## 1.1.4.2.4.5.11.2.5. ClearAccumulatedStderr

Используется для очистки накопленного стандартного вывода ошибок «stderr». После вызова этой функции свойство Накопленный стандартный вывод ошибок будет пустым.



```
void ClearAccumulatedStderr()
```

## 1.1.4.2.4.5.11.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">ID</a>	Количество окон в группе
<a href="#">Показывать окно</a>	Показывать окно процесса
<a href="#">Путь до файла приложения</a>	Путь до файла приложения
<a href="#">Аргументы запуска</a>	Аргументы запуска
<a href="#">Кодировка</a>	Кодировка ввода-вывода
<a href="#">Накапливать стандартный вывод</a>	Накапливать стандартный вывод
<a href="#">Накапливать стандартный вывод ошибок</a>	Накапливать стандартный вывод ошибок
<a href="#">Рабочий каталог</a>	Рабочий каталог текущего процесса HMI
<a href="#">Переменные среды процесса</a>	Ссылка на переменные среды процесса
<a href="#">ExecutablePath</a>	Путь к исполняемому файлу процесса
<a href="#">State</a>	Путь к исполняемому файлу процесса
<a href="#">ExitCode</a>	Код завершения
<a href="#">AccumulatedStderr</a>	Накопленный стандартный вывод
<a href="#">AccumulatedStdout</a>	Накопленный стандартный вывод ошибок

## 1.1.4.2.4.5.11.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.11.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.11.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.11.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.11.3.5. Показывать окно

Отображать ли окно запускаемого внешнего процесса при его выполнении.



bool ShowWindow

### Значение

Значение	Описание
true	Окно запускаемого процесса будет отображаться на экране
false	Запускаемый процесс будет выполняться в фоновом режиме, без отображения его окна на экране

### Примеры



```
//Запустить приложение «MyApp.exe» с видимым окном:  
Process_2.ShowWindow = true;  
Process_2.Path = "C:\\Program Files\\MyApp\\MyApp.exe";  
Process_2.Run();
```

## 1.1.4.2.4.5.11.3.6. Путь до файла приложения

Позволяет указать путь к исполняемому файлу процесса, который будет запускаться с помощью компонента Окружение: процесс. Установка значения этого свойства определяет, какое приложение должно быть запущено при вызове метода Run().



string Path

### Примеры



```
//Установить путь к исполняемому файлу «MyApp.exe»:  
Process_2.Path = "C:\\Program Files\\MyApp\\MyApp.exe";
```



## 1.1.4.2.4.5.11.3.7. Аргументы запуска

Позволяет указать дополнительные аргументы командной строки, которые будут переданы приложению при его запуске с помощью компонента Окружение: процесс. Аргументы командной строки представляют собой дополнительные параметры, которые влияют на работу запускаемого процесса.



string Arguments



Если аргументы содержат пробелы, необходимо заключить их в двойные кавычки. Это важно для корректной интерпретации параметров процессом. Например, если вам нужно передать аргумент `D:\\Project Test\\hello.py` содержащий пробел, то свойство Аргументы запуска следует установить так:

```
Process_1.Arguments = "\"D:\\Project Test\\hello.py\"";
```

### Примеры



//Открыть текстовый файл «1.txt» с помощью «notepad.exe»:

```
Process_1.Path = "C:\\Windows\\System32\\notepad.exe";
```

```
Process_1.Arguments = "C:\\Work\\1.txt";
```

```
Process_1.ShowWindow = true;
```

```
Process_1.Run();
```

## 1.1.4.2.4.5.11.3.8. Рабочий каталог

Позволяет задать рабочий каталог для запускаемого внешнего процесса.

Рабочий каталог представляет собой директорию, в которой будет выполняться запускаемый процесс. Используется, если запускаемому процессу необходим доступ к файлам или ресурсам, расположенным в определенной директории, и его работа зависит от этой локации.



string WorkingDirectory

### Примеры



//Задать рабочий каталог «C:\Program Files\Work\Process» для запускаемого процесса:

```
Process_1.WorkingDirectory = "C:\\Program Files\\Work\\Process";
```

## 1.1.4.2.4.5.11.3.9. Кодировка

Позволяет указать кодировку, которая будет использоваться при выполнении внешнего процесса. Кодировка определяет, как символы текста будут представлены в байтовом виде.



string Encoding

### Примеры



//Открыть текстовый файл 1.txt с помощью notepad.exe с содержимым в кодировке «UTF-8»:

```
Process_1.Path = "C:\\Windows\\System32\\notepad.exe";
```

```
Process_1.Arguments = "C:\\Work\\1.txt";
```

```
Process_1.ShowWindow = true;
```

```
Process_1.Encoding = "UTF-8";
```

```
Process_1.Run();
```

# 1.1.4.2.4.5.11.3.10. Накапливать стандартный вывод

Указывает, следует ли накапливать стандартный вывод «stdout» при выполнении процесса, запущенного с использованием компонента Окружение: процесс. Когда данное свойство установлено в значение «true», компонент Окружение: процесс будет накапливать данные, выводимые в стандартный поток вывода и сохранять их в свойстве Накопленный стандартный вывод.



bool AccumulateStdout

## Значение

Значение	Описание
true	Накопление стандартного вывода включено
false	Накопление стандартного вывода отключено (по умолчанию)

## 1.1.4.2.4.5.11.3.11. Накапливать стандартный вывод ошибок

Указывает, следует ли накапливать стандартный вывод ошибок «stderr» при выполнении процесса, запущенного с использованием компонента Окружение: процесс. Когда данное свойство установлено в значение «true», компонент Окружение: процесс будет накапливать данные, выводимые в стандартный поток вывода ошибок и сохранять их в свойстве Накопленный стандартный вывод ошибок.



bool AccumulateStderr

### Значение

Значение	Описание
true	Накопление стандартного вывода ошибок включено
false	Накопление стандартного вывода ошибок отключено (по умолчанию)

## 1.1.4.2.4.5.11.3.12. Переменные среды процесса

Указание ссылки дает возможность использовать настройки переменных окружения, определенных в компоненте Окружение: переменные среды процесса при запуске процесса с помощью компонента Окружение: процесс. Если ссылка на Окружение: переменные среды процесса указана, но переменные окружения не настроены, то будут использоваться значения переменных окружения по умолчанию из операционной системы.



Environment

## 1.1.4.2.4.5.11.3.13. ID

ID процесса Astra.HMI.



uint4 ID



Доступно только для чтения в режиме рантайма.

### Примеры



```
//Вывести ID процесса Astra.HMI в поле ввода:  
TextEdit.Text = Process_1.ID;
```

## 1.1.4.2.4.5.11.3.14. ExecutablePath

Путь к исполняемому файлу процесса.



string ExecutablePath



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.5.11.3.15. State

Путь к исполняемому файлу процесса.



uint1 State



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
0	Процесс выключен
1	Процесс запускается
2	Процесс запущен

## 1.1.4.2.4.5.11.3.16. ExitCode

Код завершения.



string ExitCode



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
0	Процесс успешно завершился
>0	Процесс завершился с ошибкой

## 1.1.4.2.4.5.11.3.17. AccumulatedStderr

Предоставляет доступ к накопленному стандартному выводу ошибок «stderr» после выполнения процесса, запущенного с использованием компонента Окружение: процесс. Если свойство Накапливать стандартный вывод ошибок было установлено в «true» процесс накапливает стандартный вывод ошибок, а затем сохраняет его в данном свойстве.



string AccumulatedStderr



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.11.3.18. AccumulatedStdout

Предоставляет доступ к накопленному стандартному выводу «stdout» после выполнения процесса, запущенного с использованием компонента Окружение: процесс. Если свойство Накапливать стандартный вывод установлено в значение «true», процесс накапливает стандартный вывод, а затем сохраняет его в данном свойстве.



string AccumulatedStdout



Доступно только для чтения в режиме рантайма.

### Примеры

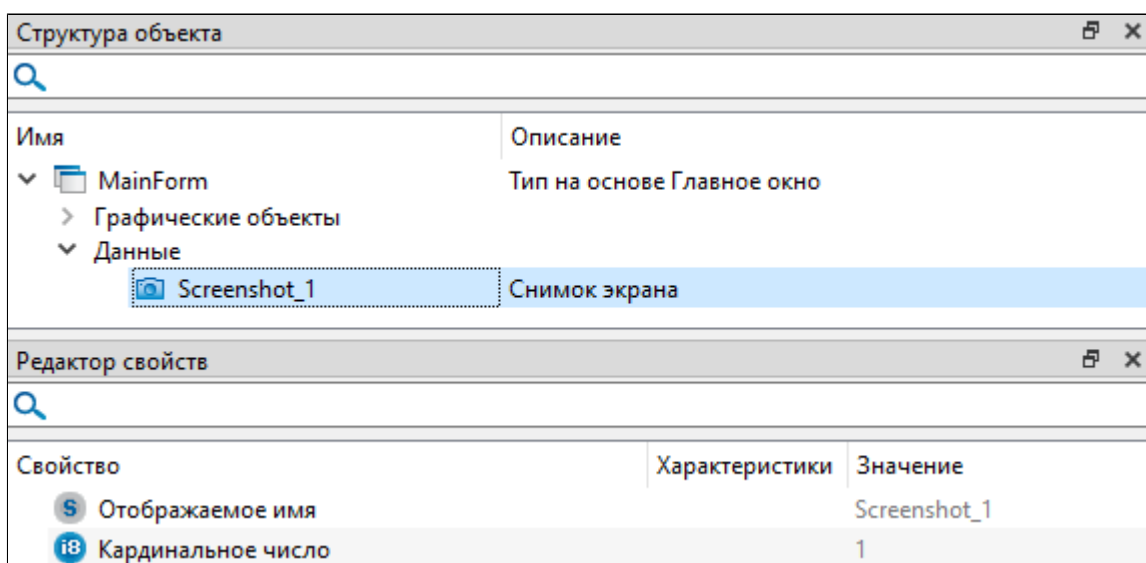


//Вывести в текстовое поле накопленный стандартный вывод после завершения процесса:

```
Text_1.Text = Process_1.AccumulatedStdout;
```

## 1.1.4.2.4.5.12. Снимок экрана

Элемент для создания скриншота экрана и сохранения его в графический файл.



## Функции

Компонент	Описание
<a href="#">Shot</a>	Сделать снимок во внутренний буфер
<a href="#">ShotRegion</a>	Сделать снимок области во внутренний буфер
<a href="#">ShotMonitor</a>	Сделать снимок монитора во внутренний буфер
<a href="#">ShotMonitorRegion</a>	Сделать снимок области монитора во внутренний буфер
<a href="#">Save</a>	Сохранить снимок в файл
<a href="#">Clear</a>	Очистить внутренний буфер

## 1.1.4.2.4.5.12.1.1. Shot

Сделать снимок во внутренний буфер.



bool Shot()

## 1.1.4.2.4.5.12.1.2. ShotRegion

Сделать снимок области во внутренний буфер.



```
bool ShotRegion(int4 x, int4 y, int4 width, int4 height)
```

### Параметры

Параметр	Тип	Описание
x	int4	Координата x начала области снимка
y	int4	Координата y начала области снимка
width	int4	Ширина области снимка
height	int4	Высота области снимка



## 1.1.4.2.4.5.12.1.3. ShotMonitor

Сделать снимок монитора во внутренний буфер.



bool ShotMonitor(int4 number)

### Параметры

Параметр	Тип	Описание
number	int4	Номер монитора

## 1.1.4.2.4.5.12.1.4. ShotMonitorRegion

Сделать снимок области монитора во внутренний буфер.



bool ShotMonitorRegion(int4 number, int4 x, int4 y, int4 width, int4 height)

### Параметры

Параметр	Тип	Описание
number	int4	Номер монитора
x	int4	Координата x начала области снимка
y	int4	Координата y начала области снимка
width	int4	Ширина области снимка
height	int4	Высота области снимка

## 1.1.4.2.4.5.12.1.5. Save

Сохранить снимок в файл.



bool Save(string path)

### Параметры

Параметр	Тип	Описание
path	string	Путь для сохраняемого файла

## 1.1.4.2.4.5.12.1.6. Clear

Очистить внутренний буфер.



`void Clear()`

## 1.1.4.2.4.5.12.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.5.12.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.12.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.12.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.5.12.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.13. Окружение: система

Элемент для получения информации об ОС, в которой запущен Astra.HMI.

The screenshot displays two windows from the Astra.HMI development environment. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project. Under the 'MainForm' folder, there is a 'Данные' (Data) folder containing a 'System\_1' object. The 'System\_1' object is selected and highlighted in blue. The description for 'System\_1' is 'Окружение: система' (Environment: system). The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties of the selected 'System\_1' object. It has a table with three columns: 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value). The table contains two rows: 'Отображаемое имя' (Display name) with value 'System\_1' and 'Кардинальное число' (Cardinality) with value '1'.

Свойство	Характеристики	Значение
Отображаемое имя		System_1
Кардинальное число		1

## 1.1.4.2.4.5.13.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Name</a>	Название
<a href="#">Family</a>	Семейство
<a href="#">Version</a>	Версия
<a href="#">Kernel</a>	Ядро
<a href="#">KernelVersion</a>	Версия ядра
<a href="#">Endianness</a>	Порядок байтов

## 1.1.4.2.4.5.13.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.5.13.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.5.13.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.5.13.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.5.13.1.5. Name

Название.



string Name



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.5.13.1.6. Family

Семейство.



string Family



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.13.1.7. Version

Версия.



string Version



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.13.1.8. Kernel

Ядро.



string Kernel



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.13.1.9. KernelVersion

Версия ядра.



string KernelVersion



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.5.13.1.10. Endianness

Порядок битов.



uint1 Endianness



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.6. OPC

Данный раздел документации устарел.

Протокол OPC DA более не применяется для построения ПТК AstraRegul, так как имеет ряд ограничений:

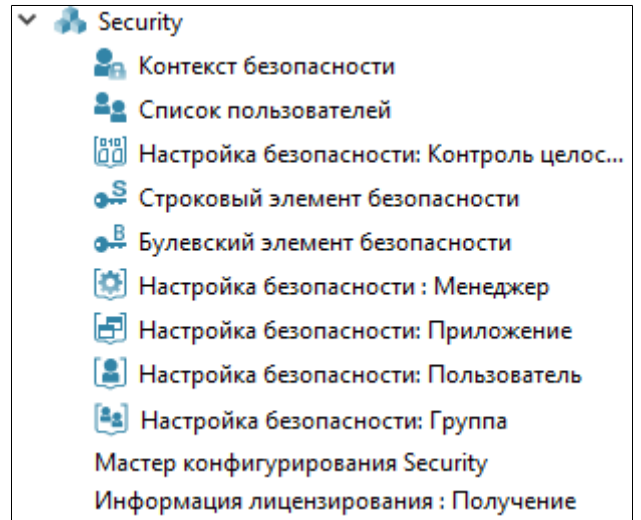
- › не является кроссплатформенным;
- › требует тонкой настройки DCOM;
- › не является безопасным.

Используйте в своих проектах элементы [AP](#).

## 1.1.4.2.4.7. Security

Набор компонентов Astra.NMI для взаимодействия с системой безопасности.

- › [Контекст безопасности](#)
- › [Список пользователей](#)
- › [Настройка безопасности: Контроль целостности](#)
- › [Строковый элемент безопасности](#)
- › [Булевский элемент безопасности](#)
- › [Настройка безопасности: Менеджер](#)
- › [Настройка безопасности: Приложение](#)
- › [Настройка безопасности: Пользователь](#)
- › [Настройка безопасности: Группа](#)
- › [Мастер конфигурирования Security](#)
- › [Информация лицензирования: Получение](#)



## 1.1.4.2.4.7.1. Контекст безопасности

Компонент позволяет взаимодействовать с подсистемой безопасности Astra.Security.



## 1.1.4.2.4.7.1.1. События

Событие	Описание
<a href="#">LoginRejected</a>	Отклонение аутентификации пользователя
<a href="#">PasswordExpiration</a>	Истечение срока действия пароля пользователя или приближение к времени истечения срока действия пароля
<a href="#">ConnectedChanged</a>	Изменение текущего подключения к Astra.Net.Agent
<a href="#">CurrentUserChanged</a>	Смена текущего пользователя
<a href="#">LoginStarted</a>	Начало аутентификации пользователя
<a href="#">LoginFailed</a>	Появление ошибки при аутентификации пользователя
<a href="#">AuditFailed</a>	Сигнал об ошибке логгирования сообщения
<a href="#">ForceStopUserSessionFinished</a>	Сигнал об успешном принудительном завершении сеанса пользователя
<a href="#">ResetUserFailedLoginCounterFinished</a>	Сигнал об успешном сбросе таймаута по превышению количества неуспешных попыток ввода пароля
<a href="#">RemoteGetLoggedUsersFinished</a>	Получение списка текущих пользователей на удаленной рабочей станции в сети Astra.Net
<a href="#">RemoteGetLoggedUsersFailed</a>	Не удалось получить список текущих пользователей на удаленной рабочей станции в сети Astra.Net



## 1.1.4.2.4.7.1.1.1. LoginRejected

Отклонение попытки входа подсистемой безопасности Astra.Security.

Активируется в момент входа при:

- › вводе неверных учетных данных;
- › истечении срока действия пароля;
- › отсутствии прав на вход.

### Параметры

Параметр	Тип	Описание
errorMessage	string	Сообщение с причиной отклонения входа

## 1.1.4.2.4.7.1.1.2. PasswordExpiration

Уведомление об истечении срока действия пароля.

Активируется при входе с учетными данными в случае, если оставшееся время действия пароля находится внутри срока уведомления о смене пароля.

### Параметры

Параметр	Тип	Описание
timeRemain	int4	Время, оставшееся до окончания действия пароля.



Событие можно использовать при наличии у текущего пользователя права Уведомление о смене пароля, дней (PasswordNotifyForChange). Назначить право можно в Конфигуратор Astra.Security.

## 1.1.4.2.4.7.1.1.3. ConnectedChanged

Изменение текущего подключения к Astra.Net.Agent.

Активируется в момент разрыва или появления соединения с Агент Astra.Security.

### Параметры

Параметр	Тип	Описание
connected	bool	Новое состояние подключения

### Значение

Значение	Описание
true	Есть соединение с Агент Astra.Security
false	Нет соединения с Агент Astra.Security

## 1.1.4.2.4.7.1.1.4. CurrentUserChanged

Смена текущего пользователя.

Активируется в момент регистрации подсистемой безопасности нового текущего пользователя.

### Параметры

Параметр	Тип	Описание
user	string	Имя нового пользователя

### Возвращаемое значение

Возвращает логин текущего пользователя в параметр user.

## 1.1.4.2.4.7.1.1.5. LoginStarted

Запуск попытки аутентификации пользователя.

Активируется в момент передачи введенных учетных данных подсистеме безопасности Astra.Security.

## 1.1.4.2.4.7.1.1.6. LoginFailed

Возникновение ошибки при попытке входа.

Активируется при проблемах соединения с Агент Astra.Security.

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки



## 1.1.4.2.4.7.1.1.7. AuditFailed

Данный сигнал уведомляет об ошибке логирования сообщения пользователя в аудит.



Аудит – это запись сообщений в базу данных на сервере.

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.1.1.8. ForceStopUserSessionFinished

Текущая сессия пользователя завершена.

Активируется в случае успешного завершения операции [ForceStopUserSession\(\)](#).

## 1.1.4.2.4.7.1.1.9.

# ResetUserFailedLoginCounterFinished

Счетчик неудачных попыток входа пользователя сброшен.

Активируется в случае успешного завершения операции [ResetUserFailedLoginCounter\(\)](#).

## 1.1.4.2.4.7.1.1.10.

### RemoteGetLoggedUsersFinished

Получен список текущих пользователей на удаленной рабочей станции в сети Astra.Net.

Активируется в случае успешного завершения операции [RemoteGetLoggedUsersList\(\)](#). Список помещается во внутреннее хранилище компонента.

Чтобы воспользоваться данными из списка, используйте функции [GetRemoteLoggedUserCount\(\)](#) и [GetRemoteLoggedUserByIndex\(\)](#).

## 1.1.4.2.4.7.1.1.11. RemoteGetLoggedUsersFailed

Не удалось получить список текущих пользователей на удаленной рабочей станции в сети Astra.Net.

Активируется в случае неуспешного завершения операции [RemoteGetLoggedUsersList\(\)](#).

### Значение

Значение	Описание
1	Не удалось отправить запрос на удаленную машину
2	Имя удаленной машины совпадает с именем локальной

## 1.1.4.2.4.7.1.2. Функции

Функция	Описание
<a href="#">GroupDisplayName</a>	Возвращает отображаемое имя группы по указанному индексу
<a href="#">Group</a>	Возвращает идентификатор группы по указанному индексу группы
<a href="#">Logout</a>	Выход пользователя из подсистемы безопасности
<a href="#">ChangePassword</a>	Меняет пароль текущего пользователя
<a href="#">AsyncLoginWithPasswordChange</a>	Асинхронная авторизация пользователя в подсистеме безопасности с последующей сменой пароля
<a href="#">AsyncLogin</a>	Асинхронная авторизация пользователя в подсистеме безопасности по логину и паролю
<a href="#">Login</a>	Авторизация пользователя в подсистеме безопасности
<a href="#">LogAudit</a>	Логировать сообщение пользователя
<a href="#">LogAuditExt</a>	Логировать сообщение пользователя
<a href="#">ForceStopUserSession</a>	Принудительно завершить пользовательскую сессию
<a href="#">ResetUserFailedLoginCounter</a>	Сброс таймаута по превышению количества неуспешных попыток ввода пароля
<a href="#">RemoteGetLoggedUsersList</a>	Запрашивает список текущих пользователей на удаленной рабочей станции в сети Astra.Net
<a href="#">GetRemoteLoggedUserCount</a>	Возвращает количество текущих пользователей на удаленной рабочей станции

<a href="#">GetRemoteLoggedUserByIndex</a>	Возвращает логин одного из текущих пользователей на удаленной рабочей станции
--	---

## 1.1.4.2.4.7.1.2.1. GroupDisplayName

Возвращает отображаемое имя группы по указанному индексу.



GroupDisplayName(Index)

### Параметры

Параметр	Тип	Описание
Index	uint8	Индекс группы



Нумерация групп пользователя в списке начинается с 0.

### Примеры



Вызов: SecurityContext.GroupDisplayName(0)



## 1.1.4.2.4.7.1.2.2. Group

Возвращает идентификатор группы по указанному индексу группы.



Group(Index)

### Параметры

Параметр	Тип	Описание
Index	uint8	Индекс группы



Нумерация групп пользователя в списке начинается с 0.

### Примеры



Вызов: SecurityContext.Group(0)

## 1.1.4.2.4.7.1.2.3. Logout

Выход пользователя из подсистемы безопасности. Функция не требует входных параметров.



void Logout()

### Примеры



Вызов: SecurityContext.Logout()

Результат: завершение сессии. Текущим пользователем становится дефолтный пользователь.

## 1.1.4.2.4.7.1.2.4. ChangePassword

Меняет пароль текущего пользователя.



ChangePassword (OldPassword, NewPassword)

### Параметры

Параметр	Тип	Описание
OldPassword	string	Пароль пользователя
NewPassword	string	Новый пароль пользователя



Изменить пароль нельзя, если не истек минимальный срок действия пароля, указанный в поле Срок действия пароля, дней (PasswordAge).

### Примеры



Вызов: `SecurityContext.ChangePassword("OldPassword", "NewPassword")`

Результат:

- пароль изменен;
- пароль не изменен.

## 1.1.4.2.4.7.1.2.5.

# AsyncLoginWithPasswordChange

Асинхронная авторизация пользователя в подсистеме безопасности с последующей сменой пароля. Активирует событие LoginStarted.



AsyncLoginWithPasswordChange (string Login, string OldPassword, NewPassword)

## Параметры

Параметр	Тип	Описание
Login	string	Логин пользователя
OldPassword	string	Старый пароль пользователя
NewPassword	string	Новый пароль пользователя



Изменить пароль нельзя, если не истек минимальный срок действия пароля, указанный в поле Срок действия пароля, дней (PasswordAge).

## Примеры



Вызов: SecurityContext.AsyncLoginWithPasswordChange("Login", "OldPassword", "NewPassword")

Результат:

- пароль изменен, вход произведен;
- пароль не изменен, вход не произведен.

## 1.1.4.2.4.7.1.2.6. AsyncLogin

Асинхронная авторизация пользователя в подсистеме безопасности по логину и паролю.



AsyncLogin(string Login, string Password)

### Параметры

Параметр	Тип	Описание
Login	string	Логин пользователя
Password	string	Пароль пользователя

### Примеры



Вызов: SecurityContext.AsyncLogin("Login", "Password")

Результат:

- › вход произведен;
- › вход не произведен.

## 1.1.4.2.4.7.1.2.7. Login

Авторизация пользователя в подсистеме безопасности. Активирует событие LoginStarted.



Login(string Login, string Password)

### Параметры

Параметр	Тип	Описание
Login	string	Логин пользователя
Password	string	Пароль пользователя



Исполнение функции останавливает другие процессы в проекте до тех пор, пока вход не будет выполнен или не возникнет ошибка входа. Поэтому проект в режиме исполнения может зависать. Для входа лучше использовать функцию AsyncLogin.

### Примеры



Вызов: SecurityContext.Login("Login", "Password")

Результат:

- › вход произведен;
- › вход не произведен.

## 1.1.4.2.4.7.1.2.8. LogAudit

Функция, записывающая сообщение аудита с указанной важностью.



Аудит – это запись сообщений от подсистемы безопасности в базу данных на сервере.



```
void LogAudit(string message, uint4 improtrance)
```

### Параметры

Параметр	Тип	Описание
message	string	Сообщение, которое необходимо отправить на аудит
improtrance	uint4	Уровень важности сообщения, оцененный по четырехбальной шкале, где 0 – наиболее важное сообщение, 3 – наименее важное сообщение

### Примеры



```
Вызов: SecurityContext.LogAudit("Обновил запись в журнале",2)
```

## 1.1.4.2.4.7.1.2.9. LogAuditExt

Функция, записывающая сообщение аудита с указанными важностью и типом.



Аудит – это запись сообщений от подсистемы безопасности в базу данных на сервере.



```
void LogAuditExt(string message, uint4 improtrance, string type)
```

### Параметры

Параметр	Тип	Описание
message	string	Сообщение, которое необходимо отправить на аудит
improtrance	uint4	Уровень важности сообщения, оцененный по четырехбальной шкале, где 0 – наиболее важное сообщение, 3 – наименее важное сообщение
type	string	Тип сообщения. Здесь следует указать одно из значений, описанных при настройке аудита в конфигурационном файле Astra.Security - astra.security.agent.xml. По умолчанию - Normal или Admin

### Примеры



```
Вызов: SecurityContext.LogAuditExt("Обновил запись в журнале",2,"Normal")
```



## 1.1.4.2.4.7.1.2.10. ForceStopUserSession

Завершает текущую сессию пользователя.

Завершить сессию можно как на локальном АРМ, так и на удаленном АРМ, входящем в сеть Astra.Net.



```
void ForceStopUserSession(string Login, string NetName)
```

### Параметры

Параметр	Тип	Описание
Login	string	Логин учетной записи пользователя
NetName	string	Имя узла в сети Astra.Net. Для завершения сессии на локальном АРМ имя узла в сети Astra.Net указывать не нужно

### Примеры



```
Вызов: SecurityContext.ForceStopUserSession("login","NetName")
```

## 1.1.4.2.4.7.1.2.11. ResetUserFailedLoginCounter

Сбрасывает счетчик неудачных попыток входа пользователя.

Сбросить счетчик можно как для попыток входа на локальном АРМ, так и на удаленном АРМ, входящем в сеть Astra.Net.



```
void ResetUserFailedLoginCounter(string Login, string NetName)
```

### Параметры

Параметр	Тип	Описание
Login	string	Логин учетной записи пользователя
NetName	string	Имя узла в сети Astra.Net. Для сброса счетчика на локальном АРМ имя узла в сети Astra.Net указывать не нужно



Для использования счетчика неудачных попыток входа пользователю должно быть назначено право Количество неуспешных попыток входа до временной блокировки, шт (MaxAttemptsCount).

### Примеры



```
Вызов: ResetUserFailedLoginCounter("Login", "NetName")
```

## 1.1.4.2.4.7.1.2.12. RemoteGetLoggedUsersList

Запрашивает список текущих пользователей на удаленной рабочей станции в сети Astra.Net.



```
void RemoteGetLoggedUsersList(string NetName)
```

В случае успешной активации события [RemoteGetLoggedUsersFinished](#) список поместится во внутреннее хранилище компонента. Чтобы воспользоваться данными из списка, используйте функции [GetRemoteLoggedUserCount\(\)](#) и [GetRemoteLoggedUserByIndex\(\)](#).

### Параметры

Параметр	Тип	Описание
NetName	string	имя узла в сети Astra.Net. Для завершения сессии на локальном АРМ имя узла в сети Astra.Net указывать не нужно

### Примеры



Вызов: `RemoteGetLoggedUsersList("NetName")`

Результат:

- › в случае успешного завершения операции активируется событие [RemoteGetLoggedUsersFinished](#);
- › в случае неуспешного завершения операции активируется событие [RemoteGetLoggedUsersFailed](#).

## 1.1.4.2.4.7.1.2.13. GetRemoteLoggedUserCount

Возвращает количество текущих пользователей на удаленной рабочей станции. Может использоваться только после вызова функции `RemoteGetLoggedUsersList()`.

Функция не требует входных параметров.



`uint4 GetRemoteLoggedUserCount()`

В случае успешной активации события [RemoteGetLoggedUsersFinished](#) список поместится во внутреннее хранилище компонента. Чтобы воспользоваться данными из списка, используйте функции [GetRemoteLoggedUserCount\(\)](#) и [GetRemoteLoggedUserByIndex\(\)](#).

### Примеры



Вызов: `SecurityContext.GetRemoteLoggedUserCount()`

## 1.1.4.2.4.7.1.2.14. GetRemoteLoggedUserByIndex

Возвращает логин одного из текущих пользователей на удаленной рабочей станции. Может использоваться только после вызова функции RemoteGetLoggedUsersList.



string GetRemoteLoggedUserByIndex(uint4 index)



Нумерация пользователей в списке начинается с 0.

### Параметры

Параметр	Тип	Описание
index	uint4	Порядковый номер пользователя в списке, полученном в результате вызова функции RemoteGetLoggedUsersList()

### Примеры



Вызов: SecurityContext.GetRemoteLoggedUserByIndex(4)

## 1.1.4.2.4.7.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">GroupCount</a>	Свойство, которое отображает количество групп текущего пользователя
<a href="#">PasswordChangeError</a>	Текст ошибки последней операции смены пароля пользователя
<a href="#">LoginRejectReason</a>	Причина отклонения последней авторизации пользователя в подсистеме безопасности
<a href="#">LoginError</a>	Текст ошибки при неудачной попытке авторизации
<a href="#">PasswordExpiresIn</a>	Остаток времени действия пароля пользователя в секундах
<a href="#">PasswordExpiresSoon</a>	Свойство уведомляет о скором сроке истечения действия пароля пользователя
<a href="#">PasswordExpires</a>	Наличие ограничений на минимальный / максимальный срок действия пароля
<a href="#">SessionExpiresIn</a>	Остаток времени сессии пользователя в секундах
<a href="#">SessionDurationLimit</a>	Максимальная длительность сессии пользователя в секундах
<a href="#">SessionStartTime</a>	Метка времени подключения текущего пользователя к подсистеме безопасности
<a href="#">ConnectionError</a>	Текст ошибки установки связи с Astra.Net.Agent
<a href="#">Connected</a>	Состояние подключения к Astra.Net.Agent
<a href="#">GuestMode</a>	Текущее состояние гостевого режима

<a href="#">CurrentUserName</a>	Отображаемое имя текущего пользователя подсистемы безопасности
<a href="#">CurrentUserId</a>	Уникальный идентификатор текущего пользователя подсистемы безопасности
<a href="#">CurrentUser</a>	Логин текущего пользователя, авторизованного в подсистеме безопасности
<a href="#">InactiveRemainTime</a>	Оставшееся время неактивности пользователя

## 1.1.4.2.4.7.1.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.7.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.1.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.1.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.7.1.3.5. GroupCount

Свойство, которое отображает количество групп текущего пользователя.



uint8 GroupCount



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `SecurityContext.GroupCount`

Пример значения: 2.

## 1.1.4.2.4.7.1.3.6. PasswordChangeError

Текст ошибки последней операции смены пароля пользователя.



string PasswordChangeError



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: SecurityContext.PasswordChangeError

Пример значения: Не вышел минимальный срок действия пароля.

## 1.1.4.2.4.7.1.3.7. LoginRejectReason

Причина отклонения последней авторизации пользователя в подсистеме безопасности.



string LoginRejectReason



Доступно только для чтения в режиме рантайма.

Подсистема безопасности отклоняет вход при:

- › вводе неверных учетных данных;
- › истечении срока действия пароля;
- › отсутствии прав на вход.

### Примеры



Вызов: SecurityContext.LoginRejectReason

Пример значения: Неверные учетные данные.

## 1.1.4.2.4.7.1.3.8. LoginError

Текст ошибки при неудачной попытке авторизации.



string LoginError



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: SecurityContext.LoginError

Пример значения: На узле 'LOCAL' зафиксирована ошибка: Служба 'SECURITYAGENT' не зарегистрирована.

## 1.1.4.2.4.7.1.3.9. PasswordExpiresIn

Остаток времени действия пароля пользователя в секундах. Если срок действия пароля неограничен, свойство равно 0.



uint4 PasswordExpiresIn



Доступно только для чтения в режиме рантайма.



В момент, когда пользователь меняет пароль, запускается таймер, отсчитывающий оставшееся время действия пароля.

Значение срока действия пароля для текущего пользователя указано в праве Срок действия пароля, дней (PasswordAge). Назначить право можно в Конфигуратор Astra.Security.

По истечении указанного в праве времени попытки входа пользователя отклоняются подсистемой безопасности Astra.Security. Узнать о том, что срок действия пароля истек, можно с помощью события [LoginRejected](#) компонента Контекст безопасности.

## Примеры



Вызов: SecurityContext.PasswordExpiresIn



## 1.1.4.2.4.7.1.3.10. PasswordExpiresSoon

Свойство уведомляет о скором сроке истечения действия пароля пользователя. Свойство связано с параметром PasswordNotifyForChange подсистемы безопасности Astra.Security.



bool PasswordExpiresSoon



Доступно только для чтения в режиме рантайма.



Значение свойства связано со значениями прав Срок действия пароля, дней (PasswordAge) и Уведомление о смене пароля, дней (PasswordNotifyForChange) текущего пользователя. Оставшееся время действия пароля сравнивается со сроком уведомления о смене пароля.

Назначить права можно в Конфигуратор Astra.Security.

### Значение

Значение	Описание
true	Оставшееся время действия пароля пользователя меньше или равно параметру PasswordNotifyForChange подсистемы безопасности Astra.Security
false	Оставшееся время действия пароля пользователя больше параметра PasswordNotifyForChange подсистемы безопасности Astra.Security

### Примеры



Вызов: SecurityContext.PasswordExpiresSoon

## 1.1.4.2.4.7.1.3.11. PasswordExpires

Наличие ограничений на минимальный/максимальный срок действия пароля.



bool PasswordExpires



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Ограничения есть
false	Ограничений нет

### Примеры



Вызов: SecurityContext.PasswordExpires

## 1.1.4.2.4.7.1.3.12. SessionExpiresIn

Остаток времени сессии пользователя в секундах.



uint4 SessionExpiresIn



Доступно только для чтения в режиме рантайма.



В момент, когда пользователь совершает вход с личными учетными данными, запускается таймер, отсчитывающий оставшееся время до завершения сессии.

Значение максимального времени длительности сессии для текущего пользователя указано в праве Максимальное время сессии, мин (SessionDurationLimit). Назначить право можно в Конфигуратор Astra.Security.

По истечении указанного в праве времени сессия текущего пользователя автоматически завершается.

## Примеры



Вызов: SecurityContext.SessionExpiresIn

## 1.1.4.2.4.7.1.3.13. SessionDurationLimit

Максимальная длительность сессии пользователя в секундах. По истечении указанного времени сессия пользователя завершится.



uint4 SessionDurationLimit



Доступно только для чтения в режиме рантайма.



Значение максимальной длительности сессии для текущего пользователя указано в праве Максимальное время сессии, мин (SessionDurationLimit). Назначить право можно в Конфигуратор Astra.Security.

### Примеры



Вызов: SecurityContext.SessionDurationLimit  
Пример значения: 120.

## 1.1.4.2.4.7.1.3.14. InactiveRemainTime

Оставшееся время неактивности пользователя, с.



uint4 InactiveRemainTime



Только для чтения в режиме рантайма.



В момент, когда пользователь перестает взаимодействовать с АРМ, запускается таймер, отсчитывающий оставшееся время неактивности.

Значение максимального времени неактивности для текущего пользователя указано в праве Максимальное время бездействия, мин (MaxIdleTime). Назначить право можно в Конфигуратор Astra.Security.

По истечении указанного в праве времени сессия текущего пользователя автоматически завершается.



Для отслеживания активности должен выполняться процесс Astra.Security CheckActivity Executable, запускаемый утилитой astra.security.useractivity.exe.

### Примеры



Вызов: SecurityContext.InactiveRemainTime

## 1.1.4.2.4.7.1.3.15. SessionStartTime

Метка времени подключения текущего пользователя к подсистеме безопасности.



timestamp SessionStartTime



Доступно только для чтения в режиме рантайма.

### Примеры



Пример вызова: `Text_Msg.Text =`

`DateTime.ToString(SecurityContext.SessionStartTime)`

Пример значения: 13.08.2021 09:20:14 после применения функции `DateTime.ToString()`.

## 1.1.4.2.4.7.1.3.16. ConnectionError

Текст ошибки установки связи с Astra.Net.Agent.



string ConnectionError



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: SecurityContext.ConnectionError

Пример значения: Соединение с Net агентом разорвано. Код ошибки1.



## 1.1.4.2.4.7.1.3.17. Connected

Состояние подключения к Astra.Net.Agent.



bool Connected



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Связь установлена
false	Связь не установлена

### Примеры



Вызов: `SecurityContext.Connected`

## 1.1.4.2.4.7.1.3.18. GuestMode

Текущее состояние гостевого режима.



bool GuestMode



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Пользователь использует подсистему безопасности как гость
false	Пользователь вошел в подсистему безопасности по логину и паролю

### Примеры



Вызов: `SecurityContext.CurrentUser`

## 1.1.4.2.4.7.1.3.19. CurrentUserName

Отображаемое имя текущего пользователя подсистемы безопасности.



string CurrentUserName



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `SecurityContext.CurrentUserName`

Пример значения: Иванов.

## 1.1.4.2.4.7.1.3.20. CurrentUserId

Уникальный идентификатор текущего пользователя подсистемы безопасности.



string CurrentUserId



Доступно только для чтения в режиме рантайма.



Уникальный идентификатор формируется подсистемой безопасности в момент создания учетной записи.

### Примеры



Вызов: SecurityContext.CurrentUserId

Вид значения: alpha:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Пример значения: alpha:c3a997b1-326e-42ee-8907-e359fc17feb7.

## 1.1.4.2.4.7.1.3.21. CurrentUser

Логин текущего пользователя, авторизованного в подсистеме безопасности.



string CurrentUser



Доступно только для чтения в режиме рантайма.



С момента ввода корректных учетных данных подсистема безопасности регистрирует вошедшего как текущего пользователя. Для всех свойств, функций и событий предоставляется контекст текущего пользователя.

### Примеры



Вызов: SecurityContext.CurrentUser

Пример значения: ivanov.

## 1.1.4.2.4.7.2. Список пользователей

Компонент позволяет:

- › извлекать список пользователей подсистемы безопасности Astra.Security;
- › обновлять извлеченный список пользователей.

## 1.1.4.2.4.7.2.1. События

Событие	Описание
<a href="#">UpdateFinished</a>	Завершение обновления списка пользователей
<a href="#">UpdateStarted</a>	Запуск обновления списка пользователей

## 1.1.4.2.4.7.2.1.1. UpdateFinished

Извлечение или обновление списка пользователей завершено.



## 1.1.4.2.4.7.2.1.2. UpdateStarted

Запущено извлечение или обновление списка пользователей.

Активируется в результате успешного завершения операции [BeginUpdate\(\)](#).

## 1.1.4.2.4.7.2.2. Функции

Компонент	Описание
<a href="#">GetLoginName</a>	Предоставляет логин пользователя по номеру в списке
<a href="#">GetDisplayName</a>	Предоставляет имя пользователя по номеру в списке
<a href="#">BeginUpdate</a>	Запускает извлечение или обновление списка пользователей

## 1.1.4.2.4.7.2.2.1. GetLoginName

Предоставляет логин пользователя по номеру в списке.



```
string GetLoginName(uint8 number)
```

### Параметры

Параметр	Тип	Описание
number	uint8	Порядковый номер логина пользователя в списке



Нумерация логинов пользователей в списке начинается с 0.

### Примеры



Вызов: `UserList.GetLoginName(0)`

```
i: var = 0;
```

```
while (i < Count)
```

```
//для всех учетных записей (Count)
```

```
{
```

```
ComboBox_UserList.AddItem(GetLoginName(i));
```

```
//добавление в выпадающий список всех логинов пользователей
```

```
i+=1;
```

```
}
```

## 1.1.4.2.4.7.2.2.2. GetDisplayName

Предоставляет имя пользователя по номеру в списке.



```
string GetDisplayName(uint8 number)
```

### Параметры

Параметр	Тип	Описание
number	uint8	Порядковый номер имени пользователя в списке



Нумерация имен пользователей в списке начинается с 0.

### Примеры



```
Вызов: userList.GetDisplayName(0)
i: var = 0;
while (i < Count)
//для всех учетных записей (Count)
{
ComboBox_UserList.AddItem(GetDisplayName(i));
//добавление в выпадающий список всех имен пользователей
i+=1;
}
```

## 1.1.4.2.4.7.2.2.3. BeginUpdate

Запускает извлечение или обновление списка пользователей.

Функция не требует входных параметров.



```
void BeginUpdate()
```

## 1.1.4.2.4.7.2.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Контекст безопасности</a>	Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">Error</a>	Текст ошибки при работе со списком пользователей
<a href="#">HasError</a>	Признак наличия ошибки при работе со списком пользователей
<a href="#">IsUpdatePending</a>	Статус выполнения процедуры обновления списка пользователей
<a href="#">Count</a>	Возвращает количество пользователей в списке

## 1.1.4.2.4.7.2.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.2.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.7.2.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.2.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.7.2.3.5. Контекст безопасности

Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом. Указывается на вкладке Редактор свойств.



Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.2.3.6. Error

Текст ошибки при работе со списком пользователей. Например, ошибка во время запроса.



string Error



Доступно только для чтения в режиме рантайма.

## Примеры



Вызов: `UserList.Error`

## 1.1.4.2.4.7.2.3.7. HasError

Наличие ошибок при извлечении или обновлении списка пользователей.



bool HasError



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `UserList.HasError`

## 1.1.4.2.4.7.2.3.8. IsUpdatePending

Статус обновления списка пользователей.



bool IsUpdatePending

### Значение

Значение	Описание
true	Список пользователей обновляется
false	Список пользователей не обновляется



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `UserList.IsUpdatePending`

## 1.1.4.2.4.7.2.3.9. Count

Количество пользователей в списке, извлеченном из подсистемы безопасности.



uint8 Count



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `UserList.Count`

## 1.1.4.2.4.7.3. Настройка безопасности: Контроль целостности

Подсистема безопасности Astra.Security позволяет контролировать целостность указанных файлов и папок. На основе контрольной суммы файлов и папок создается эталон состояния. Когда необходимо проверить их целостность, считается текущая контрольная сумма и сравнивается с эталонным значением.

Компонент Настройка безопасности: Контроль целостности предназначен для:

- › создания эталона состояния файлов и папок;
- › выполнения проверок состояния файлов и папок;
- › получения информации об изменениях в файлах и папках.



## 1.1.4.2.4.7.3.1. События

Событие	Описание
<a href="#">GetListFailed</a>	Появление ошибки получения списка контролируемых файлов
<a href="#">ListIsReady</a>	Завершение подготовки списка контролируемых файлов
<a href="#">CreateFailed</a>	Появление ошибки при создании эталонного файла
<a href="#">CreateFinished</a>	Завершение создания эталонного файла
<a href="#">UpdateFailed</a>	Появление ошибки при обновлении списка контролируемых файлов
<a href="#">UpdateFinished</a>	Завершение обновления списка контролируемых файлов
<a href="#">RemoteUpdateFinished</a>	Сигнал о завершении обновления контрольных значений на удаленном ноде
<a href="#">RemoteUpdateFailed</a>	Сигнал о неуспешном завершении операции обновления контрольных значений на удаленном ноде
<a href="#">RemoteCreateFinished</a>	Сигнал о завершении создания эталонного файла на удаленном ноде
<a href="#">RemoteCreateFailed</a>	Сигнал о неуспешном завершении операции создания эталона на удаленном ноде
<a href="#">RemoteListIsReady</a>	Результат проверки целостности файлов на удаленной рабочей станции в сети Astra.Net получен
<a href="#">RemoteGetListFailed</a>	Результат проверки целостности файлов на удаленной рабочей станции в сети Astra.Net не получен



## 1.1.4.2.4.7.3.1.1. GetListFailed

Не удалось получить результат проверки целостности файлов.

Активируется в случае неуспешного завершения операции [Get\\_IC\\_List\(\)](#).

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.3.1.2. ListIsReady

Получен результат проверки целостности файлов.

Активируется в случае успешного завершения операции [Get\\_IC\\_List\(\)](#).

Возвращает данные в JSON-формате в переменную IC\_List\_JSON.

## 1.1.4.2.4.7.3.1.3. CreateFailed

Не удалось создать эталон состояния файлов.

Активируется в случае неуспешного завершения операции [Create\\_IC\\_Etalon\(\)](#).

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.3.1.4. CreateFinished

Создан эталон состояния файлов.

Активируется в случае успешного завершения операции [Create\\_IC\\_Etalon\(\)](#).

## 1.1.4.2.4.7.3.1.5. UpdateFailed

Не удалось выполнить проверку целостности файлов.

Активируется в случае неуспешного завершения операции [Update\\_IC\(\)](#).

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.3.1.6. UpdateFinished

Выполнена проверка целостности файлов.

Активируется в случае успешного завершения операции [Update\\_IC\(\)](#).



## 1.1.4.2.4.7.3.1.7. RemoteUpdateFinished

Выполнена проверка целостности файлов на удаленной рабочей станции в сети Astra.Net.

Активируется в случае успешного завершения операции [UpdateRemote\\_IC\(\)](#).

## 1.1.4.2.4.7.3.1.8. RemoteUpdateFailed

Не удалось выполнить проверку целостности файлов на удаленной рабочей станции в сети Astra.Net.

Активируется в случае неуспешного завершения операции [UpdateRemote\\_IC\(\)](#).

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.3.1.9. RemoteCreateFinished

Создан эталон состояния файлов на удаленной рабочей станции в сети Astra.Net.

Активируется в случае успешного завершения операции [CreateRemote\\_IC\\_Etalon](#).

## 1.1.4.2.4.7.3.1.10. RemoteCreateFailed

Не удалось создать эталон состояния файлов на удаленной рабочей станции в сети Astra.Net.

Активируется в случае неуспешного завершения операции [CreateRemote\\_IC\\_Etalon](#).

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.3.1.11. RemoteListIsReady

Получен результат проверки целостности файлов на удаленной рабочей станции в сети Astra.Net.

Активируется в случае успешного завершения операции [GetRemote\\_IC\\_List](#).

Возвращает данные в JSON-формате в переменную IC\_List\_JSON.

## 1.1.4.2.4.7.3.1.12. RemoteGetListFailed

Не удалось получить результат проверки целостности файлов на удаленной рабочей станции в сети Astra.Net.

Активируется в случае неуспешного завершения операции [GetRemote\\_IC\\_List](#).

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.3.2. Функции

Компонент	Описание
<a href="#">Create_IC_Etalon</a>	Создает эталонный файл
<a href="#">Get_IC_List</a>	Вычисляет контрольную сумму файлов
<a href="#">Update_IC</a>	Вычисляет контрольную сумму файлов
<a href="#">UpdateRemote_IC</a>	Вычисляет контрольные суммы на удаленном рабочем месте
<a href="#">GetRemote_IC_List</a>	Получить список контролируемых файлов с удаленного рабочего места
<a href="#">CreateRemote_IC_Etalon</a>	Создать эталонный файл на удаленном рабочем месте

## 1.1.4.2.4.7.3.2.1. Create\_IC\_Etalon

Создает эталонный файл.



```
void Get_IC_Etalon()
```

Создает эталон на основе текущего состояния файлов и папок:

- › вычисляет текущую контрольную сумму;
- › сохраняет значение в качестве эталонного.

Функция не требует входных параметров.

### Примеры



Вызов: `SecurityIntegrityControl.Create_IC_Etalon()`

Результат:

- › в случае успешного завершения операции активируется событие [CreateFinished](#);
- › в случае неуспешного завершения операции активируется событие [CreateFailed](#).



## 1.1.4.2.4.7.3.2.2. Get\_IC\_List

Запрашивает результат проверки целостности файлов и папок.

Функция не требует входных параметров.



void Get\_IC\_List()

### Примеры



Вызов: SecurityIntegrityControl.Get\_IC\_List()

Результат:

- › в случае успешного завершения операции активируется событие [ListIsReady](#);
- › в случае неуспешного завершения операции активируется событие [GetListFailed](#).

## 1.1.4.2.4.7.3.2.3. Update\_IC

Вычисляет контрольную сумму файлов.



void Update\_IC()

Выполняет проверку целостности файлов и папок:

- › вычисляет текущую контрольную сумму;
- › сравнивает результат с эталонным значением.

Функция не требует входных параметров.

### Примеры



Вызов: SecurityIntegrityControl.Update\_IC()

Результат:

- › в случае успешного завершения операции активируется событие [UpdateFinished](#);
- › в случае неуспешного завершения операции активируется событие [UpdateFailed](#).

## 1.1.4.2.4.7.3.2.4. UpdateRemote\_IC

Вычисляет контрольные суммы на удаленном рабочем месте.



```
void UpdateRemote_IC(string netName)
```

### Параметры

Параметр	Тип	Описание
netName	string	Имя рабочей станции в сети Astra.Net

### Примеры



Вызов: `SecurityIntegrityControl.UpdateRemote_IC("Netname")`

Результат:

- › в случае успешного завершения операции активируется событие [RemoteUpdateFinished](#);
- › в случае неуспешного завершения операции активируется событие [RemoteUpdateFailed](#).

## 1.1.4.2.4.7.3.2.5. GetRemote\_IC\_List

Получить список контролируемых файлов с удаленного рабочего места. Аналогично функции [Get\\_IC\\_List\(\)](#), но на удаленной рабочей станции в сети Astra.Net.



```
void GetRemote_IC_List(string netName)
```

### Параметры

Параметр	Тип	Описание
netName	string	Имя рабочей станции в сети Astra.Net

### Примеры



Вызов: `SecurityIntegrityControl.GetRemote_IC_List("Netname")`

Результат:

- › в случае успешного завершения операции активируется событие [RemoteListIsReady](#);
- › в случае неуспешного завершения операции активируется событие [RemoteGetListFailed](#).

## 1.1.4.2.4.7.3.2.6. CreateRemote\_IC\_Etalon

Создать эталонный файл на удаленном рабочем месте. Аналогично функции [Create\\_IC\\_Etalon\(\)](#), но на удаленной рабочей станции в сети Astra.Net.



```
void CreateRemote_IC_Etalon(string netName)
```

### Параметры

Параметр	Тип	Описание
netName	string	Имя удаленного рабочего места

### Примеры



Вызов: `SecurityIntegrityControl.CreateRemote_IC_Etalon("Netname")`

Результат:

- › в случае успешного завершения операции активируется событие [RemoteCreateFinished](#);
- › в случае неуспешного завершения операции активируется событие [RemoteCreateFailed](#).

### 1.1.4.2.4.7.3.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Контекст безопасности</a>	Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">IC_List_JSON</a>	Список контролируемых файлов в формате JSON

## 1.1.4.2.4.7.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.7.3.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.3.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.7.3.3.5. Контекст безопасности

Ссылка на элемент типа Контекст безопасности, обеспечивающий взаимодействие с подсистемой безопасности Astra.Security. Указывается на вкладке Редактор свойств.



Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.3.3.6. IC\_List\_JSON

Список контролируемых файлов в формате JSON.



JSON IC\_List\_JSON



Доступно только для чтения в режиме рантайма.

### Примеры



```
{
  "date_last_check": "01.01.2021 00:00:01",
  "creator": "Guest(name)",
  "date_etalon_created": "01.01.2021 00:00:00",
  "data": [
    {
      "ID": "0",
      "ParentID": "-1",
      "Dir": "1",
      "hasRefValue": "0",
      "name": "C:\\Users\\controlled objects",
      "MD5_onStart": "",
      "date_onStart": "01.01.2021 00:00:00",
      "status": "4",
      "MD5_current": "",
      "date_current": "",
      "MD5_etalon": "",
      "date_etalon": ""
    },
    {
      "ID": "1",
```

```
"ParentID": "0",
"Dir": "0",
"hasRefValue": "0",
"name": "ex1.txt",
"MD5_onStart": "cadf2174b6ff3f2d027ee2393ff0a392",
"date_onStart": "01.01.2021 00:00:00",
"status": "4",
"MD5_current": "cadf2174b6ff3f2d027ee2393ff0a392",
"date_current": "01.01.2021 00:00:00",
"MD5_etalon": "",
"date_etalon": ""
}
]
}
```

Основной является следующая информация:

- › date\_last\_check – дата последней проверки целостности;
- › date\_etalon\_created – дата создания эталона;
- › Dir – флаг директории:
  - › 1: проверяемый объект – директория;
  - › 0: проверяемый объект – файл;
- › status – состояние проверяемого объекта:
  - › 0 – ошибок нет;
  - › 4 – файл не существует;
  - › 8 – нарушена целостность файла (контрольные суммы не совпадают).

## 1.1.4.2.4.7.4. Строковый элемент безопасности

Компонент, обеспечивающий связь со строковым правом подсистемы безопасности Astra.Security.

Связь позволяет получать разрешения и запреты для текущего пользователя.

## 1.1.4.2.4.7.4.1. События

Событие	Описание
<a href="#">ConnectedChanged</a>	Изменение подключения между Security context и Astra.Security.Agent
<a href="#">ValueChanged</a>	Изменение значения права при входе пользователя

## 1.1.4.2.4.7.4.1.1. ConnectedChanged

Смена состояния подписки на право.

### Параметры

Параметр	Тип	Описание
connected	bool	Текущее значение



## 1.1.4.2.4.7.4.1.2. ValueChanged

Смена текущего значения права.

### Параметры

Параметр	Тип	Описание
value	bool	Текущее значение

## 1.1.4.2.4.7.4.2. Функции

Компонент	Описание
<a href="#">GetForbidden</a>	Возвращает запрет строкового права по индексу запрета
<a href="#">GetAllowed</a>	Возвращает разрешение строкового права по индексу разрешения

## 1.1.4.2.4.7.4.2.1. GetForbidden

Предоставляет запрет текущего пользователя из права, связанного с элементом.



string GetForbidden(uint8 number)



Нумерация запретов начинается с 0.

### Параметры

Параметр	Тип	Описание
number	uint8	Порядковый номер запрета в списке запретов текущего пользователя

## 1.1.4.2.4.7.4.2.2. GetAllowed

Предоставляет разрешение текущего пользователя из права, связанного с элементом.



string GetAllowed(uint8 number)



Нумерация разрешений начинается с 0.

### Параметры

Параметр	Тип	Описание
number	uint8	Порядковый номер разрешения в списке разрешений текущего пользователя

## 1.1.4.2.4.7.4.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Контекст безопасности</a>	Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Error</a>	Отображает ошибки, связанные с токеном
<a href="#">Connected</a>	Свойство, которое отражает состояние подключения
<a href="#">ForbiddenCount</a>	Количество запрещений в строковом праве
<a href="#">AllowedCount</a>	Количество разрешений в строковом праве
<a href="#">Право</a>	Название булевского или строкового права в подсистеме Astra.Security
<a href="#">Приложение</a>	Приложение подсистемы Astra.Security, относительно которого применимо право

## 1.1.4.2.4.7.4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.4.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.7.4.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.7.4.3.5. Контекст безопасности

Ссылка на элемент типа Контекст безопасности, обеспечивающий взаимодействие с подсистемой безопасности Astra.Security. Указывается на вкладке Редактор свойств.



Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.4.3.6. Приложение

Приложение подсистемы Astra.Security, относительно которого применимо право.



string Application

### Примеры



Вызов: StringTokenProху.Application

## 1.1.4.2.4.7.4.3.7. Право

Имя права, с которым связан элемент.



string Right

### Примеры



Вызов: StringTokenProxy.Right

## 1.1.4.2.4.7.4.3.8. Error

Ошибка подписки.



string Error



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `StringTokenProxy.Error`

Пример значения: Не удалось получить значение токена права.  
Объект не найден в хранилище LDAP

## 1.1.4.2.4.7.4.3.9. Connected

Состояние подписки на право.

Подписка – это состояние соединения элемента с указанным правом.



bool Connected



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Связь между Security context и Astra.Security.Agent установлена
false	Связи между Security context и Astra.Security.Agent нет

### Примеры



Вызов: `StringTokenProxy.Connected`

## 1.1.4.2.4.7.4.3.10. ForbiddenCount

Количество запретов текущего пользователя в праве, связанном с элементом.



uint4 ForbiddenCount



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `StringTokenProху.ForbiddenCount`

Пример значения: 2.

## 1.1.4.2.4.7.4.3.11. AllowedCount

Количество разрешений текущего пользователя в праве, связанном с элементом.



uint4 AllowedCount



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `StringTokenProxy.AllowedCount`

Пример значения: 1.



## 1.1.4.2.4.7.5. Булевский элемент безопасности

Компонент, обеспечивающий связь с логическим правом подсистемы безопасности Astra.Security.

Связь позволяет отслеживать изменение значения права для текущего пользователя.

## 1.1.4.2.4.7.5.1. События

Событие	Описание
<a href="#">ConnectedChanged</a>	Изменение подключения между Security context и Astra.Security.Agent
<a href="#">ValueChanged</a>	Изменение значения права при входе пользователя

## 1.1.4.2.4.7.5.1.1. ConnectedChanged

Смена состояния подписки на право.

### Параметры

Параметр	Тип	Описание
connected	bool	Текущее значение

## 1.1.4.2.4.7.5.1.2. ValueChanged

Смена текущего значения права.

### Параметры

Параметр	Тип	Описание
value	bool	Текущее значение

## 1.1.4.2.4.7.5.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Контекст безопасности</a>	Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">Error</a>	Отображает ошибки, связанные с токеном
<a href="#">Connected</a>	Свойство, которое отражает состояние подключения
<a href="#">Value</a>	Значение булевского права, которое определено на сервере безопасности для текущего пользователя
<a href="#">Право</a>	Название булевского или строкового права в подсистеме Astra.Security
<a href="#">Приложение</a>	Приложение подсистемы Astra.Security, относительно которого применимо право

## 1.1.4.2.4.7.5.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.5.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.5.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.7.5.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.7.5.2.5. Контекст безопасности

Ссылка на элемент типа Контекст безопасности, обеспечивающий взаимодействие с подсистемой безопасности Astra.Security.



Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.5.2.6. Приложение

Приложение подсистемы Astra.Security, относительно которого применимо право.



string Application

### Примеры



Вызов: BoolTokenProху.Application

Пример значения: Управление состоянием оборудования.

## 1.1.4.2.4.7.5.2.7. Право

Название булевского или строкового права в подсистеме Astra.Security.



string Right

### Примеры



Вызов: BoolTokenProxy.Right

Пример значения: Управление насосом.

## 1.1.4.2.4.7.5.2.8. Error

Ошибка подписки.



string Error



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `BoolTokenProху.Error`

Пример значения: Не удалось получить значение токена права.

Объект не найден в хранилище LDAP

## 1.1.4.2.4.7.5.2.9. Connected

Состояние подписки на право.



bool Connected



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Связь между Security context и Astra.Security.Agent установлена
false	Связи между Security context и Astra.Security.Agent нет

### Примеры



Вызов: BoolTokenProxy.Connected

## 1.1.4.2.4.7.5.2.10. Value

Значение булевского права, которое определено на сервере безопасности для текущего пользователя.



bool Value



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Разрешено
false	Запрещено

### Примеры



Вызов: BoolTokenProxy.Value

## 1.1.4.2.4.7.6. Настройка безопасности: Менеджер

Компонент предназначен для:

- получения списков учетных записей, групп пользователей и приложений;
- удаления учетных записей, групп пользователей и приложений;
- создания и восстановления резервных копий конфигурации Astra.Security.

Используется для конфигурирования подсистемы безопасности Astra.Security из проектов Astra.HMI.



## 1.1.4.2.4.7.6.1. События

Событие	Описание
<a href="#">AgentStatusChanged</a>	Изменение состояния Astra.Security.Agent
<a href="#">DeleteUserFailed</a>	Появление ошибки при выполнении удаления пользователя
<a href="#">DeleteGroupFailed</a>	Появление ошибки при выполнении удаления группы
<a href="#">DeleteApplicationFailed</a>	Появление ошибки при выполнении удаления приложения
<a href="#">DeleteUserComplete</a>	Завершение удаления пользователя
<a href="#">DeleteGroupComplete</a>	Завершение удаления группы
<a href="#">DeleteApplicationComplete</a>	Завершение удаления приложения
<a href="#">RequestUsersListFailed</a>	Появление ошибки при выполнении запроса списка пользователей
<a href="#">RequestGroupListFailed</a>	Появление ошибки при выполнении запроса списка групп
<a href="#">RequestAppListFailed</a>	Появление ошибки при выполнении запроса списка приложений
<a href="#">RequestGroupListComplete</a>	Завершении выполнения запроса списка групп
<a href="#">RequestUsersListComplete</a>	Завершение выполнения запроса списка пользователей
<a href="#">RequestAppListComplete</a>	Завершение выполнения запроса списка приложений
<a href="#">GetConfigurationFinished</a>	Сигнал о завершении экспорта конфигурации LDAP
<a href="#">GetConfigurationFailed</a>	Сигнал об ошибке в процессе выполнения экспорта конфигурации LDAP
<a href="#">SetConfigurationFinished</a>	Сигнал о завершении импорта конфигурации LDAP

<a href="#">SetConfigurationFailed</a>	Сигнал об ошибке в процессе выполнения импорта конфигурации LDAP
<a href="#">LastActionError</a>	Ошибка выполнения последнего действия

## 1.1.4.2.4.7.6.1.1. AgentStatusChanged

Изменение текущего состояния Агент Astra.Security.



Все возможные состояния Агент Astra.Security описаны в свойстве AgentStatus.

## 1.1.4.2.4.7.6.1.2. DeleteUserFailed

Ошибка при удалении учетной записи из подсистемы безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [DeleteUser\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.6.1.3. DeleteGroupFailed

Ошибка при удалении группы пользователей из подсистемы безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [DeleteGroup\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.6.1.4. DeleteApplicationFailed

Ошибка при удалении приложения из подсистемы безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [DeleteApplication\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.6.1.5. DeleteUserComplete

Учетная запись удалена из подсистемы безопасности Astra.Security.

Активируется в случае успешного завершения операции [DeleteUser\(\)](#).

## 1.1.4.2.4.7.6.1.6. DeleteGroupComplete

Группа пользователей удалена из подсистемы безопасности Astra.Security.

Активируется в случае успешного завершения операции [DeleteGroup\(\)](#).



## 1.1.4.2.4.7.6.1.7. DeleteApplicationComplete

Приложение удалено из подсистемы безопасности Astra.Security.

Активируется в случае успешного завершения операции [DeleteApplication\(\)](#).

## 1.1.4.2.4.7.6.1.8. RequestUsersListFailed

Ошибка при получении списка пользователей подсистемы безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [RequestUsersList\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.6.1.9. RequestGroupListFailed

Ошибка при получении списка групп пользователей подсистемы безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [RequestGroupList\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.6.1.10. RequestAppListFailed

Ошибка при получении списка приложений подсистемы безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [RequestAppList\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.6.1.11. RequestGroupListComplete

Получен список групп пользователей, имеющихся в подсистеме безопасности Astra.Security.

Активируется в случае успешного завершения операции [RequestGroupList\(\)](#).

### Параметры

Параметр	Тип	Описание
JSONAppList	string	Список групп в JSON формате

## 1.1.4.2.4.7.6.1.12. RequestUsersListComplete

Получен список пользователей, имеющих в подсистеме безопасности Astra.Security.

Активируется в случае успешного завершения операции [RequestUsersList\(\)](#).

### Параметры

Параметр	Тип	Описание
JSONAppList	string	Список пользователей в JSON формате

## 1.1.4.2.4.7.6.1.13. RequestAppListComplete

Получен список приложений, имеющихся в подсистеме безопасности Astra.Security.

Активируется в случае успешного завершения операции [RequestAppList\(\)](#).

### Параметры

Параметр	Тип	Описание
JSONAppList	string	Список приложений в JSON формате

## 1.1.4.2.4.7.6.1.14. GetConfigurationFinished

Файл резервной копии конфигурации Astra.Security создан (или перезаписан) успешно.

Активируется в случае успешного завершения операции [ExportConfiguration\(\)](#).



## 1.1.4.2.4.7.6.1.15. GetConfigurationFailed

Ошибка при создании (или перезаписи) файла резервной копии конфигурации Astra.Security.

Активируется в случае неуспешного завершения операции [ExportConfiguration\(\)](#).

Возвращает код ошибки в переменную FailReason, если код ошибки равен:

- › 2 – не удалось экспортировать конфигурацию из LDAP (например, не существует корневого каталога и т.д.);
- › 3 – нет права на чтение конфигурации;
- › 4 – не удалось открыть файл для записи конфигурации.

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки

## 1.1.4.2.4.7.6.1.16. SetConfigurationFinished

Резервная копия конфигурации Astra.Security восстановлена успешно.

Активируется в случае успешного завершения операции [ImportConfiguration\(\)](#).

## 1.1.4.2.4.7.6.1.17. SetConfigurationFailed

Не удалось восстановить резервную копию конфигурации Astra.Security.

Активируется в случае неуспешного завершения операции [ImportConfiguration\(\)](#).

Возвращает код ошибки в переменную FailReason, если код ошибки равен:

- › 1 – файл резервной копии поврежден;
- › 2 – не удалось зачистить базу данных LDAP перед применением конфигурации;
- › 3 – произошла внутренняя ошибка LDAP;
- › 4 – произошла непредвиденная ошибка на стороне агента;
- › 5 – у пользователя нет права на редактирование конфигурации;
- › 6 – другие ошибки;
- › 7 – не удается открыть файл с конфигурацией.

## 1.1.4.2.4.7.6.1.18. LastActionError

Ошибка выполнения последнего действия.

Активируется в результате внесения в конфигурацию подсистемы безопасности таких изменений, которые могут привести к нарушению работы Astra.Security. Таким действием является, например, удаление учетной записи единственного администратора.

### Параметры

Параметр	Тип	Описание
errorMessage	string	Текст ошибки

## 1.1.4.2.4.7.6.2. Функции

Компонент	Описание
<a href="#">GetErrorDescriptionByCode</a>	Возвращает описание ошибки по коду
<a href="#">DeleteUser</a>	Удаляет пользователя
<a href="#">DeleteGroup</a>	Удаляет группу
<a href="#">DeleteApplication</a>	Удаляет приложение
<a href="#">RequestUsersList</a>	Запрашивает список пользователей
<a href="#">RequestGroupList</a>	Запрашивает список групп
<a href="#">RequestAppList</a>	Запрашивает список приложений
<a href="#">ExportConfiguration</a>	Экспортировать конфигурацию
<a href="#">ImportConfiguration</a>	Импортировать конфигурацию

## 1.1.4.2.4.7.6.2.1. GetErrorDescriptionByCode

Возвращает текстовое описание ошибок, возникающих при запросе и удалении приложений, групп пользователей и учетных записей.



```
string GetErrorDescriptionByCode(uint1 FailReason)
```

### Параметры

Параметр	Тип	Описание
FailReason	uint1	Код ошибки из любого события, говорящего об ошибке

## 1.1.4.2.4.7.6.2.2. DeleteUser

Удаляет четную запись из подсистемы безопасности.



```
void DeleteUser(string UserID)
```

### Параметры

Параметр	Тип	Описание
UserID	string	Уникальный идентификатор (uid) учетной записи



Удаление учетных записей доступно только пользователям с правами администратора.

### Примеры



Вызов: `SecurityManager.DeleteUser("alpha:d01969e0-ae05-4548-9375-2533b326a588")`

Результат:

- в случае успешного завершения операции активируется событие [DeleteUserComplete\(\)](#);
- в случае неуспешного завершения операции активируется событие [DeleteUserFailed\(\)](#).

## 1.1.4.2.4.7.6.2.3. DeleteGroup

Удаляет группу пользователей из подсистемы безопасности.



```
void DeleteGroup(string GroupUID)
```

### Параметры

Параметр	Тип	Описание
GroupID	string	Уникальный идентификатор (uid) группы



Удаление групп пользователей доступно только пользователям с правами администратора.

### Примеры



Вызов: `SecurityManager.DeleteGroup("alpha:d01969e0-ae05-4548-9375-2533b326a588")`

Результат:

- в случае успешного завершения операции активируется событие [DeleteGroupComplete](#);
- в случае неуспешного завершения операции активируется событие [DeleteGroupFailed](#).



## 1.1.4.2.4.7.6.2.4. DeleteApplication

Удаляет приложение.



```
void DeleteApplication(string appName)
```

### Параметры

Параметр	Тип	Описание
appName	string	Имя приложения

### Примеры



Вызов: `SecurityManager.DeleteApplication("Управление состоянием оборудования")`

Результат:

- › в случае успешного завершения операции активируется событие [DeleteApplicationComplete](#);
- › в случае неуспешного завершения операции активируется событие [DeleteApplicationFailed](#).

## 1.1.4.2.4.7.6.2.5. RequestUsersList

Запрашивает список всех учетных записей, имеющих в подсистеме безопасности Astra.Security.

Функция не требует входных параметров.



void RequestUsersList()



Просмотр списка учетных записей доступен только пользователям с правами администратора.

### Примеры



Вызов: SecurityManager.RequestUsersList()

Результат:

- › в случае успешного завершения операции активируется событие [RequestUsersListComplete](#);
- › в случае неуспешного завершения операции активируется событие [RequestUsersListFailed](#).

## 1.1.4.2.4.7.6.2.6. RequestGroupList

Запрашивает список всех групп пользователей, имеющих в подсистеме безопасности Astra.Security.

Функция не требует входных параметров.



void RequestGroupList()

Просмотр списка групп пользователей доступен только пользователям с правами администратора.

### Примеры



Вызов: SecurityManager.RequestGroupList()

Результат:

- › в случае успешного завершения операции активируется событие [RequestGroupListComplete](#);
- › в случае неуспешного завершения операции активируется событие [RequestGroupListFailed](#).

## 1.1.4.2.4.7.6.2.7. RequestAppList

Запрашивает список всех приложений, имеющих в подсистеме безопасности Astra.Security.

Функция не требует входных параметров.



void RequestAppList()

### Примеры



Вызов: SecurityManager.RequestAppList()

Результат:

- в случае успешного завершения операции активируется событие [RequestAppListComplete](#);
- в случае неуспешного завершения операции активируется событие [RequestAppListFailed](#).

## 1.1.4.2.4.7.6.2.8. ExportConfiguration

Создает (или перезаписывает) файл резервной копии текущей конфигурации Astra.Security в указанном месте.



```
void ExportConfiguration(string fullPath)
```

### Параметры

Параметр	Тип	Описание
fullPath	string	Полный путь к создаваемому файлу вместе с именем файла

### Примеры



Вызов: `SecurityManager.ExportConfiguration("D:/export.ldb")`

Результат:

- › в случае успешного завершения операции активируется событие [GetConfigurationFinished](#);
- › в случае неуспешного завершения операции активируется событие [GetConfigurationFailed](#).

## 1.1.4.2.4.7.6.2.9. ImportConfiguration

Восстанавливает резервную копию конфигурации Astra.Security из указанного файла.



```
void ImportConfiguration(string fullPath)
```

### Параметры

Параметр	Тип	Описание
fullPath	string	Полный путь к файлу резервной копии вместе с именем файла

### Примеры



Вызов: `SecurityManager.ImportConfiguration("D:/export.ldb")`

Результат:

- › в случае успешного завершения операции активируется событие [SetConfigurationFinished](#);
- › в случае неуспешного завершения операции активируется событие [SetConfigurationFailed](#).

## 1.1.4.2.4.7.6.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Контекст безопасности</a>	Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">AgentStatus</a>	Текущее состояние Astra.Security.Agent

## 1.1.4.2.4.7.6.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.7.6.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.6.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.6.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.7.6.3.5. Контекст безопасности

Ссылка на элемент типа Контекст безопасности, обеспечивающий взаимодействие с подсистемой безопасности Astra.Security.



Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.6.3.6. AgentStatus

Текущее состояние Astra.Security.Agent.



uint1 AgentStatus



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: SecurityManager.AgentStatus.

## 1.1.4.2.4.7.7. Настройка безопасности: Приложение

Компонент предназначен для загрузки информации о приложении:

- имеющемуся в подсистеме безопасности – для просмотра и изменения этой информации;
- созданном из проекта Astra.HMI – для дальнейшей отправки в подсистему безопасности.

## 1.1.4.2.4.7.7.1. События

Событие	Описание
<a href="#">SaveFailed</a>	Появление ошибки при сохранении приложения
<a href="#">SaveComplete</a>	Завершение сохранения приложения
<a href="#">LoadFailed</a>	Появление ошибки при загрузке приложения
<a href="#">LoadComplete</a>	Завершение выполнения загрузки приложения

## 1.1.4.2.4.7.7.1.1. SaveFailed

Информация о приложении не сохранена в подсистему безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [Save\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReasonCode	string	Номер ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).



## 1.1.4.2.4.7.7.1.2. SaveComplete

Информация о приложении сохранена в подсистему безопасности Astra.Security.

Активируется в случае успешного завершения операции [Save\(\)](#).

## 1.1.4.2.4.7.7.1.3. LoadFailed

Информация о приложении не загружена в элемент.

Активируется в случае неуспешного завершения операции [Load\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Номер ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.7.1.4. LoadComplete

Информация о приложении загружена в элемент.

Активируется в случае успешного завершения операции [Load\(\)](#).

## 1.1.4.2.4.7.7.2. Функции

Компонент	Описание
<a href="#">GetErrorDescriptionByCode</a>	Возвращает описание ошибки по коду
<a href="#">RoleDeleteRight</a>	Удаляет значение права приложения подсистемы безопасности
<a href="#">RoleChangeRight</a>	Изменяет значение права приложения подсистемы безопасности
<a href="#">RoleAddRight</a>	Шаблон описания
<a href="#">DeleteRole</a>	Удаляет одну из ролей приложения подсистемы безопасности
<a href="#">ChangeRole</a>	Изменяет имя одной из ролей приложения подсистемы безопасности
<a href="#">CreateRole</a>	Добавляет новую роль для приложения подсистемы безопасности
<a href="#">DeleteToken</a>	Удаляет право приложения подсистемы безопасности
<a href="#">ChangeToken</a>	Изменяет значение одного из прав приложения подсистемы безопасности
<a href="#">CreateToken</a>	Добавляет новое право для приложения подсистемы безопасности
<a href="#">New</a>	Создает новое приложение подсистемы безопасности
<a href="#">GetRoleRights</a>	Возвращает список прав одной из роли приложения подсистемы безопасности
<a href="#">GetRolesList</a>	Возвращает список ролей, добавленных в приложение подсистемы безопасности
<a href="#">GetTokensList</a>	Возвращает список прав, настроенных для приложения
<a href="#">Save</a>	Сохраняет данные приложения подсистемы безопасности

<a href="#">Load</a>	Загружает данные приложения подсистемы безопасности
----------------------	---

## 1.1.4.2.4.7.7.2.1. GetErrorDescriptionByCode

Возвращает текстовое описание ошибок, возникающих при загрузке и сохранении приложений.



```
string GetErrorDescriptionByCode(uint1 FailReasonCode)
```

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Код ошибки из любого события, говорящего об ошибке

## 1.1.4.2.4.7.7.2.2. RoleDeleteRight

Удаляет значение права приложения подсистемы безопасности.



```
void RoleDeleteRight(string AppName, string RoleID, string RightName)
```

### Параметры

Параметр	Тип	Описание
AppName	string	Имя приложения
RoleID	string	Уникальный идентификатор (uid) роли
RightName	string	Имя права

### Примеры



```
SecurityManagerApplication.RoleDeleteRight("appname", "alpha:4ce9483e-  
e137-41c2-bf1c-343e9b731f20", "pravo")
```

## 1.1.4.2.4.7.7.2.3. RoleChangeRight

Изменяет значение права приложения подсистемы безопасности.



```
void RoleChangeRight(string AppName, string RoleID, string RightName,  
string AllowedValue, string ForbidenValue)
```

### Параметры

Параметр	Тип	Описание
AppName	string	Имя приложения
RoleID	string	Уникальный идентификатор (uid) роли
RightName	string	Имя права
AllowedValue	string	Новое разрешающее значение права
ForbidenValue	string	Новое запрещающее значение права

### Примеры



```
SecurityManagerApplication.RoleChangeRight("appname", "alpha:4ce9483e-  
e137-41c2-bf1c-343e9b731f20", "pravo", "закрyто", "открыто")
```



## 1.1.4.2.4.7.7.2.4. RoleAddRight

Предоставляет для указанной роли право приложения, загруженного в элемент.



```
void RoleAddRight(string AppName, string RoleID, string RightName, uint1 RightType, string AllowedValue, string ForbidenValue)
```

### Параметры

Параметр	Тип	Описание
AppName	string	Имя приложения
RoleID	string	Уникальный идентификатор (uid) роли
RightName	string	Имя права
RightType	uint1	Тип права. Значение: <ul style="list-style-type: none"><li>&gt; 0 – тип string</li><li>&gt; 1 – тип bool</li></ul>
AllowedValue	string	Разрешающее значение права
ForbidenValue	string	Запрещающее значение права

### Примеры



```
SecurityManagerApplication.RoleAddRight("appname", "alpha:4ce9483e-e137-41c2-bf1c-343e9b731f20", "pravo", 0, "открыто", "закрыто")
```

## 1.1.4.2.4.7.7.2.5. DeleteRole

Удаляет одну из ролей приложения подсистемы безопасности.



```
void DeleteRole(string RoleID)
```

### Параметры

Параметр	Тип	Описание
RoleID	string	Уникальный идентификатор (uid) роли

### Примеры



```
Вызов: SecurityManagerApplication.DeleteRole("alpha:4ce9483e-  
e137-41c2-bf1c-343e9b731f20")
```

## 1.1.4.2.4.7.7.2.6. ChangeRole

Переименовывает роль из приложения, загруженного в элемент.



```
void ChangeRole(string RoleID, string RoleName)
```

### Параметры

Параметр	Тип	Описание
RoleID	string	Уникальный идентификатор (uid) роли
RoleName	string	Новое имя роли

### Примеры



```
Вызов: SecurityManagerApplication.ChangeRole("alpha:4ce9483e-  
e137-41c2-bf1c-343e9b731f20","newrolename")
```

## 1.1.4.2.4.7.7.2.7. CreateRole

Создает роль в указанном приложении.



```
void CreateRole(string RoleName, string AppName)
```

### Параметры

Параметр	Тип	Описание
RoleName	string	Имя роли
AppName	string	Имя приложения, в котором создается роль

### Примеры



```
Вызов: SecurityManagerApplication.CreateRole("role","app")
```

## 1.1.4.2.4.7.7.2.8. DeleteToken

Удаляет токен из приложения, загруженного в элемент.



```
void DeleteToken(string TokenID)
```

### Параметры

Параметр	Тип	Описание
TokenID	string	ID токена

### Примеры



```
Вызов: SecurityManagerApplication.DeleteToken("right")
```

## 1.1.4.2.4.7.7.2.9. ChangeToken

Меняет тип, имя и описание токена в приложении, загруженном в элемент.



```
void ChangeToken(string TokenID, uint1 TokenType, string TokenName, string Description)
```

### Параметры

Параметр	Тип	Описание
TokenID	string	ID токена
TokenName	string	Новое имя токена
TokenType	uint1	Тип токена. Значение: <ul style="list-style-type: none"><li>&gt; 0 – тип bool</li><li>&gt; 1 – тип string</li></ul>
Description	string	Новое описание токена



Входной параметр тип токена не будет требоваться в будущих версиях.

### Примеры



```
Вызов: SecurityManagerApplication.ChangeToken("tokenID",1,"newname","newname")
```

## 1.1.4.2.4.7.7.2.10. CreateToken

Создает токен в приложении, загруженном в элемент.



```
void CreateToken(uint1 TokenType, string TokenName, string Description)
```

### Параметры

Параметр	Тип	Описание
TokenType	uint1	Тип токена. Значение: <ul style="list-style-type: none"><li>&gt; 0 – тип bool</li><li>&gt; 1 – тип string</li></ul>
TokenName	string	Имя токена
Description	string	Описание

### Примеры



```
Вызов: SecurityManagerApplication.CreateToken(1,"right","description")
```

## 1.1.4.2.4.7.7.2.11. New

Освобождает элемент от загруженной в него информации для создания нового приложения.

Функция не требует входных параметров.



void New()



После создания нового приложения не забудьте загрузить его в подсистему безопасности с помощью функции [Save\(\)](#).

## Примеры



Вызов: SecurityManagerApplication.New()



## 1.1.4.2.4.7.7.2.12. GetRoleRights

Предоставляет список прав, назначенных текущей роли.

Возвращает данные в JSON-формате.



string GetRoleRights(stringroleUID)

### Параметры

Параметр	Тип	Описание
roleUID	string	Уникальный идентификатор (uid) роли

## 1.1.4.2.4.7.7.2.13. GetRolesList

Предоставляет список ролей из приложения, загруженного в элемент.

Функция не требует входных параметров..

Возвращает данные в JSON-формате.



string GetRolesList()

### Примеры



Вызов: SecurityManagerApplication.GetRolesList()

## 1.1.4.2.4.7.7.2.14. GetTokensList

Предоставляет список токенов и прав из приложения, загруженного в элемент.

Функция не требует входных параметров.

Возвращает данные в JSON-формате.



string GetTokensList()

### Примеры



Вызов: SecurityManagerApplication.GetTokensList()

## 1.1.4.2.4.7.7.2.15. Save

Отправляет информацию о приложении в подсистему безопасности после:

- › редактирования ранее выгруженного из подсистемы безопасности приложения;
- › создания приложения из проекта Astra.HMI с помощью функции New() и свойств элемента.

Функция не требует входных параметров.



void Save()

### Примеры



Вызов: `SecurityManagerApplication.Save()`

Результат:

- › в случае успешного завершения операции активируется событие [SaveComplete](#);
- › в случае неуспешного завершения операции активируется событие [SaveFailed](#).

## 1.1.4.2.4.7.7.2.16. Load

Загружает данные приложения подсистемы безопасности.



void Load(string applID)

### Параметры

Параметр	Тип	Описание
applID	string	ID приложения

### Примеры



Вызов: SecurityManagerApplication.Load("Управление состоянием оборудования")

Результат:

- › в случае успешного завершения операции активируется событие [LoadComplete](#);
- › в случае неуспешного завершения операции активируется событие [LoadFailed](#).

### 1.1.4.2.4.7.7.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта (поля объекта)
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">IsChanged</a>	Признак наличия изменений в параметрах приложения подсистемы безопасности
<a href="#">ApplicationID</a>	Символьный идентификатор приложения подсистемы безопасности
<a href="#">Имя приложения</a>	Имя приложения подсистемы безопасности
<a href="#">Менеджер конфигурирования безопасности</a>	Ссылка на компонент Настройка безопасности: Менеджер

## 1.1.4.2.4.7.7.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.7.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.7.7.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.7.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.7.7.3.5. Менеджер конфигурирования безопасности

Ссылка на элемент типа Настройка безопасности: Менеджер, обеспечивающий работу с учетными записями, группами пользователей и приложениями.



Содержит свойство Контекст безопасности для указания ссылки на элемент типа Контекст безопасности.

Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.7.3.6. Имя приложения

Имя приложения, загруженного в элемент.



string ApplicationName

### Примеры



Вызов: SecurityManagerApplication.ApplicationName.

## 1.1.4.2.4.7.7.3.7. IsChanged

Уведомление об изменении в приложении, загруженном в элемент.



bool IsChanged



Доступно только для чтения в режиме рантайма.



Принимает значение false при:

- › загрузке приложения в элемент;
- › сохранении приложения в подсистему безопасности.

### Примеры



Вызов: `SecurityManagerApplication.IsChanged`.

Значение:

- › true – приложение было изменено;
- › false – нет изменений в приложении.

## 1.1.4.2.4.7.7.3.8. ApplicationID

Идентификатор (ID) приложения, загруженного в элемент.

Совпадает с именем приложения.



string ApplicationID



Доступно только для чтения в режиме рантайма.

### Примеры



Вызов: `SecurityManagerApplication.ApplicationID`.

## 1.1.4.2.4.7.8. Настройка безопасности: Пользователь

Компонент предназначен для загрузки информации об учетной записи:

- имеющейся в подсистеме безопасности – для просмотра и изменения этой информации;
- созданной из проекта Astra.HMI – для дальнейшей отправки в подсистему безопасности.

## 1.1.4.2.4.7.8.1. События

Событие	Описание
<a href="#">SaveFailed</a>	Появление ошибки при сохранении приложения
<a href="#">SaveComplete</a>	Завершение сохранения приложения
<a href="#">LoadFailed</a>	Появление ошибки при загрузке приложения
<a href="#">LoadComplete</a>	Завершение выполнения загрузки приложения



## 1.1.4.2.4.7.8.1.1. SaveFailed

Информация об учетной записи не загружена в элемент.

Активируется в случае неуспешного завершения операции [Load\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Номер ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.8.1.2. SaveComplete

Информация об учетной записи сохранена в подсистему безопасности Astra.Security.

Активируется в случае успешного завершения операции [Save\(\)](#).

## 1.1.4.2.4.7.8.1.3. LoadFailed

Информация об учетной записи не загружена в элемент.

Активируется в случае неуспешного завершения операции [Load\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Номер ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.8.1.4. LoadComplete

Информация об учетной записи загружена в элемент.

Активируется в случае успешного завершения операции [Load\(\)](#).

## 1.1.4.2.4.7.8.2. Функции

Компонент	Описание
<a href="#">GetErrorDescriptionByCode</a>	Возвращает описание ошибки по коду
<a href="#">DeleteRight</a>	Удаляет значение права пользователя подсистемы безопасности
<a href="#">ChangeRight</a>	Изменяет значение права пользователя подсистемы безопасности
<a href="#">AddRight</a>	Добавляет новое право пользователю подсистемы безопасности
<a href="#">DeleteGroup</a>	Удаляет пользователя подсистемы безопасности из группы
<a href="#">AddGroup</a>	Добавляет пользователя подсистемы безопасности в группу
<a href="#">DeleteRole</a>	Удаляет одну из ролей пользователя подсистемы безопасности
<a href="#">AddRole</a>	Добавляет новую роль для пользователя подсистемы безопасности
<a href="#">New</a>	Создает нового пользователя подсистемы безопасности
<a href="#">GetEffectiveRights</a>	Возвращает список эффективных прав пользователя подсистемы безопасности
<a href="#">GetRights</a>	Возвращает список прав пользователя подсистемы безопасности
<a href="#">GetRoles</a>	Возвращает список ролей, назначенных пользователю подсистемы безопасности
<a href="#">GetApplicationsList</a>	Возвращает список приложений, на которые у пользователя подсистемы безопасности есть права
<a href="#">GetGroupsList</a>	Возвращает список групп, в которых состоит пользователь подсистемы безопасности

<a href="#">ValidatePassword</a>	Проверяет пароль на соответствие назначенным парольным политикам
<a href="#">SetPassword</a>	Задаёт пароль пользователя подсистемы безопасности
<a href="#">Save</a>	Сохраняет данные пользователя подсистемы безопасности
<a href="#">Load</a>	Загружает данные пользователя подсистемы безопасности

## 1.1.4.2.4.7.8.2.1. GetErrorDescriptionByCode

Возвращает текстовое описание ошибок, возникающих при загрузке и сохранении пользователей.



```
string GetErrorDescriptionByCode(uint1 FailReasonCode)
```

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Код ошибки из любого события, говорящего об ошибке

## 1.1.4.2.4.7.8.2.2. DeleteRight

Лишает права пользователя, чья учетная запись загружена в элемент.



```
void DeleteRight(string AppID, string RightName)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, содержащего право
RightName	string	Имя права

### Примеры



```
Вызов: SecurityManagerUser.DeleteRight("Управление состоянием  
оборудования", "Управление насосом")
```



## 1.1.4.2.4.7.8.2.3. ChangeRight

Меняет значения прав пользователя, чья учетная запись загружена в элемент.



```
void ChangeRight(string AppID, string RightName, string AllowedValue,  
string ForbidenValue)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, содержащего право
RightName	string	Имя права
AllowedValue	string	Новое разрешающее значение права
ForbidenValue	string	Новое запрещающее значение права

### Примеры



```
Вызов: SecurityManagerUser.ChangeRight("Управление состоянием  
оборудования", "Управление насосом", "false", "true")
```

## 1.1.4.2.4.7.8.2.4. AddRight

Назначает право пользователю, чья учетная запись загружена в элемент.



```
void AddRight(string AppID, string RightName, uint1 RightType, string AllowedValue, string ForbidenValue)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, содержащего право
RightName	string	Имя права
RightType	uint1	Тип права. Значение: <ul style="list-style-type: none"><li>&gt; 0 – тип string</li><li>&gt; 1 – тип bool</li></ul>
AllowedValue	string	Разрешающее значение права
ForbidenValue	string	Запрещающее значение права

### Примеры



```
Вызов: SecurityManagerUser.AddRight("Управление состоянием оборудования", "Управление насосом", 1, "true", "false")
```

## 1.1.4.2.4.7.8.2.5. DeleteGroup

Удаляет пользователя подсистемы безопасности из группы.



```
void DeleteGroup(string GroupUID)
```

### Параметры

Параметр	Тип	Описание
GroupUID	string	Уникальный идентификатор (uid) группы

### Примеры



```
Вызов: SecurityManagerUser.DeleteGroup("alpha:ffdad1f9-ae36-4980-82b8-c7180a63b451")
```

## 1.1.4.2.4.7.8.2.6. AddGroup

Добавляет в указанную группу пользователя, чья учетная запись загружена в элемент.



```
void AddGroup(string GroupUID)
```

### Параметры

Параметр	Тип	Описание
GroupUID	string	Уникальный идентификатор (uid) группы

### Примеры



```
Вызов: SecurityManagerUser.AddGroup("alpha:ffdad1f9-ae36-4980-82b8-c7180a63b451")
```

## 1.1.4.2.4.7.8.2.7. DeleteRole

Лишает указанной роли пользователя, чья учетная запись загружена в элемент.



DeleteRole(string RoleID)

### Параметры

Параметр	Тип	Описание
RoleID	string	Уникальный идентификатор (uid) роли

### Примеры



Вызов: `SecurityManagerUser.DeleteRole("alpha:4ce9483e-e137-41c2-bf1c-343e9b731f20")`

## 1.1.4.2.4.7.8.2.8. AddRole

Назначает указанную роль пользователю, чья учетная запись загружена в элемент.



```
void AddRole(string RoleID)
```

### Параметры

Параметр	Тип	Описание
RoleID	string	Уникальный идентификатор (uid) роли

### Примеры



```
Вызов: SecurityManagerUser.AddRole("alpha:4ce9483e-e137-41c2-bf1c-343e9b731f20")
```

## 1.1.4.2.4.7.8.2.9. New

Освобождает элемент от загруженной в него информации для создания нового приложения.

Функция не требует входных параметров.



void New()



После создания новой учетной записи не забудьте загрузить её в подсистему безопасности с помощью функции [Save\(\)](#).

## Примеры



Вызов: SecurityManagerUser.New()

## 1.1.4.2.4.7.8.2.10. GetEffectiveRights

Возвращает список эффективных прав пользователя подсистемы безопасности.



```
string GetEffectiveRights(string AppID)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, из которого требуется получить список эффективных прав



## 1.1.4.2.4.7.8.2.11. GetRights

Предоставляет список личных прав пользователя, чья учетная запись загружена в элемент, из указанного приложения.

Возвращает данные в JSON-формате.



string GetRights(string AppID)

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, из которого требуется получить список прав

### Примеры



Вызов: SecurityManagerUser.GetRights("Управление состоянием оборудования")

## 1.1.4.2.4.7.8.2.12. GetRoles

Предоставляет список ролей пользователя, чья учетная запись загружена в элемент, в указанном приложении.

Возвращает данные в JSON-формате.



string GetRoles(string AppID)

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, из которого требуется получить список ролей

### Примеры



Вызов: SecurityManagerUser.GetRoles("Управление состоянием оборудования")

## 1.1.4.2.4.7.8.2.13. GetApplicationsList

Предоставляет список приложений с правами пользователя, чья учетная запись загружена в элемент.

Функция не требует входных параметров.

Возвращает данные в JSON-формате.



string GetApplicationsList()

### Примеры



Вызов: SecurityManagerUser.GetApplicationsList()

## 1.1.4.2.4.7.8.2.14. GetGroupsList

Предоставляет список групп, в которых состоит пользователь учетной записи, загруженной в элемент.

Функция не требует входных параметров.

Возвращает данные в JSON-формате.



string GetGroupsList()

### Примеры



Вызов: SecurityManagerUser.GetGroupsList()

## 1.1.4.2.4.7.8.2.15. ValidatePassword

Проверить пароль на соответствие назначенным парольным политикам.



uint2 ValidatePassword(string\_1)

### Параметры

Параметр	Тип	Описание
string_1	string	Пароль

## 1.1.4.2.4.7.8.2.16. SetPassword

Устанавливает пароль для учетной записи, загруженной в элемент.



uint2 SetPassword(string Password)

### Параметры

Параметр	Тип	Описание
Password	string	Пароль



Подсистема безопасности Astra.Security не накладывает требований к содержанию пароля.

### Примеры



Вызов: `SecurityManagerUser.SetPassword("password")`

Результат:

- › в случае успешного завершения операции активируется событие [SaveComplete](#);
- › в случае неуспешного завершения операции активируется событие [SaveFailed](#).

## 1.1.4.2.4.7.8.2.17. Save

Отправляет информацию о приложении в подсистему безопасности после:

- › редактирования ранее выгруженного из подсистемы безопасности приложения;
- › создания приложения из проекта Astra.HMI с помощью функции New() и свойств элемента.

Функция не требует входных параметров.



void Save()

### Примеры



Вызов: SecurityManagerUser.Save()

Результат:

- › в случае успешного завершения операции активируется событие [SaveComplete](#);
- › в случае неуспешного завершения операции активируется событие [SaveFailed](#).

## 1.1.4.2.4.7.8.2.18. Load

Загружает информацию об учетной записи из подсистемы безопасности в элемент.



```
void Load(string UserID)
```

### Параметры

Параметр	Тип	Описание
UserID	string	уникальный идентификатор (uid) учетной записи

### Примеры



Вызов: `SecurityManagerUser.Load("alpha:ab5cc552-a515-48ab-9e82-436aabf6bdfc")`

Результат:

- в случае успешного завершения операции активируется событие [LoadComplete](#);
- в случае неуспешного завершения операции активируется событие [LoadFailed](#).



## 1.1.4.2.4.7.8.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Менеджер конфигурирования безопасности</a>	Ссылка на компонент Настройка безопасности: Менеджер
<a href="#">IsChanged</a>	Признак наличия изменений в параметрах учетной записи пользователя подсистемы безопасности
<a href="#">UserID</a>	Символьный идентификатор пользователя подсистемы безопасности
<a href="#">Смена пароля при следующем входе</a>	Требование смены пароля при следующем входе пользователя в подсистему безопасности
<a href="#">Комментарий</a>	Комментарий к учетной записи пользователя подсистемы безопасности
<a href="#">Номер телефона</a>	Номер телефона пользователя подсистемы безопасности
<a href="#">Адрес электронной почты</a>	Электронная почта пользователя подсистемы безопасности
<a href="#">Подразделение</a>	Подразделение пользователя подсистемы безопасности
<a href="#">Должность</a>	Должность пользователя подсистемы безопасности
<a href="#">Отображаемое имя</a>	Отображаемое имя пользователя подсистемы безопасности
<a href="#">Фамилия</a>	Фамилия пользователя подсистемы безопасности
<a href="#">Логин пользователя</a>	Логин пользователя подсистемы безопасности
<a href="#">Имя</a>	Имя пользователя подсистемы безопасности

<a href="#">Пользователь заблокирован</a>	Пользователь заблокирован
<a href="#">Отчество</a>	Отчество пользователя

## 1.1.4.2.4.7.8.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.8.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.8.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.8.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.7.8.3.5. Менеджер конфигурирования безопасности

Ссылка на элемент типа Настройка безопасности: Менеджер, обеспечивающий работу с учетными записями, группами пользователей и приложениями.



Содержит свойство Контекст безопасности для указания ссылки на элемент типа Контекст безопасности.

Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.8.3.6. Логин пользователя

Логин пользователя подсистемы безопасности.



string Login



## 1.1.4.2.4.7.8.3.7. Имя

Имя пользователя подсистемы безопасности.



string Name

## 1.1.4.2.4.7.8.3.8. Фамилия

Фамилия пользователя подсистемы безопасности.



string Surname

## 1.1.4.2.4.7.8.3.9. Отчество

Отчество пользователя.



string MidName

## 1.1.4.2.4.7.8.3.10. Отображаемое имя

Отображаемое имя пользователя подсистемы безопасности.



string DisplayName

## 1.1.4.2.4.7.8.3.11. Должность

Должность пользователя подсистемы безопасности.



string Position

## 1.1.4.2.4.7.8.3.12. Подразделение

Подразделение пользователя подсистемы безопасности.



string Unit

## 1.1.4.2.4.7.8.3.13. Адрес электронной почты

Электронная почта пользователя подсистемы безопасности.



string Email

## 1.1.4.2.4.7.8.3.14. Номер телефона

Номер телефона пользователя подсистемы безопасности.



string Phone



## 1.1.4.2.4.7.8.3.15. Комментарий

Комментарий к учетной записи пользователя подсистемы безопасности.



string Comment

## 1.1.4.2.4.7.8.3.16. Смена пароля при следующем входе

Требование смены пароля при следующем входе пользователя в подсистему безопасности.



bool ForcePassChange

### Примеры



Вызов: SecurityManagerUser.ForcePassChange

Значение:

- › true – требовать смену пароля;
- › false – не требовать смену пароля.

## 1.1.4.2.4.7.8.3.17. Пользователь заблокирован

Хранит информацию о том, заблокирован ли пользователь.



bool Disabled

### Примеры



Вызов: `SecurityManagerUser.Disabled`

Значение:

- › true – пользователь заблокирован;
- › false – пользователь не заблокирован.

## 1.1.4.2.4.7.8.3.18. IsChanged

Уведомление об изменении в учетной записи, загруженной в элемент.



bool IsChanged



Только для чтения в режиме рантайма.



Принимает значение false при:

- › загрузке учетной записи в элемент;
- › сохранении учетной записи в подсистему безопасности.

### Примеры



Вызов: SecurityManagerUser.IsChanged

Значение:

- › true – учетная запись была изменена;
- › false – нет изменений в учетной записи.

## 1.1.4.2.4.7.8.3.19. UserID

Уникальный идентификатор (uid) учетной записи, загруженной в элемент.



string UserID



Только для чтения в режиме рантайма.

### Примеры



Вызов: SecurityManagerUser.UserID

Вид значения: alpha:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Пример значения: alpha:ab5cc552-a515-48ab-9e82-436aabf6bdfc.

## 1.1.4.2.4.7.9. Настройка безопасности: Группа

Компонент предназначен для загрузки информации о группе:

- имеющейся в подсистеме безопасности – для просмотра и изменения этой информации;
- созданной из проекта Astra.HMI – для дальнейшей отправки в подсистему безопасности.

## 1.1.4.2.4.7.9.1. События

Событие	Описание
<a href="#">SaveFailed</a>	Появление ошибки при сохранении приложения
<a href="#">SaveComplete</a>	Завершение сохранения приложения
<a href="#">LoadFailed</a>	Появление ошибки при загрузке приложения
<a href="#">LoadComplete</a>	Завершение выполнения загрузки приложения

## 1.1.4.2.4.7.9.1.1. SaveFailed

Информация о группе не сохранена в подсистему безопасности Astra.Security.

Активируется в случае неуспешного завершения операции [Save\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Номер ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).



## 1.1.4.2.4.7.9.1.2. SaveComplete

Информация о группе сохранена в подсистему безопасности Astra.Security.

Активируется в случае успешного завершения операции [Save\(\)](#).

## 1.1.4.2.4.7.9.1.3. LoadFailed

Информация о группе не загружена в элемент.

Активируется в случае неуспешного завершения операции [Load\(\)](#).

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Номер ошибки



Получить текст ошибки можно, вызвав функцию [GetErrorDescriptionByCode\(\)](#).

## 1.1.4.2.4.7.9.1.4. LoadComplete

Информация о группе загружена в элемент.

Активируется в случае успешного завершения операции [Load\(\)](#).

## 1.1.4.2.4.7.9.2. Функции

Компонент	Описание
<a href="#">GetErrorDescriptionByCode</a>	Возвращает описание ошибки по коду
<a href="#">DeleteRight</a>	Удаляет значение права пользователя подсистемы безопасности
<a href="#">ChangeRight</a>	Изменяет значение права пользователя подсистемы безопасности
<a href="#">AddRight</a>	Добавляет новое право пользователю подсистемы безопасности
<a href="#">DeleteRole</a>	Удаляет одну из ролей пользователя подсистемы безопасности
<a href="#">AddRole</a>	Добавляет новую роль для пользователя подсистемы безопасности
<a href="#">New</a>	Создает нового пользователя подсистемы безопасности
<a href="#">GetApplicationsList</a>	Возвращает список приложений, на которые у пользователя подсистемы безопасности есть права
<a href="#">Save</a>	Сохраняет данные пользователя подсистемы безопасности
<a href="#">Load</a>	Загружает данные пользователя подсистемы безопасности
<a href="#">GroupEffectiveRights</a>	Возвращает список эффективных прав группы подсистемы безопасности
<a href="#">GroupRights</a>	Возвращает список прав группы подсистемы безопасности
<a href="#">GroupRoles</a>	Возвращает список ролей, назначенных группе подсистемы безопасности
<a href="#">GetMembersList</a>	Возвращает список пользователей, состоящих в группе



## 1.1.4.2.4.7.9.2.1. GetErrorDescriptionByCode

Возвращает текстовое описание ошибок, возникающих при загрузке и сохранении групп.



```
string GetErrorDescriptionByCode(uint1 FailReasonCode)
```

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Код ошибки из любого события, говорящего об ошибке

## 1.1.4.2.4.7.9.2.2. DeleteRight

Лишает права группу, загруженную в элемент.



```
void DeleteRight(string AppID, string RightName)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, содержащего право
RightName	string	Имя права

### Примеры



```
Вызов: SecurityManagerGroup.DeleteRight("Управление состоянием  
оборудования", "Управление насосом")
```

## 1.1.4.2.4.7.9.2.3. ChangeRight

Меняет значения прав группы, загруженной в элемент.



```
void ChangeRight(string AppID, string RightName, string AllowedValue,  
string ForbidenValue)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, содержащего право
RightName	string	Имя права
AllowedValue	string	Новое разрешающее значение права
ForbidenValue	string	Новое запрещающее значение права

### Примеры



```
Вызов: SecurityManagerGroup.ChangeRight("Управление состоянием  
оборудования", "Управление насосом", "false", "true")
```



## 1.1.4.2.4.7.9.2.4. AddRight

Назначает право группе, загруженной в элемент.



```
void AddRight(string AppID, string RightName, uint1 RightType, string AllowedValue, string ForbidenValue)
```

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, содержащего право
RightName	string	Имя права
RightType	int	Тип права. Значение: <ul style="list-style-type: none"><li>➤ 0 – тип string</li><li>➤ 1 – тип bool</li></ul>
AllowedValue	string	Разрешающее значение права
ForbidenValue	string	Запрещающее значение права

### Примеры



```
Вызов: SecurityManagerGroup.AddRight("Управление состоянием оборудования", "Управление насосом", 1, "true", "false")
```

## 1.1.4.2.4.7.9.2.5. DeleteRole

Лишает указанной роли группу, загруженную в элемент.



```
void DeleteRole(string RoleID)
```

### Параметры

Параметр	Тип	Описание
RoleID	string	Уникальный идентификатор (uid) роли

### Примеры



```
Вызов: SecurityManagerGroup.DeleteRole("alpha:4ce9483e-e137-41c2-bf1c-343e9b731f20")
```

## 1.1.4.2.4.7.9.2.6. AddRole

Назначает указанную роль группе, загруженной в элемент.



```
void AddRole(string RoleUID)
```

### Параметры

Параметр	Тип	Описание
RoleUID	string	Уникальный идентификатор (uid) роли

### Примеры



```
Вызов: SecurityManagerGroup.AddRole("alpha:4ce9483e-e137-41c2-bf1c-343e9b731f20")
```

## 1.1.4.2.4.7.9.2.7. New

Освобождает элемент от загруженной в него информации для создания новой группы.

Функция не требует входных параметров.



void New()



После создания нового приложения не забудьте загрузить его в подсистему безопасности с помощью функции [Save\(\)](#).

## Примеры



Вызов: SecurityManagerGroup.New()

## 1.1.4.2.4.7.9.2.8. GetApplicationsList

Предоставляет список приложений с правами группы, загруженной в элемент.

Функция не требует входных параметров.

Возвращает данные в JSON-формате.



string GetApplicationsList()

### Примеры



Вызов: SecurityManagerGroup.GetApplicationsList()

## 1.1.4.2.4.7.9.2.9. Save

Отправляет информацию о приложении в подсистему безопасности после:

- › редактирования ранее выгруженного из подсистемы безопасности приложения;
- › создания приложения из проекта Astra.HMI с помощью функции [New\(\)](#) и свойств элемента.

Функция не требует входных параметров.



void Save()

### Примеры



Вызов: `SecurityManagerGroup.Save()`

Результат:

- › в случае успешного завершения операции активируется событие [SaveComplete](#);
- › в случае неуспешного завершения операции активируется событие [SaveFailed](#).

## 1.1.4.2.4.7.9.2.10. Load

Загружает информацию о группе из подсистемы безопасности в элемент.



void Load(string AppID)

### Параметры

Параметр	Тип	Описание
AppID	string	Уникальный идентификатор (uid) группы

### Примеры



Вызов: `SecurityManagerGroup.Load("alpha:d01969e0-ae05-4548-9375-2533b326a588")`

Результат:

- › в случае успешного завершения операции активируется событие [LoadComplete](#);
- › в случае неуспешного завершения операции активируется событие [LoadFailed](#).

## 1.1.4.2.4.7.9.2.11. GroupEffectiveRights

Предоставляет список эффективных прав группы, загруженной в элемент, из указанного приложения.

Возвращает данные в JSON-формате.



string GroupEffectiveRights(string AppID)

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, из которого требуется получить список эффективных прав

### Примеры



Вызов: SecurityManagerGroup.GroupEffectiveRights("Управление состоянием оборудования")



## 1.1.4.2.4.7.9.2.12. GroupRights

Предоставляет список прав группы, загруженной в элемент, из указанного приложения.

Возвращает данные в JSON-формате.



string GroupRights(string AppID)

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, из которого требуется получить список прав



Для получения списка эффективных прав используйте функцию [GroupEffectiveRights\(\)](#).

### Примеры



Вызов: SecurityManagerGroup.GroupRights("Управление состоянием оборудования")

## 1.1.4.2.4.7.9.2.13. GroupRoles

Предоставляет список ролей группы, загруженной в элемент, в указанном приложении.

Возвращает данные в JSON-формате.



string GroupRoles(string AppID)

### Параметры

Параметр	Тип	Описание
AppID	string	ID приложения, из которого требуется получить список ролей

### Примеры



Вызов: SecurityManagerGroup.GroupRoles("Управление состоянием оборудования")

## 1.1.4.2.4.7.9.2.14. GetMembersList

Предоставляет список участников группы, загруженной в элемент.

Функция не требует входных параметров.

Возвращает данные в JSON-формате.



string GetMembersList()

### Примеры



Вызов: SecurityManagerGroup.GetMembersList()

## 1.1.4.2.4.7.9.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Менеджер конфигурирования безопасности</a>	Ссылка на компонент Настройка безопасности: Менеджер
<a href="#">IsChanged</a>	Признак наличия изменений в параметрах группы подсистемы безопасности
<a href="#">GroupID</a>	Символьный идентификатор группы подсистемы безопасности
<a href="#">Описание группы</a>	Описание группы подсистемы безопасности
<a href="#">Имя группы</a>	Имя группы подсистемы безопасности
<a href="#">Группа заблокирована</a>	Пользователь заблокирован

## 1.1.4.2.4.7.9.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.9.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.9.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.9.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.7.9.3.5. Менеджер конфигурирования безопасности

Ссылка на элемент типа Настройка безопасности: Менеджер, обеспечивающий работу с учетными записями, группами пользователей и приложениями.



Содержит свойство Контекст безопасности для указания ссылки на элемент типа Контекст безопасности.

Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.9.3.6. Имя группы

Название группы, загруженной в элемент.



string GroupName

### Примеры



Вызов: SecurityManagerGroup.GroupName

Пример значения: dispatchers.

## 1.1.4.2.4.7.9.3.7. Описание группы

Описание группы, загруженной в элемент.



string GroupDescription

### Примеры



Вызов: SecurityManagerGroup.GroupDescription

## 1.1.4.2.4.7.9.3.8. Группа заблокирована

Хранит информацию о том, заблокирована ли группа.



bool Disabled

### Примеры



Вызов: `SecurityManagerGroup.Disabled`

Значение:

- › true – группа заблокирована;
- › false – группа не заблокирована.

## 1.1.4.2.4.7.9.3.9. IsChanged

Уведомление об изменении в группе, загруженной в элемент.



bool IsChanged



Доступно только для чтения в режиме рантайма.



Принимает значение false при:

- › загрузке группы в элемент;
- › сохранении группы в подсистему безопасности.

### Примеры



Вызов: SecurityManagerGroup.IsChanged

Значение:

- › true – группа была изменена;
- › false – нет изменений в группе.

## 1.1.4.2.4.7.9.3.10. GroupID

Уникальный идентификатор (uid) группы, загруженной в элемент.



string GroupID



Доступно только для чтения в режиме рантайма.



Уникальный идентификатор формируется подсистемой безопасности в момент создания группы.

### Примеры



Вызов: SecurityManagerGroup.GroupID

Вид значения: alpha:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Пример значения: alpha:d01969e0-ae05-4548-9375-2533b326a588.

## 1.1.4.2.4.7.10. Мастер конфигурирования Security

Компонент предназначен для чтения и обновления конфигурации службы Агент Astra.Security.

После установки подсистемы безопасности Astra.Security необходимо настроить ее компоненты – в том числе, службу Агент Astra.Security. Вносить изменения в конфигурационный файл можно вручную, либо из проекта Astra.NMI с помощью свойств, функций и событий экземпляра Мастер конфигурирования Security.

Компонент следует использовать следующим образом:

- Для чтения конфигурации необходимо вызвать функцию [Read\(\)](#). Она поместит данные из конфигурационного файла в компонент. После этого можно будет обратиться к данным, прочитав значения свойств компонента или вызвав функции, название которых начинается с Get ([GetLdapHost\(\)](#), [GetLdapPort\(\)](#) и пр.).
- Для создания новой конфигурации необходимо сначала указать новые данные, поместив их в значения свойств и во входные параметры функций, название которых начинается с Add ([AddLdap\(\)](#), [AddLogConsumer\(\)](#) и пр.). После этого следует вызвать функцию [Generate\(\)](#), которая создаст текст нового конфигурационного файла.

## 1.1.4.2.4.7.10.1. События

Событие	Описание
<a href="#">ConsumersListChanged</a>	Изменение списка потребителей аудита
<a href="#">LdapListChanged</a>	Изменение списка LDAP-серверов
<a href="#">ReadingFailure</a>	Появление ошибки считывания конфигурации Astra.Security.Agent
<a href="#">ReadingFinished</a>	Завершение считывания конфигурации Astra.Security.Agent
<a href="#">ReadingStarted</a>	Начало считывания конфигурации Astra.Security.Agent
<a href="#">GenerationFailure</a>	Появление ошибки генерации конфигурации Astra.Security.Agent
<a href="#">GenerationFinished</a>	Завершение генерации конфигурации Astra.Security.Agent
<a href="#">GenerationStarted</a>	Начало генерации конфигурации Astra.Security.Agent



## 1.1.4.2.4.7.10.1.1. ConsumersListChanged

Изменен список серверов-потребителей сообщений аудита во внутреннем массиве компонента.

Активируется в результате успешного выполнения одной из функций – [AddLogConsumer\(\)](#) или [ClearLogConsumersList\(\)](#).

## 1.1.4.2.4.7.10.1.2. LdapListChanged

Изменен список LDAP-серверов во внутреннем массиве компонента.

Активируется в результате успешного выполнения одной из функций – [AddLdap\(\)](#) или [ClearLdapList\(\)](#).

## 1.1.4.2.4.7.10.1.3. ReadingFailure

Не удалось прочитать конфигурационный файл Агент Astra.Security.

Активируется в результате неуспешного выполнения функции [Read\(\)](#).

Чтобы ознакомиться с текстом ошибки, обратитесь к значению внутренней переменной Error этого события, либо к значению свойства [Ошибка чтения](#) (ReadError) компонента.

## 1.1.4.2.4.7.10.1.4. ReadingFinished

Чтение конфигурационного файла Агент Astra.Security завершено успешно.

Активируется в результате успешного выполнения функции [Read\(\)](#).

Чтобы ознакомиться с полученными настройками, обратитесь к значениям свойств компонента.

## 1.1.4.2.4.7.10.1.5. ReadingStarted

Начато чтение конфигурационного файла Агент Astra.Security.

Активируется сразу после вызова функции [Read\(\)](#).

## 1.1.4.2.4.7.10.1.6. GenerationFailure

Не удалось создать текст конфигурационного файла Агент Astra.Security.

Активируется в результате неуспешного выполнения функции [Generate\(\)](#).

Чтобы ознакомиться с текстом ошибки, обратитесь к значению внутренней переменной `Error` этого события, либо к значению свойства [Ошибка конфигурирования](#) компонента.

## 1.1.4.2.4.7.10.1.7. GenerationFinished

Создан текст конфигурационного файла Агент Astra.Security. Результат помещен в свойство [GeneratedString](#).

Активируется в результате успешного выполнения функции [Generate\(\)](#).

## 1.1.4.2.4.7.10.1.8. GenerationStarted

Начато создание текста конфигурационного файла Агент Astra.Security.

Активируется сразу после вызова функции [Generate\(\)](#).



## 1.1.4.2.4.7.10.2. Функции

Компонент	Описание
<a href="#">AddSignal</a>	Добавляет новый сигнал потребителю аудита
<a href="#">AddSeverity</a>	Добавляет новую категорию важности потребителю аудита
<a href="#">AddLogConsumer</a>	Добавляет нового потребителя аудита в конфигурацию Astra.Security.Agent
<a href="#">ClearLogConsumersList</a>	Очищает список потребителей аудита
<a href="#">GetSignalMode</a>	Возвращает режим сигнала аудита по индексу сервера аудита и индексу сигнала
<a href="#">GetSignalName</a>	Возвращает имя сигнала аудита по индексу сервера аудита и индексу сигнала
<a href="#">GetSeverityValue</a>	Возвращает значение важности аудита по индексу сервера аудита и индексу категории
<a href="#">GetSeverityCategory</a>	Возвращает категорию важности аудита по индексу сервера аудита и индексу категории
<a href="#">GetSignalsCount</a>	Возвращает количество сигналов аудита по индексу сервера аудита
<a href="#">GetSeverityCount</a>	Возвращает количество категорий аудита по индексу сервера аудита
<a href="#">GetServerType</a>	Предоставляет информацию о типе одного из указанных в конфигурационном файле серверов-потребителей аудита
<a href="#">GetServerProgId</a>	Возвращает программный идентификатор (ProgId) сервера аудита по индексу
<a href="#">GetAuditServerPort</a>	Возвращает порт сервера аудита по индексу
<a href="#">GetAuditServerHost</a>	Возвращает IP адрес или имя компьютера с сервером аудита по индексу
<a href="#">GetLdapPort</a>	Возвращает порт LDAP-сервера по индексу

<a href="#">GetLdapHost</a>	Возвращает IP адрес или имя компьютера с LDAP-сервером по индексу
<a href="#">ClearLdapList</a>	Очищает список LDAP-серверов в конфигурации Astra.Security.Agent
<a href="#">AddLdap</a>	Добавляет новый LDAP-сервер в конфигурацию Astra.Security.Agent
<a href="#">Read</a>	Запускает чтение конфигурации Astra.Security.Agent
<a href="#">Generate</a>	Запускает генерацию конфигурации Astra.Security.Agent
<a href="#">GetSignalType</a>	Предоставляет название типа сообщений в указанном сигнале для указанного сервера-потребителя

## 1.1.4.2.4.7.10.2.1. AddSignal

Добавляет в компонент информацию о сигнале, предназначенном для записи сообщений, для указанного сервера-потребителя.



```
bool AddSignal(int4 ConsumerIndex, string SignalName, int4 SignalMode, string SignalType)
```

### Параметры

Параметр	Тип	Описание
ConsumerIndex	int4	Индекс сервера-потребителя во внутреннем массиве компонента
SignalName	string	Название сигнала
SignalMode	int4	Режим записи сообщения в сигнал: <ul style="list-style-type: none"><li>&gt; 1 – соответствует DynamicEvent (запись сообщения с xml-конструкцией, описывающей динамическое событие)</li><li>&gt; 2 – соответствует Value (обычная запись сообщения)</li></ul>
SignalType	string	Тип сообщений, записывающихся в сигнал

Функция возвращает результат добавления информации о сигнале в компонент:

- > true – информация добавлена;
- > false – информацию не удалось добавить. В таком случае следует проверить значение индекса – оно может быть больше, чем размер массива в компоненте.

## 1.1.4.2.4.7.10.2.2. AddSeverity

Добавляет в компонент категорию важности сообщений для указанного сервера-потребителя.



```
bool AddSeverity(int4 ConsumerIndex, string SeverityCategory, int4 SeverityValue)
```

### Параметры

Параметр	Тип	Описание
ConsumerIndex	int4	Индекс сервера-потребителя во внутреннем массиве компонента
SeverityCategory	string	Название категории важности
SeverityValue	int4	Значение категории важности

Функция возвращает результат добавления категории в компонент:

- › true – категория добавлена;
- › false – категорию не удалось добавить. В таком случае следует проверить значение индекса – оно может быть больше, чем размер массива в компоненте.

## 1.1.4.2.4.7.10.2.3. AddLogConsumer

Добавляет описание сервера-потребителя аудита сообщений. Добавленное описание хранится во внутреннем массиве компонента.

В случае успешного завершения операции активируется событие ConsumersListChanged.



```
void AddLogConsumer(string Host, string HostTcpReserve, int4  
TCPServerPort, string ProgID, string Type)
```

### Параметры

Параметр	Тип	Описание
Host	string	Имя или IP-адрес сервера
HostTcpReserve	string	Имя или IP-адрес сервера для резервного канала связи с сервером TCP (значение может быть пустым)
TCPServerPort	int4	Порт для подключения к TCP-серверу (если выбран TCP-сервер)
ProgID	string	Строковый идентификатор OPC-сервера (если выбран OPC-сервер)
Type	string	Тип сервера: <ul style="list-style-type: none"><li>➤ для Windows возможны значения OPC (по умолчанию) и TCP</li><li>➤ для других ОС допустимо только TCP</li></ul>

## 1.1.4.2.4.7.10.2.4. ClearLogConsumersList

Очищает внутренний массив серверов-потребителей в компоненте.

Функция не требует входных параметров.

В случае успешного завершения операции активируется событие [ConsumersListChanged](#).



```
void ClearLogConsumersList()
```

## 1.1.4.2.4.7.10.2.5. GetSignalMode

Возвращает режим сигнала аудита по индексу сервера аудита и индексу сигнала.



int4 GetSignalMode(ServerIndex, SignalIndex)

### Параметры

Параметр	Тип	Описание
ServerIndex	int4	Индекс сервера аудита
SignalIndex	int4	Индекс сигнала

## 1.1.4.2.4.7.10.2.6. GetSignalName

Предоставляет информацию о режиме записи сообщения в указанный сигнал в указанном сервере-потребителе.



```
string GetSignalName(int4 i, int4 k)
```

### Параметры

Параметр	Тип	Описание
i	int4	Индекс сервера аудита
k	int4	Индекс категории важности

Возвращает значение в числовом виде, где:

- › 1 – соответствует DynamicEvent (запись сообщения с xml-конструкцией, описывающей динамическое событие);
- › 2 – соответствует Value (обычная запись сообщения).

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, каждый из них описывается в компоненте в виде элемента массива A[i]. Каждый элемент массива A[i] представляет собой массив B, каждый элемент B[i,k] которого описывает один из сигналов в описываемом сервере A[i].



Массив A, описывающий сервера-потребители = [[Массив B, описывающий первый сервер], [Массив B, описывающий второй сервер]]

Массив B, описывающий первый сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Массив B, описывающий второй сервер = [[Описание первой категории важности], [Описание второй категории важности]]



Обратиться к конкретному сигналу можно по номеру элемента в массивах A и B (i, k).

Каждый элемент массива B[i,k] соответствует одному из значений атрибута Mode xml-элемента <Signal>, вложенного в xml-элемент <SignalMap> одного из xml-элементов <OpcDaLogConsumer> в конфигурационном файле агента безопасности.

Пример использования приведен в описании функции [GetSignalType\(\)](#).

## 1.1.4.2.4.7.10.2.7. GetSeverityValue

Предоставляет числовое значение выбранной категории важности сообщений.



int4 GetSeverityValue(int4 i, int4 k)

### Параметры

Параметр	Тип	Описание
i	int4	Индекс сервера аудита
k	int4	Индекс категории важности

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, каждый из них описывается в компоненте в виде элемента массива A[i]. Каждый элемент массива A[i] представляет собой массив B, каждый элемент B[i,k] которого описывает одну из категорий важности описываемого сервера A[i].



Массив A, описывающий сервера-потребители = [[Массив B, описывающий первый сервер], [Массив B, описывающий второй сервер]]

Массив B, описывающий первый сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Массив B, описывающий второй сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Обратиться к конкретной категории важности можно по номеру элемента в массивах A и B (i, k).

Каждый элемент массива  $B[i,k]$  соответствует одному из значений атрибута Value xml-элемента <Severity>, вложенного в xml-элемент <SeverityMap> одного из xml-элементов <OpcDaLogConsumer> в конфигурационном файле агента безопасности.

## Примеры



Допустим, в конфигурационном файле описан один сервер-потребитель сообщений (один элемент <OpcDaLogConsumer>), для которого описано четыре категории важности сообщений аудита:



```
<AuditLogConsumers TraceAudit="1">
<OpcDaLogConsumer>
<Server Host="127.0.0.1" Type="OPC" ...>
<SeverityMap>
  <Severity Category="Critical"
Value="800"/>
  <Severity
Category="Important"
Value="200"/>
  <Severity
Category="Info"
Value="100"/>
  <Severity
Category="Debug"
Value="0"/>
</SeverityMap>
</Server>
</OpcDaLogConsumer>
</AuditLogConsumers
```

Чтобы получить список категорий важности сообщений с их значениями в каждом сервере-потребителе, вызовите нужные функции в коде, выполняющемся в случае успешного чтения конфигурации Агент Astra.Security (например, в обработчике

события [ReadingFinished](#)). Укажите в качестве входных параметров индексы *i* (индекс в массиве серверов-потребителей) и *k* (индекс в массиве категорий важности). Приведенный ниже пример написан на языке Astra.Оm, в нем итоговый список записывается в лог:



```
i: int4 = 0;
while (i < Configurator.ConsumersCount) //цикл
выполняется, пока в массиве А не будут описаны все
сервера-потребители
{
    k: int4 = 0;
    while (k < Configurator.GetSeverityCount(i)) //цикл
выполняется, пока в массив В не будут записаны все
категории важности i-го сервера-потребителя
    {
        DebugTool.Log("Категория важности:
"+Configurator.GetSeverityCategory(i,k)+"; Значение:
"+String.ToString(
Configurator.GetSeverityValue(i,k))); //в Журнал
времени исполнения записываются названия и
значения категорий важности для i-го сервера
        k += 1;
    }
    i += 1;
}
```

В результате вызова функций в Журнал времени исполнения запишутся названия категорий важности с их значениями:

```
Журнал времени исполнения
Категория важности: Critical; Значение: 800
Категория важности: Important; Значение: 200
Категория важности: Info; Значение: 100
Категория важности: Debug; Значение: 0
```

## 1.1.4.2.4.7.10.2.8. GetSeverityCategory

Возвращает категорию важности аудита по индексу сервера аудита и индексу категории.



```
string GetSeverityCategory(int4 i, int4 k)
```

### Параметры

Параметр	Тип	Описание
i	int4	Индекс сервера аудита
k	int4	Индекс категории важности

Предоставляет название выбранной категории важности сообщений.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, каждый из них описывается в компоненте в виде элемента массива A[i]. Каждый элемент массива A[i] представляет собой массив B, каждый элемент B[i,k] которого описывает одну из категорий важности описываемого сервера A[i].



Массив A, описывающий сервера-потребители = [[Массив B, описывающий первый сервер], [Массив B, описывающий второй сервер]]

Массив B, описывающий первый сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Массив B, описывающий второй сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Обратиться к конкретной категории важности можно по номеру элемента в массивах A и B (i, k).

Каждый элемент массива  $B[i,k]$  соответствует одному из значений атрибута `Category` xml-элемента `<Severity>`, вложенного в xml-элемент `<SeverityMap>` одного из xml-элементов `<OpcDaLogConsumer>` в конфигурационном файле агента безопасности.

Пример использования приведен в описании функции [GetSeverityValue\(\)](#).

## 1.1.4.2.4.7.10.2.9. GetSignalsCount

Предоставляет количество сигналов, предназначенных для записи сообщений аудита, в одном из указанных в конфигурационном файле серверов-потребителей аудита.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, значение для каждого из них записывается в компонент в виде элемента массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует количеству элементов <Signal> внутри элемента <SignalMap>, вложенного в один из элементов <OpcDaLogConsumer>, в конфигурационном файле агента безопасности.



int4 GetSignalsCount(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс сервера аудита

## 1.1.4.2.4.7.10.2.10. GetSeverityCount

Предоставляет количество категорий важности сообщений для одного из указанных в конфигурационном файле серверов-потребителей аудита.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, значение от каждого из них записывается в компонент в виде элемента массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует количеству элементов <Severity> внутри элемента <SeverityMap>, вложенного в один из элементов <OpсDaLogConsumer>, в конфигурационном файле агента безопасности.



int4 GetSeverityCount(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс сервера аудита



## 1.1.4.2.4.7.10.2.11. GetServerType

Предоставляет информацию о типе одного из указанных в конфигурационном файле серверов-потребителей аудита.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, перечень их типов записывается в компонент в виде массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует значению атрибута `Type` xml-элемента `<Server>`, вложенного в один из xml-элементов `<OpcDaLogConsumer>`, в конфигурационном файле агента безопасности.



```
string GetServerType(int4 Index)
```

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс сервера аудита

## 1.1.4.2.4.7.10.2.12. GetServerProgId

Предоставляет ProgID (строковый идентификатор) одного из указанных в конфигурационном файле серверов-потребителей аудита.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, перечень их ProgID записывается в компонент в виде массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует значению атрибута ProgId xml-элемента <Server>, вложенного в один из xml-элементов <OpсDaLogConsumer>, в конфигурационном файле агента безопасности.



```
string GetServerProgId(int4 Index)
```

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс сервера аудита

## 1.1.4.2.4.7.10.2.13. GetAuditServerPort

Предоставляет порт для подключения к одному из указанных в конфигурационном файле серверов-потребителей аудита.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, перечень портов для подключения к ним записывается в компонент в виде массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует значению атрибута TCPServerPort xml-элемента <Server>, вложенного в один из xml-элементов <OpсDaLogConsumer>, в конфигурационном файле агента безопасности.



int4 GetAuditServerPort(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс сервера аудита

## 1.1.4.2.4.7.10.2.14. GetAuditServerHost

Предоставляет IP-адрес одного из указанных в конфигурационном файле серверов-потребителей аудита.

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, перечень их IP-адресов записывается в компонент в виде массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует значению атрибута Host xml-элемента <Server>, вложенного в один из xml-элементов <OpcDaLogConsumer>, в конфигурационном файле агента безопасности.



string GetAuditServerHost(int Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс сервера аудита

## 1.1.4.2.4.7.10.2.15. GetLdapPort

Предоставляет номер порта для подключения к одному из указанных в конфигурационном файле LDAP-серверов.

Поскольку в конфигурационном файле может быть указано несколько LDAP-серверов, перечень портов для подключения к ним записывается в компонент в виде массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует значению атрибута Port одного из xml-элементов <LDAPServer>, вложенного в xml-элемент <LdapHosts>, в конфигурационном файле агента безопасности.



int4 GetLdapPort(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс LDAP-сервера

## 1.1.4.2.4.7.10.2.16. GetLdapHost

Предоставляет IP-адрес одного из указанных в конфигурационном файле LDAP-серверов.

Поскольку в конфигурационном файле может быть указано несколько LDAP-серверов, перечень их IP-адресов записывается в компонент в виде массива. Обратиться к конкретному значению можно по номеру его записи в массив.

Каждый элемент массива соответствует значению атрибута Address одного из xml-элементов <LDAPServer>, вложенного в xml-элемент <LdapHosts>, в конфигурационном файле агента безопасности.



string GetLdapHost(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс LDAP-сервера

## 1.1.4.2.4.7.10.2.17. ClearLdapList

Очищает внутренний массив LDAP-серверов компонента.

Функция не требует входных параметров.

В случае успешного завершения операции активируется событие [LdapListChanged](#).



void ClearLdapList()

## 1.1.4.2.4.7.10.2.18. AddLdap

Добавляет описание LDAP-сервера, к которому должен подключаться агент безопасности. Добавленное описание хранится во внутреннем массиве компонента.



bool AddLdap(string LDAPHost, int4 LDAPPort)

### Параметры

Параметр	Тип	Описание
LDAPHost	string	IP адрес или имя компьютера, где расположен добавляемый LDAP-сервер
LDAPPort	int	Порт для подключения к добавляемому LDAP-серверу

### Значение

Значение	Описание
true	Информация добавлена
false	Информацию не удалось добавить

В случае успешного завершения операции активируется событие [LdapListChanged](#).



## 1.1.4.2.4.7.10.2.19. Read

Выполняет чтение конфигурации из конфигурационного файла Агент Astra.Security.



```
void Read(string XML)
```

### Параметры

<u>Параметр</u>	<u>Тип</u>	<u>Описание</u>
XML	string	Строка, содержащая полный текст конфигурационного файла Агент Astra.Security в xml-формате

Чтобы получить такую строку из файла с помощью компонентов Astra.HMI, используйте функцию `ReadTextFile()` компонента Окружение: файлы (File System). В качестве входного параметра функции укажите полный путь к файлу конфигурации.

Результат:

- сразу после вызова функции активируется событие [ReadingStarted](#);
- в случае успешного завершения операции активируется событие [ReadingFinished](#);
- в случае неуспешного завершения операции активируется событие [ReadingFailure](#).

### Примеры



```
Configurator.Read(FileSystem.ReadTextFile("C:/Program Files/  
AstraRegul/Astra.Security/astra.security.agent.xml"));
```

## 1.1.4.2.4.7.10.2.20. Generate

Создает текст конфигурационного файла Агент Astra.Security.



int4 Generate(bool ActivationFlag)

### Параметры

Параметр	Тип	Описание
ActivationFlag	bool	Параметр отвечает за совместимость создаваемой конфигурации со старыми версиями Astra.Security. В качестве значения следует указывать true.

Результат:

- › сразу после вызова функции активируется событие [GenerationStarted](#);
- › в случае успешного завершения операции активируется событие [GenerationFinished](#), а строка-результат записывается в свойство [GeneratedString](#);
- › в случае неуспешного завершения операции активируется событие [GenerationFailure](#).

Для создания конфигурации должны быть заполнены все обязательные поля. Обязательно следует указать:

- › адрес и порт Net-агента – в соответствующих свойствах;
- › параметры хотя бы одного LDAP-сервера – с помощью функции [AddLdap\(\)](#);
- › учетную запись администратора LDAP (в стандартном для LDAP виде) и пароль;
- › пароль пользователя, чья учетная запись используется по умолчанию;
- › адрес корневой папки LDAP-сервера;
- › защищено ли соединение агента безопасности с LDAP-сервером.

Если в процессе создания конфигурации возникнет ошибка, функция вернет ее код. Чтобы ознакомиться с текстом ошибки, обратитесь ко значению

свойства Ошибка конфигурирования (Error) компонента, либо ко значению внутренней переменной Error события [GenerationFailure](#).

## 1.1.4.2.4.7.10.2.21. GetSignalType

Предоставляет название типа сообщений в указанном сигнале для указанного сервера-потребителя.



```
string GetSignalType(int4 i, int4 k)
```

### Параметры

Параметр	Тип	Описание
i	int4	Индекс сервера аудита
k	int4	Индекс категории важности

Поскольку в конфигурационном файле может быть указано несколько серверов-потребителей, каждый из них описывается в компоненте в виде элемента массива  $A[i]$ . Каждый элемент массива  $A[i]$  представляет собой массив  $B$ , каждый элемент  $B[i,k]$  которого описывает один из сигналов в описываемом сервере  $A[i]$ .



Массив  $A$ , описывающий сервера-потребители = [[Массив  $B$ , описывающий первый сервер], [Массив  $B$ , описывающий второй сервер]]

Массив  $B$ , описывающий первый сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Массив  $B$ , описывающий второй сервер = [[Описание первой категории важности], [Описание второй категории важности]]

Обратиться к конкретному сигналу можно по номеру элемента в массивах  $A$  и  $B$  ( $i, k$ ).

Каждый элемент массива  $B[i,k]$  соответствует одному из значений атрибута `Type` xml-элемента `<Signal>`, вложенного в xml-элемент `<SignalMap>` одного из xml-элементов `<OpcDaLogConsumer>` в конфигурационном файле агента безопасности..

## Примеры



Допустим, в конфигурационном файле описан один сервер-потребитель сообщений (один элемент `<OpcDaLogConsumer>`), в котором для записи сообщений аудита предназначено несколько сигналов:



```
<AuditLogConsumers TraceAudit="1">
  <OpcDaLogConsumer>
    <Server Host="127.0.0.1" Type="OPC" ...>
      <SignalMap>
        <Signal Name="DynEvents.NormalDynSignal"
          Mode="DynamicEvent"
          Type="Normal"/>
          <Signal Name="DynEvents.AdminDynSignal"
            Mode="DynamicEvent" Type="Admin"/>
            <Signal Name="DynEvents.UserNameDynSignal"
              Mode="DynamicEvent" Type="UserName"/>
              <Signal Name="DynEvents.DisplayNameDynSignal"
                Mode="DynamicEvent" Type="DisplayName"/>
                <Signal Name="DynEvents.GroupNameDynSignal"
                  Mode="DynamicEvent" Type="GroupName"/>
                  <Signal Name="DynEvents.WorkstationNameDynSignal"
                    Mode="DynamicEvent" Type="WorkstationName"/>
                    <Signal Name="DynEvents.NormalMessage" Mode="Value"
                      Type="Normal"/>
                      <Signal Name="DynEvents.AdminMessage" Mode="Value"
                        Type="Admin"/>
```


```

        <Signal Name="DynEvents.UserNameMessage"
Mode="Value" Type="UserName"/>
        <Signal Name="DynEvents.DisplayNameMessage"
Mode="Value" Type="DisplayName"/>
        <Signal Name="DynEvents.GroupNameMessage"
Mode="Value" Type="GroupName"/>
        <Signal Name="DynEvents.WorkstationNameMessage"
Mode="Value" Type="WorkstationName"/>
    </SignalMap>
</Server>
</OpcDaLogConsumer>
</AuditLogConsumers>

```

Чтобы получить список параметров сигналов в каждом сервере-потребителе, вызовите нужные функции в коде, выполняющемся в случае успешного чтения конфигурации Агент Astra.Security (например, в обработчике события [ReadingFinished\(\)](#)). Укажите в качестве входных параметров индексы *i* (индекс в массиве серверов-потребителей) и *k* (индекс в массиве сигналов). Приведенный ниже пример написан на языке Astra.Om, в нем список сигналов с их параметрами записывается в лог:

```

 i: int4 = 0;
while (i < Configurator.ConsumersCount) //цикл выполняется,
пока в массиве А не будут описаны все сервера-
потребители
{
    k: int4 = 0;
    while (k < Configurator.GetSignalsCount(i)) //цикл
выполняется, пока в массив В не будут записаны все
сигналы i-го сервера-потребителя
    {

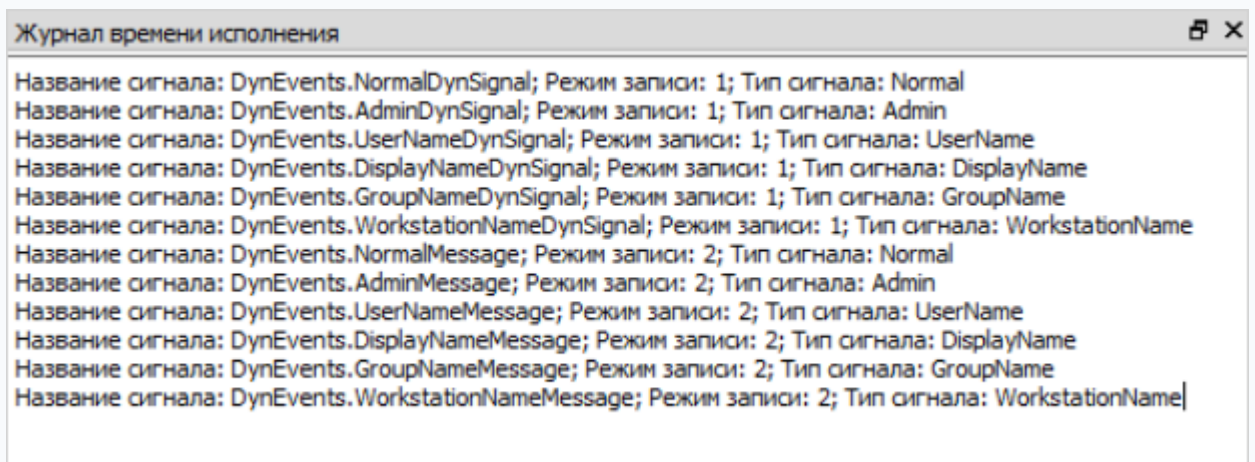
```

```

        DebugTool.Log("Название сигнала:
"+Configurator.GetSignalName(i,k)+"; Режим записи:
"+String.ToString(
Configurator.GetSignalMode(i,k))+"; Тип сообщения: "+
Configurator.GetSignalType(i,k)); //в Журнал времени
исполнения записывается список сигналов i-го сервера
        k += 1;
    }
    i += 1;
}

```

В результате вызова функций в Журнал времени исполнения запишется список сигналов с их параметрами:



```

Журнал времени исполнения
Название сигнала: DynEvents.NormalDynSignal; Режим записи: 1; Тип сигнала: Normal
Название сигнала: DynEvents.AdminDynSignal; Режим записи: 1; Тип сигнала: Admin
Название сигнала: DynEvents.UserNameDynSignal; Режим записи: 1; Тип сигнала: UserName
Название сигнала: DynEvents.DisplayNameDynSignal; Режим записи: 1; Тип сигнала: DisplayName
Название сигнала: DynEvents.GroupNameDynSignal; Режим записи: 1; Тип сигнала: GroupName
Название сигнала: DynEvents.WorkstationNameDynSignal; Режим записи: 1; Тип сигнала: WorkstationName
Название сигнала: DynEvents.NormalMessage; Режим записи: 2; Тип сигнала: Normal
Название сигнала: DynEvents.AdminMessage; Режим записи: 2; Тип сигнала: Admin
Название сигнала: DynEvents.UserNameMessage; Режим записи: 2; Тип сигнала: UserName
Название сигнала: DynEvents.DisplayNameMessage; Режим записи: 2; Тип сигнала: DisplayName
Название сигнала: DynEvents.GroupNameMessage; Режим записи: 2; Тип сигнала: GroupName
Название сигнала: DynEvents.WorkstationNameMessage; Режим записи: 2; Тип сигнала: WorkstationName

```

## 1.1.4.2.4.7.10.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">ConsumersCount</a>	Количество потребителей аудита, указанных в конфигурации Astra.Security.Agent
<a href="#">LdapCount</a>	Количество LDAP-серверов, указанных в конфигурации Astra.Security.Agent
<a href="#">ReadError</a>	Текст ошибки, возникшей при чтении конфигурации Astra.Security.Agent
<a href="#">GeneratedString</a>	Текст готовой конфигурации Astra.Security.Agent в формате .xml
<a href="#">Error</a>	Текст ошибки, возникшей при конфигурировании Astra.Security.Agent
<a href="#">Уровень логирования</a>	Степень вывода сообщений о работе в журнал приложений
<a href="#">Пароль пользователя по умолчанию</a>	Пароль учетной записи, используемой по умолчанию
<a href="#">Пользователь по умолчанию</a>	Любой пользователь, существующий на LDAP-сервере
<a href="#">Имя гостевой УЗ</a>	Произвольное имя гостевой учетной записи
<a href="#">Адрес папки системы безопасности</a>	Путь к папке с конфигурацией системы безопасности на LDAP-сервере
<a href="#">Пароль пользователя LDAP</a>	Пароль пользователя для подключения к LDAP-серверу



<a href="#">Пользователь LDAP</a>	Путь к пользовательской папке на LDAP-сервере
<a href="#">Режим работы контроля целостности</a>	Включение/выключение режима работы системы контроля целостности файлов и папок
<a href="#">Использование защищенного соединения</a>	Использование/отказ от использования защищенного соединения при подключении к LDAP-серверу
<a href="#">Порт LDAP сервера</a>	Порт для подключения к LDAP-серверу
<a href="#">Адрес LDAP сервера</a>	IP адрес или имя компьютера, где расположен LDAP-сервер
<a href="#">Порт Net агента</a>	Порт для подключения к Astra.Net.Agent
<a href="#">Адрес Net агента</a>	IP адрес или имя компьютера, где расположен Astra.Net.Agent
<a href="#">Комбинации блокировок</a>	Комбинация клавиш, блокируемые драйвером клавиатуры
<a href="#">Трассировка аудита</a>	Логировать ли сообщения аудита в журнал приложений
<a href="#">Кэш прав пользователя</a>	Использовать ли кэширование значения прав текущего пользователя
<a href="#">Префикс сообщений аудита</a>	Префикс для сообщений аудита для идентификации источника

## 1.1.4.2.4.7.10.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.7.10.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.10.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.10.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.7.10.3.5. Адрес Net агента

IP-адрес или имя компьютера, на котором установлен Net-агент, к которому подключается агент безопасности. Под Net-агентом подразумевается служба Astra.Net.Agent (на ОС Windows) или Astra.Net.service (на ОС Linux).

Зачастую Net-агент установлен на локальном компьютере.

Значение свойства соответствует значению атрибута Address элемента <EntryPointNetAgent> в конфигурационном файле агента безопасности.



string NetHost

## 1.1.4.2.4.7.10.3.6. Порт Net агента

Номер порта для подключения к Net-агенту, к которому подключается агент безопасности.

Значение свойства соответствует значению атрибута Port элемента <EntryPointNetAgent> в конфигурационном файле агента безопасности.



int4 NetPort

## 1.1.4.2.4.7.10.3.7. Адрес LDAP сервера

IP-адрес или имя компьютера, на котором развернут LDAP-сервер, к которому подключается агент безопасности.

Свойство устарело. Теперь при настройке агента может быть указано несколько LDAP-серверов, поэтому значения адресов записываются в компонент в виде массива.

Чтобы прочитать конкретное значение в массиве:

1. Вызовите функцию [Read\(\)](#), предназначенную для записи текущей конфигурации агента в массив.
2. Вызовите функцию [GetLdapHost\(\)](#), указав в качестве входного параметра номер элемента в массиве.

Количество элементов в массиве (LDAP-серверов) записывается в значение свойства `LdapCount`. Каждый элемент массива соответствует значению атрибута `Address` одного из элементов `<LDAPServer>` в конфигурационном файле агента безопасности.

Чтобы добавить описание нового LDAP-сервера, используйте функцию [AddLdap\(\)](#).



string LdapHost



## 1.1.4.2.4.7.10.3.8. Порт LDAP сервера

Номер порта для подключения к LDAP-серверу.

Свойство устарело. Теперь при настройке агента может быть указано несколько LDAP-серверов, поэтому значения портов записываются в компонент в виде массива.

Чтобы прочитать конкретное значение в массиве:

1. Вызовите функцию [Read\(\)](#), предназначенную для записи текущей конфигурации агента в массив.
2. Вызовите функцию [GetLdapPort\(\)](#), указав в качестве входного параметра номер элемента в массиве.

Количество элементов в массиве (LDAP-серверов) записывается в значение свойства `LdapCount`. Каждый элемент массива соответствует значению атрибута `Port` одного из элементов `<LDAPServer>` в конфигурационном файле агента безопасности.

Чтобы добавить описание нового LDAP-сервера, используйте функцию [AddLdap\(\)](#).



string LdapPort

## 1.1.4.2.4.7.10.3.9. Использование защищенного соединения

Указывает агенту безопасности необходимость использования защищенного соединения с LDAP-сервером.

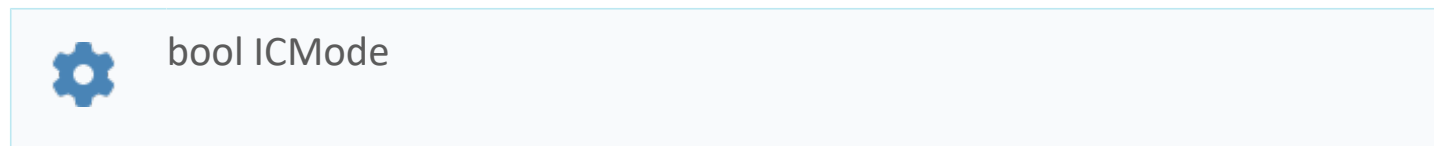
Значение свойства соответствует значению атрибута `value` элемента `<LdapSecure>` в конфигурационном файле агента безопасности.



bool UseSecureConnection

## 1.1.4.2.4.7.10.3.10. Режим работы контроля целостности

Указывает, включен ли режим контроля целостности файлов и папок.



Значение свойства соответствует значению атрибута ICMode элемента <Options> в конфигурационном файле агента безопасности:

- 0 – контроль целостности отключен, значение свойства – false;
- 1 – контроль целостности включен, значение свойства – true.

## 1.1.4.2.4.7.10.3.11. Пользователь LDAP

Учетная запись администратора LDAP-сервера.



string LdapUser

Поскольку учетная запись хранится на LDAP-сервере в виде каталога, обращение к ней происходит в стандартном для LDAP виде:



cn=ЛогинПользователя,dc=ДоменLDAPсервера

Значение свойства соответствует значению атрибута value элемента <LdapUser> в конфигурационном файле агента безопасности.

### Примеры



Пример значения: cn=Manager,dc=maxcrc,dc=com

## 1.1.4.2.4.7.10.3.12. Пароль пользователя LDAP

Пароль учетной записи администратора LDAP-сервера.

Поскольку пароли являются секретной информацией, просмотреть текущее значение при чтении конфигурации нельзя. При создании новой конфигурации свойство используется для записи нового значения. Значение пароля, введенное в открытом виде, будет зашифровано.



string LdapUserPass

## 1.1.4.2.4.7.10.3.13. Адрес папки системы безопасности

Название корневого каталога на LDAP-сервере.



string LdapDN

Обращение к каталогу происходит в стандартном для LDAP виде:



ou=НазваниеПапки,dc=ДоменLDAPсервера

Значение свойства соответствует значению атрибута value элемента <SecurityDn> в конфигурационном файле агента безопасности.

### Примеры



Пример значения: ou=AstraSecurity,dc=maxcsc,dc=com

## 1.1.4.2.4.7.10.3.14. Имя гостевой УЗ

Имя гостевой учетной записи – записи, чьи права используются, когда нет активной пользовательской сессии.

Значение свойства соответствует значению атрибута `value` элемента `<GuestDisplayName>` в конфигурационном файле агента безопасности.



string GuestName

## 1.1.4.2.4.7.10.3.15. Пользователь по умолчанию

Имя учетной записи пользователя по умолчанию – пользователя, чья сессия становится активной, если другие пользователи не авторизовались в подсистеме.

Значение свойства соответствует значению атрибута value элемента <DefaultUser> в конфигурационном файле агента безопасности.



string DefaultUser



## 1.1.4.2.4.7.10.3.16. Пароль пользователя по умолчанию

Пароль пользователя по умолчанию.

Значение свойства соответствует значению атрибута value элемента <DefaultUserPassword> в конфигурационном файле агента безопасности.



string DefaultUserPass

## 1.1.4.2.4.7.10.3.17. Уровень логирования

Означает количество информации, выводимой в лог подсистемы безопасности.

Значение свойства соответствует значению атрибута `LogLevel` элемента `<Options>` в конфигурационном файле агента безопасности:

- 0 – в лог выводится минимум информации;
- 2 – в лог выводится вся основная информация о работе Astra.Security;
- 5 – в лог выводится дополнительная информация о работе Astra.Security помимо основной.



int4 LogLevel

## 1.1.4.2.4.7.10.3.18. Комбинации блокировки

Перечень клавиш, заблокированных для использования пользователем.

Заблокированные сочетания указаны в виде SCAN-кодов клавиш, разделенных внутри сочетания символом "+", а между сочетаниями – символом ";".

Значение свойства соответствует значению атрибута kbDriverString элемента <Options> в конфигурационном файле агента безопасности.



string DriverString

### Примеры



Пример значения: 0x1D+0x38+0x53;0x1D+0x2A+0x01;

В данном примере заблокированы сочетания Ctrl+Alt+Del и Ctrl+Shift+Esc.

## 1.1.4.2.4.7.10.3.19. Трассировка аудита

Включает трансляцию сообщений аудита в системный журнал.

Значение свойства `TraceAudit` соответствует значению атрибута `<AuditLogConsumers>` в конфигурационном файле агента безопасности:

- 0 – сообщения не транслируются в системный журнал, значение свойства – `false`;
- 1 – сообщения транслируются в системный журнал, значение свойства – `true`.



bool TraceAudit

## 1.1.4.2.4.7.10.3.20. Кэш прав пользователя

Указывает, используются ли кэшированные значения прав пользователя.

Значение свойства соответствует значению атрибута `UseRightsCacheStorage` элемента `<Options>` в конфигурационном файле агента безопасности:

- 0 – актуальные значения прав пользователя запрашиваются с LDAP-сервера постоянно, значение свойства – `false`;
- 1 – значения прав пользователя запрашиваются с LDAP-сервера только в момент входа пользователя в подсистему, а в процессе его работы значения запрашиваются из кэша. Значение свойства в таком случае – `true`.



`bool UseRightsStorageCache`

## 1.1.4.2.4.7.10.3.21. Префикс сообщений аудита

Префикс сообщений аудита.

Значение свойства соответствует значению атрибута `value` элемента `<mesPrefix>` в конфигурационном файле агента безопасности.



string MesPrefix

## 1.1.4.2.4.7.10.3.22. UnderfinedListItems

Список полей (свойств), для которых обязательно указать значение для создания новой конфигурации Агент Astra.Security с помощью функции [Generate\(\)](#).



string UnderfinedListItems



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.7.10.3.23. ConsumersCount

Количество серверов-потребителей аудита, указанных в конфигурационном файле Агент Astra.Security.



int4 ConsumersCount



Доступно только для чтения в режиме рантайма.



## 1.1.4.2.4.7.10.3.24. LdapCount

Количество LDAP-серверов, указанных в конфигурационном файле Агент Astra.Security.



int4 LdapCount



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.7.10.3.25. ReadError

Текст ошибки, возникшей при попытке чтения конфигурации Агент Astra.Security с помощью функции [Read\(\)](#).



string ReadError



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.7.10.3.26. GeneratedString

Полный текст конфигурационного файла Агент Astra.Security в xml-формате, созданного в результате вызова функции [Generate\(\)](#).



string GeneratedString



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.7.10.3.27. Error

Текст ошибки, возникшей при попытке создания новой конфигурации Агент Astra.Security с помощью функции [Generate\(\)](#).



string Error



Доступно только для чтения в режиме рантайма.

## **1.1.4.2.4.7.11. Информация лицензирования: Получение**

Компонент предназначен для получения информации о лицензировании.

## 1.1.4.2.4.7.11.1. События

Событие	Описание
<a href="#">RequestLicenseInfoComplete</a>	Сигнал об успешном получении LicenseInfo, определенной сервером лицензирования на устройстве
<a href="#">RequestRemoteLicenseInfoComplete</a>	Сигнал об успешном получении LicenseInfo, определенной сервером лицензирования на устройстве с удаленного узла
<a href="#">RequestLicenseInfoFailed</a>	Сигнал об ошибке в процессе получения LicenseInfo, определенной сервером лицензирования на устройстве
<a href="#">RequestRemoteLicenseInfoFailed</a>	Сигнал об ошибке в процессе получения LicenseInfo, определенной сервером лицензирования на устройстве с удаленного узла

## 1.1.4.2.4.7.11.1.1. RequestLicenseInfoComplete

Сигнал об успешном получении LicenseInfo, определенной сервером лицензирования на устройстве.

## **1.1.4.2.4.7.11.1.2.**

### **RequestRemoteLicenseInfoComplete**

Сигнал об успешном получении LicenseInfo, определенной сервером лицензирования на устройстве с удаленного узла.



## 1.1.4.2.4.7.11.1.3. RequestLicenseInfoFailed

Сигнал об ошибке в процессе получения LicenseInfo, определенной сервером лицензирования на устройстве.

## 1.1.4.2.4.7.11.1.4.

### **RequestRemoteLicenseInfoFailed**

Сигнал об ошибке в процессе получения LicenseInfo, определенной сервером лицензирования на устройстве с удаленного узла.

## 1.1.4.2.4.7.11.2. Функции

Функция	Описание
<a href="#">RequestLicenseInfo</a>	Асинхронный запрос на получение лицензионной информации
<a href="#">RequestRemoteLicenseInfo</a>	Асинхронный запрос на получение лицензионной информации с удаленного узла
<a href="#">GetErrorDescriptionByCode</a>	Возвращает текстовое описание ошибок, возникающих при загрузке и сохранении приложений

## 1.1.4.2.4.7.11.2.1. RequestLicenseInfo

Асинхронный запрос на получение лицензионной информации.



```
void RequestLicenseInfo(string jsonRequest)
```

### Параметры

Параметр	Тип	Описание
jsonRequest	string	Запрос

## 1.1.4.2.4.7.11.2.2. RequestRemoteLicenseInfo

Асинхронный запрос на получение лицензионной информации с удаленного узла.



```
void RequestRemoteLicenseInfo(string jsonRequest, string aNetNode)
```

### Параметры

Параметр	Тип	Описание
jsonRequest	string	Запрос
aNetNode	string	Имя удаленного узла

## 1.1.4.2.4.7.11.2.3. GetErrorDescriptionByCode

Возвращает текстовое описание ошибок, возникающих при загрузке и сохранении приложений.



```
string GetErrorDescriptionByCode(uint1 FailReasonCode)
```

### Параметры

Параметр	Тип	Описание
FailReasonCode	uint1	Код ошибки из любого события, говорящего об ошибке

## 1.1.4.2.4.7.11.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Контекст безопасности</a>	Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом
<a href="#">LicenseInfo</a>	Уведомление об изменении в группе, загруженной в элемент

## 1.1.4.2.4.7.11.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.7.11.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.7.11.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.7.11.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.7.11.3.5. Контекст безопасности

Ссылка на компонент Контекст безопасности, относительно которого будет происходить дальнейшая работа с компонентом. Указывается на вкладке Редактор свойств.



Необходимо заполнить для взаимодействия с подсистемой безопасности Astra.Security.

## 1.1.4.2.4.7.11.3.6. LicenseInfo

Уведомление об изменении в группе, загруженной в элемент.

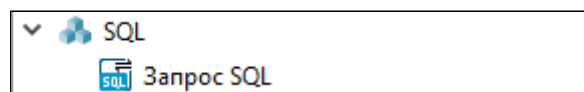


string LicenseInfo

## 1.1.4.2.4.8. SQL

Набор компонентов Astra.NMI для взаимодействия с базами данных.

› [Запрос SQL](#)



## 1.1.4.2.4.8.1. Запрос SQL

Компонент Запрос SQL предназначен для выполнения SQL-запросов к базе данных (PostgreSQL, MySQL, Microsoft SQL Server и другие).

## 1.1.4.2.4.8.1.1. События

Событие	Описание
<u>OnOperationFinished</u>	Завершение выполнения запроса
<u>OnConnectionStateChanged</u>	Изменение состояния подключения к базе данных



## 1.1.4.2.4.8.1.1.1. OnOperationFinished

Завершение выполнения запроса.

### Параметры

Параметр	Тип	Описание
operationType	uint1	Тип операции: <ul style="list-style-type: none"><li>› «0» – Подготовка запроса</li><li>› «1» – Выполнение запроса</li></ul>
operationResult	uint1	Результат выполнения операции: <ul style="list-style-type: none"><li>› «0» – Операция успешно выполнена</li><li>› «1» – При выполнении операции произошла ошибка</li></ul>
message	string	Текст сообщения по результатам выполнения операции

### Примеры



//Используя параметры «operationResult» и «message» события OnOperationFinished вывести в текстовое поле результат выполнения SQL-запроса:

```
if (operationType == 1) {  
    if ( operationResult == 0 )  
        Text_1.Text = "Success:" + message;  
    else  
        Text_1.Text = "Error" + message;  
}
```

## 1.1.4.2.4.8.1.1.2. OnConnectionStateChanged

Изменение состояния подключения к базе данных.

### Параметры

Параметр	Тип	Описание
state	uint1	Состояние соединения: <ul style="list-style-type: none"><li>› «0» – Соединение не установлено</li><li>› «1» – Соединение установлено</li></ul>
stateAsString	string	Состояние соединения в строковом представлении

### Примеры



//Используя параметры «state» и «stateAsString» события OnConnectionStateChanged вывести в текстовое поле состояние соединения с базой данных. В случае отсутствия соединения вывести уведомление об ошибке:

```
if ( state == 0 )
    Text_1.Text = "ConnectError:" + stateAsString;
else
    Text_1.Text = "Connected!";
```

## 1.1.4.2.4.8.1.2. Функции

Компонент	Описание
<a href="#">Connect</a>	Устанавливает соединение с базой данных
<a href="#">Execute</a>	Выполняет SQL-запрос, переданный через свойство Текст запроса
<a href="#">Prepare</a>	Выполняет подготовку SQL-запроса, переданного через свойство Текст запроса

## 1.1.4.2.4.8.1.2.1. Connect

Устанавливает соединение с базой данных.



```
void Connect();
```

## 1.1.4.2.4.8.1.2.2. Execute

Выполняет SQL-запрос, переданный через свойство Текст запроса.



```
void Execute();
```

## 1.1.4.2.4.8.1.2.3. Prepare

Выполняет подготовку SQL-запроса, переданного через свойство Текст запроса. Подготовка включает в себя синтаксическую проверку SQL-запроса, построение плана выполнения запроса и выделение под него вычислительных ресурсов.



```
void Prepare();
```

## 1.1.4.2.4.8.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Строка подключения</a>	Строка, которая содержит информацию, необходимую ODBC драйверу для подключения к базе данных
<a href="#">Текст запроса</a>	Текст SQL-запроса
<a href="#">Состояние подключения к БД</a>	Текущий статус соединения с базой данных

## 1.1.4.2.4.8.1.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.8.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.8.1.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.8.1.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.8.1.3.5. Строка подключения

Строка, которая содержит информацию, необходимую ODBC драйверу для подключения к базе данных.



string ConnectionString



Системные источники данных можно посмотреть в Панель управления → Администрирование → Администратор источника данных ODBC → Вкладка Системный DSN.

### Примеры



В примере ниже используется строка подключения, которая содержит информацию о драйвере «Driver» для PostgreSQL, адресе сервера «Server», порте «Port», базе «Database», а также учетные данные пользователя «Uid - имя пользователя и пароль».

```
Driver={PostgreSQL};Server=localhost;Port=5432;Database=postgres;Uid=postgres
```

Вместо использования целого ряда атрибутов, определяющих способ подключения к источнику данных, можно указать только имя системного источника (DSN):

```
DSN=PostgreSQL35W
```



//Настроить подключение к БД, если имя системного источника данных «PostgreSQL35W»:

```
SqlQuery_1.ConnectionString = "DSN=PostgreSQL35W";
```

## 1.1.4.2.4.8.1.3.6. Текст запроса

Текст SQL-запроса.



string Text

### Примеры



//Выполнить «SELECT»запрос для выборки данных по всем столбцам из таблицы «Numbers»:

```
Table_1.TableModel_1.SqlQuery_1.Text = "SELECT * FROM Numbers";
```

## 1.1.4.2.4.8.1.3.7. Состояние подключения к БД

Текущий статус соединения с базой данных.



uint1 ConnectionState



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
1	Соединение установлено
0	Соединение не установлено

### Примеры



//Подключиться к базе данных, если соединение с ней не установлено:

```
if ( Table_1.TableModel_1.SqlQuery_1.ConnectionState == 0 )  
Table_1.TableModel_1.SqlQuery_1.Connect( );
```

## 1.1.4.2.4.9. Таблицы

Чтобы получить возможность:

- › помещать данные, полученные от источника по TCP, в таблицу
- › формировать собственные таблицы данных

следует подключить внешнюю библиотеку `astra.hmi.tables` (для ОС Windows) и `libastra.hmi.tables` (для ОС Linux).

Чтобы подключить библиотеки:

- › ОС Windows

Разместите файл библиотеки в папке установки Astra.HMI.

- › ОС Linux

Скопируйте библиотеку в папку Astra.HMI командой:



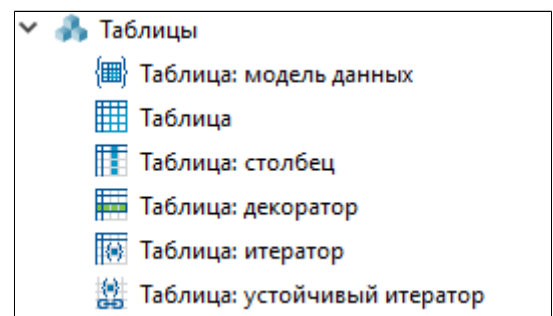
```
cp <имя библиотеки>.so /opt/AstraRegul/Astra.HMI
```



Версия SDK, на которой выполнялась сборка библиотеки, должна быть такой же, какая используется в Astra.HMI.

После подключения библиотеки в библиотеку компонентов Astra.HMI добавится юнит Таблицы с компонентами:

- › [Таблица: модель данных](#)
- › [Таблица](#)
- › [Таблица: столбец](#)
- › [Таблица: декоратор](#)
- › [Таблица: итератор](#)
- › [Таблица: устойчивый итератор](#)



Чтобы сформировать таблицу данных, предварительно выберите способ заполнения таблицы. Таблица может заполняться данными, полученными от источника, либо данными, введенными вручную с помощью скриптов Astra.Om или JavaScript.

В любом из выбранных способов используется компонент Таблица: модель данных. Модель данных хранит в себе полученные данные в табличном виде.

Чтобы графически представить на экранной форме данные, хранящиеся в модели, используйте компонент Таблица.

## Заполнение таблицы данными от источника

В качестве источника, к примеру, используйте Astra.Server, с которого можно запросить оперативные и исторические события по TCP. Каждое событие, сгенерированное на источнике, будет помещаться в таблицу модели данных.



Вы не можете изменить количество строк и столбцов в таблице модели данных, так как это выполняет компонент Запрос алармов на основе данных, полученных от источника.

Чтобы настроить заполнение таблицы модели данных событиями, полученными с источника:

1. Добавьте компонент Источник AP. Задайте значения в свойствах Хост, Порт, Порт истории (если планируете запрашивать исторические события), Активность.



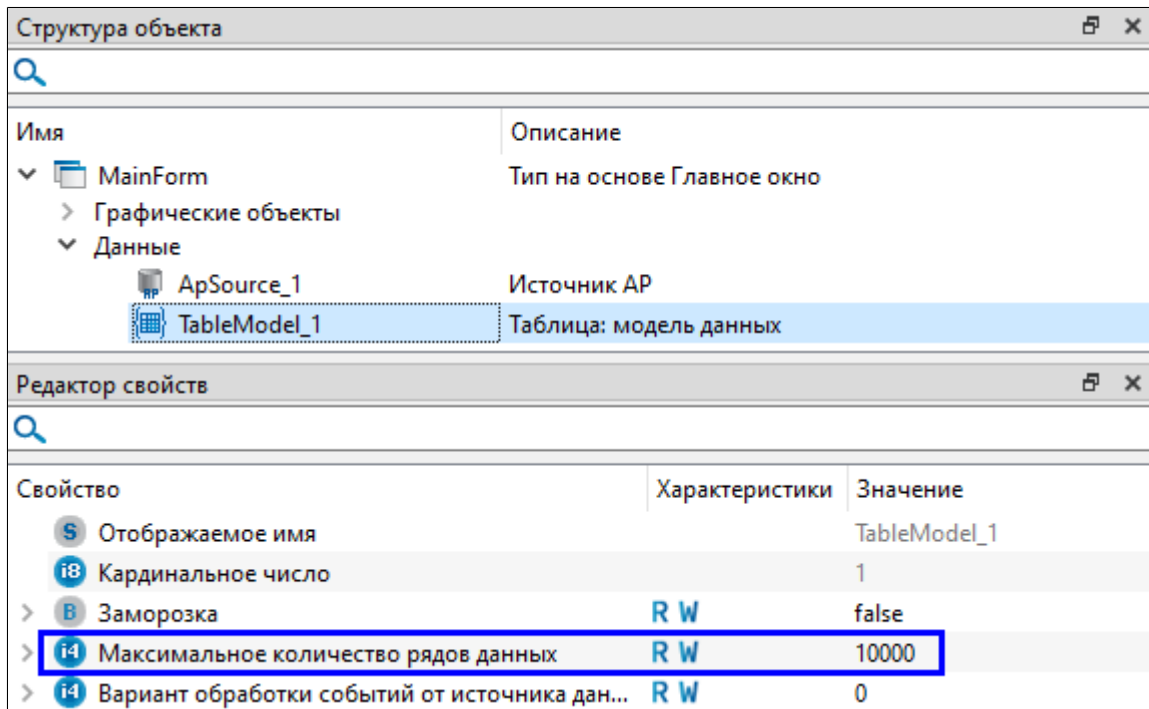
Структура объекта

Имя	Описание
Графические объекты	
Button_1	Кнопка
Данные	
ApSource_1	Источник AP

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		ApSource_1
Кардинальное число		1
Родительский источник	R W → 0	<не определено>
Источник учетных данных	R W → 0	<не определено>
Таймаут операций с БД	R ∞	<не определено>
Таймаут установления соединения с БД	R ∞	<не определено>
Максимальный размер пула буферов	R ∞	<не определено>
Общее соединение	R ∞	<не определено>
> Хост	R W	localhost
> Порт	R W	4388
> Порт истории	R W	4950
> Путь	R W	
> Активность	R W	true
> Период переподписки	R W	0
> Имя клиента	R W	HMI
> Идентификатор клиента	R W	HMI

2. Добавьте компонент Таблица: модель данных. Укажите максимальное число строк таблицы в свойстве Максимальное количество рядов данных.



3. Добавьте компонент Запрос алармов дочерним компоненту Таблица: модель данных. Укажите источник данных в свойстве Источник, задайте режим работы компонента в зависимости от вида запрашиваемых событий (1 - исторические события, 2 - оперативные события) и активируйте компонент в свойстве Активность.

Структура объекта

Имя	Описание
Данные	
ApSource_1	Источник AP
TableModel_1	Таблица: модель данных
Данные	
ApAlarmQuery_1	Запрос алармов

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		ApAlarmQuery_1
Кардинальное число		1
Источник	R $\leq$ $\rightarrow$ 0 ✓	ApSource_1
Режим работы	R W	2
Начало исторического интервала	R W	0
Конец исторического интервала	R W	0
Фильтр в JSON-формате	R W	
Фильтр исторических данных в JSON-формате	R W	
Активность	R W	true
Удалять квитируемые и деактивированные с...	R W	false
Запрашивать список активных событий при п...	R W	false
Интервал предзагрузки истории	R W	0
Расширяющий интервал	R W	0
Хронология исторических событий	R W	0
Режим сведения событий	R W	0
Размер буфера оперативных событий	R W	0
Параметры обработки данных	R W	
Спецификация пользовательских атрибутов	R W	

4. Пропишите скрипт, который будет выполнять запрос данных от источника (функция `Reload()` компонента Запрос алармов) и чтение полученных данных моделью данных (функция `BeginReadAsync()` компонента Таблица: модель данных).
5. Добавьте компонент Таблица для визуализации данных из модели (см. ниже).

Для самостоятельного воспроизведения стандартных функций `Astra.Alarms` (подгрузка исторических данных, удаление квитируемых событий и т.д.) используйте свойства и функции компонента Запрос алармов.

## Ручное заполнение таблицы

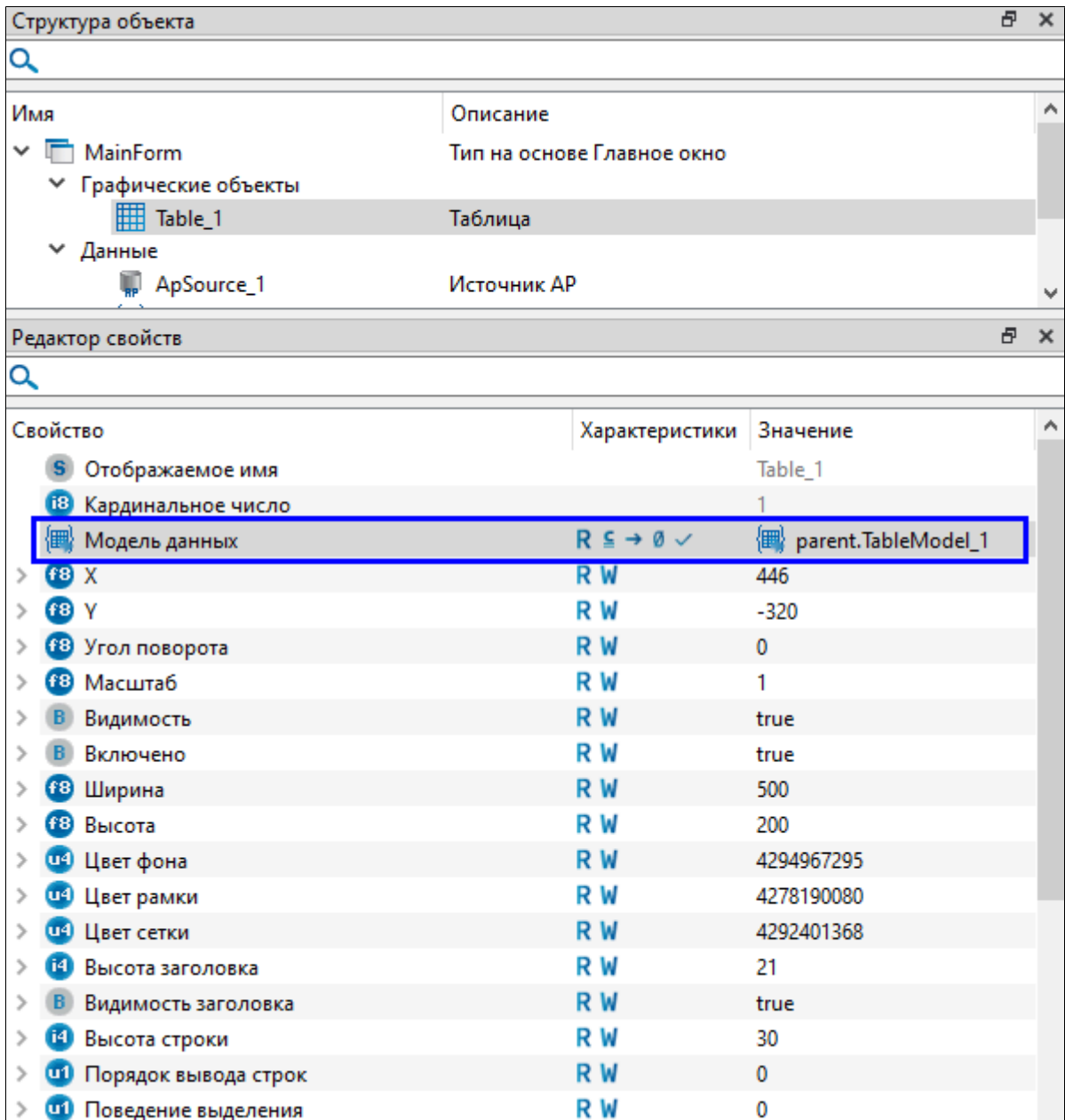
Чтобы наполнять таблицу модели данных самостоятельно, используйте функции компонента Таблица: модель данных.

Что сформировать таблицу в модели данных, выполните:

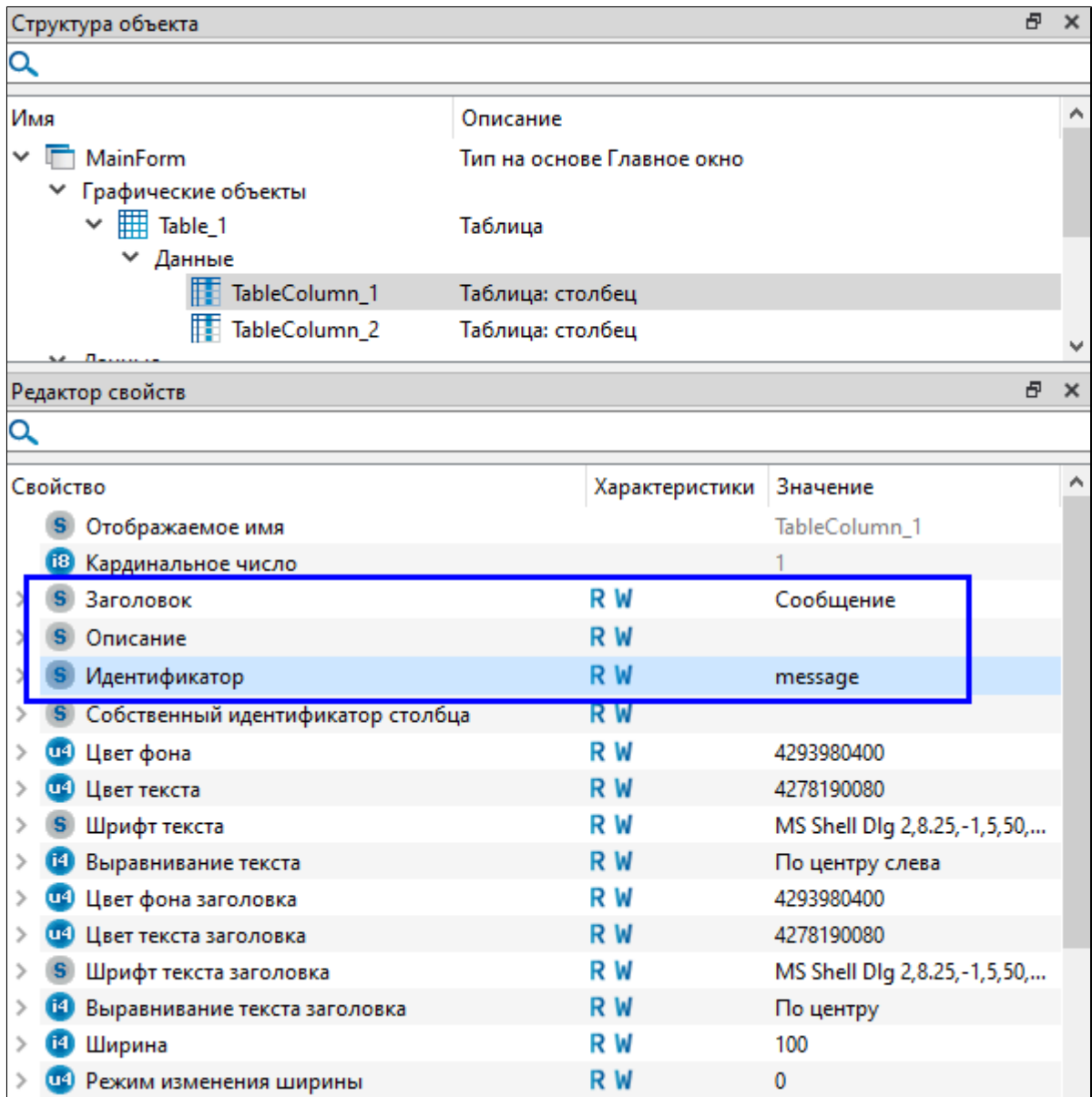
1. Добавьте нужно количество столбцов с помощью функции AddColumn.
2. Добавьте нужно количество строк с помощью функции AddRow.
3. Заполните ячейки таблицы данными с помощью функции SetCellData.
4. Добавьте компонент Таблица для визуализации данных из модели (см. ниже).

## Графическое представление данных из таблицы модели данных

Чтобы визуализировать данные из таблицы модели, добавьте на экранную форму компонент Таблица. В свойстве Модель данных свяжите таблицу с моделью данных, значения которой будут визуализироваться.



Вы можете настроить отображение в таблице на форме всех данных модели либо только отдельных столбцов. Добавьте компоненту Таблица дочерние компоненты Таблица: столбец. Для каждого столбца графической таблицы (свойство Идентификатор) запишите идентификатор столбца таблицы в модели данных. Идентификатор позволяет указать столбец, из которого в графическую таблицу будут подгружаться данные.



Идентификаторы столбцов таблицы модели соответствуют идентификаторам столбцов в источнике либо указываются вручную при конфигурировании таблицы с помощью скриптов.

Возможные значения идентификатора при получении данных от источника:

Значение	Описание
source	Источник события
time	Время генерации уведомления о событии
message	Сообщение
severity	Уровень важности события

condition_name	Имя условия генерации события
subcondition_name	Имя подусловия генерации события
quality	Текущее качество сигнала, изменение которого привело к генерации события
active_time	Время перехода состояния события в активное
actor_id	Имя пользователя, выполнившего квитирование сообщения о событии
ack	Признак квитирования события
active	Признак активности подусловия, по которому было сгенерировано событие
cookie	Специальный идентификатор события, который имеет служебное назначение. Необходим для сведения квитанций, деактиваций с событиями
ack_time	Время квитирования
ack_required	Требование квитирования
value	Значение сигнала, изменение которого привело к генерации события
sound	Звуковой файл, исполняемый при выполнении подусловия генерации сообщения о событии
area_path	Относительный тег объекта
object_id	Идентификатор объекта на сервере, по которому было сгенерировано событие
deactive_time	Время, когда событие перешло из активного состояния в неактивное

## 1.1.4.2.4.9.1. Таблица: модель данных

Компонент позволяет сформировать невизуальную таблицу, предназначенную для хранения данных. Данные могут поступать в таблицу от источника AP либо вводиться вручную с помощью скриптов Astra.OM или JavaScript.



## 1.1.4.2.4.9.1.1. События

Событие	Описание
<a href="#">RowsRemoved</a>	Удаление строки
<a href="#">RowsInserted</a>	Вставка строки
<a href="#">ReadFinished</a>	Завершение чтения данных моделью
<a href="#">DataChanged</a>	Изменение содержимого в таблице модели данных
<a href="#">GetContentAsJsonFinished</a>	Сигнал об окончании получения данных модели в формате JSON
<a href="#">GetContentAsJsonFailed</a>	Сигнал об ошибке получения данных модели в формате JSON
<a href="#">SetContentAsJsonFinished</a>	Сигнал об окончании установки данных в формате JSON в модель
<a href="#">SetContentAsJsonFailed</a>	Сигнал об ошибке установки данных в формате JSON в модель

## 1.1.4.2.4.9.1.1.1. RowsRemoved

Удаление строки.

### Параметры

Параметр	Тип	Описание
first	int4	Индекс первой удаленной строки
last	int4	Индекс последней удаленной строки

## 1.1.4.2.4.9.1.1.2. RowsInserted

Вставка строки.

### Параметры

Параметр	Тип	Описание
first	int4	Индекс первой вставленной строки
last	int4	Индекс последней вставленной строки

## 1.1.4.2.4.9.1.1.3. ReadFinished

Завершение чтения данных моделью.

### Параметры

Параметр	Тип	Описание
rowsRead	uint8	Количество прочитанных строк
elapsedSec	float	Общее время чтения (в секундах)

## 1.1.4.2.4.9.1.1.4. DataChanged

Изменение содержимого в таблице модели данных.

### Параметры

Параметр	Тип	Описание
top	int4	Порядковый номер первой строки. в которой были изменения. Нумерация начинается с 0
left	int4	Порядковый номер столбца в первой строке, в котором были изменения. Нумерация начинается с 0
bottom	int4	Порядковый номер последней строки. в которой были изменения. Нумерация начинается с 0
right	int4	Порядковый номер столбца в последней строке, в котором были изменения. Нумерация начинается с 0

## 1.1.4.2.4.9.1.1.5. GetContentAsJsonFinished

Сигнал об окончании получения данных модели в формате JSON.

## 1.1.4.2.4.9.1.1.6. GetContentAsJsonFailed

Сигнал об ошибке получения данных модели в формате JSON.

## **1.1.4.2.4.9.1.1.7. SetContentAsJsonFinished**

Сигнал об окончании установки данных в формате JSON в модель.



## 1.1.4.2.4.9.1.1.8. SetContentAsJsonFailed

Сигнал об ошибке установки данных в формате JSON в модель.

## 1.1.4.2.4.9.1.2. Функции

Компонент	Описание
<a href="#">Resert</a>	Удаляет все строки и столбцы из таблицы модели данных
<a href="#">Clear</a>	Удаляет все строки из таблицы модели данных
<a href="#">CancelReadAsync</a>	Прерывает чтение данных из источника
<a href="#">BeginReadAsync</a>	Заполняет таблицу модели данными из источника
<a href="#">RemoveRows</a>	Удаляет указанный диапазон строк
<a href="#">RemoveRow</a>	Удаляет указанную строку из таблицы модели данных
<a href="#">InsertRows</a>	Добавляет несколько новых строк после указанной строки в таблице модели данных
<a href="#">InsertRow</a>	Добавляет новую строку после указанной строки в таблице модели данных
<a href="#">AddRow</a>	Добавляет новую строку в конец таблицы модели данных
<a href="#">AddColumn</a>	Добавляет новый столбец в таблицу модели данных
<a href="#">SetRowsDataAsJson</a>	Задаёт данные указанного диапазона строк в формате JSON
<a href="#">SetRowDataAsJson</a>	Задаёт данные в формате JSON
<a href="#">SetDataAsJson</a>	Задаёт данные в формате JSON
<a href="#">SetCellData</a>	Задаёт значение указанной ячейке
<a href="#">GetRowCount</a>	Возвращает количество строк, существующих в таблице на момент вызова функции
<a href="#">GetColumnCount</a>	Возвращает количество столбцов, существующих в таблице на момент вызова функции

<a href="#">GetRowDataAsJson</a>	Возвращает JSON-объект с данными указанной строки
<a href="#">GetRowsDataAsJson</a>	Возвращает JSON-объект с данными указанного диапазона строк
<a href="#">GetCellData</a>	Возвращает значение указанной ячейки
<a href="#">GetDataAsJson</a>	Получить данные в формате JSON
<a href="#">GetColumnSpecAsJson</a>	Получить спецификацию столбцов в формате JSON в модель
<a href="#">SetContentAsJson</a>	Установить данные в формате JSON в модель
<a href="#">GetContentAsJson</a>	Получить данные модели в формате JSON

## 1.1.4.2.4.9.1.2.1. Reset

Удаляет все строки и столбцы из таблицы модели данных.



```
void Reset()
```

## 1.1.4.2.4.9.1.2.2. Clear

Удаляет все строки из таблицы модели данных.



void Clear()

## 1.1.4.2.4.9.1.2.3. CancelReadAsync

Прерывает чтение данных из источника.



```
void CancelReadAsync()
```

## 1.1.4.2.4.9.1.2.4. BeginReadAsync

Заполняет таблицу модели данными из источника.



```
void BeginReadAsync()
```

## 1.1.4.2.4.9.1.2.5. RemoveRows

Удаляет указанный диапазон строк.



```
void RemoveRows(int4 Index1, int4 Index2)
```

### Параметры

Параметр	Тип	Описание
Index1	int4	Индекс первой строки из списка удаляемых
Index2	int4	Индекс последней строки из списка удаляемых



## 1.1.4.2.4.9.1.2.6. RemoveRow

Удаляет указанную строку из таблицы модели данных.



```
void RemoveRow(int4 Row)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0

## 1.1.4.2.4.9.1.2.7. InsertRows

Добавляет несколько новых строк после указанной строки в таблице модели данных.



```
void InsertRows(int4 Row, int4 Count)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, после которой будут вставлены новые строки. Нумерация начинается с 0
Count	int4	Количество добавляемых строк

## 1.1.4.2.4.9.1.2.8. InsertRow

Добавляет новую строку после указанной строки в таблице модели данных.



```
void InsertRow(int4 Row)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0

## 1.1.4.2.4.9.1.2.9. AddRow

Добавляет новую строку в конец таблицы модели данных.



```
void AddRow()
```

## 1.1.4.2.4.9.1.2.10. AddColumn

Добавляет новый столбец в таблицу модели данных.



```
void AddColumn(string ColumnName, string Title, string Description,  
string EUnit, string Type, bool ReadOnly, bool Emptiable)
```

### Параметры

Параметр	Тип	Описание
ColumnName	string	идентификатор столбца, возможные значения: <ul style="list-style-type: none"><li>&gt; source - источник события</li><li>&gt; time - время генерации уведомления о событии</li><li>&gt; message - сообщение<ul style="list-style-type: none"><li>&gt; source - источник события</li><li>&gt; time - время генерации уведомления о событии</li><li>&gt; message - сообщение</li><li>&gt; severity - уровень важности события</li><li>&gt; condition_name - имя условия генерации события</li><li>&gt; subcondition_name - имя подусловия генерации события</li><li>&gt; quality - текущее качество сигнала, изменение которого привело к генерации события</li><li>&gt; active_time - время перехода состояния события в активное</li><li>&gt; actor_id - имя пользователя, выполнившего квитирование сообщения о событии</li><li>&gt; ack - признак квитирования события</li><li>&gt; active - признак активности подусловия, по которому было сгенерировано событие</li></ul></li></ul>

		<ul style="list-style-type: none"> <li>› cookie - специальный идентификатор события, который имеет служебное назначение. Необходим для сведения квитанций, деактиваций с событиями</li> <li>› ack_time - время квитирования</li> <li>› ack_comment - комментарий квитирования</li> <li>› ack_required - требование квитирования</li> <li>› value - значение сигнала, изменение которого привело к генерации события</li> <li>› sound - звуковой файл, исполняемый при выполнении подусловия генерации сообщения о событии</li> <li>› area_path - относительный тег объекта</li> <li>› object_id - идентификатор объекта на сервере, по которому было сгенерировано событие</li> <li>› deactive_time - время, когда событие перешло из активного состояния в неактивное</li> </ul>
Title	string	заголовок столбца
Description	string	описание столбца
EUnit	string	единицы измерения значения в столбце
Type	string	тип данных значения в столбце
ReadOnly	bool	модификатор доступа "только чтение"
Emptiable	bool	разрешение/запрет ячейкам в столбце не содержать значений

## 1.1.4.2.4.9.1.2.11. SetRowsDataAsJson

Задает данные указанного диапазона строк в формате JSON.



```
void SetRowsDataAsJson(int4 Row_1, int4 Row_2, string Data)
```

### Параметры

Параметр	Тип	Описание
Row_1	int4	Порядковый номер начала диапазона строк
Row_2	int4	Порядковый номер окончания диапазона строк
Data	string	Данные, добавляемые в строку

## 1.1.4.2.4.9.1.2.12. SetRowDataAsJson

Задает данные в формате JSON.



```
void SetRowDataAsJson(int4 Row, string Data)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0
Data	string	Данные, добавляемые в строку



## 1.1.4.2.4.9.1.2.13. SetDataAsJson

Задает данные в формате JSON.



```
void SetDataAsJson(string Data)
```

### Параметры

Параметр	Тип	Описание
Data	string	Объект в формате JSON

## 1.1.4.2.4.9.1.2.14. SetCellData

Задает значение указанной ячейке.



```
void SetCellData(int4 Row, int4 Column, variant Value)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0
Column	int4	Порядковый номер столбца. Нумерация начинается с 0
Value	variant	Значение, добавляемое в ячейку

## 1.1.4.2.4.9.1.2.15. GetRowCount

Возвращает количество строк, существующих в таблице на момент вызова функции.



```
int4 GetRowCount()
```

## 1.1.4.2.4.9.1.2.16. GetColumnCount

Возвращает количество столбцов, существующих в таблице на момент вызова функции.



```
string GetColumnCount()
```

## 1.1.4.2.4.9.1.2.17. GetRowDataAsJson

Возвращает JSON-объект с данными указанной строки.



string GetRowDataAsJson(int4 Row)

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0

## 1.1.4.2.4.9.1.2.18. GetRowsDataAsJson

Возвращает JSON-объект с данными указанного диапазона строк.



```
string GetRowsDataAsJson(int4 Row_1, Int4 Row_2)
```

### Параметры

Параметр	Тип	Описание
Row_1	int4	Порядковый номер начала диапазона строк
Row_2	int4	Порядковый номер окончания диапазона строк

## 1.1.4.2.4.9.1.2.19. GetCellData

Возвращает значение указанной ячейки.



variant GetCellData(int4 Row, int4 Column)

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0
Column	int4	Порядковый номер столбца. Нумерация начинается с 0

## 1.1.4.2.4.9.1.2.20. GetDataAsJson

Получить данные в формате JSON.



```
string GetDataAsJson()
```



## 1.1.4.2.4.9.1.2.21. GetColumnSpecAsJson

Получить спецификацию столбцов в формате JSON в модель.



```
string GetColumnSpecAsJson()
```

## 1.1.4.2.4.9.1.2.22. SetContentAsJson

Установить данные в формате JSON в модель.



```
void SetContentAsJson(string _1)
```

### Параметры

Параметр	Тип	Описание
string_1	string	Объект в формате JSON

## 1.1.4.2.4.9.1.2.23. GetContentAsJson

Получить данные модели в формате JSON.



```
void GetContentAsJson()
```

## 1.1.4.2.4.9.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Максимальное количество рядов данных</a>	Максимальное число строк таблицы
<a href="#">Заморозка</a>	Запрещает/разрешает обновление таблицы данными, поступившими от источника или заданными в скрипте
<a href="#">Вариант обработки событий от источника данных</a>	Вариант обработки событий от источника данных

## 1.1.4.2.4.9.1.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.9.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.9.1.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.9.1.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.9.1.3.5. Заморозка

Запрещает/разрешает обновление таблицы данными, поступившими от источника или заданными в скрипте.



bool Frozen

## 1.1.4.2.4.9.1.3.6. Максимальное количество рядов данных

Максимальное число строк таблицы.



int4 MaxRowCount

## 1.1.4.2.4.9.1.3.7. Вариант обработки событий от источника данных

Вариант обработки событий от источника данных.



int4 OptOfProcessingEvents

## 1.1.4.2.4.9.2. Таблица

Компонент позволяет графически представить данные, хранящиеся в таблице модели данных.

## 1.1.4.2.4.9.2.1. События

Событие	Описание
<a href="#">SelectionChanged</a>	Изменение выделения строк
<a href="#">CurrentChanged</a>	Выделение ячейки
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области строки или столбца
<a href="#">MouseDown</a>	Нажатие кнопки мыши в области строки или столбца
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">GetAvailablePrintersFinished</a>	Сигнал об окончании получения списка доступных принтеров
<a href="#">GetAvailablePrintersFailed</a>	Сигнал об ошибке получения списка доступных принтеров
<a href="#">ScrollMove</a>	Сигнал о перемещении скролл бара
<a href="#">KeyRelease</a>	Отпускание определённой клавиши или сочетания клавиш на форме
<a href="#">KeyPress</a>	Нажатие определённой клавиши или сочетания клавиш на форме
<a href="#">PreparePrintStart</a>	Сигнал о начале получения списка доступных принтеров
<a href="#">PreparePrintFinished</a>	Сигнал об окончании получения списка доступных принтеров
<a href="#">PrintProcess</a>	Сигнал о начале процесса печати
<a href="#">PrintProcessFinished</a>	Сигнал об окончании процесса печати
<a href="#">PrintProcessAborted</a>	Сигнал о прерывании процесса печати

## 1.1.4.2.4.9.2.1.1. SelectionChanged

Изменение выделения строк.

## 1.1.4.2.4.9.2.1.2. CurrentChanged

Выделение ячейки.

### Параметры

Параметр	Тип	Описание
row	int4	Порядковый номер строки. Нумерация начинается с 0
column	int4	Порядковый номер столбца. Нумерация начинается с 0

## 1.1.4.2.4.9.2.1.3. MouseRelease

Нажатая кнопка мыши отпущена в области строки или столбца.

### Параметры

Параметр	Тип	Описание
row	int4	Порядковый номер строки. Нумерация начинается с 0
column	int4	Порядковый номер столбца. Нумерация начинается с 0



## 1.1.4.2.4.9.2.1.4. MousePress

Нажатие кнопки мыши в области строки или столбца.

### Параметры

Параметр	Тип	Описание
row	int4	Порядковый номер строки. Нумерация начинается с 0
column	int4	Порядковый номер столбца. Нумерация начинается с 0

## 1.1.4.2.4.9.2.1.5. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

## 1.1.4.2.4.9.2.1.6. GetAvailablePrintersFinished

Сигнал об окончании получения списка доступных принтеров.

## 1.1.4.2.4.9.2.1.7. GetAvailablePrintersFailed

Сигнал об ошибке получения списка доступных принтеров.

## 1.1.4.2.4.9.2.1.8. ScrollMove

Сигнал о перемещении скролл бара.

## 1.1.4.2.4.9.2.1.9. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
Text_1.Text = event.Text;
```

## 1.1.4.2.4.9.2.1.10. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпуске пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпуске клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```





//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## **1.1.4.2.4.9.2.1.11. PreparePrintStart**

Сигнал о начале получения списка доступных принтеров.

## **1.1.4.2.4.9.2.1.12. PreparePrintFinished**

Сигнал об окончании получения списка доступных принтеров.

## **1.1.4.2.4.9.2.1.13. PrintProcess**

Сигнал о начале процесса печати.

## 1.1.4.2.4.9.2.1.14. PrintProcessFinished

Сигнал об окончании процесса печати.

## 1.1.4.2.4.9.2.1.15. **PrintProcessAborted**

Сигнал о прерывании процесса печати.

## 1.1.4.2.4.9.2.2. Функции

Компонент	Описание
<a href="#">Save</a>	Сохраняет таблицу в файл
<a href="#">PrintDialog</a>	Открывает диалог печати
<a href="#">Print</a>	Запускает печать таблицы
<a href="#">SetPageNumberFont</a>	Задаёт шрифт номера страницы
<a href="#">GetPageNumberFont</a>	Возвращает шрифт номера страницы
<a href="#">SetPageFooterText</a>	Записать текст в нижний колонтитул
<a href="#">GetPageFooterText</a>	Возвращает текст нижнего колонтитула
<a href="#">SetPageFooterFont</a>	Задаёт шрифт текста нижнего колонтитула
<a href="#">GetPageFooterFont</a>	Возвращает шрифт текста нижнего колонтитула
<a href="#">SetPageFooterTextAlignment</a>	Задаёт выравнивание текста нижнего колонтитула
<a href="#">GetPageFooterTextAlignment</a>	Возвращает выравнивание текста нижнего колонтитула
<a href="#">GetPageHeaderText</a>	Возвращает текст верхнего колонтитула
<a href="#">SetPageHeaderText</a>	Записывает текст в верхний колонтитул
<a href="#">SetPageHeaderFont</a>	Задаёт шрифт текста верхнего колонтитула
<a href="#">GetPageHeaderFont</a>	Возвращает шрифт текста верхнего колонтитула
<a href="#">SetPageHeaderTextAlignment</a>	Задаёт выравнивание текста верхнего колонтитула
<a href="#">GetPageHeaderTextAlignment</a>	Возвращает выравнивание текста верхнего колонтитула
<a href="#">ScrollTo</a>	Прокручивает таблицу к указанной ячейке
<a href="#">ScrollToTop</a>	Прокручивает таблицу к началу
<a href="#">ScrollToBottom</a>	Прокручивает таблицу к концу
<a href="#">MapRowToModel</a>	Возвращает соответствующий индекс строки из таблицы в модели данных

<a href="#">MapRowFromModel</a>	Возвращает соответствующий индекс строки из модели данных в таблице
<a href="#">MapColumnToModel</a>	Возвращает соответствующий индекс столбца из таблицы в модели данных
<a href="#">MapColumnFromModel</a>	Возвращает соответствующий индекс столбца из модели данных в таблице
<a href="#">ApplySort</a>	Применяет сортировку, настроенную в столбцах
<a href="#">MoveColumn</a>	Перемещает столбец из одной позиции в указанную позицию
<a href="#">IsRowSelected</a>	Возвращает true, если указанная строка выделена
<a href="#">RemoveRow</a>	Удаляет указанную строку из таблицы
<a href="#">InsertRow</a>	Добавляет новую строку после указанной строки в таблице
<a href="#">AddRow</a>	Добавляет новую строку в конец таблицы
<a href="#">SetSelectedRows</a>	Задать JSON-объект с номерами выделенных строк
<a href="#">GetSelectedRows</a>	Возвращает JSON-объект с номерами выделенных строк
<a href="#">GetCellData</a>	Возвращает значение указанной ячейки
<a href="#">GetRowCount</a>	Возвращает количество строк, существующих в таблице на момент вызова функции
<a href="#">GetRowsDataAsJson</a>	Получить JSON-объект с данными указанного диапазона строк
<a href="#">GetRowDataAsJson</a>	Получить JSON-объект с данными указанной строки
<a href="#">GetColumnCount</a>	Возвращает количество столбцов, существующих в таблице на момент вызова функции
<a href="#">SetDefaultPrinter</a>	Задать используемый для печати принтер
<a href="#">GetDefaultPrinter</a>	Получить используемый для печати принтер



<a href="#">GetAvailablePrinters</a>	Получить список доступных принтеров
<a href="#">SetPageOrientation</a>	Задать ориентацию страницы
<a href="#">GetPageOrientation</a>	Получить ориентацию страницы
<a href="#">SetCurrentCell</a>	Задать текущую ячейку
<a href="#">GetColumnWidth</a>	Получить ширину столбца
<a href="#">GetRowTop</a>	Координаты верхней границы строки по оси Y в пикселях
<a href="#">GetColumnLeft</a>	Координаты левой границы столбца по оси X в пикселях
<a href="#">GetSelectedCells</a>	Получить JSON-объект со списком выделенных ячеек

## 1.1.4.2.4.9.2.2.1. Save

Сохраняет таблицу в файл.



```
void Save(string FileName, string Format)
```

### Параметры

Параметр	Тип	Описание
FileName	string	Имя сохраняемого файла
Format	string	Формат сохраняемого файла

## 1.1.4.2.4.9.2.2.2. PrintDialog

Открывает диалог печати.



```
void PrintDialog()
```

## 1.1.4.2.4.9.2.2.3. Print

Запускает печать таблицы.



void Print()

## 1.1.4.2.4.9.2.2.4. SetPageNumberFont

Задает шрифт номера страницы. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
void SetPageNumberFont(string Font)
```

### Параметры

Параметр	Тип	Описание
Font	string	Наименование шрифта

## 1.1.4.2.4.9.2.2.5. GetPageNumberFont

Возвращает шрифт номера страницы. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
string GetPageNumberFont()
```

## 1.1.4.2.4.9.2.2.6. SetPageFooterText

Записать текст в нижний колонтитул. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
string SetPageFooterText(string Text)
```

### Параметры

Параметр	Тип	Описание
Text	string	Текст для нижнего колонтитула

## 1.1.4.2.4.9.2.2.7. GetPageFooterText

Возвращает текст нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
string GetPageFooterText()
```



## 1.1.4.2.4.9.2.2.8. SetPageFooterFont

Задает шрифт текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
void SetPageFooterFont(string Font)
```

### Параметры

Параметр	Тип	Описание
Font	string	Наименование шрифта

## 1.1.4.2.4.9.2.2.9. GetPageFooterFont

Возвращает шрифт текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
string GetPageFooterFont()
```

## 1.1.4.2.4.9.2.2.10. SetPageFooterTextAlignment

Задает выравнивание текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
void SetPageFooterTextAlignment(string Alignment)
```

### Параметры

Параметр	Тип	Описание
Alignment	string	Возможные значения: <ul style="list-style-type: none"><li>&gt; 1 – по левому краю</li><li>&gt; 2 – по правому краю</li><li>&gt; 4 – по центру</li></ul>

## 1.1.4.2.4.9.2.2.11. GetPageFooterTextAlignment

Возвращает выравнивание текста нижнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



int4 GetPageFooterTextAlignment()

## 1.1.4.2.4.9.2.2.12. GetPageHeaderText

Возвращает текст верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
string GetPageHeaderText()
```

## 1.1.4.2.4.9.2.2.13. SetPageHeaderText

Записывает текст в верхний колонтитул. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
void SetPageHeaderText(string Text)
```

### Параметры

Параметр	Тип	Описание
Text	string	Текст

## 1.1.4.2.4.9.2.2.14. SetPageHeaderFont

Задает шрифт текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



int4 SetPageHeaderFont(string Font)

### Параметры

Параметр	Тип	Описание
Font	string	Наименование шрифта

## 1.1.4.2.4.9.2.2.15. GetPageHeaderFont

Возвращает шрифт текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
string GetPageHeaderFont()
```



## 1.1.4.2.4.9.2.2.16. SetPageHeaderTextAlignment

Задает выравнивание текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



```
void SetPageHeaderTextAlignment(string Alignment)
```

### Параметры

Параметр	Тип	Описание
Alignment	string	Возможные значения: <ul style="list-style-type: none"><li>&gt; 1 – по левому краю</li><li>&gt; 2 – по правому краю</li><li>&gt; 4 – по центру</li></ul>

## 1.1.4.2.4.9.2.2.17. GetPageHeaderTextAlignment

Прокручивает таблицу к началу.

Возвращает выравнивание текста верхнего колонтитула. Используется для дополнительного оформления таблицы перед экспортом в файл .pdf.



int4 GetPageHeaderTextAlignment()

## 1.1.4.2.4.9.2.2.18. ScrollTo

Прокручивает таблицу к указанной ячейке.



```
void ScrollTo(int4 Row, int4 Column, uint1 Hint)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0
Column	int4	Порядковый номер столбца. Нумерация начинается с 0
Hint	uint1	Месторасположение ячейки в видимой области после смещения. Возможные значения: <ul style="list-style-type: none"><li>› 0 – ячейка должна быть видима</li><li>› 1 – ячейка расположена сверху</li><li>› 2 – ячейка расположена снизу</li><li>› 3 – ячейка расположена по центру</li></ul>

## 1.1.4.2.4.9.2.2.19. ScrollToTop

Прокручивает таблицу к началу.



```
void ScrollToTop()
```

## 1.1.4.2.4.9.2.2.20. ScrollToBottom

Прокручивает таблицу к концу.



```
void ScrollToBottom()
```

## 1.1.4.2.4.9.2.2.21. MapRowToModel

Возвращает соответствующий индекс строки из таблицы в модели данных.



int4 MapRowToModel(int4 Row)

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.22. MapRowFromModel

Возвращает соответствующий индекс строки из модели данных в таблице.



int4 MapRowFromModel(int4 Row)

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.23. MapColumnToModel

Возвращает соответствующий индекс столбца из таблицы в модели данных.



int4 MapColumnToModel(int4 Column)

### Параметры

Параметр	Тип	Описание
Column	int4	Порядковый номер столбца. Нумерация начинается с 0



## 1.1.4.2.4.9.2.2.24. MapColumnFromModel

Возвращает соответствующий индекс столбца из модели данных в таблице.



int4 MapColumnFromModel(int4 Column)

### Параметры

Параметр	Тип	Описание
Column	int4	Порядковый номер столбца. Нумерация начинается с 0

## 1.1.4.2.4.9.2.25. ApplySort

Применяет сортировку, настроенную в столбцах.



void ApplySort()

## 1.1.4.2.4.9.2.26. MoveColumn

Перемещает столбец из одной позиции в указанную позицию.



```
void MoveColumn(int4 Column, int4 Position)
```

### Параметры

Параметр	Тип	Описание
Column	int4	Порядковый номер столбца. Нумерация начинается с 0
Position	int4	Номер позиции, в которую будет перемещен столбец

## 1.1.4.2.4.9.2.2.27. IsRowSelected

Возвращает true, если указанная строка выделена.



bool IsRowSelected(Row)

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.28. RemoveRow

Удаляет указанную строку из таблицы.



```
void RemoveRow(int4 Row)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.29. InsertRow

Добавляет новую строку после указанной строки в таблице.



```
void InsertRow(int4 Row)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, после которой будет вставлена новая строка. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.30. AddRow

Добавляет новую строку в конец таблицы.



```
void AddRow()
```

## 1.1.4.2.4.9.2.2.31. SetSelectedRows

Задать JSON-объект с номерами выделенных строк.



```
void SetSelectedRows(string Rows)
```

### Параметры

Параметр	Тип	Описание
Row	string	JSON-объект



## 1.1.4.2.4.9.2.2.32. GetSelectedRows

Возвращает JSON-объект с номерами выделенных строк.



```
string GetSelectedRows()
```

## 1.1.4.2.4.9.2.2.33. GetCellData

Возвращает значение указанной ячейки.



```
string GetCellData(int4 Row, int4 Column)
```

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0
Column	int4	Порядковый номер столбца. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.34. GetRowCount

Возвращает количество строк, существующих в таблице на момент вызова функции.



```
int4 GetRowCount()
```

## 1.1.4.2.4.9.2.2.35. GetRowDataAsJson

Получить JSON-объект с данными указанной строки.



string GetRowDataAsJson(int4 Row)

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки. Нумерация начинается с 0

## 1.1.4.2.4.9.2.2.36. GetRowsDataAsJson

Получить JSON-объект с данными указанного диапазона строк.



```
string GetRowDataAsJson(int4 Row_1, int4 Row_2, bool _3)
```

### Параметры

Параметр	Тип	Описание
Row_1	int4	Порядковый номер начала диапазона строк
Row_2	int4	Порядковый номер окончания диапазона строк
_3	bool	

## 1.1.4.2.4.9.2.2.37. GetColumnCount

Возвращает количество столбцов, существующих в таблице на момент вызова функции.



```
int4 GetColumnCount()
```

## 1.1.4.2.4.9.2.2.38. SetDefaultPrinter

Задать используемый для печати принтер.



```
void SetDefaultPrinter(string _1)
```

### Параметры

Параметр	Тип	Описание
_1	string	Имя принтера

## 1.1.4.2.4.9.2.2.39. GetDefaultPrinter

Получить используемый для печати принтер.



```
string GetDefaultPrinter()
```



## 1.1.4.2.4.9.2.2.40. GetAvailablePrinters

Получить список доступных принтеров.



```
string GetAvailablePrinters()
```

## 1.1.4.2.4.9.2.2.41. SetPageOrientation

Задать ориентацию страницы.



```
void SetPageOrientation(uint1 _1)
```

### Параметры

Параметр	Тип	Описание
_1	uint1	

## 1.1.4.2.4.9.2.2.42. GetPageOrientation

Получить ориентацию страницы.



uint1 GetPageOrientation()

## 1.1.4.2.4.9.2.2.43. SetCurrentCell

Задать текущую ячейку.



```
void SetCurrentCell(int4 _1, int4 _2)
```

### Параметры

Параметр	Тип	Описание
int4 _1	int4	Столбец
int4 _2	int4	Строка

## 1.1.4.2.4.9.2.2.44. GetColumnWidth

Получить ширину столбца.



int4 GetColumnWidth(int4 \_1)

### Параметры

Параметр	Тип	Описание
_1	int4	Номер столбца

## 1.1.4.2.4.9.2.2.45. GetRowTop

Координаты верхней границы строки по оси Y в пикселях.



int4 GetRowTop(int4 \_1)

### Параметры

Параметр	Тип	Описание
_1	int4	Значение координаты верхней границы строки по оси Y

## 1.1.4.2.4.9.2.2.46. GetColumnLeft

Координаты левой границы столбца по оси X в пикселях.



int4 GetColumnLeft(int4 \_1)

### Параметры

Параметр	Тип	Описание
_1	int4	Значение координаты левой границы столбца по оси X

## 1.1.4.2.4.9.2.2.47. GetSelectedCells

Получить JSON-объект со списком выделенных ячеек.



string GetSelectedCells()



## 1.1.4.2.4.9.2.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Следить за новой строкой</a>	Включение/отключение постоянного отображения в поле видимости последней добавленной строки
<a href="#">Режим выделения</a>	Комбинация клавиш для выделения нескольких ячеек
<a href="#">Поведение выделения</a>	Количество выделенных ячеек при нажатии на одну из них
<a href="#">Порядок вывода строк</a>	Расположение новых строк в таблице
<a href="#">Высота строки</a>	Высота строки таблицы (в пикселях)
<a href="#">Видимость заголовка</a>	Скрывает/отображает заголовок таблицы
<a href="#">Высота заголовка</a>	Высота заголовка таблицы (в пикселях)
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Показать оверлей выделения</a>	Показывать оверлей выделения

<a href="#">Цвет фона</a>	Цвет фона
<a href="#">Цвет рамки</a>	Цвет рамки
<a href="#">Цвет сетки</a>	Цвет сетки
<a href="#">Отложенная сортировка</a>	Использовать отложенную сортировку
<a href="#">Режим горизонтальной прокрутки</a>	Режим горизонтальной прокрутки
<a href="#">Режим вертикальной прокрутки</a>	Режим вертикальной прокрутки
<a href="#">Политика горизонтальной полосы прокрутки</a>	Политика горизонтальной полосы прокрутки
<a href="#">Политика вертикальной полосы прокрутки</a>	Политика вертикальной полосы прокрутки
<a href="#">Модель данных</a>	Ссылка на модель данных
<a href="#">Режим изменения высоты строк</a>	Режим изменения высоты строк

## 1.1.4.2.4.9.2.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.9.2.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.9.2.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.9.2.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.9.2.3.5. Модель данных

Ссылка на модель данных.

## 1.1.4.2.4.9.2.3.6. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```



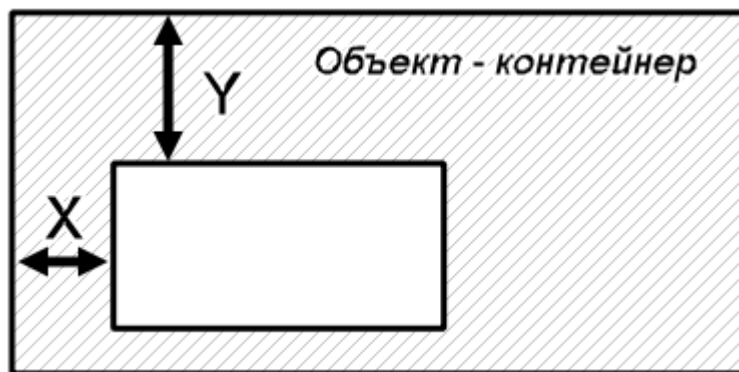
## 1.1.4.2.4.9.2.3.7. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

## 1.1.4.2.4.9.2.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.9.2.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.9.2.3.10. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.9.2.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



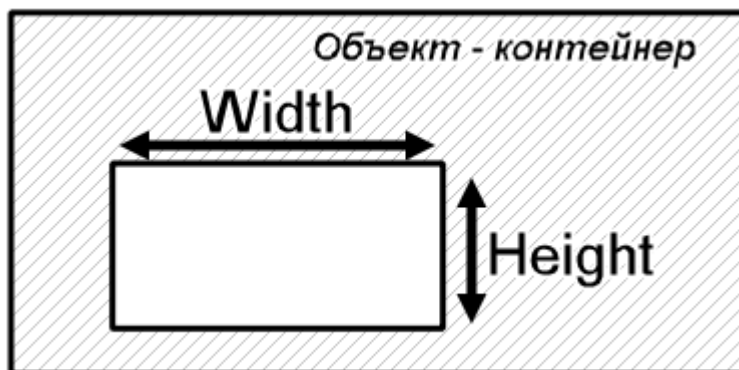
```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.9.2.3.12. Ширина

Ширина объекта в пикселях.



float8 Width

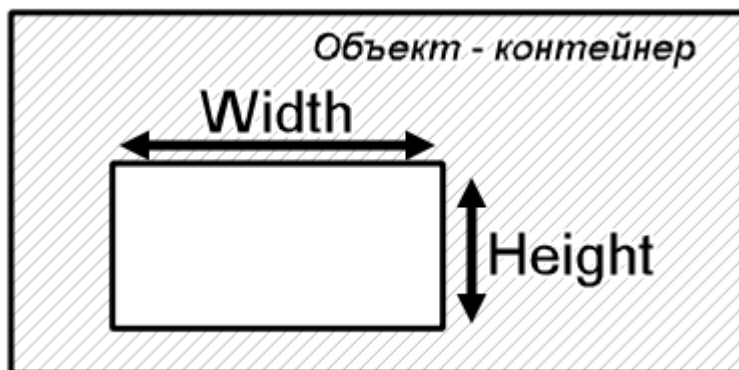


## 1.1.4.2.4.9.2.3.13. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.9.2.3.14. Цвет фона

Цвет фона.



uint4 BackgroundColor



## 1.1.4.2.4.9.2.3.15. Цвет рамки

Цвет рамки.



uint4 FrameColor

## 1.1.4.2.4.9.2.3.16. Цвет сетки

Цвет сетки.



uint4 GridColor

## 1.1.4.2.4.9.2.3.17. Высота заголовка

Высота заголовка таблицы (в пикселях). Может превышать указанное значение, если разрешен перенос слов.



int4 HeaderHeight

## 1.1.4.2.4.9.2.3.18. Видимость заголовка

Скрывает/отображает заголовок таблицы.



bool HeaderVisible

### Значение

Значение	Описание
true	Заголовок отображается
false	Заголовок скрыт

## 1.1.4.2.4.9.2.3.19. Высота строки

Высота строки таблицы (в пикселях). Может превышать указанное значение, если разрешен перенос слов.



int4 RowHeight

## 1.1.4.2.4.9.2.3.20. Порядок вывода строк

Расположение новых строк в таблице.



uint1 RowOrder

### Значение

Значение	Описание
0	Каждая новая строка добавляется в конец таблицы
1	Каждая новая строка добавляется в начало таблицы

## 1.1.4.2.4.9.2.3.21. Поведение выделения

Количество выделенных ячеек при нажатии на одну из них.



uint1 SelectionBehavior

### Значение

Значение	Описание
0	Выделяется только ячейка
1	Выделяется строка
3	Выделяется столбец

## 1.1.4.2.4.9.2.3.22. Режим выделения

Комбинация клавиш для выделения нескольких ячеек.



uint1 SelectionMode

### Значение

Значение	Описание
0	Выделение ячеек отключено
1	Возможность единовременного выделения только одной ячейки кликом мыши
2	Возможность одновременного выделения нескольких ячеек и индивидуального снятия выделения
3	Возможность выделения смежных элементов кликом мыши и клавишей Ctrl
4	Возможность выделения смежных элементов кликом мыши и клавишей Shift



## 1.1.4.2.4.9.2.3.23. выделения

## Показывать оверлей

Показывать оверлей выделения.



bool SelectionOverlayEnabled

## 1.1.4.2.4.9.2.3.24. Следить за новой строкой

Включение/отключение постоянного отображения в поле видимости последней добавленной строки.



bool LatestRowTracking

### Значение

Значение	Описание
0	Режим слежения включен
1	Режим слежения выключен

## 1.1.4.2.4.9.2.3.25. Отложенная сортировка

Использовать отложенную сортировку.



bool DelayedSortingEnabled

# 1.1.4.2.4.9.2.3.26. Режим горизонтальной прокрутки

Режим горизонтальной прокрутки.



uint1 HorizontalScrollMode

# 1.1.4.2.4.9.2.3.27. Режим вертикальной прокрутки

Режим вертикальной прокрутки.



uint1 VerticalScrollMode

## 1.1.4.2.4.9.2.3.28. Политика горизонтальной полосы прокрутки

Политика горизонтальной полосы прокрутки.



uint1 HorizontalScrollBarPolicy

# 1.1.4.2.4.9.2.3.29. Политика вертикальной полосы прокрутки

Политика вертикальной полосы прокрутки.



uint1 VerticalScrollBarPolicy

## 1.1.4.2.4.9.2.3.30. Режим изменения высоты строк

Режим изменения высоты строк.



uint4 RowHeightResizeMode



### 1.1.4.2.4.9.3. Таблица: столбец

Компонент позволяет сформировать таблицу для отображения данных из модели.

## 1.1.4.2.4.9.3.1. События

Событие	Описание
<a href="#">WidthChanged</a>	Изменение ширины столбца
<a href="#">MouseRightTitleClick</a>	Щелчок правой кнопкой мыши по заголовку столбца
<a href="#">MouseTitleClick</a>	Щелчок левой кнопкой мыши по заголовку столбца
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по ячейке столбца
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по ячейке
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по ячейке

## 1.1.4.2.4.9.3.1.1. WidthChanged

Изменение ширины столбца.

## 1.1.4.2.4.9.3.1.2. MouseRightTitleClick

Щелчок правой кнопкой мыши по заголовку столбца.

## 1.1.4.2.4.9.3.1.3. MouseTitleClick

Щелчок левой кнопкой мыши по заголовку столбца.

## 1.1.4.2.4.9.3.1.4. MouseRightClick

Щелчок правой кнопкой мыши по ячейке столбца.

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, по ячейке которой был выполнен щелчок

## 1.1.4.2.4.9.3.1.5. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по ячейке.

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, по ячейке которой был выполнен щелчок

## 1.1.4.2.4.9.3.1.6. MouseClick

Щелчок левой кнопкой мыши по ячейке.

### Параметры

Параметр	Тип	Описание
Row	int4	Порядковый номер строки, по ячейке которой был выполнен щелчок



## 1.1.4.2.4.9.3.2. Функции

Компонент	Описание
<a href="#">GetCellData</a>	Возвращает значение указанной ячейки
<a href="#">GetIndex</a>	Возвращает индекс столбца в таблице
<a href="#">GetTitleTextSizePx</a>	Получить высоту и длину текста заголовка для заданного шрифта в пикселях
<a href="#">GetMaxTextSizePx</a>	Получить максимальную высоту и длину текста в ячейках колонки для заданного шрифта в пикселях

## 1.1.4.2.4.9.3.2.1. GetCellData

Возвращает значение указанной ячейки.



```
string GetCellData(int4 Cell)
```

### Параметры

Параметр	Тип	Описание
Cell	int4	Порядковый номер ячейки

## 1.1.4.2.4.9.3.2.2. GetIndex

Возвращает индекс столбца в таблице.



```
int4 GetIndex()
```

## 1.1.4.2.4.9.3.2.3. GetTitleTextSizePx

Получить высоту и длину текста заголовка для заданного шрифта в пикселях.



```
string GetTitleTextSizePx(string _1)
```

### Параметры

Параметр	Тип	Описание
_1	string	Текст заголовка

## 1.1.4.2.4.9.3.2.4. GetMaxTextSizePx

Получить максимальную высоту и длину текста в ячейках колонки для заданного шрифта в пикселях.



```
string GetMaxTextSizePx(string _1)
```

### Параметры

Параметр	Тип	Описание
_1	string	

## 1.1.4.2.4.9.3.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Разрешить перенос слов</a>	Разрешение/запрет переноса слов в столбце
<a href="#">Приоритет сортировки</a>	Порядок применения сортировки
<a href="#">Порядок сортировки</a>	Вид сортировки данных в столбце
<a href="#">Сортировать</a>	Разрешение/запрет сортировки данных в столбце
<a href="#">Фильтр в JSON-ajhvfnt</a>	Условия фильтрации отображаемых данных в таблице
<a href="#">Временной формат</a>	Формат выводимой даты полученных данных
<a href="#">Числовой формат</a>	Формат поступающих числовых данных
<a href="#">Скрывать</a>	Скрывает/отображает столбец
<a href="#">Может не иметь значения</a>	Разрешает/запрещает ячейкам столбца не содержать значений
<a href="#">Режим изменения ширины</a>	Изменение ширины столбца при изменении содержимого
<a href="#">Ширина</a>	Ширина столбца (в пикселях)
<a href="#">Шрифт текста заголовка</a>	Шрифт текста заголовка столбца
<a href="#">Цвет текста заголовка</a>	Цвет текста заголовка столбца

<a href="#">Цвет фона заголовка</a>	Цвет фона заголовка столбца
<a href="#">Выравнивание текста</a>	Вид выравнивания текста в столбце
<a href="#">Шрифт текста</a>	Шрифт текста в столбце
<a href="#">Цвет текста</a>	Цвет текста в столбце
<a href="#">Цвет фона</a>	Цвета фона столбца
<a href="#">Идентификатор</a>	Идентификатор столбца
<a href="#">Описание</a>	Описание столбца
<a href="#">Заголовок</a>	Заголовок столбца
<a href="#">Режим сортировки</a>	Режим сортировки
<a href="#">Выравнивание текста заголовка</a>	Выравнивание текста заголовка
<a href="#">Собственный идентификатор столбца</a>	Собственный идентификатор столбца

## 1.1.4.2.4.9.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.9.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.9.3.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

## Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.9.3.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.9.3.3.5. Заголовок

Заголовок столбца.



string Title

## 1.1.4.2.4.9.3.3.6. Описание

Описание столбца.



string Description

## 1.1.4.2.4.9.3.3.7. Идентификатор

Идентификатор столбца.



string Id

### Значение

Значение	Описание
source	Источник события
time	Время генерации уведомления о событии
message	Сообщение
severity	Уровень важности события
condition_name	Имя условия генерации события
subcondition_name	Имя подусловия генерации события
quality	Текущее качество сигнала, изменение которого привело к генерации события
active_time	Время перехода состояния события в активное
actor_id	Имя пользователя, выполнившего квитирование сообщения о событии
ack	Признак квитирования события
active	Признак активности подусловия, по которому было сгенерировано событие
cookie	Специальный идентификатор события, который имеет служебное назначение. Необходим для сведения квитанций, деактиваций с событиями
ack_time	Время квитирования
ack_comment	Комментарий квитирования

ack_required	Требование квитирования
value	Значение сигнала, изменение которого привело к генерации события
sound	Звуковой файл, исполняемый при выполнении подусловия генерации сообщения о событии
area_path	Относительный тег объекта
object_id	Идентификатор объекта на сервере, по которому было сгенерировано событие
deactive_time	Время, когда событие перешло из активного состояния в неактивное

## 1.1.4.2.4.9.3.3.8. Собственный идентификатор столбца

Собственный идентификатор столбца.



string ColumnOwnId



## 1.1.4.2.4.9.3.3.9. Цвет фона

Цвета фона столбца.



uint4 BackgroundColor

## 1.1.4.2.4.9.3.3.10. Цвет текста

Цвет текста в столбце.



uint4 ForegroundColor

# 1.1.4.2.4.9.3.3.11. Шрифт текста

Шрифт текста в столбце.



string Font

## 1.1.4.2.4.9.3.3.12. Выравнивание текста

Вид выравнивания текста в столбце.



int4 TextAlignment

## 1.1.4.2.4.9.3.3.13. Цвет фона заголовка

Цвет фона заголовка столбца.



uint4 TitleBackgroundColor

## 1.1.4.2.4.9.3.3.14. Цвет текста заголовка

Цвет текста заголовка столбца.



uint4 TitleForegroundColor

## 1.1.4.2.4.9.3.3.15. Шрифт текста заголовка

Шрифт текста заголовка столбца.



string TitleFont

# 1.1.4.2.4.9.3.3.16. Выравнивание текста заголовка

Выравнивание текста заголовка.



int4 TitleTextAlignment



## 1.1.4.2.4.9.3.3.17. Ширина

Ширина столбца (в пикселях).



int4 Width

## 1.1.4.2.4.9.3.3.18. Режим изменения ширины

Изменение ширины столбца при изменении содержимого.



uint4 WidthSizeMode

### Значение

Значение	Описание
0	Ручное изменение ширины с помощью мыши
1	Максимальное растяжение столбца по полю таблицы
2	Фиксированная ширина столбца, заданная в свойствах столбца
3	Автоматическое изменение ширины столбца в соответствии с шириной содержимого

## 1.1.4.2.4.9.3.3.19. Может не иметь значения

Разрешает/запрещает ячейкам столбца не содержать значений.



bool Emptiable

### Значение

Значение	Описание
true	Ячейки могут быть пустыми
false	В ячейки следует записать значения

## 1.1.4.2.4.9.3.3.20. Скрывать

Скрывает/отображает столбец.



bool Hidden

### Значение

Значение	Описание
true	Столбец скрыт
false	Столбец отображается

## 1.1.4.2.4.9.3.3.21. Числовой формат

Формат поступающих числовых данных.



string NumberFormat

## 1.1.4.2.4.9.3.3.22. Временной формат

Формат выводимой даты полученных данных.



string TimeFormat

## 1.1.4.2.4.9.3.3.23. Фильтр в JSON-формате

Условия фильтрации отображаемых данных в таблице. Задаются в виде строки - объекта JSON.



string JsonFilter

### Правила формирования условий фильтрации

Фильтрация возможна только для столбцов с идентификаторами (переменными): severity, message, source, area, acked.

Условие фильтрации представляет собой одиночный ({"ключ": "значение"}) или составной (состоящий из нескольких объектов) объект JSON.

Составной объект формируется с помощью:

- › логических операций:
  - › конъюнкция - "and" или "&")
  - › дизъюнкция - "or" или "|")
  - › инверсия - "not" или "!"
- › операций сравнения:
  - › больше - ">"
- › больше либо равно - ">="
- › меньше - "<"
- › меньше либо равно - "<="
- › соответствует - "=="
- › не соответствует - "!="

В зависимости от используемой операции составной объект JSON может иметь формат:

- › {"and": [ARRAY]}
- › {"or": [ARRAY]}
- › {"not": { OBJECT } }

> { "операция сравнения": { VARIABLE } }(операцию сравнения "==" можно не указывать, т.е. формат объекта - { VARIABLE })

где:

- > VARIABLE - переменная. Обозначается структурой <имя переменной>: <значение переменной>
- > OBJECT - переменная или составной объект JSON любого из указанных выше форматов
- > ARRAY - массив. Может состоять из переменных и составных объектов JSON (любого формата) в различных комбинациях

## Примеры



```
//Отфильтровать по содержимому столбца с числовыми данными (например, важность). Отобразить события с важностью 700. При фильтрации по столбцу имя переменной должно быть пустым, т.е. указанное условие имеет вид: {"==": {"": 700}} или {"":700}
```



```
//Отфильтровать по содержимому столбца с текстовыми данными. Отобразить события, содержащие текст "поставлен" или "снят":  
{  
    "or": [  
        {"": "*поставлен*"},  
        {"": "*снят*"}  
    ]  
}
```



```
//Отфильтровать данные по содержимому столбца с числовыми данными (например, важность). Отобразить события с важностью от 100 до 300:  
{  
    "and": [  
        {">": 100},  
        {"<": 300}
```



```
    {">=": {"": 100}},  
    {"<=": {"": 300}}  
  ]  
}
```

## 1.1.4.2.4.9.3.3.24. Сортировать

Разрешение/запрет сортировки данных в столбце.



bool SortingEnabled

## 1.1.4.2.4.9.3.3.25. Порядок сортировки

Вид сортировки данных в столбце.



uint1 SortOrder

### Значение

Значение	Описание
0	По возрастанию (прямо по алфавиту)
1	По убыванию (обратно по алфавиту)

## 1.1.4.2.4.9.3.3.26. Приоритет сортировки

Порядок применения сортировки.



uint4 SortPriority

## 1.1.4.2.4.9.3.3.27. Режим сортировки

Режим сортировки.



uint1 SortMode

## 1.1.4.2.4.9.3.3.28. Разрешить перенос слов

Разрешение/запрет переноса слов в столбце.



bool WordWrap

### Значение

Значение	Описание
true	Перенос слов разрешен
false	Перенос слов запрещен

## 1.1.4.2.4.9.4. Таблица: декоратор

Компонент позволяет выделять цветом отдельные строки или столбцы таблицы.

## 1.1.4.2.4.9.4.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Настройки декоратора</a>	Задаются в виде строки - объекта JSON
<a href="#">ErrorDescription</a>	Описание ошибки



## 1.1.4.2.4.9.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.9.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.9.4.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.9.4.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.9.4.1.5. Настройки декоратора

Задаются в виде строки - объекта JSON.



string JsonDecorator

## 1.1.4.2.4.9.4.1.6. ErrorDescription

Описание ошибки.

## 1.1.4.2.4.9.5. Таблица: итератор

Компонент позволяет выполнять перебор строк от первой строки до последней. Устаревший компонент, вместо него используйте компонент Таблица: устойчивый итератор.

## 1.1.4.2.4.9.5.1. Функции

Компонент	Описание
<a href="#">End</a>	Возвращает true, если достигнут конец итерации
<a href="#">Next</a>	Переходит к следующей строке
<a href="#">FirstCustom</a>	Инициализирует итератор с пользовательским фильтром
<a href="#">FirstCelected</a>	Инициализирует итератор
<a href="#">CurrentRow</a>	Индекс текущей строки



## 1.1.4.2.4.9.5.1.1. End

Возвращает true, если достигнут конец итерации.



End()

## 1.1.4.2.4.9.5.1.2. Next

Переходит к следующей строке.



Next()

## 1.1.4.2.4.9.5.1.3. FirstCustom

Инициализирует итератор с пользовательским фильтром.



FirstCustom(Filter)

### Параметры

Параметр	Тип	Описание
Filter	string	Наименование фильтра

## 1.1.4.2.4.9.5.1.4. FirstCelected

Инициализирует итератор.



FirstCelected()

## 1.1.4.2.4.9.5.1.5. CurrentRow

Индекс текущей строки.

## 1.1.4.2.4.9.5.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Индекс текущей строки</a>	Порядковый номер строки, с которой будет проведена обработка данных

## 1.1.4.2.4.9.5.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.9.5.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.9.5.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.9.5.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.9.5.2.5. Индекс текущей строки

Порядковый номер строки, с которой будет проведена обработка данных.



int4 CurrentRow

## 1.1.4.2.4.9.6. Таблица: устойчивый итератор

Компонент расширяет возможности компонента Таблица: итератор. Позволяет выполнять перебор строк от первой строки до последней, а также перебор выделенных строк таблицы.

## 1.1.4.2.4.9.6.1. Функции

Компонент	Описание
<a href="#">Close</a>	Деинициализирует итератор
<a href="#">RemoveRow</a>	Удаляет текущую строку из таблицы
<a href="#">HasNext</a>	Выводит true, если текущая строка является последней для выбранного режима итератора
<a href="#">Next</a>	Переходит к следующей строке, в зависимости от режима
<a href="#">FirstSelectedRow</a>	Инициализирует итератор для прохода по выделенным строкам таблицы, начиная с первой выделенной
<a href="#">FirstRow</a>	Инициализирует итератор для прохода по всем строкам таблицы, начиная с первой
<a href="#">SetRowSelected</a>	Выделяет или снимает выделение с текущей строки
<a href="#">IsRowSelected</a>	Возвращает true, если текущая строка выделена
<a href="#">GetStatus</a>	Возвращает код состояния строки, на которую указывает итератор
<a href="#">GetRow</a>	Возвращает номер текущей строки
<a href="#">GetText</a>	Возвращает значение ячейки строки, на которую указывает итератор, в виде текстовой строки
<a href="#">GetData</a>	Возвращает значение (с типом variant) ячейки строки, на которую указывает итератор
<a href="#">FirstVisibleRow</a>	Инициализация итератора для прохода по видимым строкам таблицы

## 1.1.4.2.4.9.6.1.1. Close

Деинициализирует итератор.



`void Close()`

## 1.1.4.2.4.9.6.1.2. RemoveRow

Удаляет текущую строку из таблицы.



```
void RemoveRow()
```

## 1.1.4.2.4.9.6.1.3. hasNext

Выводит true, если текущая строка является последней для выбранного режима итератора.



```
bool hasNext()
```



## 1.1.4.2.4.9.6.1.4. Next

Переходит к следующей строке, в зависимости от режима (проход по выделенным или проход по всем).



```
void Next()
```

## 1.1.4.2.4.9.6.1.5. FirstSelectedRow

Инициализирует итератор для прохода по выделенным строкам таблицы, начиная с первой выделенной.



```
void FirstSelectedRow()
```

## 1.1.4.2.4.9.6.1.6. FirstRow

Инициализирует итератор для прохода по всем строкам таблицы, начиная с первой.



```
void FirstRow()
```

## 1.1.4.2.4.9.6.1.7. SetRowSelected

Выделяет или снимает выделение с текущей строки.



```
void SetRowSelected(bool FlagBool)
```

### Параметры

Параметр	Тип	Описание
FlagBool	bool	Флаг выделения/снятия выделения

## 1.1.4.2.4.9.6.1.8. IsRowSelected

Возвращает true, если текущая строка выделена.



bool IsRowSelected()

## 1.1.4.2.4.9.6.1.9. GetStatus

Возвращает код состояния строки, на которую указывает итератор.



```
uint1 GetStatus()
```

## 1.1.4.2.4.9.6.1.10. GetRow

Возвращает номер текущей строки.



int4 GetRow()

## 1.1.4.2.4.9.6.1.11. GetText

Возвращает значение ячейки строки, на которую указывает итератор, в виде текстовой строки.



```
string GetText(int4 Index)
```

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс строки, на которой установлен итератор



## 1.1.4.2.4.9.6.1.12. GetData

Возвращает значение (с типом variant) ячейки строки, на которую указывает итератор.



variant GetData(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Индекс строки, на которой установлен итератор

## 1.1.4.2.4.9.6.1.13. FirstVisibleRow

Инициализация итератора для прохода по видимым строкам таблицы, начиная с первой видимой.



```
void FirstVisibleRow()
```

## 1.1.4.2.4.9.6.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.9.6.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.9.6.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.9.6.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.9.6.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

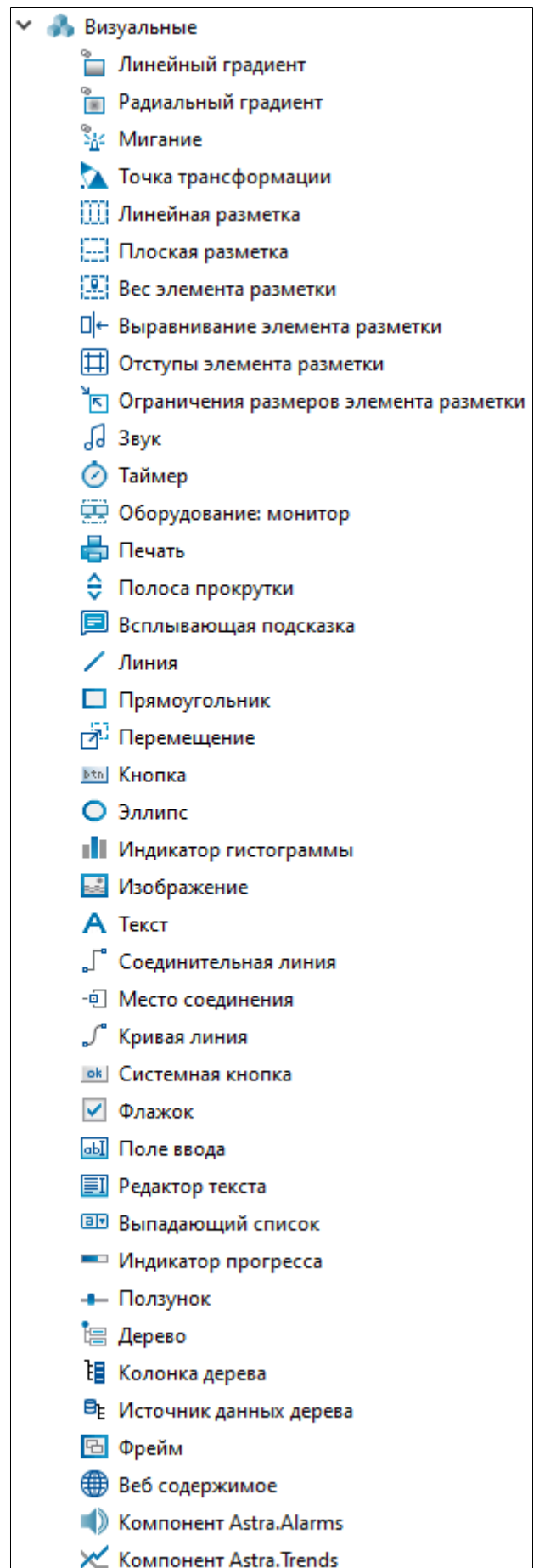
## 1.1.4.2.4.10. Визуальные

Графические примитивы используются в проекте для рисования отдельных фигур и объектов технологического процесса, а также для организации взаимодействия пользователя с мнемосхемой.



Компонент содержит:

- › [Линейный градиент](#)
- › [Радиальный градиент](#)
- › [Мигание](#)
- › [Точка трансформации](#)
- › [Линейная разметка](#)
- › [Плоская разметка](#)
- › [Вес элемента разметки](#)
- › [Выравнивание элемента разметки](#)
- › [Отступы элемента разметки](#)
- › [Ограничения размеров элемента разметки](#)
- › [Звук](#)
- › [Таймер](#)
- › [Оборудование: монитор](#)
- › [Печать](#)
- › [Полоса прокрутки](#)
- › [Всплывающая подсказка](#)
- › [Линия](#)
- › [Прямоугольник](#)
- › [Перемещение](#)
- › [Кнопка](#)
- › [Эллипс](#)
- › [Индикатор гистограммы](#)
- › [Изображение](#)
- › [Текст](#)
- › [Соединительная линия](#)
- › [Место соединения](#)
- › [Кривая линия](#)
- › [Системная кнопка](#)
- › [Флажок](#)
- › [Поле ввода](#)
- › [Редактор текста](#)
- › [Выпадающий список](#)
- › [Индикатор прогресса](#)
- › [Ползунок](#)
- › [Дерево](#)
- › [Колонка дерева](#)
- › [Источник данных дерева](#)
- › [Фрейм](#)
- › [Веб содержимое](#)
- › [Компонент Astra.Alarms](#)
- › [Компонент Astra.Trends](#)

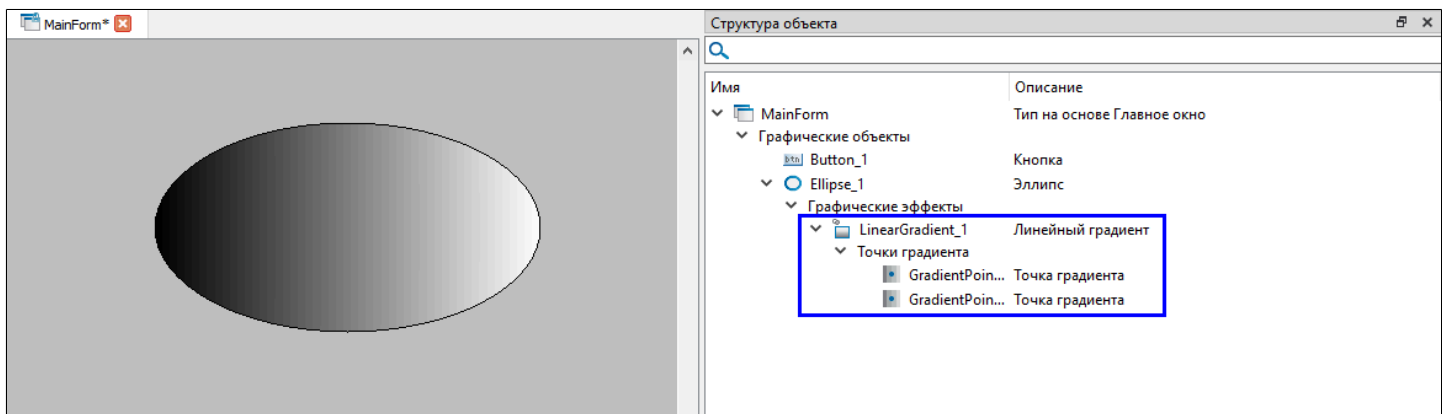


- › [Ползунок](#)
- › [Дерево](#)
- › [Колонка дерева](#)
- › [Источник данных дерева](#)
- › [Фрейм](#)
- › [Веб содержимое](#)
- › [Компонент Astra.Alarms](#)
- › [Компонент Astra.Trends](#)

## 1.1.4.2.4.10.1. Линейный градиент

Стиль заливки объекта, при котором создается плавный линейный переход между двумя цветами. Каждый линейный градиент имеет две точки, определяющие начало и конец градиента.

Чтобы выполнить заливку графического объекта в виде плавного линейного перехода между двумя цветами, добавьте компонент *Линейный градиент* на графический объект. Обычно с помощью градиентной заливки создается объемность в изображении объектов. После применения градиента к графическому объекту создаются точки градиента – первая точка содержит свойства начала градиента, а вторая - свойства конца градиента.



- цвет в точках градиента определяется свойством [Цвет](#);
- угол наклона градиентной линии определяется свойством [Угол поворота](#).

# Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет</a>	Цвет в начальной или конечной точке градиента
<a href="#">Положение</a>	Положение точек начала или конца цветового перехода
<a href="#">Угол поворота</a>	Угол наклона градиентной линии, которая задает направление градиента
<a href="#">Видимость</a>	Видимость
<a href="#">Непрозрачность</a>	Непрозрачность

## 1.1.4.2.4.10.1.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.1.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.1.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.1.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

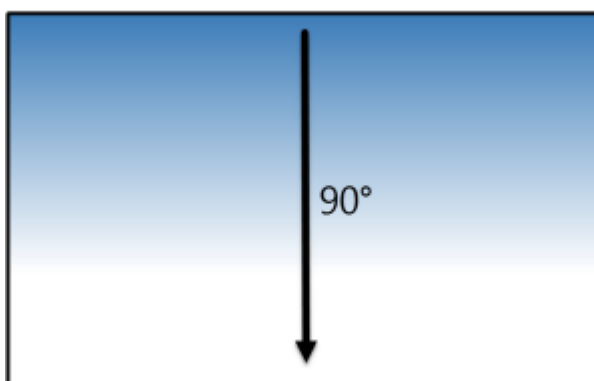
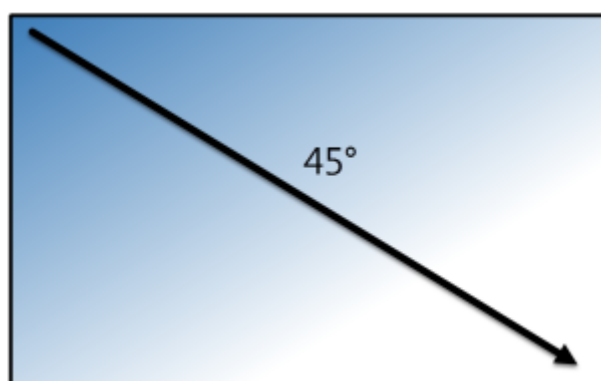
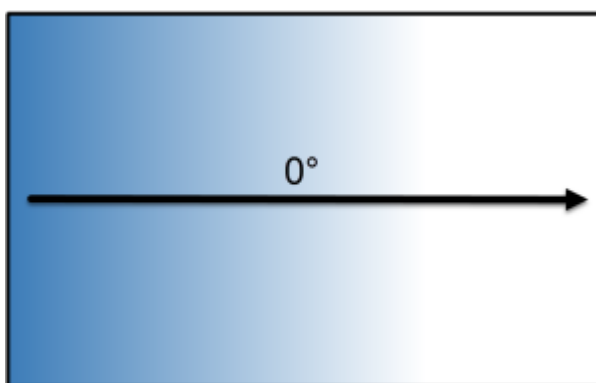


## 1.1.4.2.4.10.1.1.5. Угол поворота

Угол наклона градиентной линии, которая задает направление градиента. 0 градусов соответствует линейному переходу слева направо. С увеличением градуса линия градиента поворачивается по часовой стрелке.



float8 Rotation



## 1.1.4.2.4.10.1.1.6. Видимость

Видимость.



bool Visible

## 1.1.4.2.4.10.1.1.7. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.1.1.8. Цвет

Свойства точек градиента – цвет в начальной или конечной точке градиента. Выбирается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 Color

Скриншот интерфейса разработки, состоящий из двух панелей: "Структура объекта" и "Редактор свойств".

**Структура объекта:**

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Button_1	Кнопка
Ellipse_1	Эллипс
Графические эффекты	
LinearGradient_1	Линейный градиент
Точки градиента	
GradientPoin...	Точка градиента
GradientPoin...	Точка градиента

**Редактор свойств:**

Свойство	Характеристики	Значение
Отображаемое имя		GradientPoint_1
Кардинальное число		1
Положение	R W	0
Начальное значение		<не определено>
Вычисляемое значение		<не определено>
Цвет	R W	4278190080

## Примеры



//Установить начальный цвет градиента - черный и конечный цвет градиента - белый:

```
Object.Gradient.GradientPoint_1.Color = 0xff000000;
```

```
Object.Gradient.GradientPoint_2.Color = 0xffffffff;
```

## 1.1.4.2.4.10.1.1.9. Положение

Свойства точек градиента – положение точек начала или конца цветового перехода. Измеряется от 0 до 1, где 0 - начало объекта, 1 - конец объекта. На картинке и в примере ниже показано, что старт линейного градиента находится в самом начале объекта (значение 0), а конец градиента наступает уже к середине объекта (значение 0.5).

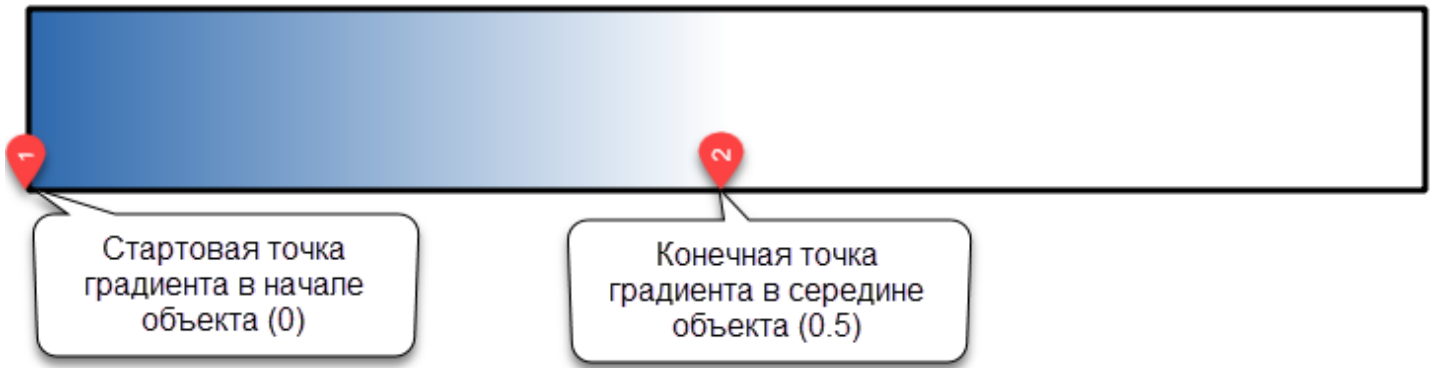


float8 Position

The screenshot displays two windows from the Visual Studio IDE:

- Структура объекта (Object Structure):** A tree view showing the hierarchy of the object. The selected item is `GradientPoin...` (GradientPoint\_1) under `Точки градиента` (Gradient Points) of `LinearGradient_1` (Linear Gradient).
- Редактор свойств (Properties):** A table showing the properties of the selected `GradientPoint_1` object.

Свойство (Property)	Характеристики (Characteristics)	Значение (Value)
Отображаемое имя (Display Name)		GradientPoint_1
Кардинальное число (Cardinal Number)		1
Положение (Position)	R W	0
Начальное значение (Initial Value)		<не определено> (<not defined>)
Вычисляемое значение (Computed Value)		<не определено> (<not defined>)
Цвет (Color)	R W	4278190080



## Примеры



//Задать стартовую позицию градиента в начале объекта и конечную позицию градиента в середине объекта:

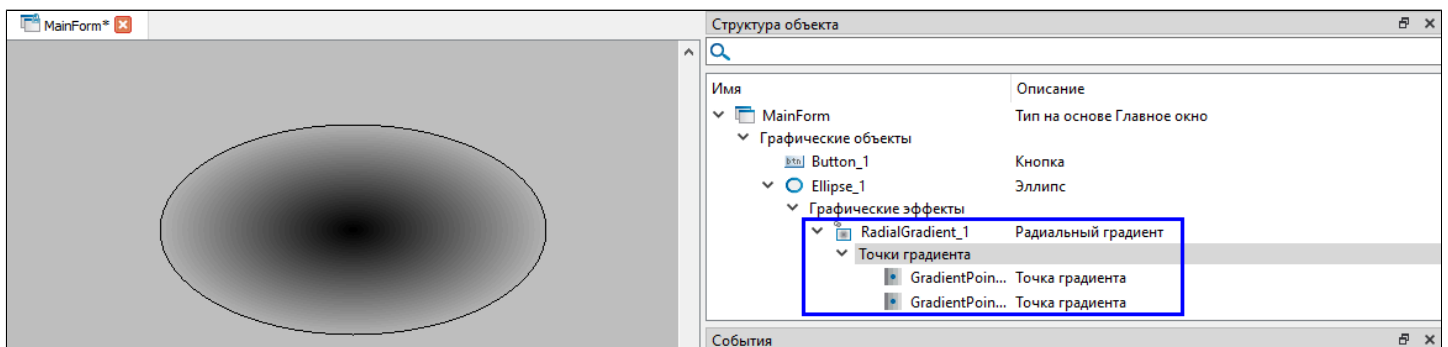
```
Object.Gradient.GradientPoint_1.Position = 0;
```

```
Object.Gradient.GradientPoint_2.Position = 0.5;
```

## 1.1.4.2.4.10.2. Радиальный градиент

Стиль заливки объекта, при котором выполняется затенение от начальной до конечной точки в виде кругового узора. Каждый радиальный градиент имеет две точки, определяющие начало и конец градиента.

Компонент Радиальный градиент, также как и линейный градиент, применяется на мнемосхемах при заливке графических объектов. Обычно с помощью градиентной заливки создается объемность в изображении объектов. При добавлении радиального градиента на объект создаются точки градиента – первая точка содержит свойства начала градиента, а вторая – свойства конца градиента. Заливка объекта происходит с помощью затемнения от начальной до конечной точки в виде кругового узора. На рисунке ниже показан объект мнемосхемы, залитый радиальным градиентом и структура радиального градиента.



- цвет в точках градиента устанавливается свойством [Цвет](#);
- координаты центра радиального градиента по горизонтали и вертикали определяются свойствами [X центра](#), [Y центра](#).

# Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет</a>	Цвет в начальной или конечной точке градиента
<a href="#">Положение</a>	Положение точек начала или конца цветового перехода
<a href="#">X центра</a>	Координаты центра радиального градиента по горизонтали
<a href="#">Y центра</a>	Координаты центра радиального градиента по вертикали
<a href="#">Видимость</a>	Видимость
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Размер</a>	Размер



## 1.1.4.2.4.10.2.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.2.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.2.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.2.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.2.1.5. X центра

Координаты центра радиального градиента по горизонтали.



float8 XCenter



## 1.1.4.2.4.10.2.1.6. Y центра

Координаты центра радиального градиента по вертикали.



float8 YCenter



## 1.1.4.2.4.10.2.1.7. Видимость

Видимость.



bool Visible

## 1.1.4.2.4.10.2.1.8. Непрозрачность

Непрозрачность.



float8 Opacity



## 1.1.4.2.4.10.2.1.9. Размер

Размер.



uint1 Size

## 1.1.4.2.4.10.2.1.10. Цвет

Свойства точек градиента – цвет в начальной или конечной точке градиента. Выбирается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 Color

## 1.1.4.2.4.10.2.1.11. Положение

Свойства точек градиента – положение точек начала или конца цветового перехода. Измеряется от 0 до 1, где 0 - начало объекта, 1 - конец объекта.



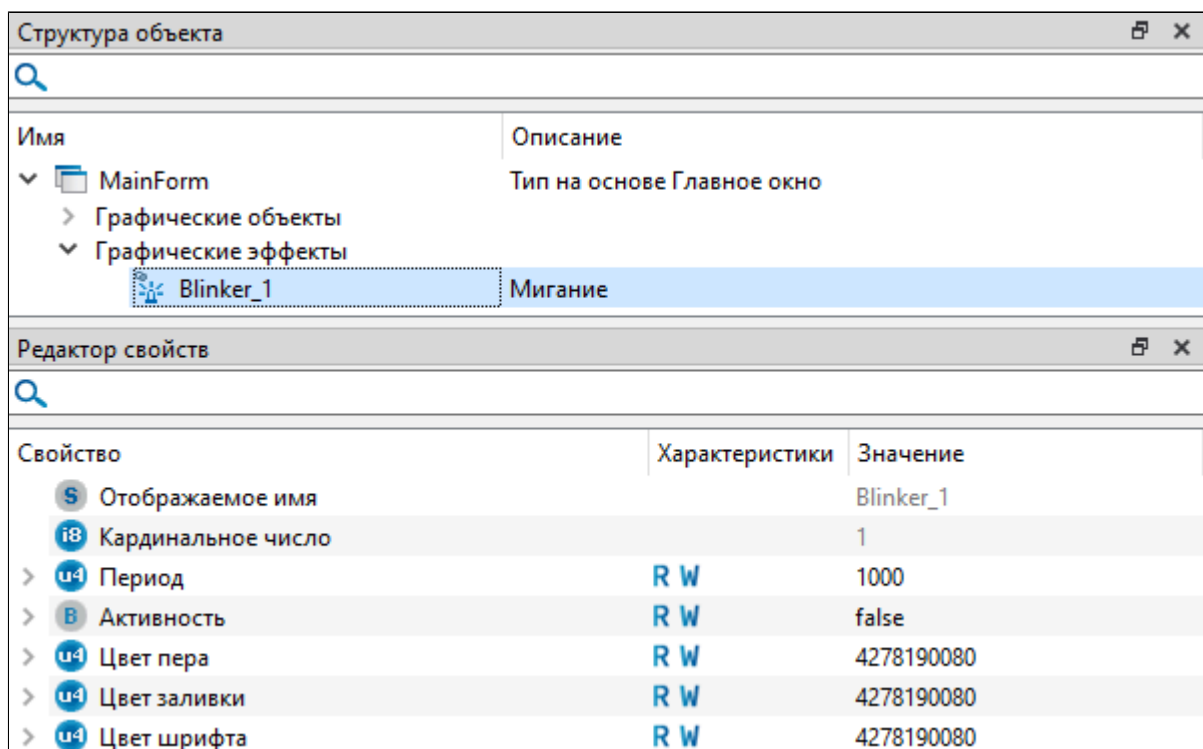
float8 Position

## 1.1.4.2.4.10.3. Мигание

Невизуальный компонент позволяет графическим компонентам циклически с заданной периодичностью менять цвет заливки, пера или шрифта.

Цвет попеременно изменяется от стандартного цвета графического компонента к цвету, который настроен в свойствах компонента Мигание.

Чтобы графические объекты получили возможность мигания, добавьте на фигуру или экранную форму компонент Мигание.



The screenshot shows two windows from a development environment. The top window, titled 'Структура объекта' (Object Structure), displays a tree view of the project components. Under 'Графические эффекты' (Graphic Effects), the 'Blinker\_1' component is selected, with the description 'Мигание' (Blinking). The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties for the selected 'Blinker\_1' component. The properties are listed in a table with columns for the property name, characteristics, and the current value.

Свойство	Характеристики	Значение
Отображаемое имя		Blinker_1
Кардинальное число		1
Период	R W	1000
Активность	R W	false
Цвет пера	R W	4278190080
Цвет заливки	R W	4278190080
Цвет шрифта	R W	4278190080

## 1.1.4.2.4.10.3.1. События

Событие	Описание
<a href="#">Blink</a>	Мигание объекта

## 1.1.4.2.4.10.3.1.1. Blink

Мигание объекта.

### Параметры

Параметр	Тип	Описание
value	bool	Отражает факт смены цвета пера/заливки/шрифта при мигании: <ul style="list-style-type: none"><li>› true – цвет сменился на конечный</li><li>› false – цвет вернулся к начальному значению</li></ul>

# Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Цвет заливки</a>	Цвет внутренней заливки мигающего объекта
<a href="#">Цвет пера</a>	Цвет внешней границы мигающего объекта
<a href="#">Активность</a>	Активность мигания
<a href="#">Период</a>	Период мигания графического объекта (мс)
<a href="#">Цвет шрифта</a>	Цвет шрифта

## 1.1.4.2.4.10.3.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.10.3.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.3.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.3.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.3.2.5. Период

Период мигания графического объекта (мс).



uint4 Period

## 1.1.4.2.4.10.3.2.6. Активность

Активность мигания.



bool Active

### Значение

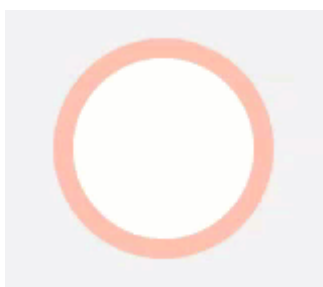
Значение	Описание
true	Мигание включено
false	Мигание выключено

## 1.1.4.2.4.10.3.2.7. Цвет пера

Цвет внешней границы мигающего объекта. При мигании объекта цвет внешней границы меняется с цвета, указанного в свойствах объекта, на цвет, заданный в свойствах объекта Мигание.



uint4 PenColor



## 1.1.4.2.4.10.3.2.8. Цвет заливки

Цвет внутренней заливки мигающего объекта. При мигании объекта цвет заливки меняется с цвета, указанного в свойствах объекта, на цвет, заданный в свойствах объекта Мигание.



uint4 BrushColor



## 1.1.4.2.4.10.3.2.9. Цвет шрифта

Цвет шрифта мигающего объекта. При мигании объекта цвет шрифта меняется с цвета, указанного в свойствах объекта, на цвет, заданный в свойствах объекта типа Мигание.

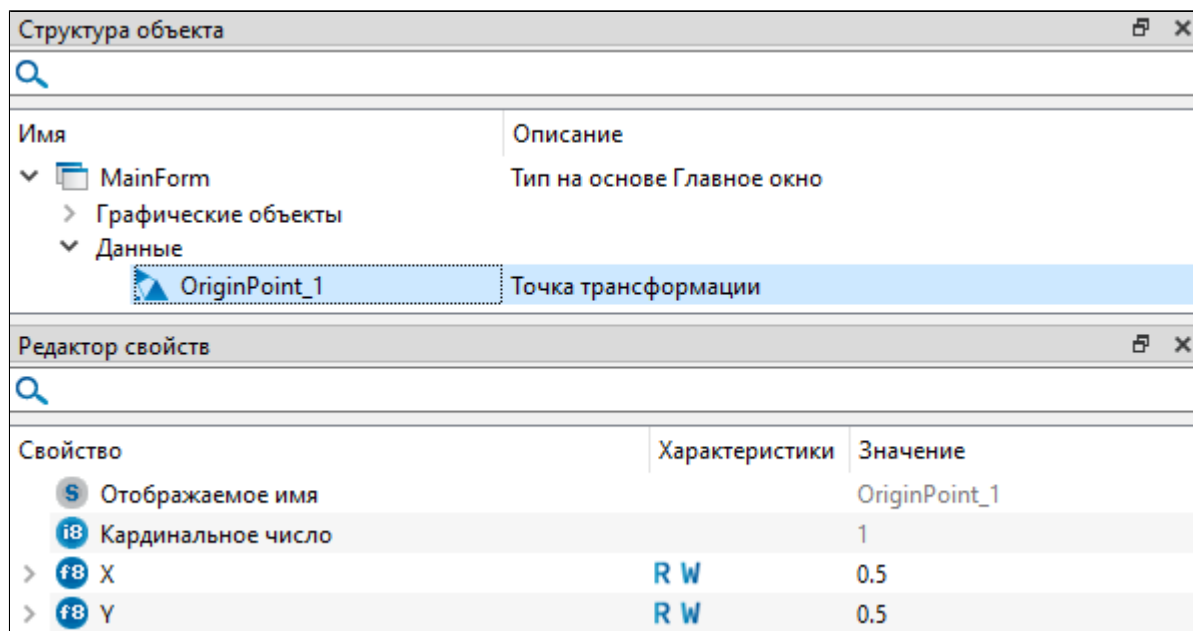


uint4 FontColor



# Точка трансформации

Компонент предназначен для возможности масштабирования и вращения графических объектов относительно данной точки, а не его центра.



Компонент представляет из себя точку, которая размещается на графическом объекте (родительский объект) формы. Масштабирование и вращение родительского объекта будет происходить относительно координат данной точки. Координаты точки вычисляются относительно того объекта, внутри которого точка находится. Значение точки может варьироваться от 0 до 1. Координата точки 0;0 обозначает верхний левый угол объекта.

## 1.1.4.2.4.10.4.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали

## 1.1.4.2.4.10.4.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.4.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.4.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.4.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
TextEdit\_2.Text = String.ToString(Index);



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
TextEdit\_3.Text = String.ToString(penwidth[Index]);



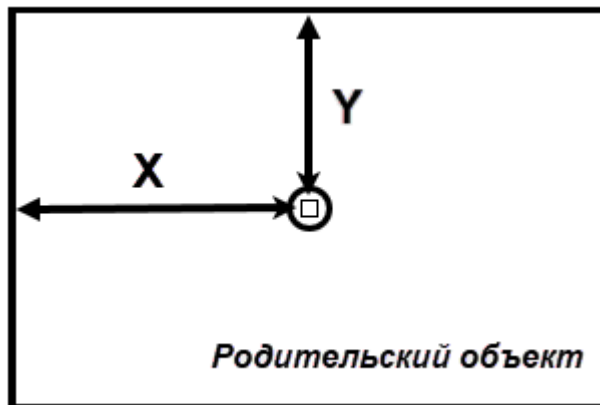
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
if (SW\_1.Mode.Index == 0) {  
    TextEdit\_1.Text = "Включен режим 1";}  
else  
    {TextEdit\_1.Text = "Включен режим 2";}

## 1.1.4.2.4.10.4.1.5. X

Позиция точки по горизонтали и по вертикали, относительно координат родительского объекта. Могут принимать значения от 0 до 1.



float8 X



### Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



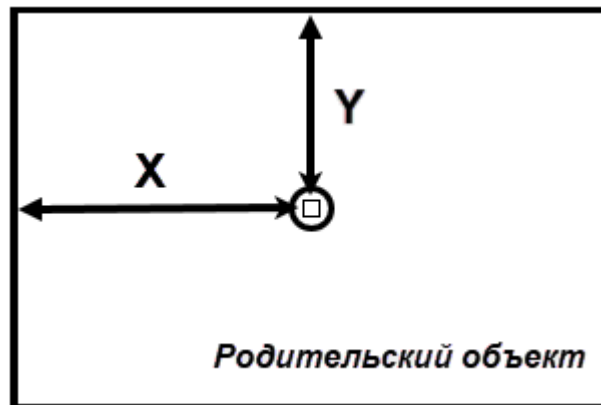
```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.2.4.10.4.1.6. Y

Позиция точки по вертикали, относительно координат родительского объекта.  
Могут принимать значения от 0 до 1.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



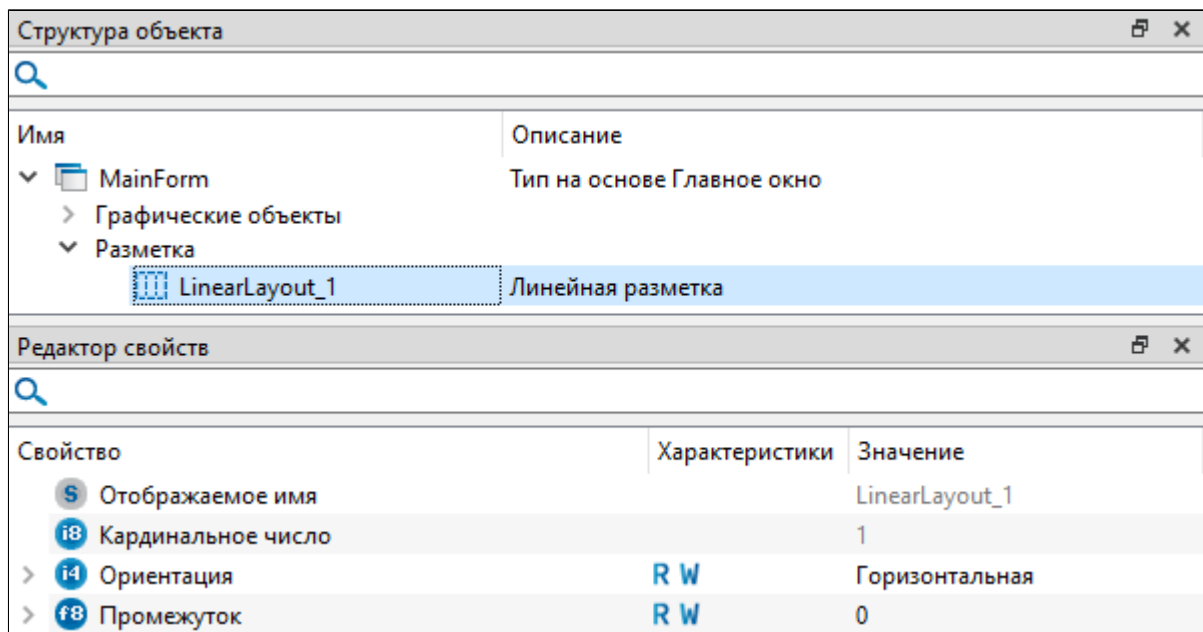
```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



## 1.1.4.2.4.10.5. Линейная разметка

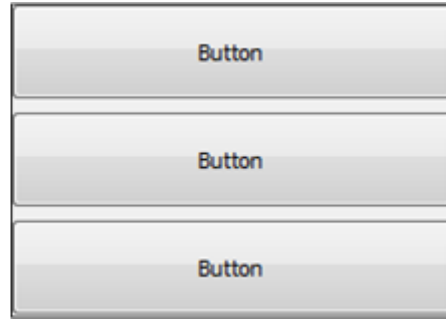
Позволяет линейно выравнять объекты внутри контейнера (экранной формы или фигуры).

Чтобы линейно выравнять объекты внутри контейнера (экранной формы или фигуры), добавьте в контейнер компонент *Линейная разметка*. Компонент не визуальный и виден только в области Структура объекта. На рисунке ниже показано, как на прямоугольник добавлен компонент *Линейная разметка*.



Все элементы, входящие в прямоугольник будут выравнены линейно по горизонтали или вертикали в зависимости от настройки свойств компонента:

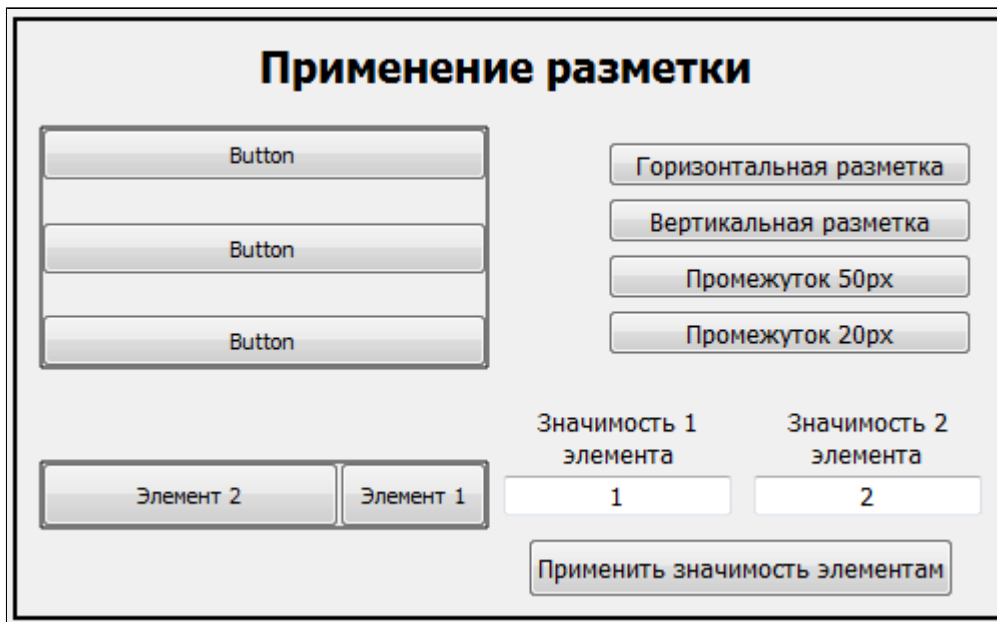
- Ориентация (int4) – направление выравнивания объектов внутри фигуры или формы:
  - 1 – по горизонтали;
  - 2 – по вертикали.



› Промежуток (double) – расстояние между объектами внутри фигуры или формы.



## Разметка мнемосхемы



Для разметки кнопок на первый контейнер (прямоугольник Rectangle\_1) добавлен компонент Линейная разметка, на второй (прямоугольник Rectangle\_2) – компонент Плоская разметка.

Имя	Описание
Form_1	Тип на основе Форма
Графические объекты	
Rectangle_1	Прямоугольник
Графические объекты	
Разметка	
LinearLayout_1	Линейная разметка
Rectangle_2	Прямоугольник
Графические объекты	
Разметка	
TreeMapLayout_1	Плоская разметка

Чтобы установить вес разметки кнопкам второго контейнера, на каждую кнопку добавлен компонент Вес элемента разметки.

Rectangle_2	Прямоугольник
Графические объекты	
Button_1	Кнопка
Разметка	
LayoutWeight_1	Вес элемента разметки
Button_2	Кнопка
Разметка	
LayoutWeight_1	Вес элемента разметки
Разметка	
TreeMapLayout_1	Плоская разметка

Чтобы элементы первого контейнера выстроились в горизонтальной ориентации, для кнопки Горизонтальная разметка создайте обработчик событий. Код обработчика на языке Astra.Om:



```
Rectangle_1.LinearLayout_1.Orientation = 1;
```

Чтобы элементы первого контейнера выстроились в вертикальной ориентации, для кнопки Вертикальная разметка создайте обработчик событий. Код обработчика на языке Astra.Om:



```
Rectangle_1.LinearLayout_1.Orientation = 2;
```

Чтобы промежуток между элементами в первом прямоугольнике принял значение 50px и 20px, для кнопок Промежуток 50px и Промежуток 20px создайте обработчики событий. Код обработчиков на языке Astra.Om:



```
Rectangle_1.LinearLayout_1.Gap = 50;  
Rectangle_1.LinearLayout_1.Gap = 20;
```

Чтобы элементам второго контейнера установить значимость разметки, для кнопки Применить значимость элементам создайте обработчик событий. Обработчик конвертирует и передает значение из текстового поля в значение свойства Значимость. Код обработчика на языке Astra.Om:



```
Rectangle_2.Button_1.LayoutWeight_1 =  
String.ToDouble(TextEdit_1.Text,1);  
Rectangle_2.Button_2.LayoutWeight_1 =  
String.ToDouble(TextEdit_2.Text,1);
```

## 1.1.4.2.4.10.5.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Промежуток</a>	Расстояние между объектами внутри фигуры или формы
<a href="#">Ориентация</a>	Направление выравнивания объектов внутри фигуры или формы

## 1.1.4.2.4.10.5.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.5.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.5.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.10.5.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.5.1.5. Ориентация

Направление выравнивания объектов внутри фигуры или формы.



int4 Orientation

### Значение

Значение	Описание
1	Ориентация по горизонтали
2	Ориентация по вертикали

## 1.1.4.2.4.10.5.1.6. Промежуток

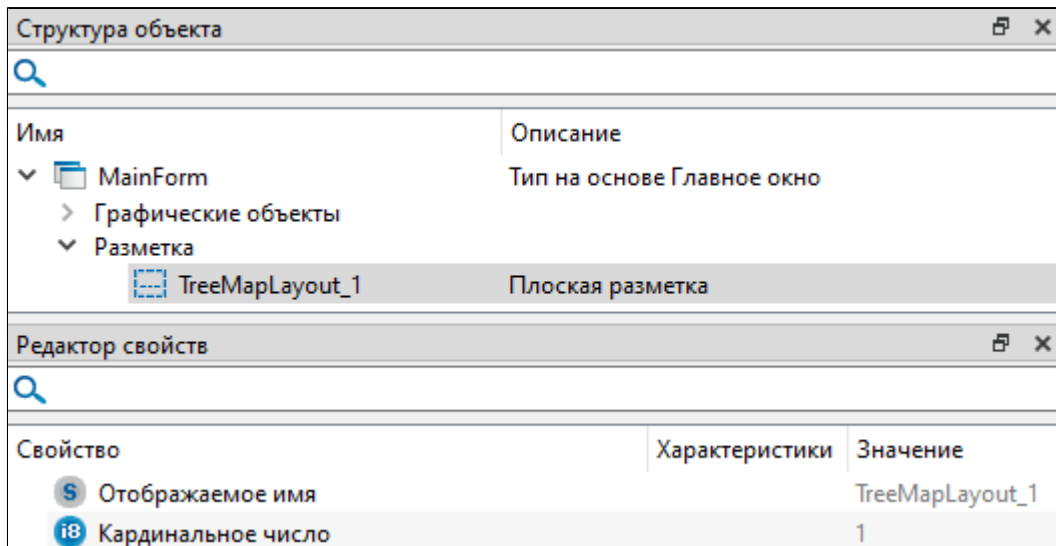
Расстояние между объектами внутри фигуры или формы.



float8 Gap

## 1.1.4.2.4.10.6. Плоская разметка

Позволяет расположить внутри фигуры или формы объекты с разной ориентацией. Ориентация объектов зависит от размеров и положения фигуры или формы. Промежутки между объектами невозможно изменить.



Компонент необходимо использовать именно на конкретную фигуру или форму, внутри которой необходимо разместить объекты в виде вложенных объектов. Компонент не визуальный и виден только в области Структура объекта. На рисунке ниже показано, как на прямоугольник добавлен компонент Плоская разметка.

Компонент позволяет расположить объекты внутри фигуры или формы в виде вложенных объектов. Соотношения сторон между фигурой и вложенными объектами оптимизировано, т.е. размер и расположение вложенных объектов зависит от размера и расположения фигуры, в которую объекты добавлены. Компонент эффективно распределяет свободное пространство фигуры, что позволяет корректно отображать большое количество вложенных объектов.

## 1.1.4.2.4.10.6.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.6.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.6.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.6.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.10.6.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

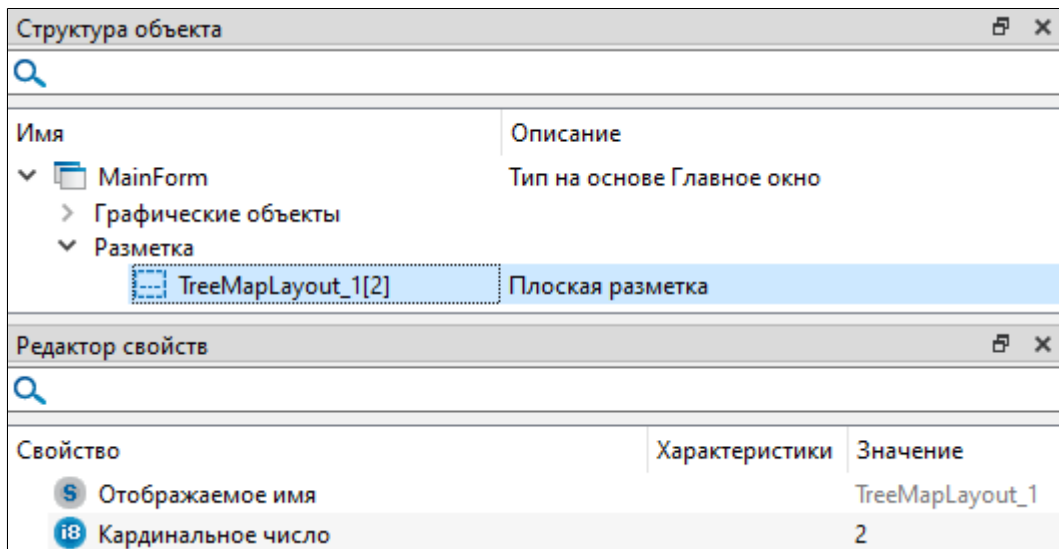


```
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
} else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.7. Вес элемента разметки

Позволяет задавать объектам разметки приоритет размера по отношению к другим объектам фигуры или формы.

Компонент работает как дополнение компонентов Плоская разметка и Линейная разметка.



## 1.1.4.2.4.10.7.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Значимость</a>	Позволяет задавать объектам разметки приоритет размера по отношению к другим объектам фигуры или формы

## 1.1.4.2.4.10.7.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.7.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.7.1.3. Значимость

Позволяет задавать объектам разметки приоритет размера по отношению к другим объектам фигуры или формы.



float8 Weight

### Пример

Имя	Описание
Form_30	Тип на основе Форма
Графические объекты	
Rectangle_1	Прямоугольник
Графические объекты	
Button_1	Кнопка
Разметка	
LayoutW...	Вес элемента разметки
Button_2	Кнопка
Button_3	Кнопка
Разметка	
TreeMapLayout_1	Плоская разметка

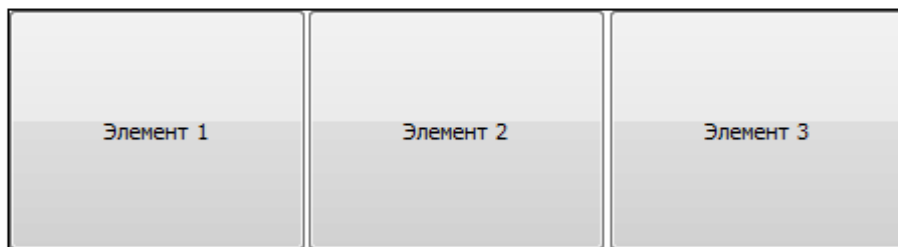
Свойство Значимость для первого элемента 1, для второго – 1, для третьего – 3.

Объект с наибольшей значимостью занимает первое место и имеет размер в трое превосходящий размеры элементов с меньшим весом.

### С использованием веса элемента разметки



### Без использования веса элемента разметки



## 1.1.4.2.4.10.7.1.4. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.10.7.1.5. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

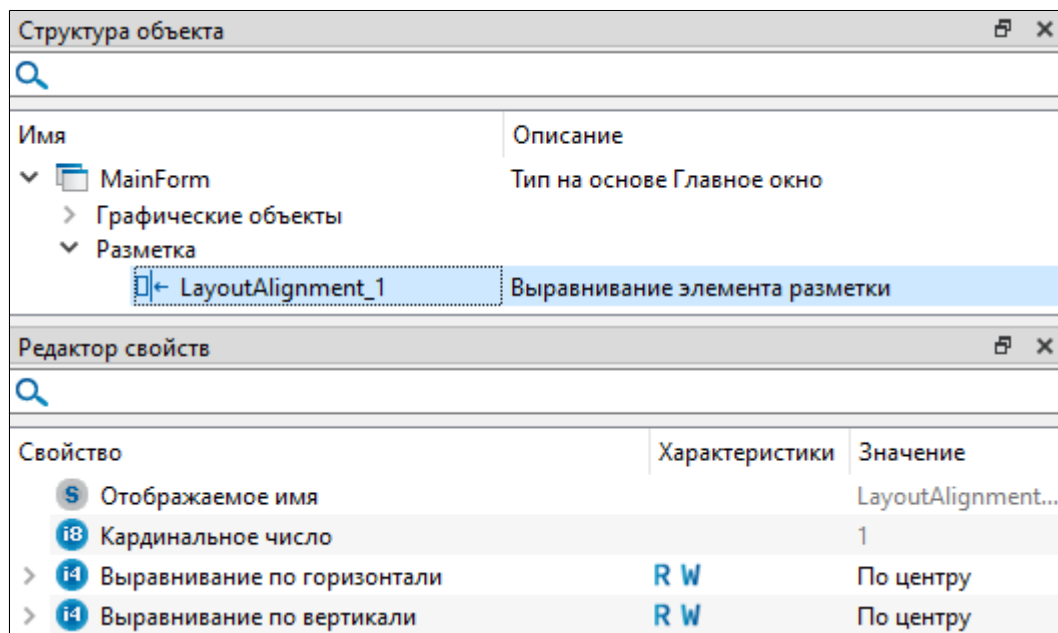


//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.8. Выравнивание элемента разметки

Позволяет выровнять объекты по вертикали или горизонтали.



## 1.1.4.2.4.10.8.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Выравнивание по горизонтали</a>	Выравнивание по горизонтали элемента
<a href="#">Выравнивание по вертикали</a>	Выравнивание по вертикали элемента
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.8.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.8.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.2.4.10.8.1.3. горизонтали

## Выравнивание

по

Выравнивание по горизонтали элемента.



int4 HorizontalAlignment

## 1.1.4.2.4.10.8.1.4. Выравнивание по вертикали

Выравнивание по вертикали элемента.



int4 VerticalAlignment

## 1.1.4.2.4.10.8.1.5. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.10.8.1.6. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



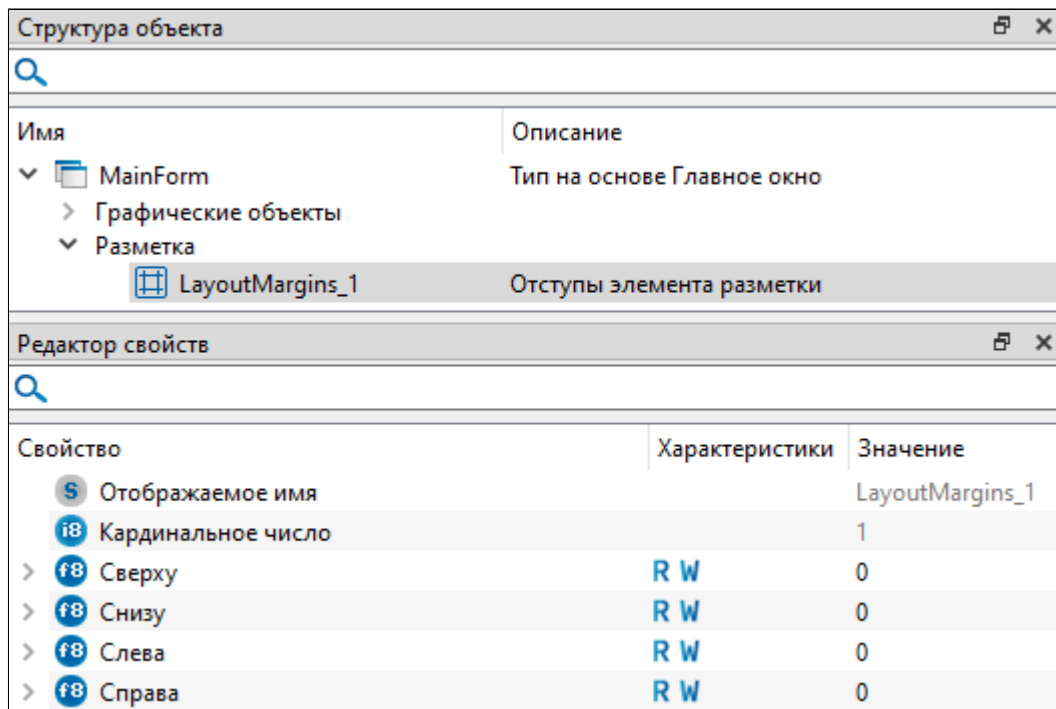
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.9. Отступы элемента разметки

Позволяет задать отступы от краёв для элементов разметки.



The screenshot shows two windows from Visual Studio. The top window, titled 'Структура объекта' (Object Structure), displays a tree view of the project structure. Under 'MainForm', there is a 'Разметка' (Markup) folder containing 'LayoutMargins\_1'. The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties for 'LayoutMargins\_1'. The properties are listed in a table with columns for 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		LayoutMargins_1
Кардинальное число		1
Сверху	R W	0
Снизу	R W	0
Слева	R W	0
Справа	R W	0

## 1.1.4.2.4.10.9.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Сверху</a>	Отступ сверху
<a href="#">Снизу</a>	Отступ снизу
<a href="#">Слева</a>	Отступ слева
<a href="#">Справа</a>	Отступ справа
<a href="#">Lenght</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.9.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.9.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.9.1.3. Сверху

Отступ сверху.



float8 Top

## 1.1.4.2.4.10.9.1.4. Снизу

Отступ снизу.



float8 Bottom

## 1.1.4.2.4.10.9.1.5. Слева

Отступ слева.



float8 Left



## 1.1.4.2.4.10.9.1.6. Справа

Отступ справа.



float8 Right

## 1.1.4.2.4.10.9.1.7. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.9.1.8. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



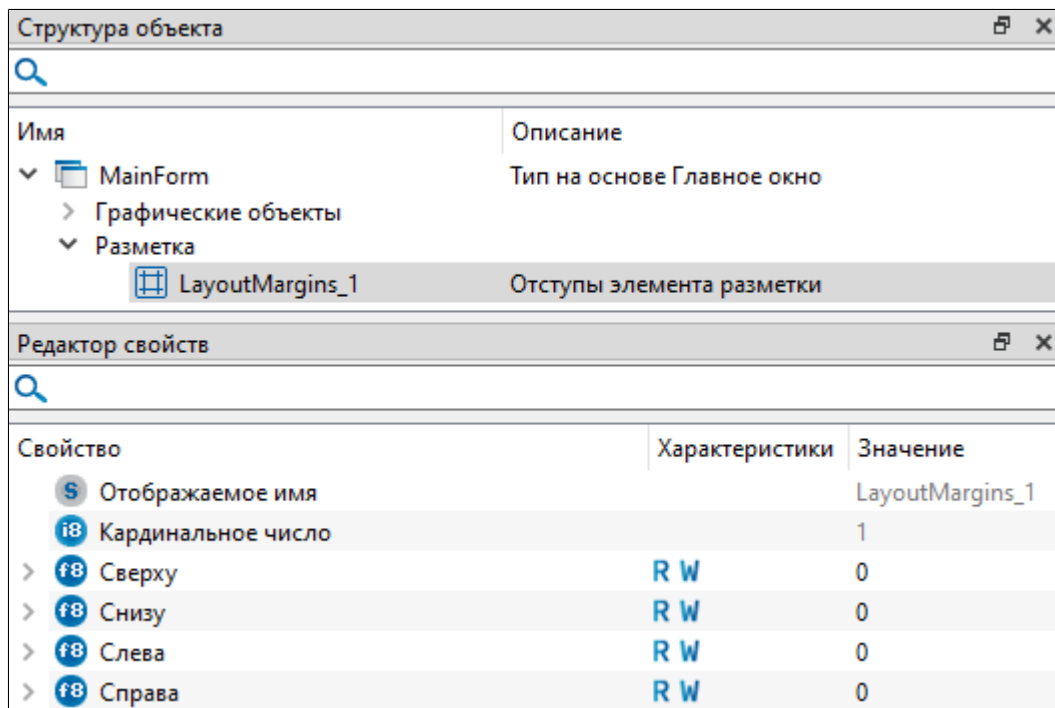
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.10. Ограничения размеров элемента разметки

Позволяет задать минимальные/максимальные значения ширины и высоты элементов разметки.



The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project hierarchy. The 'LayoutMargins\_1' control is selected, with its description 'Отступы элемента разметки' (Layout element margins) visible. The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties of the selected control. The properties are listed in a table with columns for 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		LayoutMargins_1
Кардинальное число		1
Сверху	R W	0
Снизу	R W	0
Слева	R W	0
Справа	R W	0

## 1.1.4.2.4.10.10.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Минимальная ширина</a>	Минимальная ширина элемента
<a href="#">Минимальная высота</a>	Минимальная высота элемента
<a href="#">Максимальная ширина</a>	Максимальная ширина элемента
<a href="#">Максимальная высота</a>	Максимальная высота элемента
<a href="#">Lenght</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.10.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.10.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.10.1.3. Минимальная ширина

Минимальная ширина элемента.



float8 MinWidth



## 1.1.4.2.4.10.10.1.4. Минимальная высота

Минимальная высота элемента.



float8 MinHeight

## 1.1.4.2.4.10.10.1.5. Максимальная ширина

Максимальная ширина элемента.



float8 MaxWidth

## 1.1.4.2.4.10.10.1.6. Максимальная высота

Максимальная высота элемента.



float8 MaxHeight

## 1.1.4.2.4.10.10.1.7. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.10.1.8. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

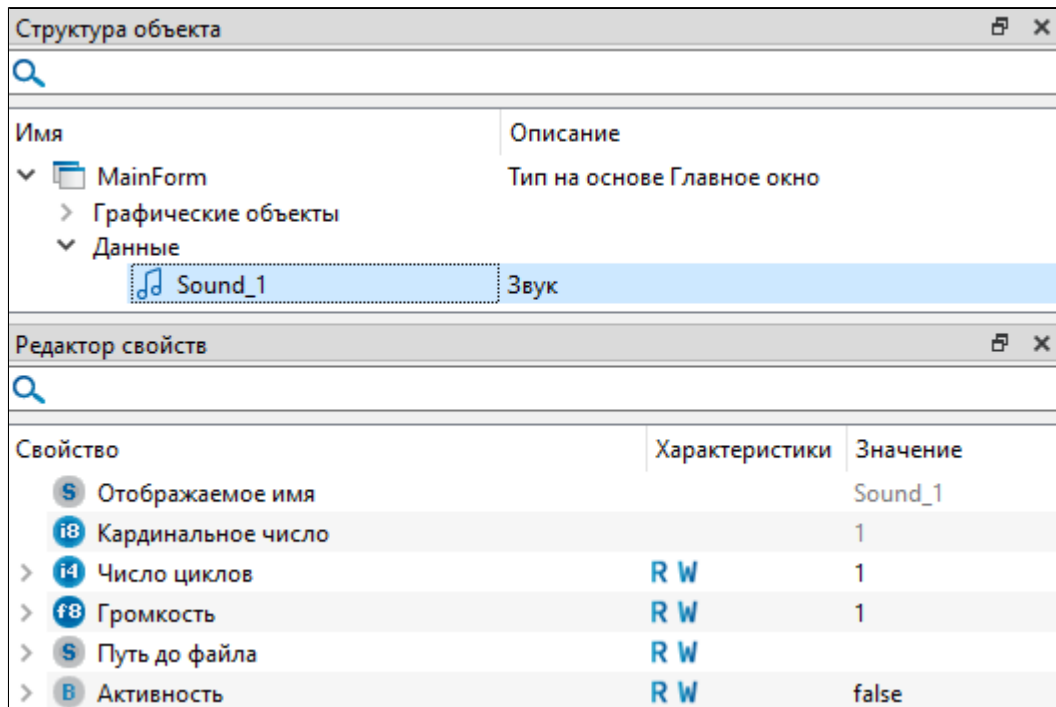


```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.11. Звук

Компонент для проигрывания звуковых файлов. Звуковые файлы должны быть в формате \*.wav и находиться в папке проекта resources.

Компонент не визуальный и виден только в области Структура объекта.



После добавления компонента Звук на экранную форму, укажите в свойстве Путь до файла название файла в формате \*.wav, который следует проигрывать. Звуковой файл должен находиться в папке проекта resources.

Чтобы звук начал проигрываться в режиме исполнения, переключите свойство Активность в состояние true.

Для проигрывания очереди звуков используйте обработчики событий [LoopPlayed](#) (завершение одного цикла проигрывания звука) или [AllLoopsPlayed](#) (завершение всех циклов проигрывания звука).

## 1.1.4.2.4.10.11.1. События

Событие	Описание
<a href="#">AllLoopsPlayed</a>	Завершение всех циклов проигрывания звука
<a href="#">LoopPlayed</a>	Завершение одного цикла проигрывания звука

## 1.1.4.2.4.10.11.1.1. AllLoopsPlayed

Завершение всех циклов проигрывания звука.



## 1.1.4.2.4.10.11.1.2. LoopPlayed

Завершение одного цикла проигрывания звука.

## 1.1.4.2.4.10.11.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Активность</a>	Начинает/прекращает проигрывание звукового файла
<a href="#">Путь до файла</a>	Путь к звуковому файлу относительно папки resources
<a href="#">Громкость</a>	Громкость проигрывания звукового файла
<a href="#">Число циклов</a>	Число повторений звукового файла

## 1.1.4.2.4.10.11.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.11.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.11.2.3. Число циклов

Число повторений звукового файла.



int4 LoopCount

## 1.1.4.2.4.10.11.2.4. Громкость

Громкость проигрывания звукового файла.



float8 Volume

Может принимать значения от 0 до 1.

- › 0 – отсутствие звука;
- › 1 – максимальная громкость, совпадающая с громкостью динамиков.

### Примеры



//Уменьшить громкость звука наполовину:

```
Sound.Volume = 0.5;
```

## 1.1.4.2.4.10.11.2.5. Путь до файла

Путь к звуковому файлу относительно папки resources.



string Path

## 1.1.4.2.4.10.11.2.6. АКТИВНОСТЬ

Начинает/прекращает проигрывание звукового файла.



bool Active

### Значение

Значение	Описание
true	Звук проигрывается
false	Звук не проигрывается



## 1.1.4.2.4.10.11.2.7. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.11.2.8. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

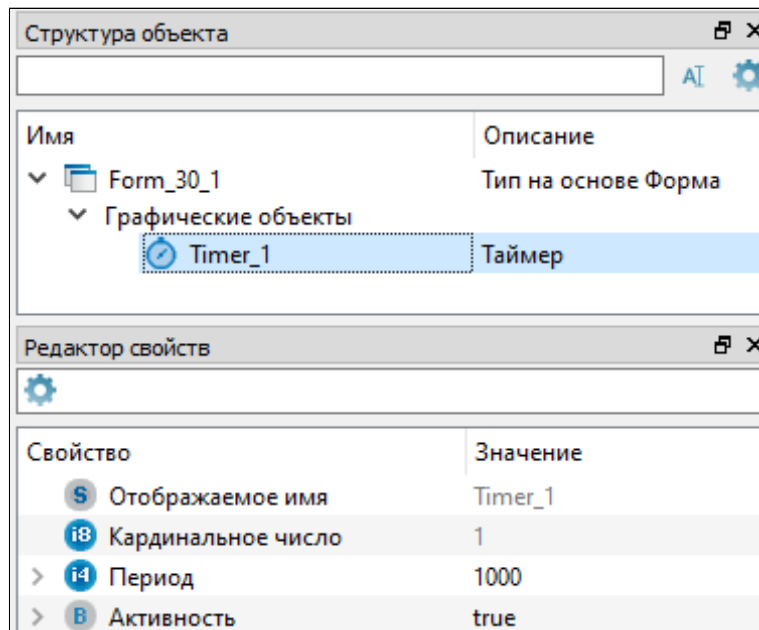


```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.12. Таймер

Компонент позволяет циклически выполнять процедуру (определяется в обработчике события OnTimer), которая повторяются с заданной периодичностью (свойство Период).

Компонент невизуальный и виден только в области Структура объекта.



## 1.1.4.2.4.10.12.1. События

Событие	Описание
<a href="#">OnTimer</a>	Событие, выполняемое по таймеру с заданным периодом

## 1.1.4.2.4.10.12.1.1. OnTimer

Событие, выполняемое по таймеру с заданным периодом.

### Параметры

Параметр	Тип	Описание
time	time stamp	Текущее время UTC

### Примеры



Применить анимацию в виде меняющихся цветов для линейного градиента. Скорость анимации зависит от периодичности таймера.

```
Object.LinearGradient_1.GradientPoint_1.Color +=1;  
Object.LinearGradient_1.GradientPoint_1.Color -=1;
```

## 1.1.4.2.4.10.12.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Now</a>	Текущее локальное время
<a href="#">Активность</a>	Активность таймера
<a href="#">Период</a>	Период исполнения события по таймеру

## 1.1.4.2.4.10.12.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.12.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.10.12.2.3. Период

Период исполнения события по таймеру.



int4 Period

## 1.1.4.2.4.10.12.2.4. Активность

Активность таймера.



bool Active

### Значение

Значение	Описание
true	Событие по таймеру будет исполняться
false	Событие по таймеру не будет исполняться

## 1.1.4.2.4.10.12.2.5. Now

Текущее локальное время.



Now



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.10.12.2.6. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.12.2.7. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

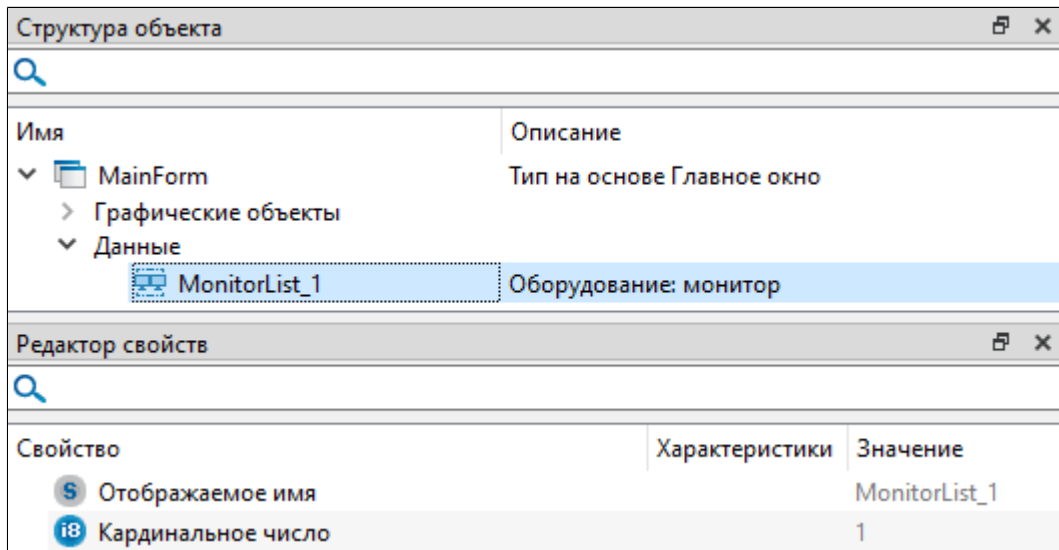


```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.13. Оборудование: монитор

Компонент позволяет получить доступ к функциям взаимодействия с мониторами.

Компонент не визуальный и виден только в области Структура объекта.



## 1.1.4.2.4.10.13.1. События

Событие	Описание
<a href="#">MonitorListChanged</a>	Обновление списка мониторов

## 1.1.4.2.4.10.13.1.1. MonitorListChanged

Обновление списка мониторов.



MonitorListChanged



# Функции

Компонент	Описание
<a href="#">Count</a>	Количество мониторов
<a href="#">GetResolutionX</a>	Разрешение монитора по X
<a href="#">GetResolutionY</a>	Разрешение монитора по Y
<a href="#">GetX</a>	Абсолютное положение по X
<a href="#">GetY</a>	Абсолютное положение по Y
<a href="#">GetAvailableWidth</a>	Ширина допустимой для пользователя области экрана
<a href="#">GetAvailableHeight</a>	Высота допустимой для пользователя области экрана
<a href="#">GetAvailableX</a>	Координата по оси X левого верхнего угла доступной для пользователя области экрана
<a href="#">GetAvailableY</a>	Координата по оси Y левого верхнего угла доступной для пользователя области экрана

## 1.1.4.2.4.10.13.2.1. Count

Возвращает количество мониторов на рабочем месте.



int4 Count()

### Примеры



```
//Отобразить в поле ввода количество мониторов на рабочем месте:  
TextEdit.Text = MonitorList.Count();
```

## 1.1.4.2.4.10.13.2.2. GetResolutionX

Вычисляют разрешение монитора по горизонтали в пикселях.



int4 GetResolutionX(int4 Number)

Нумерация мониторов начинается с 0 (0 - это крайний левый верхний монитор) и не зависит от нумерации мониторов в операционной системе. Направление нумерации происходит «сверху-вниз» и «слева-направо».



### Параметры

Параметр	Тип	Описание
Number	int4	Номер монитора

### Примеры



```
//Вывести в поля ввода разрешение монитора под номером 2:  
TextEdit1.Text = MonitorList.GetResolutionX(0);
```

## 1.1.4.2.4.10.13.2.3. GetResolutionY

Вычисляют разрешение монитора с номером Number по вертикали в пикселях.



int4 GetResolutionY(int4 Number)

Нумерация мониторов начинается с 0 (0 - это крайний левый верхний монитор) и не зависит от нумерации мониторов в операционной системе. Направление нумерации происходит «сверху-вниз» и «слева-направо».



### Параметры

Параметр	Тип	Описание
Number	int4	Номер монитора

### Примеры



```
//Вывести в поля ввода разрешение монитора под номером 2  
TextEdit2.Text = MonitorList.GetResolutionY(0);
```

## 1.1.4.2.4.10.13.2.4. GetX

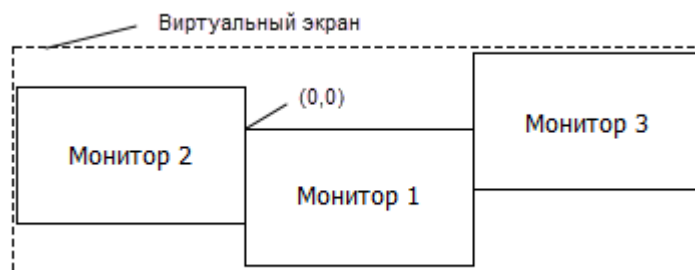
Вычисляет абсолютную координату монитора относительно многомониторной системы отображения.



int4 GetX(int4 Number)

Нумерация мониторов начинается с 0 (0 - это крайний левый верхний монитор) и не зависит от нумерации мониторов в операционной системе. Направление нумерации происходит «сверху-вниз» и «слева-направо».

Первый монитор в операционной системе содержит начало координат (0,0).



### Параметры

Параметр	Тип	Описание
Number	int4	Номер монитора

### Примеры



//Вывести координаты монитора 2 относительно многомониторной системы отображения:

```
TextEdit1.Text = MonitorList.GetX(0);
```

## 1.1.4.2.4.10.13.2.5. GetY

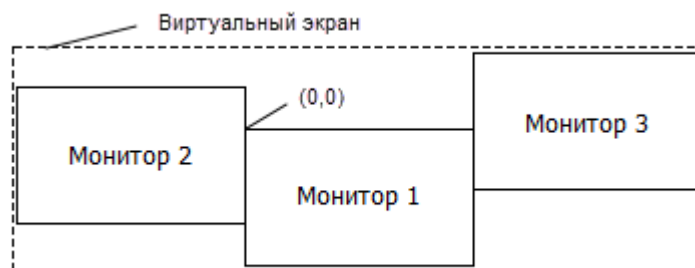
Вычисляют абсолютную координату монитора относительно многомониторной системы отображения.



int4 GetY(int4 Number)

Нумерация мониторов начинается с 0 (0 - это крайний левый верхний монитор) и не зависит от нумерации мониторов в операционной системе. Направление нумерации происходит «сверху-вниз» и «слева-направо».

Первый монитор в операционной системе содержит начало координат (0,0).



### Параметры

Параметр	Тип	Описание
Number	int4	Номер монитора

### Примеры



//Вывести координаты монитора 2 относительно многомониторной системы отображения:

```
TextEdit2.Text = MonitorList.GetY(0);
```

## 1.1.4.2.4.10.13.2.6. GetAvailableWidth

Ширина допустимой для пользователя области экрана. Это разрешение монитора по оси X без областей, занятых оконным менеджером, таких как панель задач и т. д.



int4 GetAvailableWidth(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	допустимая ширина экрана

## 1.1.4.2.4.10.13.2.7. GetAvailableHeight

Высота допустимой для пользователя области экрана. Это разрешение монитора по оси Y без областей, занятых оконным менеджером, таких как панель задач и т. д.



int4 GetAvailableHeight(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	допустимая высота экрана



## 1.1.4.2.4.10.13.2.8. GetAvailableX

Координата по оси X левого верхнего угла доступной для пользователя области экрана.



int4 GetAvailableX(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	координата по оси X левого верхнего угла

## 1.1.4.2.4.10.13.2.9. GetAvailableY

Координата по оси Y левого верхнего угла доступной для пользователя области экрана.



int4 GetAvailableY(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	координата по оси Y левого верхнего угла

## 1.1.4.2.4.10.13.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.13.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.13.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.13.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.13.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



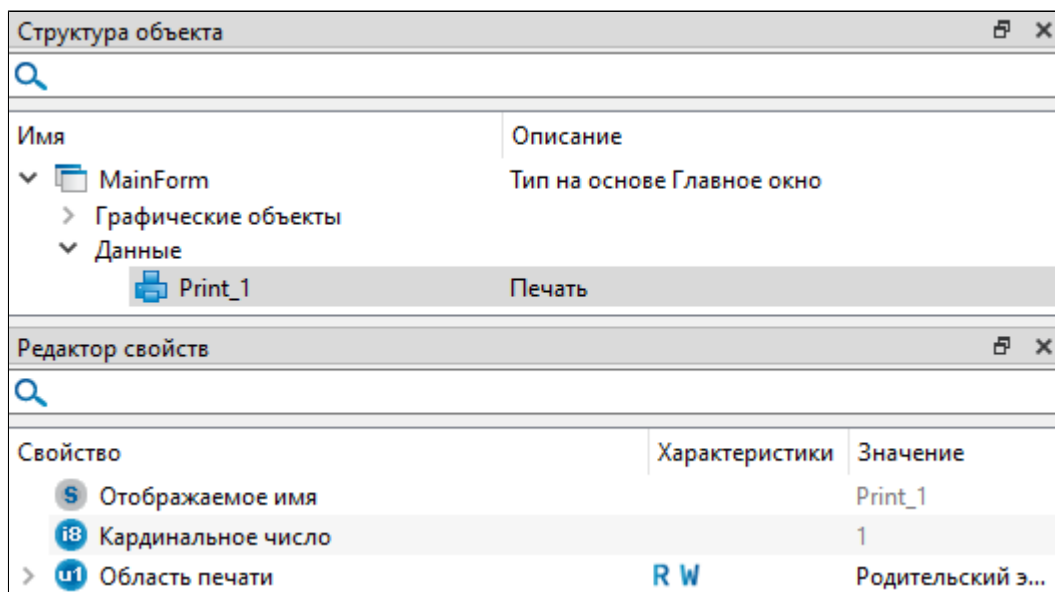
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.14. Печать

Компонент для вывода на печать.



The image shows two windows from Visual Studio. The top window, titled "Структура объекта", displays a tree view of the project structure. Under "MainForm", there is a folder "Графические объекты" containing a data item "Print\_1" with a printer icon and the description "Печать". The bottom window, titled "Редактор свойств", shows the properties for the selected "Print\_1" component. It has three properties: "Отображаемое имя" (Print\_1), "Кардинальное число" (1), and "Область печати" (Parental э...).

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Print_1	Печать

Свойство	Характеристики	Значение
Отображаемое имя		Print_1
Кардинальное число		1
Область печати	R W	Родительский э...



## Функции

Компонент	Описание
<a href="#">Print</a>	Печать на принтере по умолчанию
<a href="#">PrintDialog</a>	Показать диалог печати
<a href="#">PrintPreviewDialog</a>	Показать диалог предпросмотра печати

## 1.1.4.2.4.10.14.1.1. Print

Печать на принтере по умолчанию.



void Print()

## 1.1.4.2.4.10.14.1.2. PrintDialog

Показать диалог печати.



void PrintDialog()

## 1.1.4.2.4.10.14.1.3. PrintPreviewDialog

Показать диалог предпросмотра печати.



void PrintPreviewDialog(int4 Index)

### Параметры

Параметр	Тип	Описание
Index	int4	Значение расширения монитора по Y

## 1.1.4.2.4.10.14.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Область печати</a>	Область, отправляемая на печать

## 1.1.4.2.4.10.14.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.14.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.14.2.3. Область печати

Область, отправляемая на печать.



uint1 Area



## 1.1.4.2.4.10.14.2.4. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.14.2.5. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



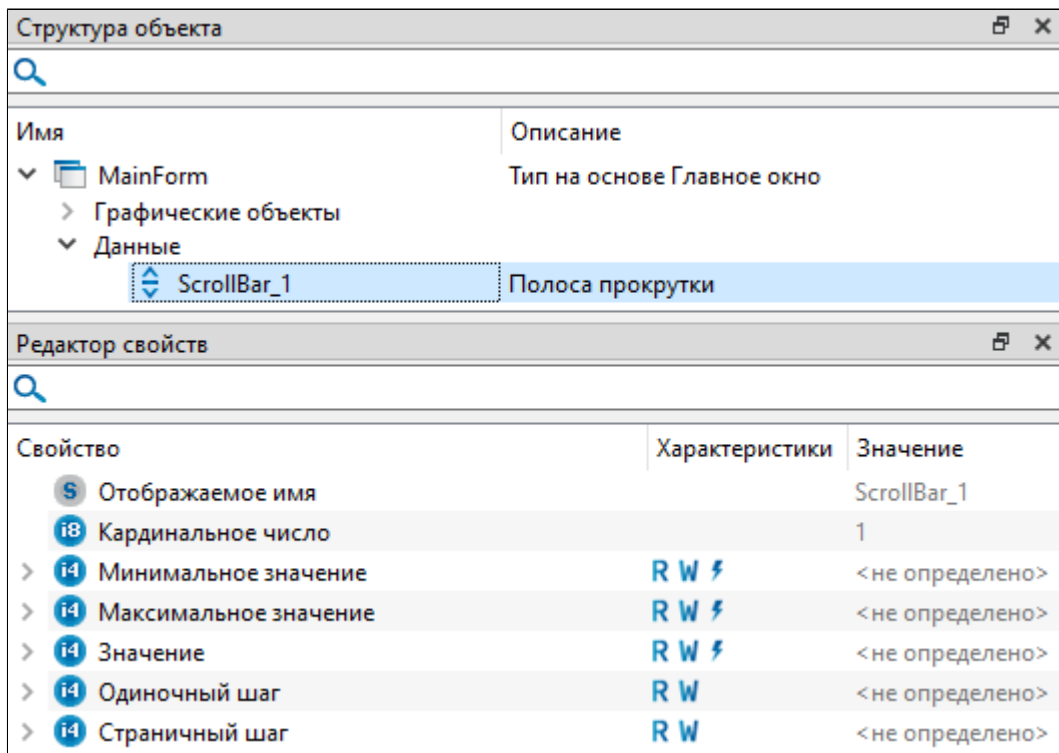
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.15. Полоса прокрутки

Компонент для прокрутки в области рабочего окна.



The image shows two windows from a development environment. The top window, titled "Структура объекта" (Object Structure), displays a tree view of the project. Under "MainForm", there are "Графические объекты" (Visual Objects) and "Данные" (Data). Under "Данные", the "ScrollBar\_1" component is selected and highlighted in blue. The bottom window, titled "Редактор свойств" (Properties Editor), shows the properties for the selected "ScrollBar\_1" component. It has a search bar and a table with columns for "Свойство" (Property), "Характеристики" (Characteristics), and "Значение" (Value).

Свойство	Характеристики	Значение
Отображаемое имя		ScrollBar_1
Кардинальное число		1
Минимальное значение	R W ⚡	< не определено >
Максимальное значение	R W ⚡	< не определено >
Значение	R W ⚡	< не определено >
Одиночный шаг	R W	< не определено >
Страничный шаг	R W	< не определено >

## 1.1.4.2.4.10.15.1. События

Событие	Описание
<a href="#">MinimumChanged</a>	Изменение минимального значения
<a href="#">MaximumChanged</a>	Изменение максимального значения
<a href="#">ValueChanged</a>	Изменение значения

## 1.1.4.2.4.10.15.1.1. MinimumChanged

Изменение минимального значения.



MinimumChanged

## 1.1.4.2.4.10.15.1.2. MaximumChanged

Изменение максимального значения.



MaximumChanged

## 1.1.4.2.4.10.15.1.3. ValueChanged

Изменение значения.



ValueChanged

## 1.1.4.2.4.10.15.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Минимальное значение</a>	Минимальное значение полосы прокрутки
<a href="#">Максимальное значение</a>	Максимальное значение полосы прокрутки
<a href="#">Значение</a>	Значение полосы прокрутки
<a href="#">Одиночный шаг</a>	Одиночный шаг прокрутки
<a href="#">Страничный шаг</a>	Страничный шаг прокрутки
<a href="#">Orientation</a>	Ориентация



## 1.1.4.2.4.10.15.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.15.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.15.2.3. Минимальное значение

Минимальное значение полосы прокрутки.



int4 Minimum

## 1.1.4.2.4.10.15.2.4. Максимальное значение

Максимальное значение полосы прокрутки.



int4 Maximum

## 1.1.4.2.4.10.15.2.5. Значение

Значение полосы прокрутки.



int4 Value

## 1.1.4.2.4.10.15.2.6. Одиночный шаг

Одиночный шаг прокрутки.



int4 SingleStep

## 1.1.4.2.4.10.15.2.7. Страничный шаг

Страничный шаг прокрутки.



int4 PageStep

## 1.1.4.2.4.10.15.2.8. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.10.15.2.9. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.15.2.10. Orientation

Ориентация.



int4 Orientation



Доступно только для чтения в режиме рантайма.

## 1.1.4.2.4.10.16. Линия

Компонент Линия применяется на мнемосхемах при рисовании объектов технологического процесса.

Можно создавать следующие типы объектов:

- › прямые линии;
- › ломаные линии;
- › многоугольники.



Каждый объект, созданный с помощью элемента Линия представлен набором точек, связь которых образует объект.

Структура объекта

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Button_1	Кнопка
Line_1	Линия
Точки	
Point_1	Точка
Point_2	Точка

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		Line_1
Кардинальное число		1
X	R W	982
Y	R W	235
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	190
Высота	R W	115
Фокус ввода	R W	<не определено>
Цвет пера	R W	4278190080
Стиль пера	R W	Сплошная линия
Толщина пера	R W	1
Цвет заливки	R W	4278190080
Стиль заливки	R W	Нет заливки

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана
<a href="#">TouchPress</a>	Нажатие на объект

## 1.1.4.2.4.10.16.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.16.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.16.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.16.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.2.4.10.16.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;
```

```
Ellipse_1.Y = 341;
```

```
SW_1.OpenSW = 1;
```

## 1.1.4.2.4.10.16.1.12. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.16.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Свойства точек</a>	Свойство X и свойство Y точек
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Отражение</a>	Отражение объекта

## 1.1.4.2.4.10.16.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.16.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.16.2.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

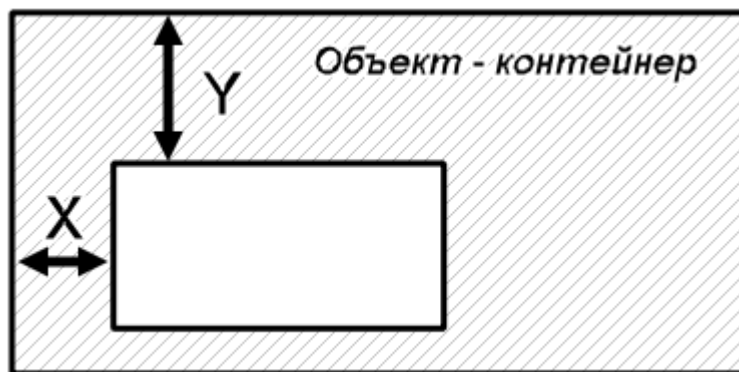
## 1.1.4.2.4.10.16.2.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.16.2.5. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.2.4.10.16.2.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

# 1.1.4.2.4.10.16.2.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

## Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.16.2.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.16.2.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.10.16.2.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.16.2.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.16.2.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

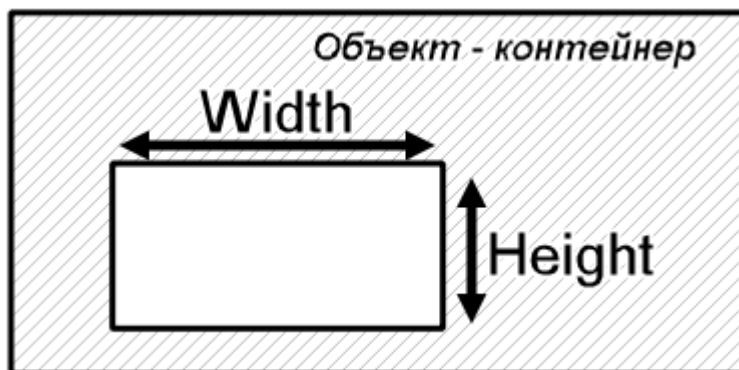
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.16.2.13. Ширина

Ширина объекта в пикселях.



float8 Width



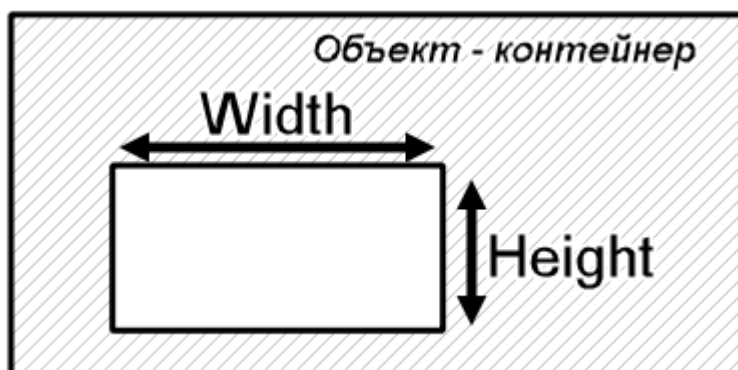


## 1.1.4.2.4.10.16.2.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.16.2.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.16.2.16. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.2.4.10.16.2.17. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;
```

## 1.1.4.2.4.10.16.2.18. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.16.2.19. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.BrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.BrushColor = 4294901760;
```



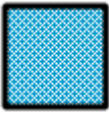

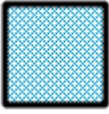

## 1.1.4.2.4.10.16.2.20. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры

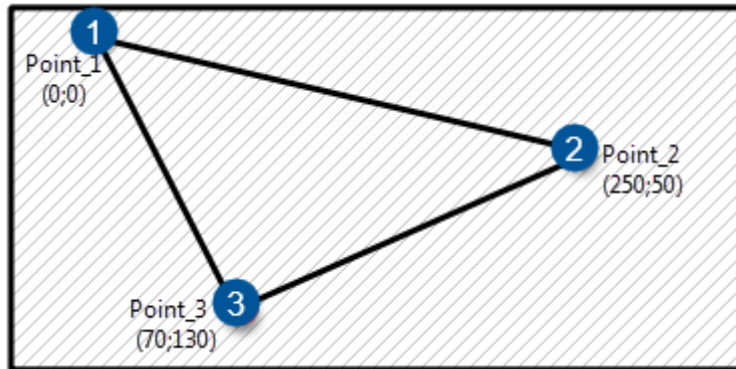


//Залить объект сплошным цветом:  
Object.BrushStyle = 1;



## 1.1.4.2.4.10.16.2.21. Свойства точек

Каждая точка, в составе многоугольника имеет координаты: свойство X и свойство Y. Началом отсчета координат (точка - 0;0) считается первая точка многоугольника. Координаты остальных точек отсчитываются относительно первой точки.



### Примеры



```
//Изменить координаты точки 2 относительно координат точки 1:  
Figure.Point_2.X = Figure.Point_1.X + 250;  
Figure.Point_2.Y = Figure.Point_1.Y + 50;
```

## 1.1.4.2.4.10.16.2.22. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.16.2.23. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.17. Прямоугольник

Компонент Прямоугольник применяется на мнемосхемах при рисовании объектов технологического процесса и, в некоторых случаях, как альтернатива Кнопке, так как имеет более широкие возможности для обработчиков событий.

The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project's components. Under 'Графические объекты' (Graphic Objects), 'Rectangle\_1' is selected. The bottom window, titled 'Редактор свойств' (Properties Editor), shows a list of properties for the selected 'Rectangle\_1' object.

Свойство	Характеристики	Значение
Отображаемое имя		Rectangle_1
Кардинальное число		1
X	R W	821
Y	R W	220
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	100
Высота	R W	100
Фокус ввода	R W	<не определено>
Радиус скругления	R W	0
Цвет пера	R W	4278190080
Стиль пера	R W	Сплошная линия
Толщина пера	R W	1
Цвет заливки	R W	4278190080
Стиль заливки	R W	Нет заливки

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана
<a href="#">TouchPress</a>	Нажатие на объект

## 1.1.4.2.4.10.17.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.17.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.17.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.17.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.2.4.10.17.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.2.4.10.17.1.12. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

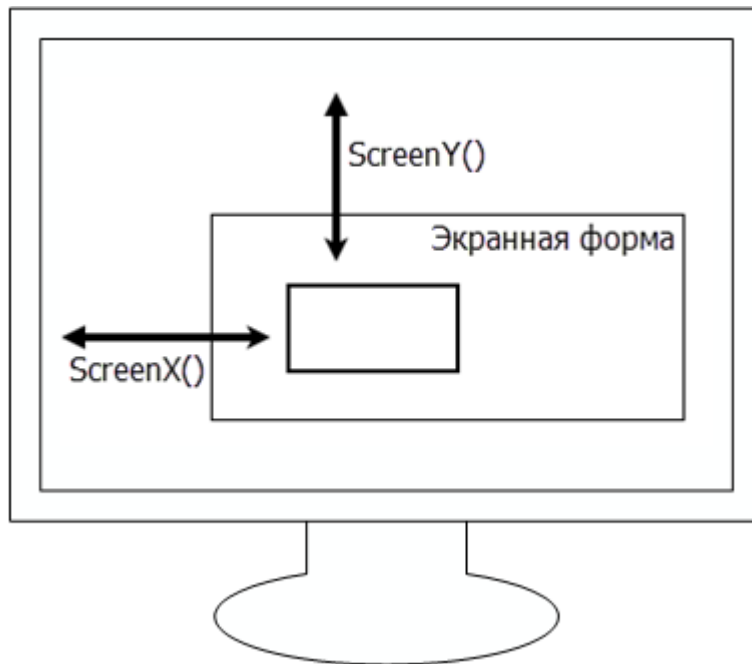
```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.17.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



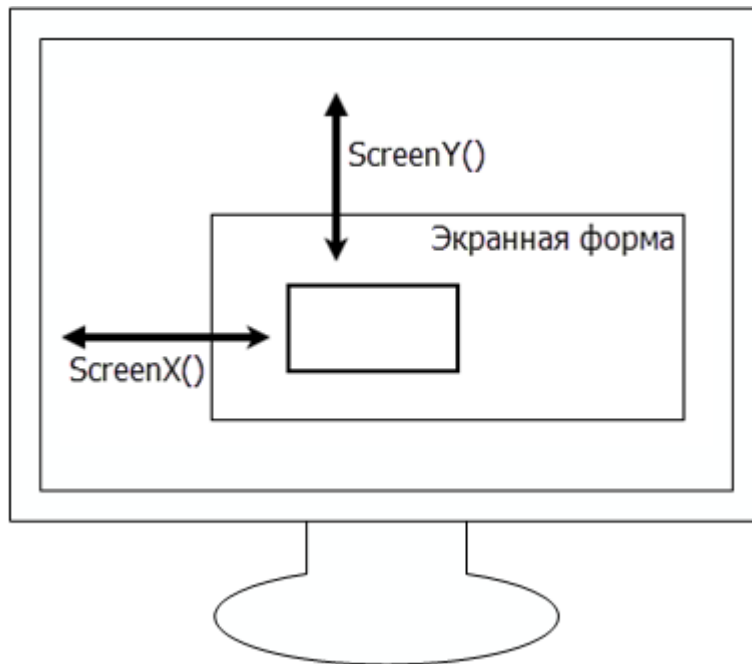
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.17.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



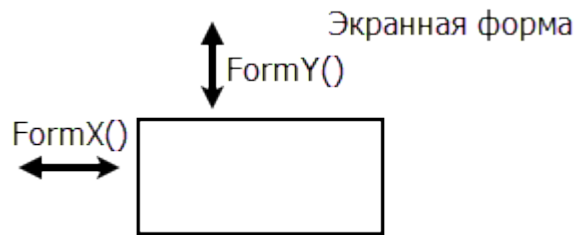
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.17.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



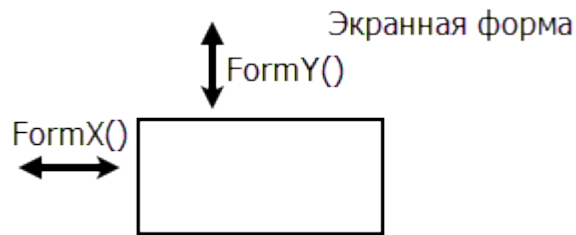
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.17.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.17.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника. Измеряется в пикселях
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта



<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.17.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.17.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

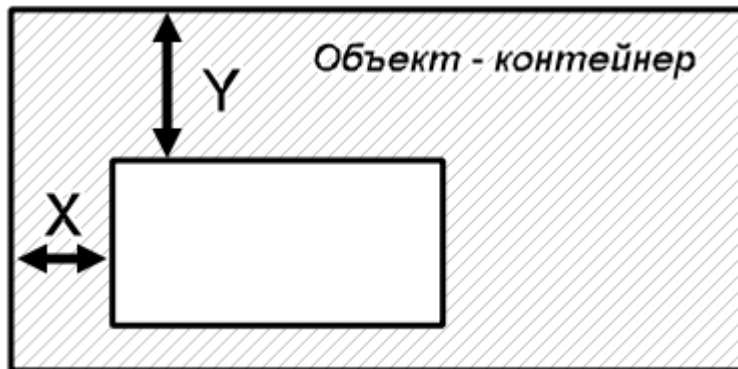
## 1.1.4.2.4.10.17.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

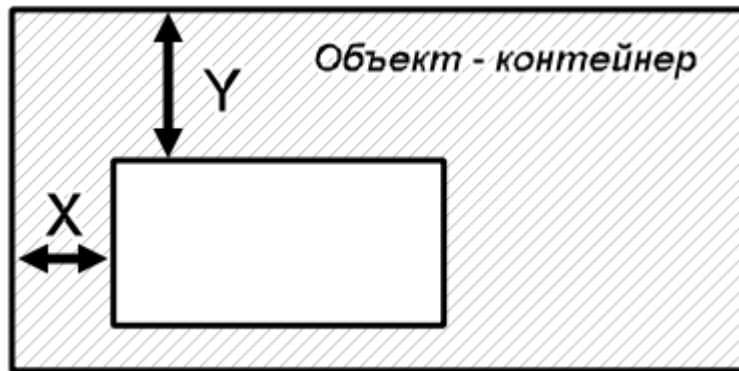
## 1.1.4.2.4.10.17.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.17.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.17.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.17.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```



## 1.1.4.2.4.10.17.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.17.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.10.17.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.17.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.17.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

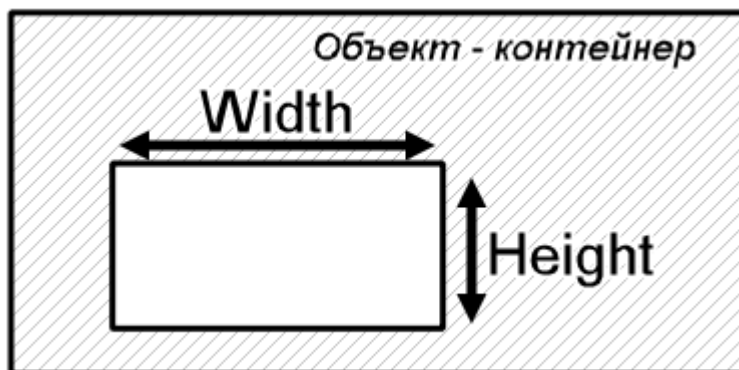
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.17.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

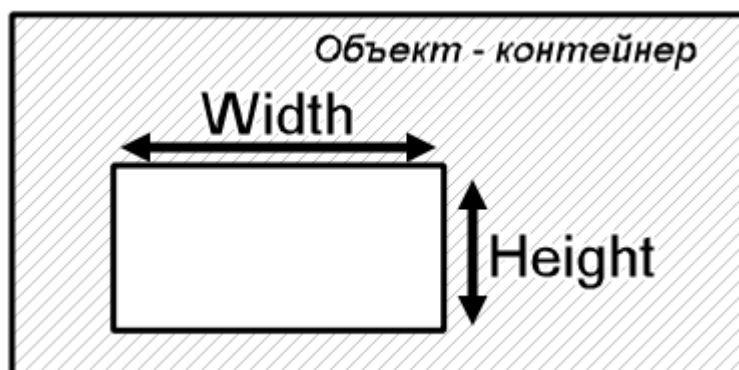


## 1.1.4.2.4.10.17.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.17.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

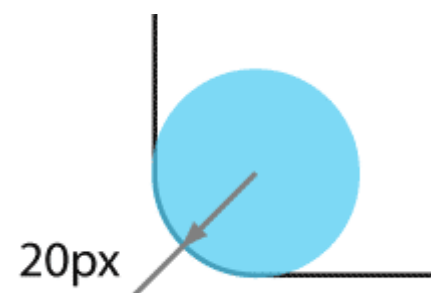


## 1.1.4.2.4.10.17.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.2.4.10.17.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.2.4.10.17.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.17.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.17.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.BrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.BrushColor = 4294901760;
```




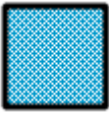

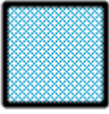

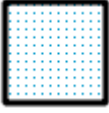
## 1.1.4.2.4.10.17.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



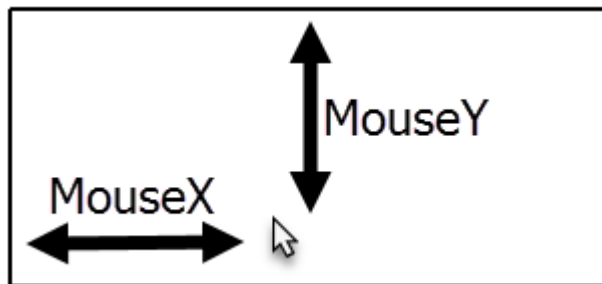
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.4.10.17.3.22. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

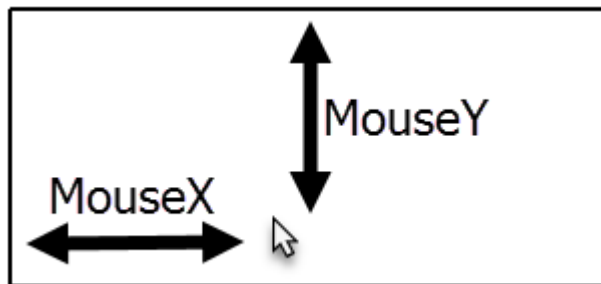


## 1.1.4.2.4.10.17.3.23. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.17.3.24. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.17.3.25. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.18. Перемещение

При работе в рантайм из объекта в объект можно перетаскивать данные. Перетаскивать данные можно внутри одного проекта либо между объектами разных проектов.

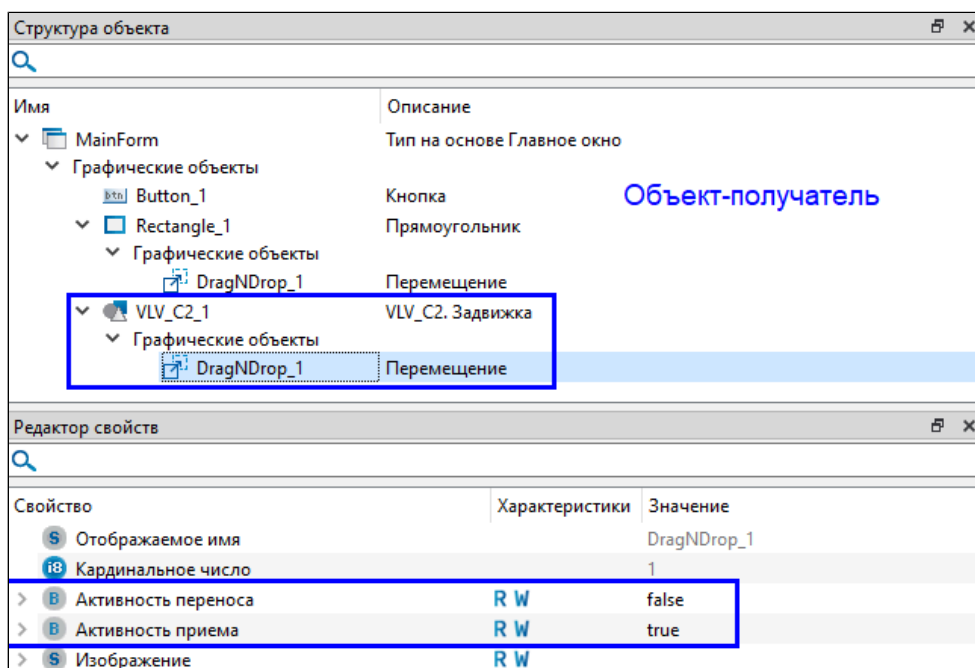
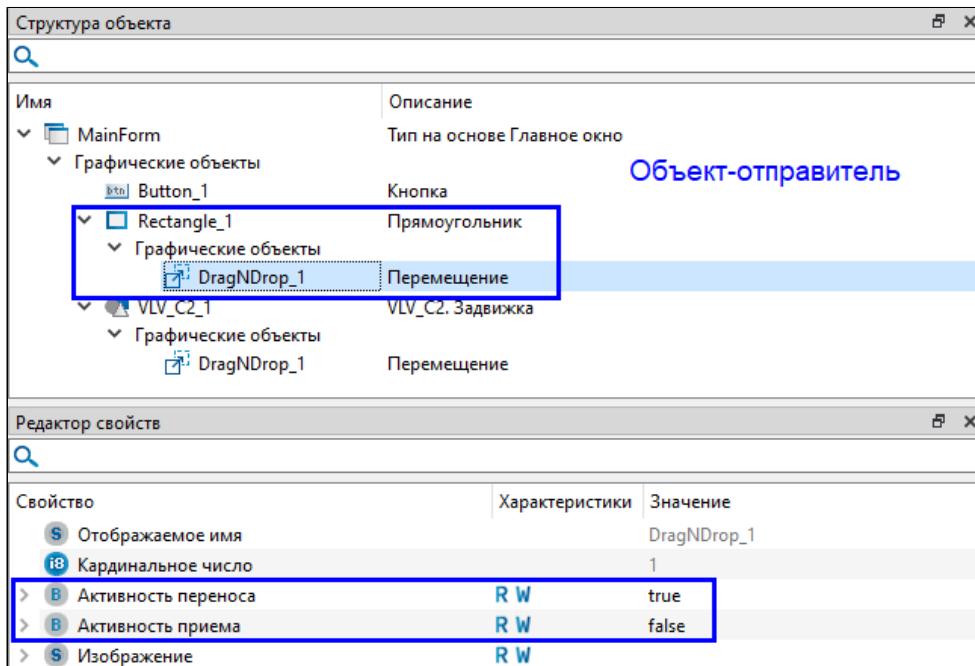
Вы можете передавать между объектами:

- › значения свойств объектов/отдельных элементов проекта;
- › произвольный текст.

Данные, полученные от другого объекта, можно использовать как угодно, настроив их обработку.

Вы можете перетаскивать одни и те же данные с одного объекта-отправителя на разные объекты-получатели. При этом любой объект может одновременно быть и отправителем и получателем.

Чтобы добавить объекту возможность передавать/принимать данные, добавьте ему дочерний элемент Перемещение и разрешите отправку/прием данных с помощью свойств Активность переноса и Активность приема элемента Перемещение.



Чтобы указать изображение, отображаемое рядом с курсором при перетаскивании, перейдите к объекту-отправителю и укажите название файла в свойстве Изображение элемента Перемещение. По умолчанию при переносе отображается изображение объекта-отправителя.

Чтобы создать иллюзию перетаскивания данных, используйте обработчики событий MousePress объекта-отправителя и OnDrop элемента Перемещение на стороне объекта-получателя. Укажите передаваемые данные в обработчике события MousePress и настройте обработку принимаемых данных в обработчике события OnDrop. Для перетаскивания щелкните на объекте-

отправителе и не отпуская клавишу мыши, перетащите данные на объект-получатель.

При перетаскивании на сторону объекта-получателя "сбрасываются" абсолютно все данные, заложенные в обработчик события `MouseDown` объекта-отправителя. Какие из "сброшенных" данных будут приняты объектом-получателем, зависит от настройки обработчика события `OnDrop`.

Перед указанием передаваемых данных на стороне объекта-отправителя оцените объем и содержание передаваемых данных и выберите способ передачи.

Способы передачи данных между объектами:

- › передача данных единым блоком;
- › передача данных частями;
- › передача массива бинарных данных.

## Передача данных единым блоком

Если вы намерены передать большой блок данных (например, произвольный текст), который целиком нужен каждому объекту-получателю, передавайте данные одним блоком. Для передачи используйте функции [SetText](#) ([SetHtml](#)) элемента Перемещение на стороне объекта-отправителя и функции [GetText](#) ([GetHtml](#)) параметра `data` события [OnDrop](#) элемента Перемещение на стороне объекта-получателя.

Чтобы настроить передачу единого блока данных, выполните:

1. Для объекта-отправителя и объекта-получателя добавьте дочерний элемент Перемещение.
2. В обработчике события `MouseDown` объекта-отправителя укажите передаваемые данные с помощью функции `SetText` (`SetHtml`) элемента Перемещение.

```
1 //передать теги сигналов датчика P_01
2 P_01.DragNDrop_1.SetText("P_01.Pressure.Value\r
  \nP_01.Pressure.Delta.Value\r\nP_01.Pressure.Speed.Value\r
  \nP_01.SensorQ.Q\r\nP_01.SensorQ.Qdiff\r\nP_01.SensorDensity.Density");
3 SW_1.DragNDrop_1.DoDragDrop();
```

100% < >

Язык: Om OK Отмена

```
1 //передать высоту прямоугольника
2 DragNDrop_1.SetText(Str.ToString(Rectangle_1.Height));
3 DragNDrop_1.DoDragDrop();
```

100% < >

Язык: Om OK Отмена

3. Для запуска перемещения данных используйте функцию [DoDragDrop](#) элемента Перемещение. Вызывайте функцию сразу же после вызова функции [SetText \(SetHtml\)](#) (см. рисунок выше).

4. В обработчике события [OnDrop](#) элемента Перемещение разрешите объекту-получателю принимать любой текстовый блок с помощью функции [GetText \(GetHtml\)](#) параметра data.

Пропишите как вы будете использовать полученный блок данных. Вы можете по своему усмотрению настроить обработку полученного блока так, чтобы блок после получения разбивался на части. Для этого на стороне объекта-получателя запишите передаваемые данные в виде JSON-структуры или строки с разделителями.

```
1 //записывать текст, полученный от любого объекта, в
поле ввода
2 TextEdit_2.Text := data.GetText();
```

100% ▾

Язык: От ▾

OK Отмена

## Передача данных частями

Если каждому из объектов-получателей нужна только определённая часть из всего передаваемого блока, либо вы хотите разбить блок данных на части перед отправкой, используйте функцию [SetData](#) элемента Перемещение на стороне объекта-отправителя. С помощью функции [SetData](#) каждой отдельной части можно присвоить тип (идентификатор). Типы позволяют ограничить объем принимаемых данных на объекте-получателе.

Используйте функцию [AddAcceptedFormat](#) элемента Перемещение, чтобы указать типы, разрешённые для приёма. При получении блока данных объект-получатель ориентируется на список разрешённых принимаемых типов и принимает только типы из списка. Для приёма типов на стороне объекта-получателя используйте функцию [GetData](#) параметра data события [OnDrop](#) элемента Перемещение.

Чтобы настроить передачу данных по частям, выполните:

1. Для объекта-отправителя и объекта-получателя добавьте дочерний элемент Перемещение.
2. В обработчике события MousePress объекта-отправителя укажите передаваемые данные с помощью функции [SetData](#) элемента Перемещение. Для каждой отдельной части используйте свою функцию [SetData](#) и присваивайте свой тип (идентификатор).



```
1 Rectangle_1.Text_3.DragNDrop_out.SetData("value",Text_3.Text);¶
2 Rectangle_1.Text_3.DragNDrop_out.SetData("width",Str.ToString(Text_3.Width));¶
3 Rectangle_1.Text_3.DragNDrop_out.SetData("height",Str.ToString(Text_3.Height));¶
4 Text_3.DragNDrop_out.DoDragDrop();
```

100% ▾

Язык: Om ▾ OK Отмена

3. Для запуска перемещения данных используйте функцию [DoDragDrop](#) элемента Перемещение. Вызывайте функцию сразу же после вызова функции [SetData](#) (см. рисунок выше).

4. Составьте список разрешённых принимаемых типов для объекта-получателя с помощью функции [AddAcceptedFormat](#) элемента Перемещение. Необходимо, чтобы список был обработан перед сбросом данных на объект-получатель. К примеру, укажите список принимаемых типов в обработчике события `Opened` для формы.

```
1 Rectangle_2.DragNDrop_in.AddAcceptedFormat("value");¶
2 Rectangle_2.DragNDrop_in.AddAcceptedFormat("width");¶
3 Rectangle_2.DragNDrop_in.AddAcceptedFormat("height");
```

100% ▾

Язык: Om ▾ OK Отмена

5. В обработчике события [OnDrop](#) элемента Перемещение настройте обработку полученных типов с помощью функции `GetData` параметра `data`. Для каждого принимаемого типа используйте свою функцию `GetData`.

```
1  TextEdit_1.Text := .data.GetData("value");
2  TextEdit_2.Text := .data.GetData("width");
3  TextEdit_3.Text := .data.GetData("Height");
```

100% ▾

Язык:

## Передача бинарных данных

Если вы хотите передать блок бинарных данных, передавайте такие данные в виде массива. Имя массива и его размер указывайте с помощью функции [SetDataSize](#). Для указания передаваемых данных используйте функцию [SetDataAt](#) элемента Перемещение на стороне объекта-отправителя. Для получения размера массива на стороне объекта-получателя используйте функцию [GetDataSize](#) параметра data события [OnDrop](#) элемента Перемещение. Для получения значений из массива используйте функцию [GetDataSize](#) параметра data события [OnDrop](#) элемента Перемещение.

Чтобы настроить передачу данных в виде массива, выполните:

1. Для объекта-отправителя и объекта-получателя добавьте дочерний элемент Перемещение.
2. В обработчике события `MouseDown` объекта-отправителя укажите размер и имя массива с помощью функции [SetDataSize](#). Укажите передаваемые данные с помощью функции [SetDataAt](#) элемента Перемещение. Для каждого элемента массива используйте свою функцию [SetDataAt](#).

```
1 Rectangle_3.DragNDrop_1.SetDataSize(Type, -4);  
2 Rectangle_3.DragNDrop_1.SetDataAt(Type, 0, -State);  
3 Rectangle_3.DragNDrop_1.SetDataAt(Type, 1, -Energy);  
4 Rectangle_3.DragNDrop_1.SetDataAt(Type, 2, -Model);  
5 Rectangle_3.DragNDrop_1.SetDataAt(Type, 3, -Mode2);  
6 Rectangle_3.DragNDrop_1.DoDragDrop();
```

100% ▾

Язык: JavaScript ▾

3. Для запуска перемещения данных используйте функцию [DoDragDrop](#) элемента Перемещение. Вызывайте функцию сразу же после вызова функции [SetData](#) (см. рисунок выше).

4. На стороне объекта-получателя в обработчике события [OnDrop](#) элемента Перемещение настройте обработку полученного массива с помощью функции [GetDataAt](#) параметра data. Для каждого элемента массива используйте свою функцию [GetDataAt](#).

```
1 State := ..data.GetDataAt(Type, -0) != -0 ? true : false;  
2 Energy := ..data.GetDataAt(Type, -1) != -0 ? true : false;  
3 Model := ..data.GetDataAt(Type, -2) != -0 ? true : false;  
4 Mode2 := ..data.GetDataAt(Type, -3) != -0 ? true : false;
```

100% ▾

Язык: Om ▾

# События

Событие	Описание
<a href="#">OnDrop</a>	Сброс данных

## 1.1.4.2.4.10.18.1.1. OnDrop

Данные сброшены.

## Функции

Компонент	Описание
<a href="#">AddAcceptedFormat</a>	Добавить применяемый тип
<a href="#">ResetAcceptedFormat</a>	Очистить список принимаемых типов
<a href="#">SetText</a>	Установить текст
<a href="#">SetHtml</a>	Установить Html
<a href="#">SetData</a>	Установить данные
<a href="#">SetDataSize</a>	Установить размер данных
<a href="#">SetDataAt</a>	Установить значение данных по индексу
<a href="#">Append</a>	Добавить значение в конец данных
<a href="#">Clear</a>	Очистить содержимое
<a href="#">DoDragDrop</a>	Совершить перенос

## 1.1.4.2.4.10.18.2.1. AddAcceptedFormat

Добавить применяемый тип.



```
void AddAcceptedFormat(string _1)
```

### Параметры

Параметр	Тип	Описание
_1	string	

## 1.1.4.2.4.10.18.2.2. ResetAcceptedFormat

Очистить список принимаемых типов.



```
void ResetAcceptedFormat()
```



## 1.1.4.2.4.10.18.2.3. SetText

Установить текст.



```
void SetText(string _1)
```

### Параметры

Параметр	Тип	Описание
_1	string	

## 1.1.4.2.4.10.18.2.4. SetHtml

Установить Html.



void SetHtml(string \_1)

### Параметры

Параметр	Тип	Описание
_1	string	

## 1.1.4.2.4.10.18.2.5. SetData

Установить данные.



```
void SetData(string _1, string _2)
```

### Параметры

Параметр	Тип	Описание
_1	string	
_2	string	

## 1.1.4.2.4.10.18.2.6. SetDataSize

Установить размер данных.



```
void SetDataSize(string _1, uint4 _2)
```

### Параметры

Параметр	Тип	Описание
_1	string	
_2	uint4	

## 1.1.4.2.4.10.18.2.7. SetDataAt

Установить значение данных по индексу.



```
void SetDataAt(string _1, uint4 _2, uint1 _3)
```

### Параметры

Параметр	Тип	Описание
_1	string	
_2	uint4	
_3	uint1	

## 1.1.4.2.4.10.18.2.8. Append

Добавить значение в конец данных.



```
void Append(string _1, uint1 _2)
```

### Параметры

Параметр	Тип	Описание
_1	string	
_2	uint1	

## 1.1.4.2.4.10.18.2.9. Clear

Очистить содержимое.



void Clear()

## 1.1.4.2.4.10.18.2.10. DoDragDrop

Совершить перенос.



```
void DoDragDrop()
```



## 1.1.4.2.4.10.18.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Активность переноса</a>	Активность переноса
<a href="#">Активность приема</a>	Активность приема
<a href="#">Изображение</a>	Изображение, отображаемое при переносе
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.18.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.18.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.18.3.3. Активность переноса

Активность переноса.



bool DragEnabled

## 1.1.4.2.4.10.18.3.4. Активность приема

Активность приема.



bool DropEnabled

## 1.1.4.2.4.10.18.3.5. Изображение

Изображение, отображаемое при переносе.



string Image

## 1.1.4.2.4.10.18.3.6. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.18.3.7. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.10.19. Кнопка

Компонент Кнопка применяется на мнемосхемах для подачи управляющих команд или взаимодействия с пользователем в диалоговых окнах. Действия, происходящие при нажатии на кнопку, определяются в обработчике события [ButtonPressed](#).

The screenshot displays two windows from the Visual Studio IDE:

- Структура объекта (Object Structure):** Shows a tree view where 'MainForm' is expanded to 'Графические объекты' (Graphics Objects), and 'Button\_1' is selected. The description for 'Button\_1' is 'Кнопка' (Button).
- Редактор свойств (Properties):** Shows a list of properties for the selected 'Button\_1' control. The properties are organized into columns: 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		Button_1
Кардинальное число		1
X	R W	676
Y	R W	-190
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	75
Высота	R W	23
Фокус ввода	R W	< не определено >
Удерживаемая	R W	false
Нажата	R W	false
Текст	R W	Кнопка
Выравнивание текста	R W	По центру
Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0,0
Цвет шрифта	R W	4278190080
Цвет шрифта при нажатии	R W	4278190080
Цвет шрифта при наведении мыши	R W	4278190080

# События

Событие	Описание
<a href="#">ButtonPressed</a>	Нажатие на кнопку
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.19.1.1. ButtonPressed

Нажатие на кнопку.

### Параметры

Параметр	Тип	Описание
checked	bool	Указывает была ли нажата (true) или отпущена (false) кнопка

## 1.1.4.2.4.10.19.1.2. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.19.1.3. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.4. MouseEnter

[Ввод курсора в область объекта.](#)

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.5. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.19.1.6. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.7. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.8. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.9. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.10. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.19.1.11. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.19.1.12. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.19.1.13. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

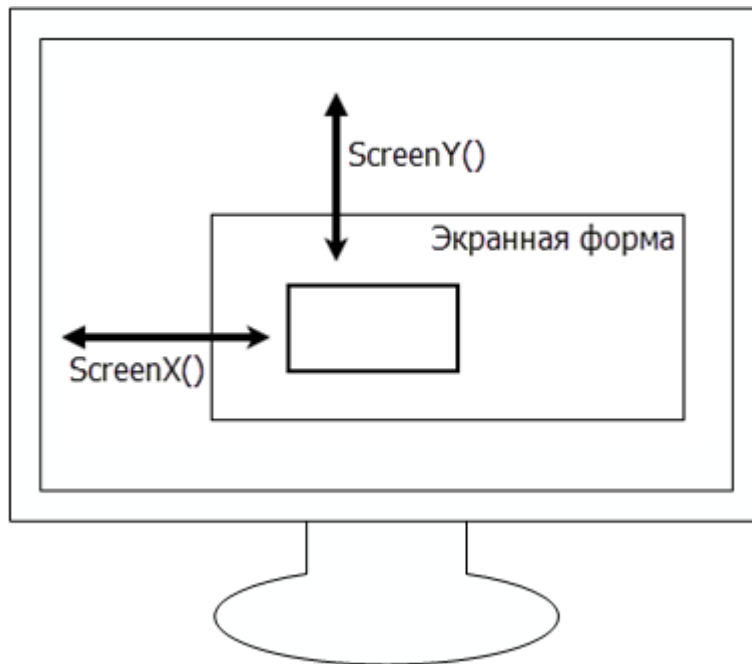


# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">Click</a>	Имитировать нажатие кнопки

## 1.1.4.2.4.10.19.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



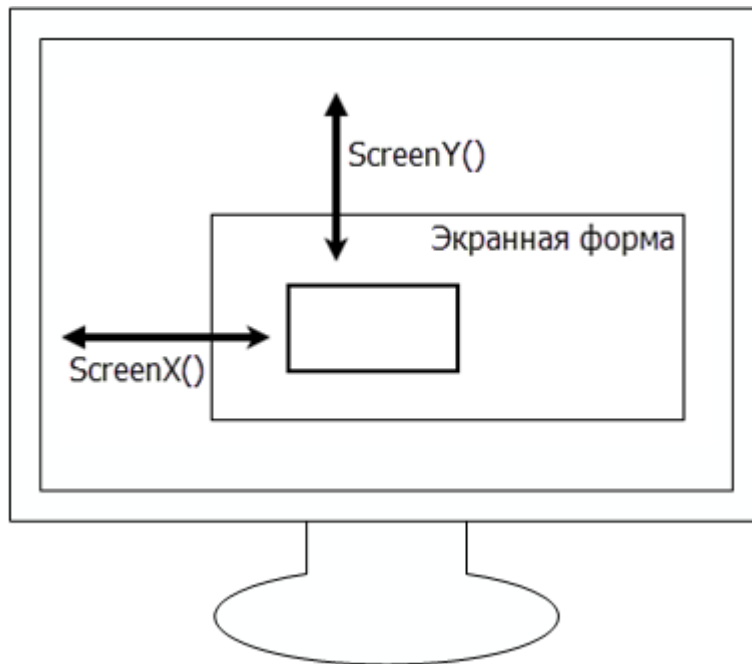
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.19.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



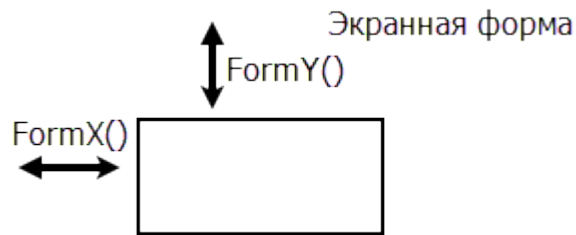
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.19.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



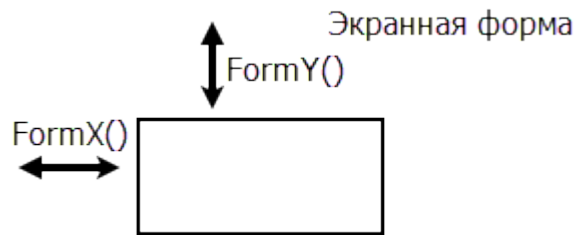
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.19.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.19.2.5. Click

Имитировать нажатие кнопки.



void Click()

## 1.1.4.2.4.10.19.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Толщина заливки</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта

<a href="#">Стиль пера при наведении мыши</a>	Стиль внешней границы объекта при вводе курсора в область объекта
<a href="#">Цвет пера при наведении мыши</a>	Цвет внешней границы объекта при вводе курсора в область объекта
<a href="#">Толщина пера при нажатии</a>	Толщина внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Стиль пера при нажатии</a>	Стиль внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Цвет пера при нажатии</a>	Цвет внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Стиль заливки при наведении мыши</a>	Заливает объект одним из стилей заливки при вводе курсора в область объекта
<a href="#">Цвет заливки при наведении мыши</a>	Цвет внутренней заливки объекта при вводе курсора в область объекта
<a href="#">Стиль заливки при нажатии</a>	Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту
<a href="#">Цвет заливки при нажатии</a>	Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту
<a href="#">Цвет шрифта при наведении мыши</a>	Цвет шрифта при наведении мыши
<a href="#">Цвет шрифта при нажатии</a>	Цвет шрифта при нажатии
<a href="#">Цвет шрифта</a>	Цвет шрифта
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Выравнивание текста</a>	Выравнивание текста
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Удерживаемая</a>	Оставляет кнопку удерживаемой/отпускаемой после нажатия
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Нажата</a>	Кнопка нажата



<a href="#">Цвет шрифта при выключении</a>	Цвет шрифта при выключении
<a href="#">Цвет заливки при выключении</a>	Цвет заливки при выключении
<a href="#">Стиль заливки при выключении</a>	Стиль заливки при выключении
<a href="#">Толщина пера при наведении мыши</a>	Толщина пера при наведении мыши
<a href="#">Цвет пера при выключении</a>	Цвет пера при выключении
<a href="#">Стиль пера при выключении</a>	Стиль пера при выключении
<a href="#">Толщина пера при выключении</a>	Толщина пера при выключении
<a href="#">Z-значение</a>	Z-значение
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.19.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.19.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

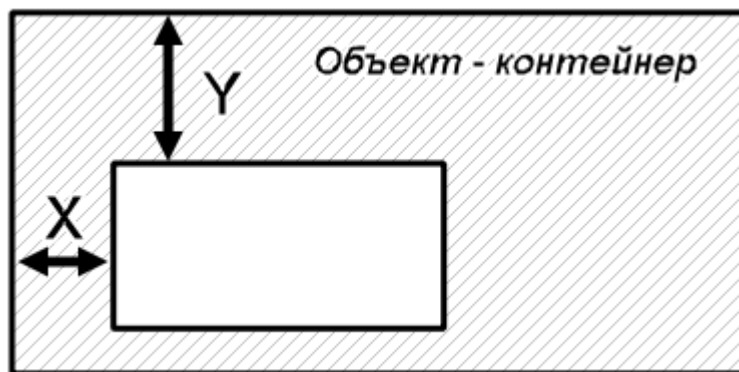
## 1.1.4.2.4.10.19.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

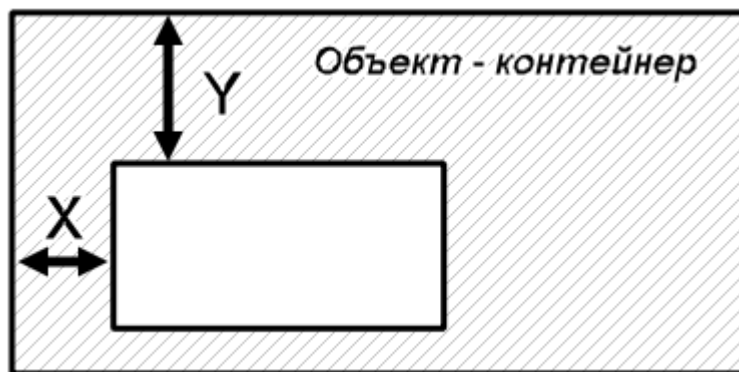
## 1.1.4.2.4.10.19.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.19.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.19.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.19.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```



## 1.1.4.2.4.10.19.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.19.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.10.19.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.19.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.19.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

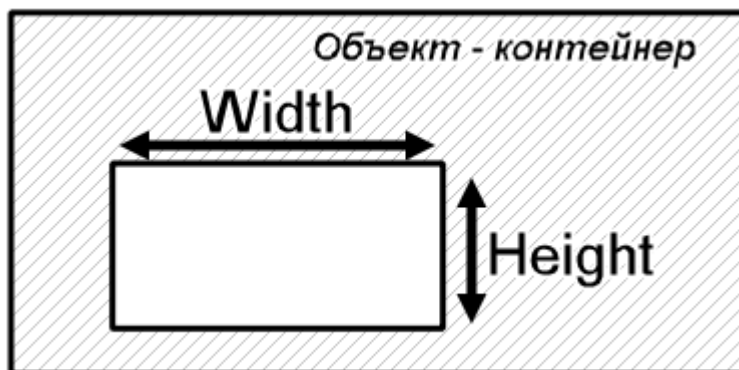
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.19.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

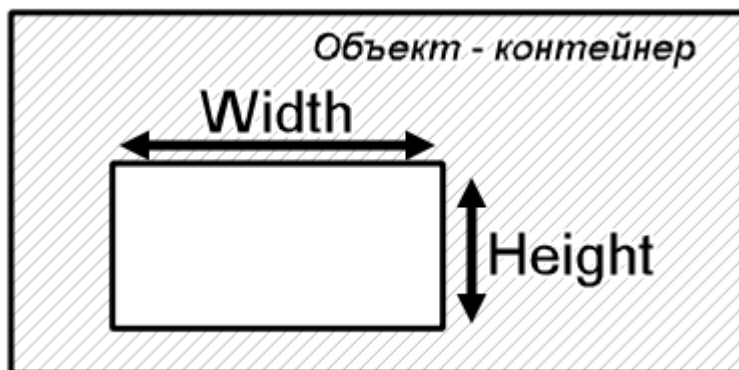


## 1.1.4.2.4.10.19.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.19.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```



## 1.1.4.2.4.10.19.3.16. Удерживаемая

Оставляет кнопку удерживаемой/отпускаемой после нажатия.



bool Checkable

### Значение

Значение	Описание
true	Удерживать кнопку после нажатия
false	Отпускать кнопку после нажатия

## 1.1.4.2.4.10.19.3.17. Нажата

Нажата ли кнопка.



bool Checked

## 1.1.4.2.4.10.19.3.18. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.2.4.10.19.3.19. Выравнивание текста



int4 TextAlignment

### Значение

Значение	Описание
33	Слева сверху
65	Слева снизу
129	Слева по центру
34	Справа сверху
66	Справа снизу
130	Справа по центру
36	По центру сверху
68	По центру снизу
132	По центру объекта
40	По всей ширине сверху
72	По всей ширине снизу
136	По всей ширине по центру

## 1.1.4.2.4.10.19.3.20. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку.



string Font

## 1.1.4.2.4.10.19.3.21. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

## 1.1.4.2.4.10.19.3.22. Цвет шрифта при нажатии

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickFontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.OnClickFontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.OnClickFontColor = 4294901760;
```

## 1.1.4.2.4.10.19.3.23. Цвет шрифта при наведении мыши

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverFontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.OnHoverFontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.OnHoverFontColor = 4294901760;
```



# 1.1.4.2.4.10.19.3.24. Цвет шрифта при ВЫКЛЮЧЕНИИ

Цвет шрифта при выключении.



uint4 DisabledFontColor

## 1.1.4.2.4.10.19.3.25. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.BrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.BrushColor = 4294901760;
```

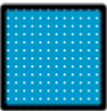
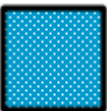
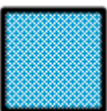

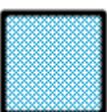
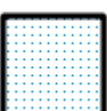
## 1.1.4.2.4.10.19.3.26. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.2.4.10.19.3.27. Цвет заливки при нажатии

Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickBrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.OnClickBrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.OnClickBrushColor = 4294901760;
```

## 1.1.4.2.4.10.19.3.28. Стиль заливки при нажатии

Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту.



uint2 OnClickBrushStyle

### Значение

Значение в дизайнере	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.OnClickBrushStyle = 1;

## 1.1.4.2.4.10.19.3.29. Цвет заливки при наведении мыши

Цвет внутренней заливки объекта при вводе курсора в область объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverBrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.OnHoverBrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.OnHoverBrushColor = 4294901760;
```






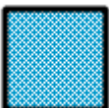

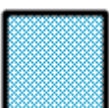
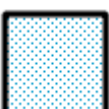
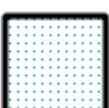
## 1.1.4.2.4.10.19.3.30. Стиль заливки при наведении мыши

Заливает объект одним из стилей заливки при вводе курсора в область объекта.



uint2 OnHoverBrushStyle

### Значение

Значение в дизайнере	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.OnHoverBrushStyle = 1;

# 1.1.4.2.4.10.19.3.31. Цвет заливки при выключении

Цвет заливки при выключении.



uint4 DisabledBrushColor

# 1.1.4.2.4.10.19.3.32.    Стил ь    заливки    при выключении

Стил ь заливки при выключении.



uint2 DisabledBrushStyle

## 1.1.4.2.4.10.19.3.33. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.2.4.10.19.3.34. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.19.3.35. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.19.3.36. Цвет пера при нажатии

Цвет внешней границы объекта при щелчке левой кнопкой по объекту. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickPenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:

```
Object.OnClickPenColor = 0xff000000;
```



//Задать черный цвет пера в десятичной системе счисления:

```
Object.OnClickPenColor = 4278190080;
```





## 1.1.4.2.4.10.19.3.37. Стиль пера при нажатии

Стиль внешней границы объекта при щелчке левой кнопкой по объекту.



uint2 OnClickPenStyle

### Значение

Значение	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.OnClickPenStyle = 3;
```

## 1.1.4.2.4.10.19.3.38. Толщина пера при нажатии

Толщина внешней границы объекта при щелчке левой кнопкой по объекту. Измеряется в пикселях.



float8 OnClickPenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.OnClickPenWidth = 5;
```

## 1.1.4.2.4.10.19.3.39. Цвет пера при наведении МЫШИ

Цвет внешней границы объекта при вводе курсора в область объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverPenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.OnHoverPenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.OnHoverPenColor = 4278190080;
```



## 1.1.4.2.4.10.19.3.40. Стиль пера при наведении МЫШИ

Стиль внешней границы объекта при вводе курсора в область объекта.



uint2 OnHoverPenStyle

### Значение

Значение	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.OnHoverPenStyle = 3;
```

## 1.1.4.2.4.10.19.3.41. Толщина пера при наведении мыши

Толщина пера при наведении мыши.



float8 OnHoverPenWidth

## 1.1.4.2.4.10.19.3.42. Цвет пера при выключении

Цвет пера при выключении.



uint4 DisabledPenColor

# 1.1.4.2.4.10.19.3.43.      Стил ь      пера      при выключении

Стил ь пера при выключении.



uint2 DisabledPenStyle

# 1.1.4.2.4.10.19.3.44. Толщина пера при выключении

Толщина пера при выключении.



float8 DisabledPenWidth

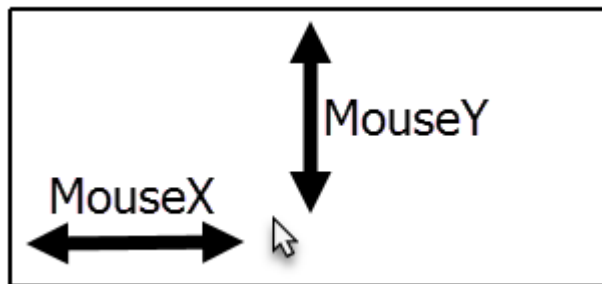


## 1.1.4.2.4.10.19.3.45. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

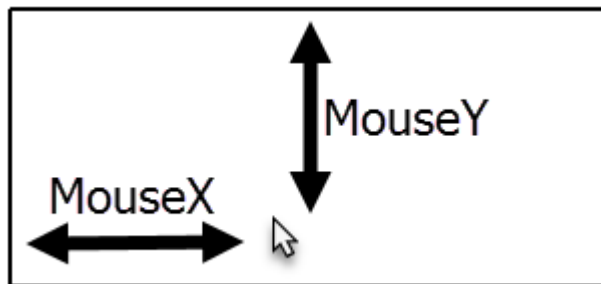
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.19.3.46. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.19.3.47. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.19.3.48. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.20. Эллипс

Компонент Эллипс применяется на мнемосхемах при рисовании объектов технологического процесса.

The screenshot displays two windows from a software application. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the object hierarchy. Under 'Графические объекты' (Graphic objects), 'Ellipse\_1' is selected. The bottom window, titled 'Редактор свойств' (Properties Editor), shows a list of properties for the selected object.

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
btn Button_1	Кнопка
○ Ellipse_1	Эллипс

Свойство	Характеристики	Значение
Отображаемое имя		Ellipse_1
Кардинальное число		1
X	R W	807
Y	R W	251
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	163
Высота	R W	100
Фокус ввода	R W	< не определено >
Цвет пера	R W	4278190080
Стиль пера	R W	Сплошная линия
Толщина пера	R W	1
Цвет заливки	R W	4278190080
Стиль заливки	R W	Нет заливки
Размер сектора	R W	360
Начало сектора	R W	360
Размер отверстия	R W	0

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.20.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.20.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	локальная координата объекта по оси X, в которой возникло событие
Y	float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.20.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.20.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.20.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.20.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

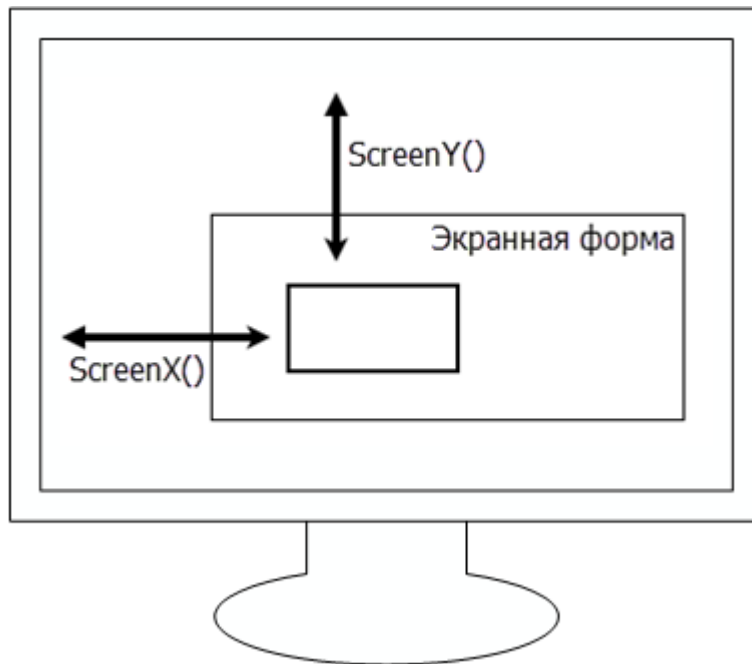
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.20.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



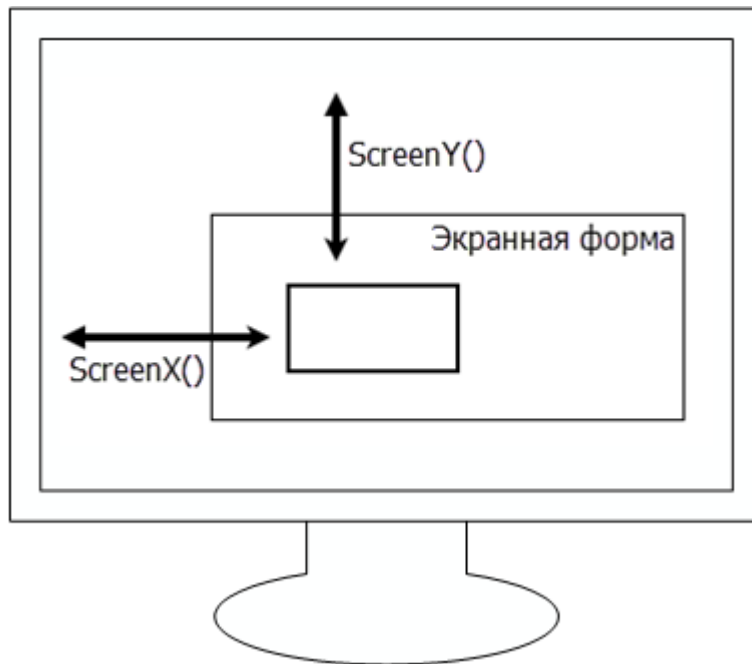
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.20.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



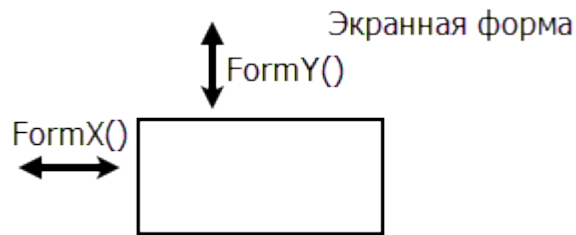
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.20.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



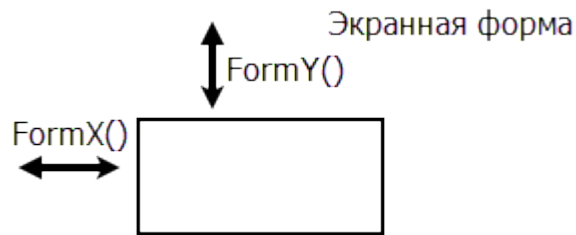
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.20.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.20.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта

<a href="#">Начало сектора</a>	Угол, который считается точкой отсчета при очерчивании сектора эллипса
<a href="#">Размер сектора</a>	Угол отображаемого сектора эллипса
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Размер отверстия</a>	Размер внутреннего отверстия
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.20.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.20.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.20.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.2.4.10.20.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.20.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.20.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```



## 1.1.4.2.4.10.20.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.20.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.20.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.20.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.20.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.20.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

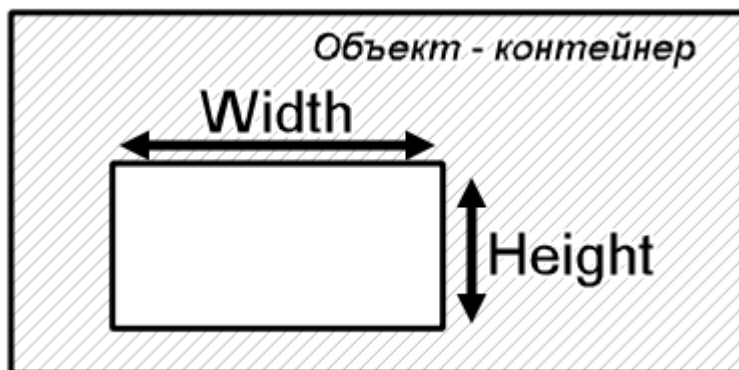
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.20.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

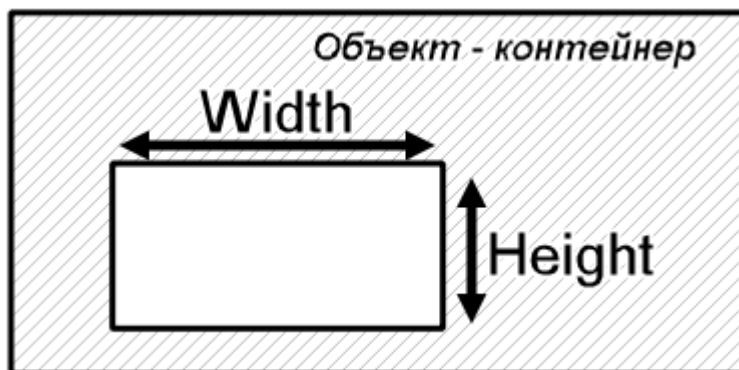


## 1.1.4.2.4.10.20.3.14. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.2.4.10.20.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.20.3.16. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.2.4.10.20.3.17. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.20.3.18. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.20.3.19. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.2.4.10.20.3.20. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



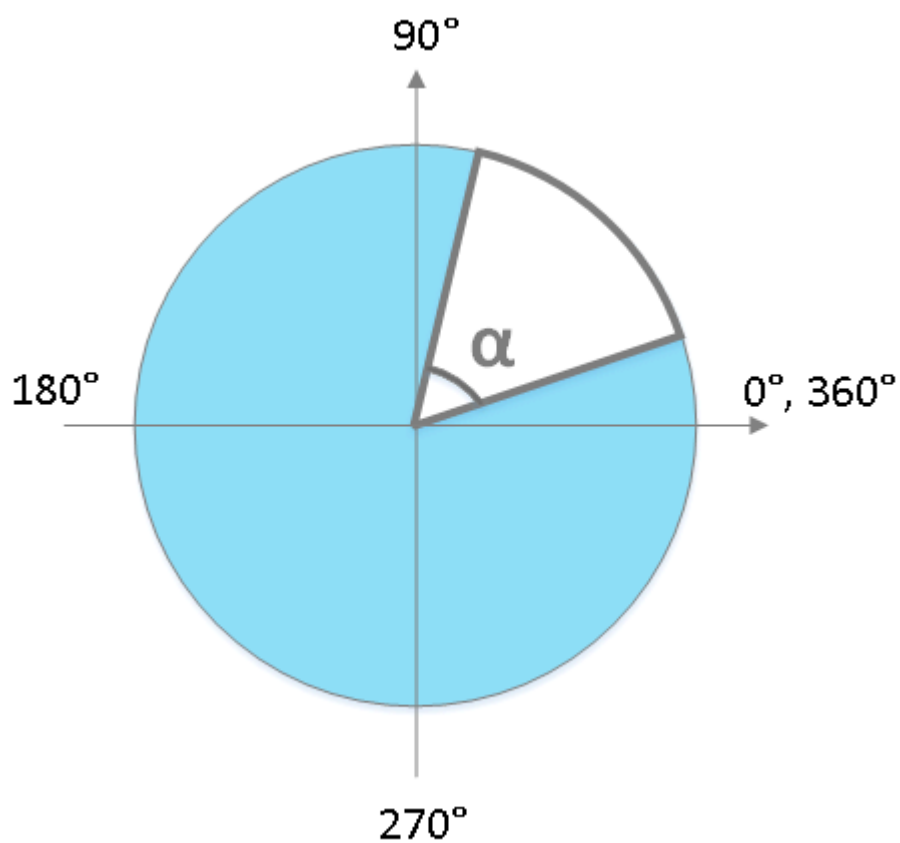
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.4.10.20.3.21. Размер сектора

Угол отображаемого сектора эллипса. Измеряется в градусах.



float4 PieAngle



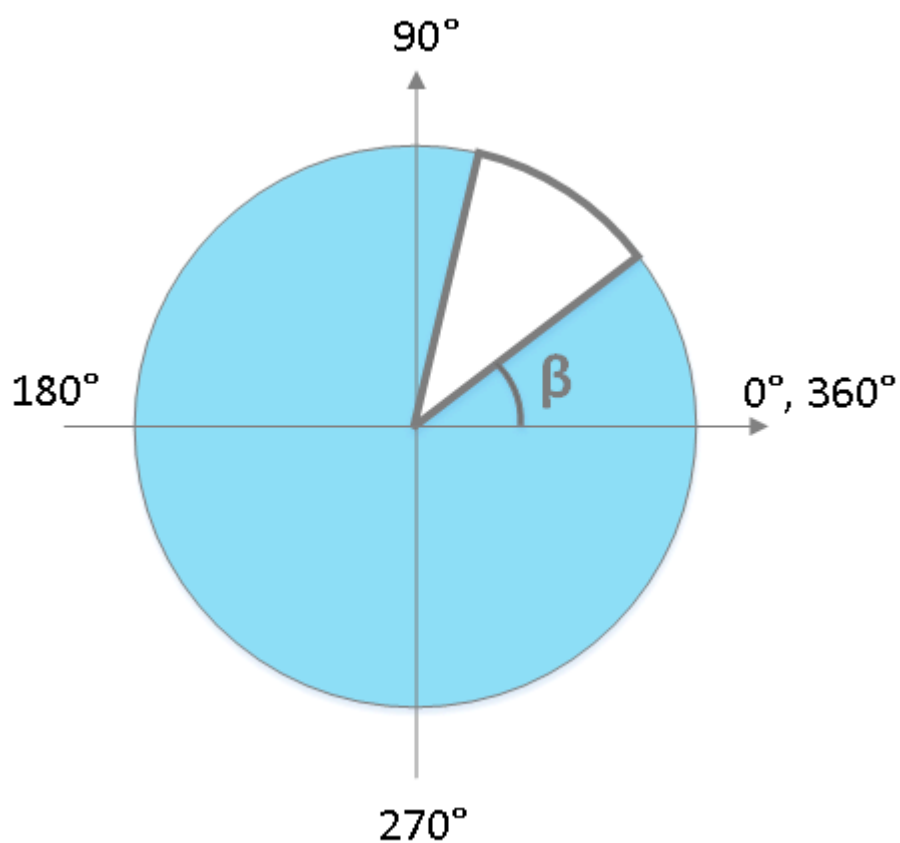


## 1.1.4.2.4.10.20.3.22. Начало сектора

Угол, который считается точкой отсчета при очерчивании сектора эллипса. Измеряется в градусах.



float4 PieAngleStart



## 1.1.4.2.4.10.20.3.23. Размер отверстия

Размер внутреннего отверстия.



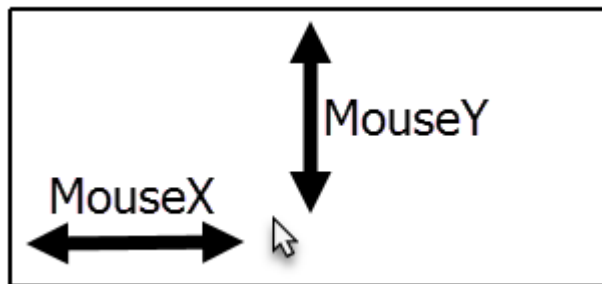
float8 HoleSize

## 1.1.4.2.4.10.20.3.24. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

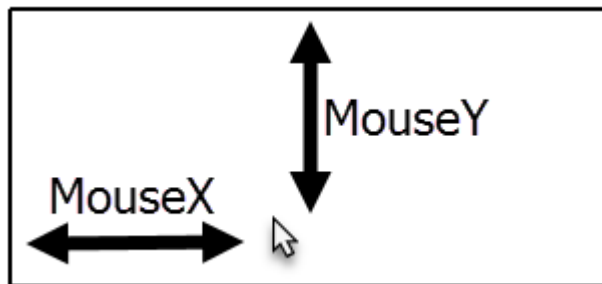
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.20.3.25. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.20.3.26. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.20.3.27. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.21. Индикатор гистограммы

Компонент Индикатор гистограммы применяется на мнемосхемах для демонстрации значений некоторых динамических параметров, у которых есть нижнее и верхнее пороговое значение. Текущее значение индикатора гистограммы определяется свойством Значение.

The screenshot displays two windows from a software development environment:

- Структура объекта (Object Structure):** A tree view showing the hierarchy of components. Under 'Графические объекты' (Graphic Objects), 'HistogramBar\_1' is selected, with the description 'Индикатор гистограммы' (Histogram Indicator).
- Редактор свойств (Properties Editor):** A table listing the properties of the selected component.

Свойство	Характеристики	Значение
Отображаемое имя		HistogramBar_1
Кардинальное число		1
X	R W	801
Y	R W	213
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	100
Высота	R W	20
Фокус ввода	R W ⚡ ⓘ	<не определено>
Минимальное значение	R W	0
Максимальное значение	R W	1
Нижнее пороговое значение	R W	0.25
Верхнее пороговое значение	R W	0.75
Значение	R W ⚙	0.5
Ориентация	R W	Горизонтальная
Цвет пера	R W	4278190080

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана



## 1.1.4.2.4.10.21.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.21.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.21.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.21.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.21.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.21.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

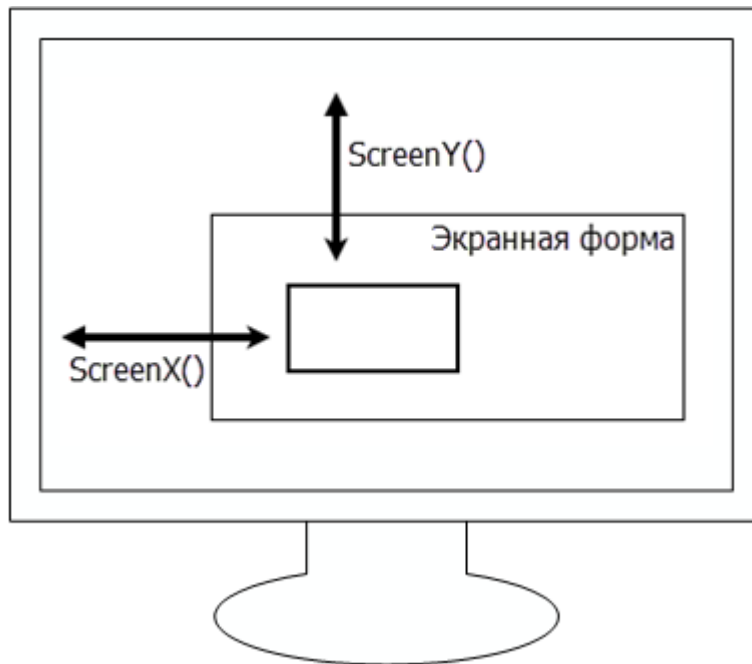
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.21.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



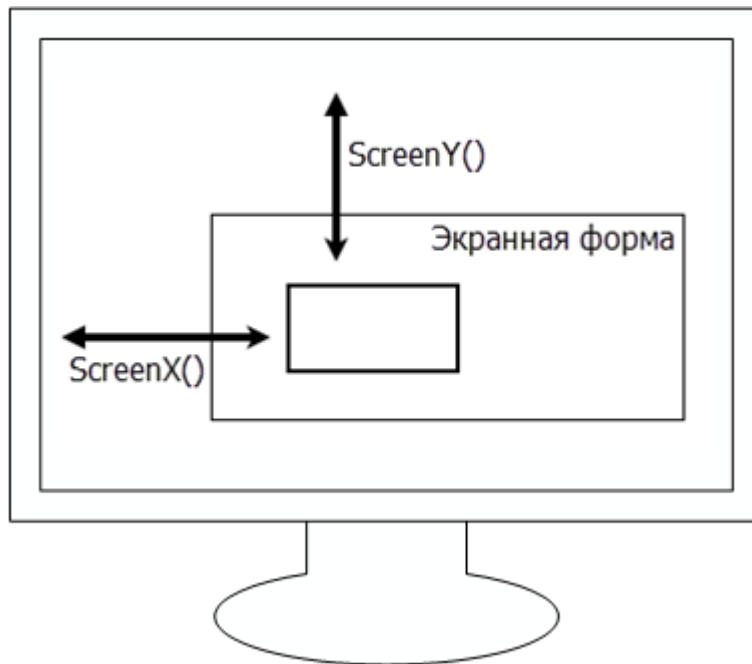
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.21.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

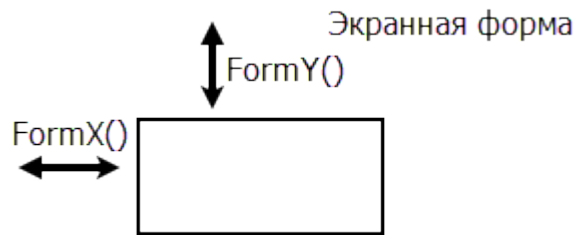
```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.21.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



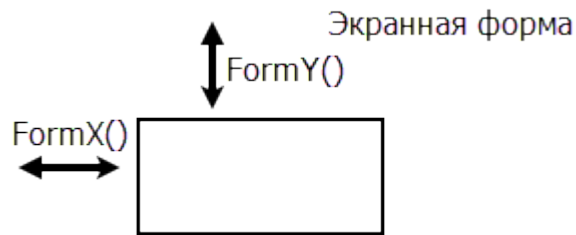
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.21.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.21.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта

<a href="#">Стиль заливки превышения уровня</a>	Стиль заливки полосы индикатора, если значение превышает Верхнее пороговое значение
<a href="#">Цвет заливки превышения уровня</a>	Цвет заливки полосы индикатора, если значение превышает Верхнее пороговое значение
<a href="#">Стиль заливки пониженного модуля</a>	Стиль заливки полосы индикатора, если значение ниже, чем Нижнее пороговое значение
<a href="#">Цвет заливки пониженного уровня</a>	Цвет заливки полосы индикатора, если значение ниже, чем Нижнее пороговое значение
<a href="#">Стиль заливки полосы</a>	Стиль заливки основного уровня гистограммы
<a href="#">Цвет заливки полосы</a>	Цвет заливки основного уровня гистограммы
<a href="#">Ориентация</a>	Ориентация гистограммы
<a href="#">Значение</a>	Текущее значение уровня на гистограмме
<a href="#">Верхнее пороговое значение</a>	Значение уровня гистограммы, при превышении которого полоска индикатора изменяет цвет и стиль заливки
<a href="#">Нижнее пороговое значение</a>	Значение уровня гистограммы, при понижении которого полоска индикатора изменяет цвет и стиль заливки
<a href="#">Максимальное задание</a>	Максимальное значение шкалы гистограммы
<a href="#">Минимальное значение</a>	Минимальное значение шкалы гистограммы
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.21.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.21.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.21.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

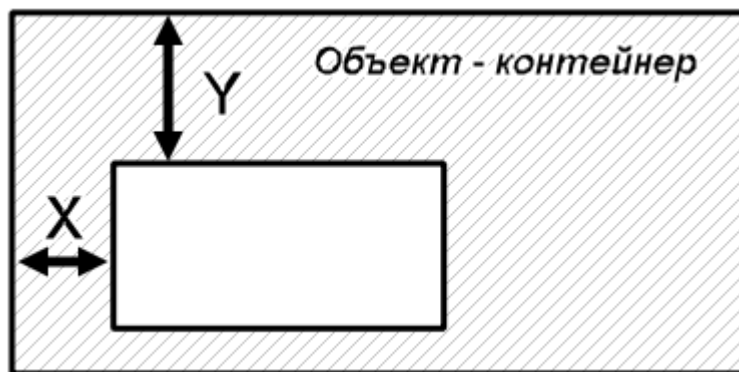
## 1.1.4.2.4.10.21.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.2.4.10.21.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.21.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.21.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.21.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.21.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.21.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.21.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.21.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

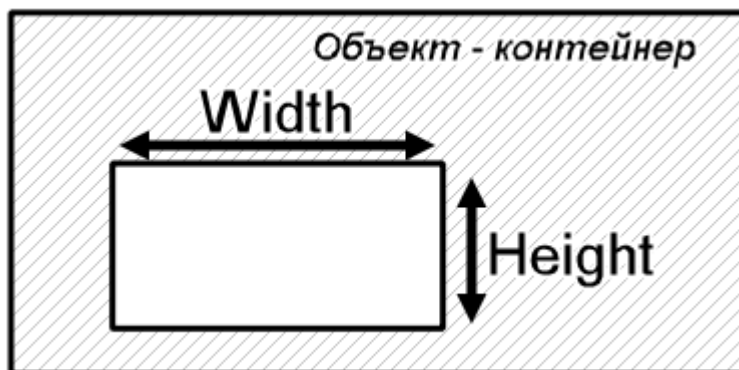


## 1.1.4.2.4.10.21.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

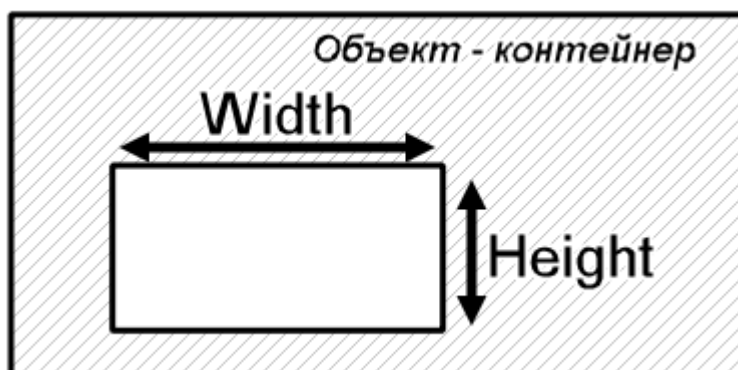


## 1.1.4.2.4.10.21.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.21.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.21.3.16. Минимальное значение

Минимальное значение шкалы гистограммы.



float8 MinLimit

## 1.1.4.2.4.10.21.3.17. Максимальное значение

Максимальное значение шкалы гистограммы.



float8 MaxLimit

## 1.1.4.2.4.10.21.3.18. Нижнее пороговое значение

Значение уровня гистограммы, при понижении которого полоска индикатора изменяет цвет и стиль заливки (Цвет заливки пониженного уровня и Стиль заливки пониженного уровня).



float8 LowValue

## 1.1.4.2.4.10.21.3.19. Верхнее пороговое значение

Значение уровня гистограммы, при превышении которого полоска индикатора изменяет цвет и стиль заливки (Цвет заливки превышения уровня и Стиль заливки превышения уровня).



float8 HighValue

## 1.1.4.2.4.10.21.3.20. Значение

Текущее значение уровня на гистограмме.



float8 Value



## 1.1.4.2.4.10.21.3.21. Ориентация

Ориентация гистограммы.



int4 Orientation

### Значение

Значение	Описание
0	Горизонтальная ориентация гистограммы
1	Вертикальная ориентация гистограммы

## 1.1.4.2.4.10.21.3.22. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.2.4.10.21.3.23. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.21.3.24. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.21.3.25. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.2.4.10.21.3.26. Стилъ заливки

Заливаеят обект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.4.10.21.3.27. Цвет заливки полосы

Цвет заливки основного уровня гистограммы. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BarBrushColor



## 1.1.4.2.4.10.21.3.28. Стиль заливки полосы

Стиль заливки основного уровня гистограммы. Доступные стили аналогичны стилям свойства Стиль заливки.



uint2 BarBrushStyle

## 1.1.4.2.4.10.21.3.29. Цвет заливки пониженного уровня

Цвет заливки полосы индикатора, если значение ниже, чем Нижнее пороговое значение.



uint4 BarLowBrushColor

## 1.1.4.2.4.10.21.3.30. пониженного уровня

Стиль

заливки

Стиль заливки полосы индикатора, если значение ниже, чем Нижнее пороговое значение. Доступные стили аналогичны стилям свойства Стиль заливки.



uint2 BarLowBrushStyle

## 1.1.4.2.4.10.21.3.31. Стиль заливки превышения уровня

Стиль заливки полосы индикатора, если значение превышает Верхнее пороговое значение. Доступные стили аналогичны стилям свойства Стиль заливки.



uint2 BarHighBrushStyle

## 1.1.4.2.4.10.21.3.32. Цвет заливки превышения уровня

Цвет заливки полосы индикатора, если значение превышает Верхнее пороговое значение.



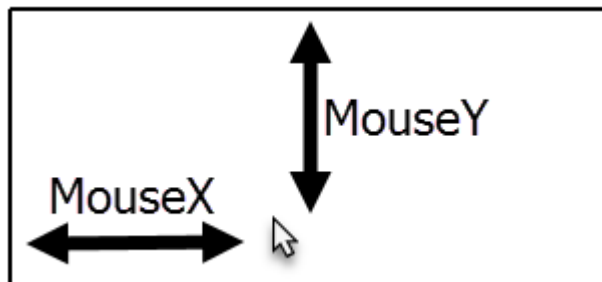
uint4 BarHighBrushColor

## 1.1.4.2.4.10.21.3.33. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

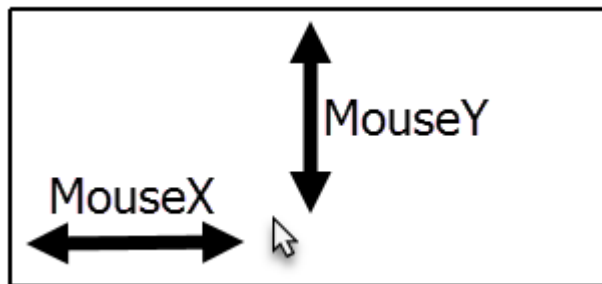
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.21.3.34. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.21.3.35. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.2.4.10.21.3.36. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



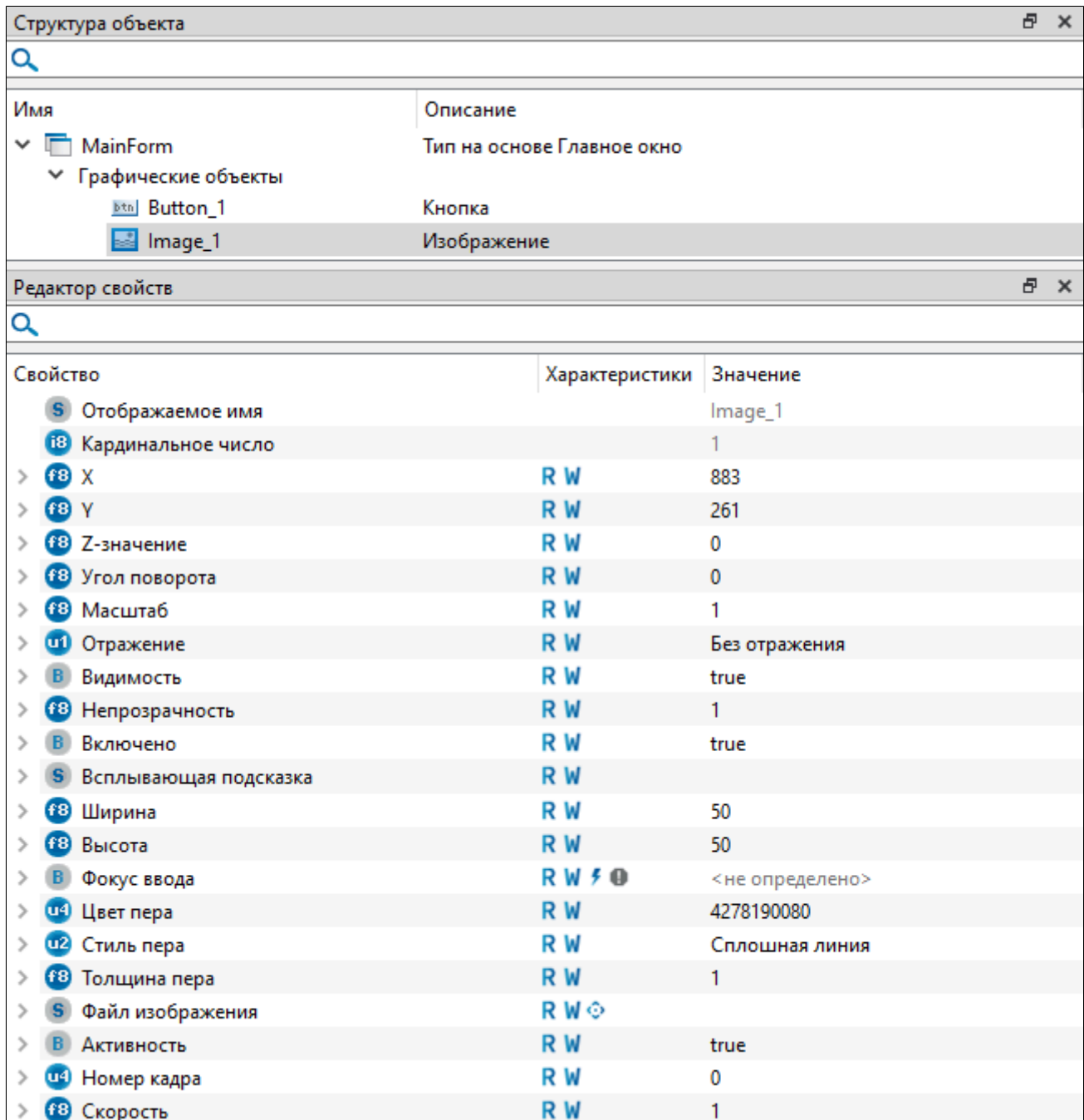
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.22. Изображение

Компонент для вставки растрового изображения или анимации из файла.

Изображения используются для вставки логотипов компаний и визуального представления используемого оборудования. Также изображения можно использовать для воспроизведения анимации (gif-файлов).

Чтобы добавить изображение, поместите его в папку проекта resources и укажите имя файла в свойстве Файл изображения. Для управления анимацией используйте свойства Активность, Номер кадра и Скорость.



На мнемосхемах можно применять картинки с поддержкой прозрачности.

**Допустимые форматы:** dds, gif, ics, ico, jpeg, svg, tga, tiff, bmp и webp.

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.22.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.22.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.22.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.22.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.22.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```



## 1.1.4.2.4.10.22.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

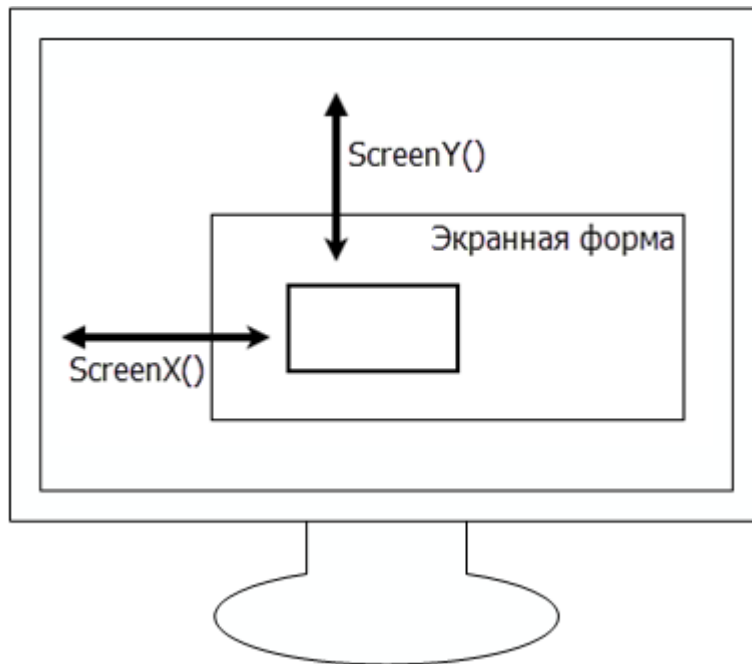
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.22.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



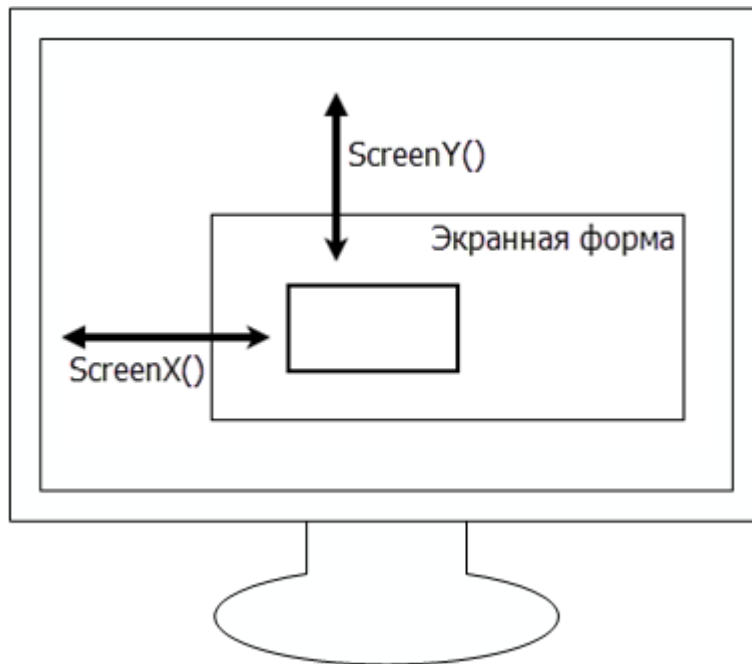
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.22.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



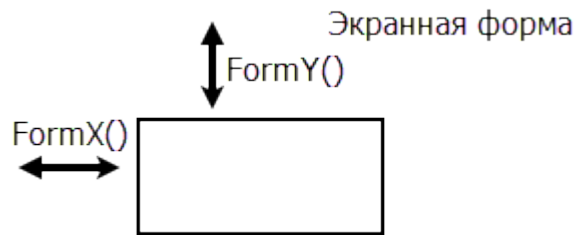
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.22.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



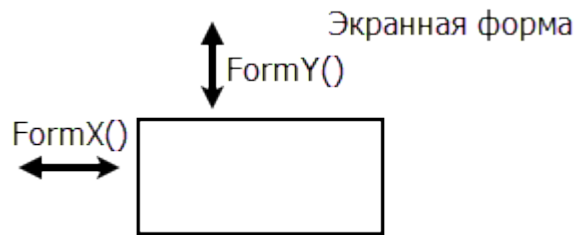
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.22.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.22.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Скорость</a>	Управляет скоростью, с которой сменяются кадры анимации

<a href="#">Номер кадра</a>	Позволяет выбрать кадр gif-файла, начиная с которого будет воспроизводиться анимация
<a href="#">Активность</a>	Управляет активностью gif-файла
<a href="#">Файл изображения</a>	Путь к файлу изображения относительно папки resources
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Отражение</a>	Отражение



## 1.1.4.2.4.10.22.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.22.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.22.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

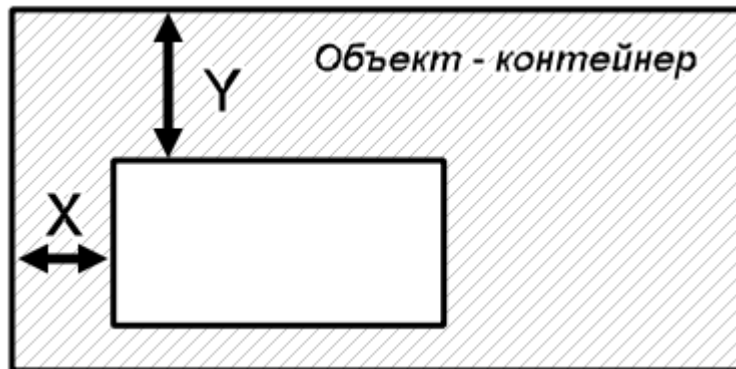
## 1.1.4.2.4.10.22.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.22.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.22.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.22.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.22.3.8. Отражение

Отражение объекта.



uint1 Flip



## 1.1.4.2.4.10.22.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.22.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.22.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.22.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

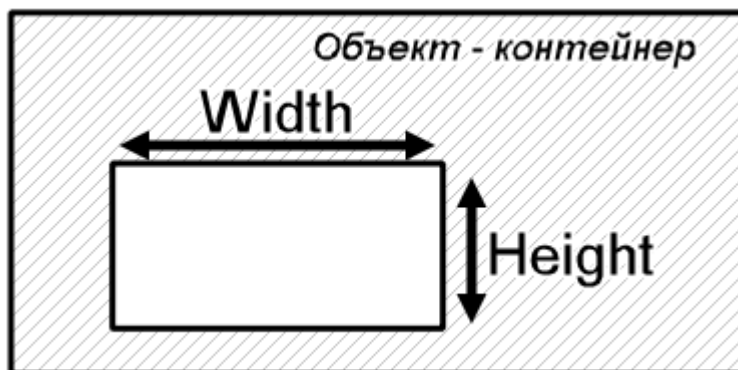
```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.2.4.10.22.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

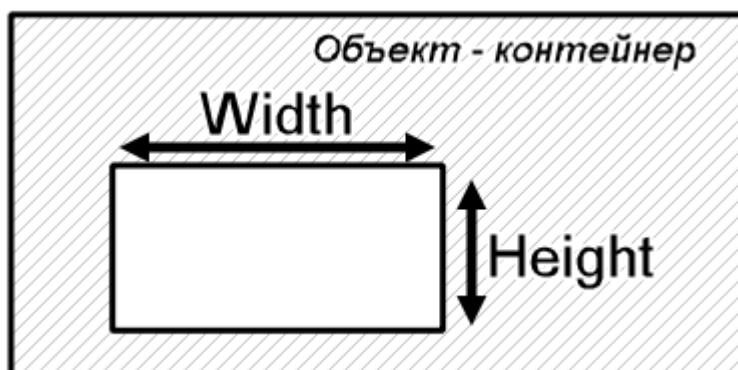


## 1.1.4.2.4.10.22.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.22.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.22.3.16. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;




## 1.1.4.2.4.10.22.3.17. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.22.3.18. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.22.3.19. Файл изображения

Путь к файлу изображения относительно папки resources.



string ImageFile

## 1.1.4.2.4.10.22.3.20. Активность

Управляет активностью gif-файла.



bool Active

### Значение

Значение	Описание
true	Разрешить воспроизведение анимации
false	Запретить воспроизведение анимации

## 1.1.4.2.4.10.22.3.21. Номер кадра

Позволяет выбрать кадр gif-файла, начиная с которого будет воспроизводиться анимация.



uint4 CurrentFrameNumber

## 1.1.4.2.4.10.22.3.22. Скорость

Управляет скоростью, с которой сменяются кадры анимации.



float8 Speed

### Значение

Значение	Описание
0	Минимальная скорость смены кадров
от 0 до 1	Замедление скорости смены кадров
1	Стандартная скорость смены кадров
>1	Увеличение скорости воспроизведения кадров

### Примеры



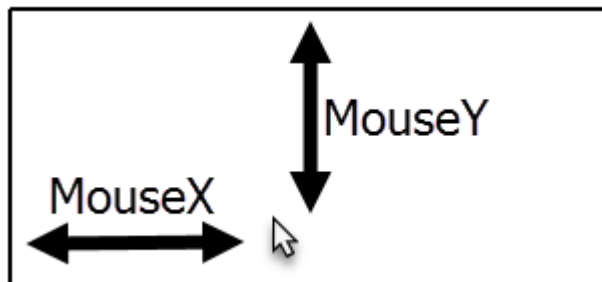
//Замедлить стандартную скорость смены кадров наполовину:  
Image.Speed = 0.5;

## 1.1.4.2.4.10.22.3.23. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

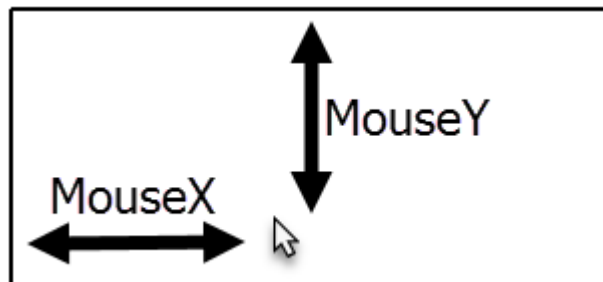
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.22.3.24. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```



## 1.1.4.2.4.10.22.3.25. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.22.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.23. Текст

Компонент для создания областей с текстом. Применяется на мнемосхемах для пояснительных целей.

The screenshot displays two windows from a software development environment. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project hierarchy. Under 'Графические объекты' (Graphic Objects), 'Text\_1' is selected. The bottom window, titled 'Редактор свойств' (Properties Editor), shows a list of properties for the selected 'Text\_1' component.

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
btn Button_1	Кнопка
A Text_1	Текст

Свойство	Характеристики	Значение
Отображаемое имя		Text_1
Кардинальное число		1
>  X	R W	795
>  Y	R W	272
>  Z-значение	R W	0
>  Угол поворота	R W	0
>  Масштаб	R W	1
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Ширина	R W	50
>  Высота	R W	20
>  Фокус ввода	R W	<не определено>
>  Текст	R W	Текст
>  Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0
>  Цвет шрифта	R W	4278190080
>  Выравнивание текста	R W	По верху слева

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.23.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.23.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.23.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	локальная координата объекта по оси X, в которой возникло событие
Y	float	локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.23.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.23.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```



## 1.1.4.2.4.10.23.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

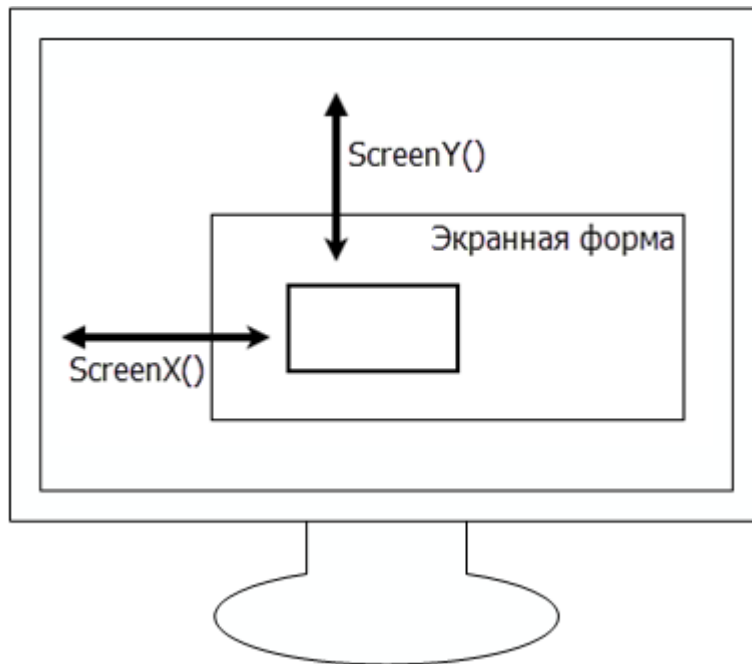
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.23.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



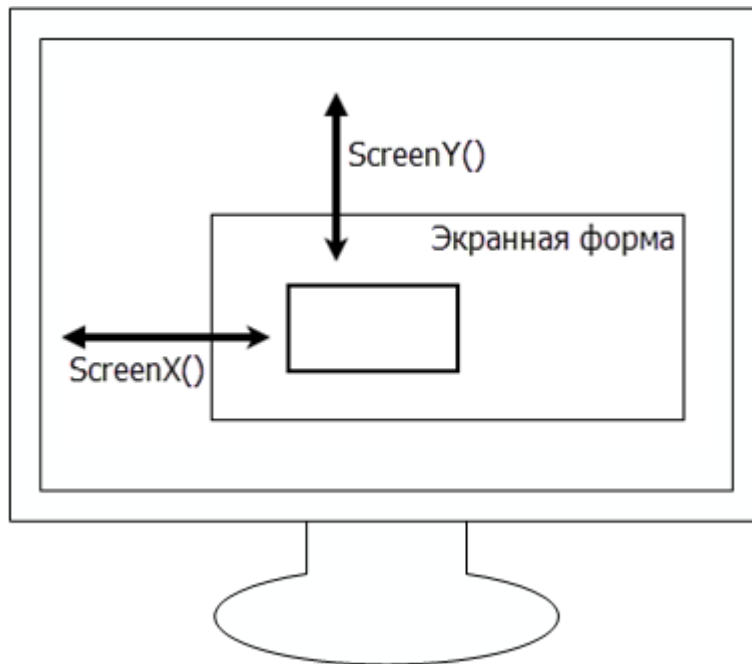
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.23.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



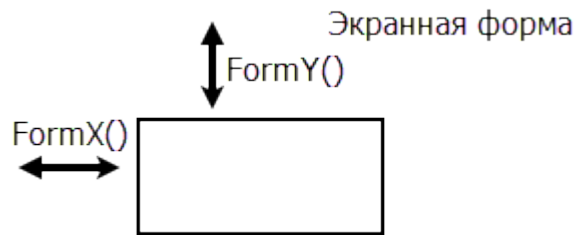
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.23.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



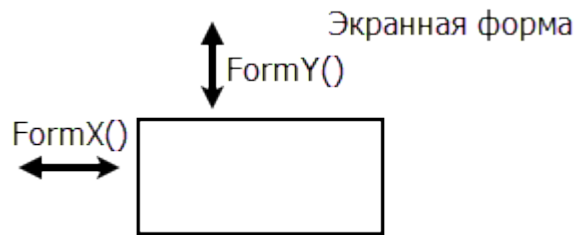
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.23.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.23.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Выравнивание текста</a>	Выравнивание текста
<a href="#">Z-значение</a>	Z-значение

<a href="#">Непрозрачность</a>	Непрозрачность объектов
<a href="#">Отражение</a>	Отражение



## 1.1.4.2.4.10.23.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.23.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

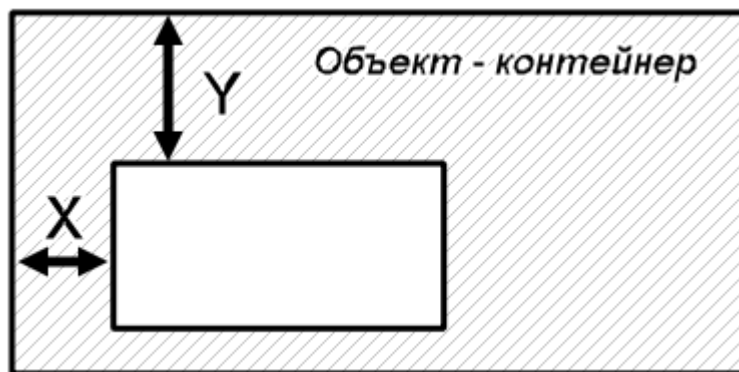
## 1.1.4.2.4.10.23.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

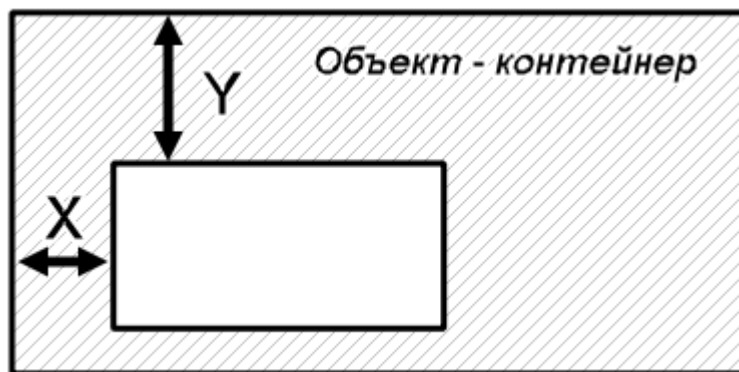
## 1.1.4.2.4.10.23.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.23.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.23.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.23.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.2.4.10.23.3.8. Отражение

Отражение объекта.



uint1 Flip



## 1.1.4.2.4.10.23.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.23.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.23.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.23.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

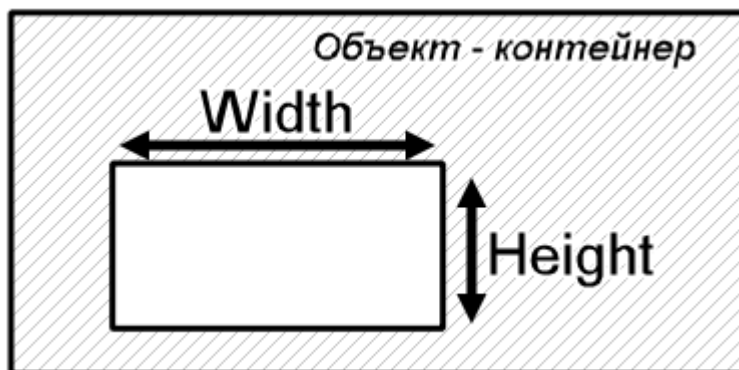
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.23.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

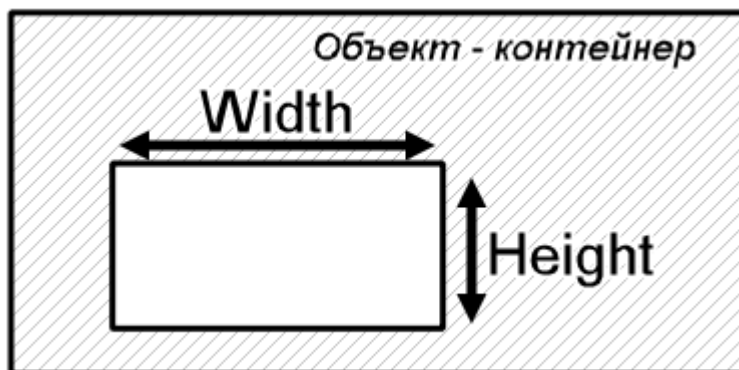


## 1.1.4.2.4.10.23.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.23.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.23.3.16. Текст

Содержимое текстовой области объекта.



string Text



## 1.1.4.2.4.10.23.3.17. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.23.3.18. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

# 1.1.4.2.4.10.23.3.19. Выравнивание текста



int4 TextAlignment

## Значение

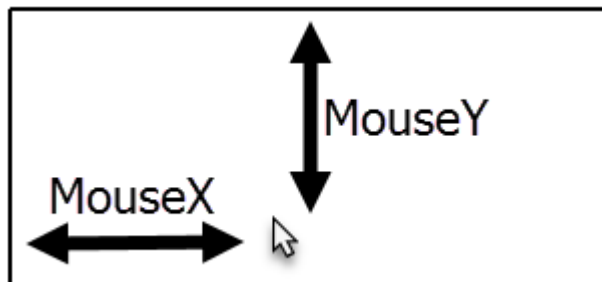
Значение	Описание
33	Слева сверху
65	Слева снизу
129	Слева по центру
34	Справа сверху
66	Справа снизу
130	Справа по центру
36	По центру сверху
68	По центру снизу
132	По центру объекта
40	По всей ширине сверху
72	По всей ширине снизу
136	По всей ширине по центру

## 1.1.4.2.4.10.23.3.20. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

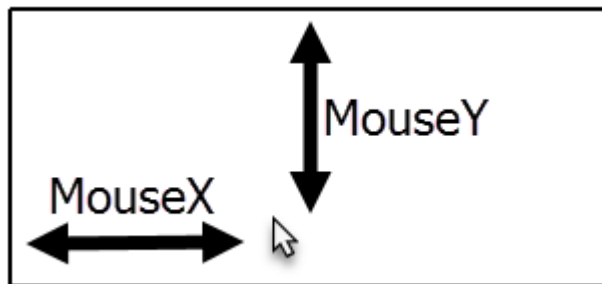
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.23.3.21. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.23.3.22. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.23.3.23. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.24. Соединительная линия

Чтобы нарисовать сегменты трубопровода и соединители труб, используйте элементы Соединительная линия и Место соединения. Основное отличие элемента Соединительная линия от элемента Линия – возможность перемещения отдельных отрезков линии. Такая возможность позволяет построить сложный участок трубопровода из одной линии.

The screenshot displays two windows from a software application:

- Структура объекта (Object Structure):** A tree view showing the hierarchy of objects. The selected object is `ConnectionLine_1`, which is a `Соединительная линия` (Connection Line) containing three points: `Point_3`, `Point_1`, and `Point_2`.
- Редактор свойств (Properties Editor):** A table listing the properties of the selected object.

Свойство	Характеристики	Значение
Отображаемое имя		ConnectionLine_1
Кардинальное число		1
X	R W	784
Y	R W	213
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	95
Высота	R W	69
Фокус ввода	R W ⚡ ⓘ	<не определено>
Декоратор	R W	false
Цвет пера	R W	4278190080
Стиль пера	R W	Сплошная линия
Толщина пера	R W	1



Чтобы перемещать точки и отрезки между точками, перейдите в режим редактирования линии. Выделите линию, перейдите в контекстное меню и выберите команду Редактировать линию или нажмите клавишу F2. Чтобы выйти из режима редактирования, нажмите клавишу Esc.

Чтобы изменить внешний вид трубы, используйте свойства элемента Соединительная линия:

1. Включите декоратор линии в свойстве Декоратор. Декоратор позволяет добавить объемность в изображение трубы.
2. Задайте цвет линии в свойстве Цвет пера. Линия окрашивается в градиентный цвет при выборе любого цвета, кроме черного.

Чтобы менять толщину линии, используйте свойство Масштаб.

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.24.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.24.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.24.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.24.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.24.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.24.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

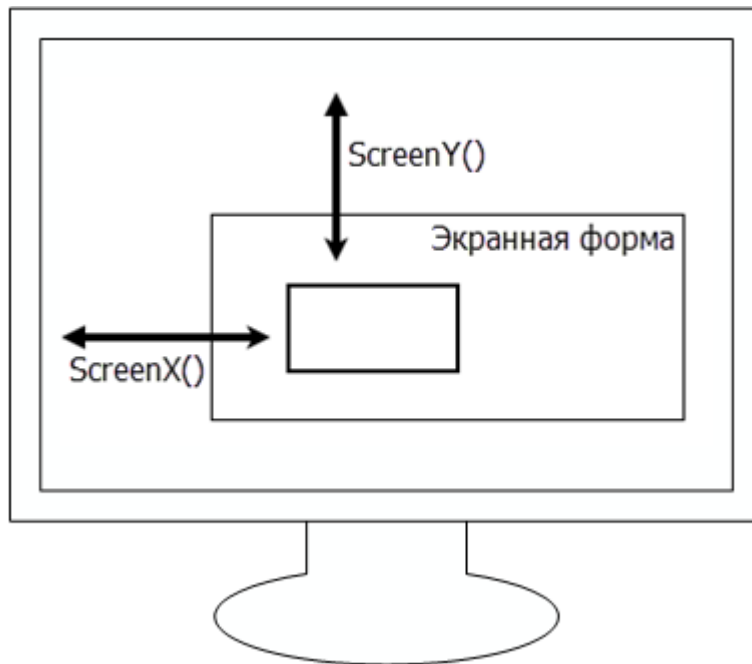
## Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме



## 1.1.4.2.4.10.24.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



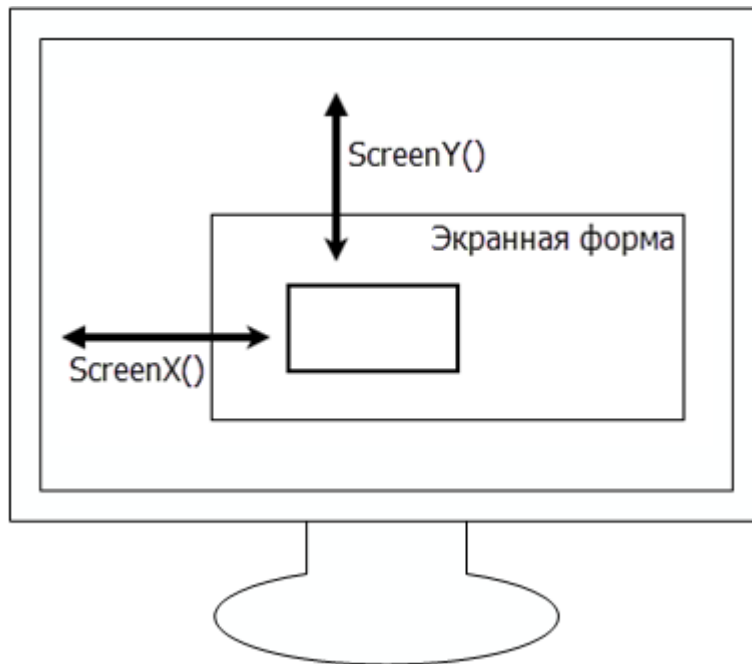
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.24.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	значение свойства Y объекта

### Примеры



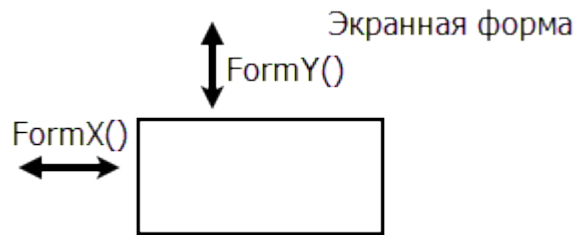
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.24.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



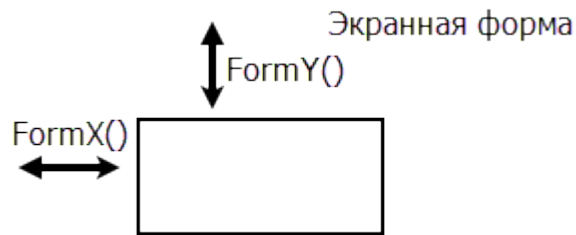
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.24.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.24.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Декоратор</a>	Декоратор линии
<a href="#">Отражение</a>	Отражение
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта

Толщина пера

Толщина внешней границы объекта

## 1.1.4.2.4.10.24.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.24.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.2.4.10.24.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

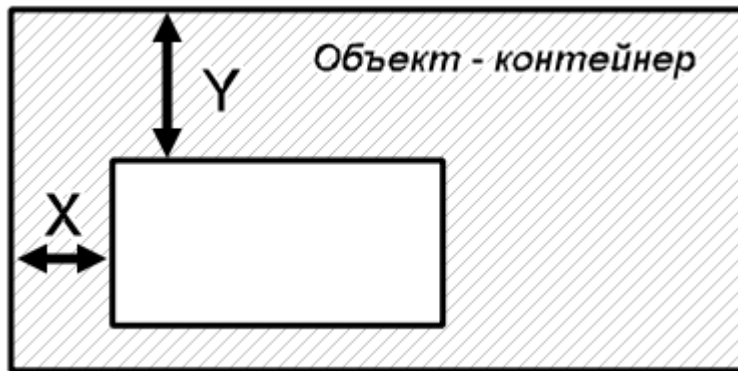
## 1.1.4.2.4.10.24.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.24.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.24.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.24.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.24.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.24.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.24.3.10. Непрозрачность

Непрозрачность.



float8 Opacity



## 1.1.4.2.4.10.24.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.24.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

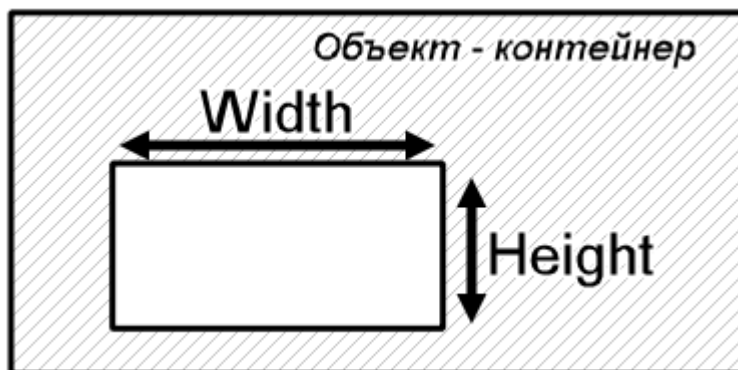
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.24.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

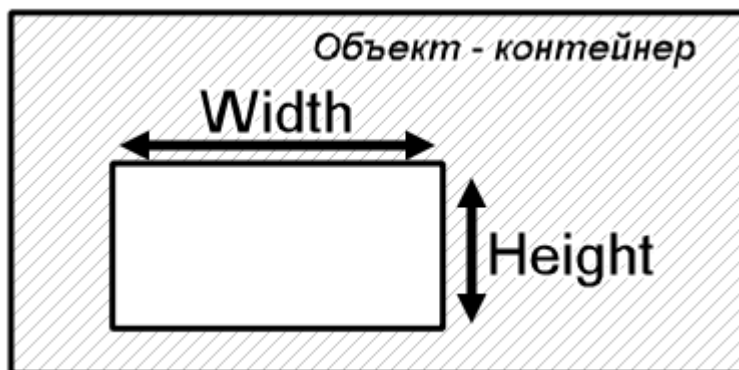


## 1.1.4.2.4.10.24.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.24.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.24.3.16. Декоратор

Декоратор линии.



bool EnableDecoration

## 1.1.4.2.4.10.24.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.2.4.10.24.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;
```



## 1.1.4.2.4.10.24.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



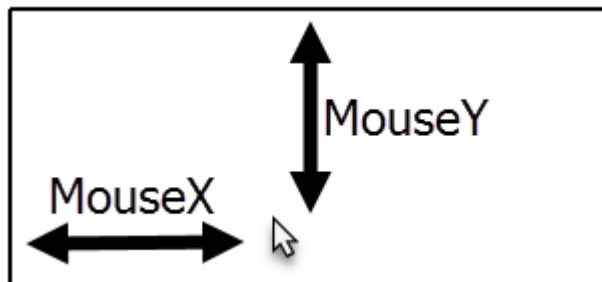
```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.24.3.20. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

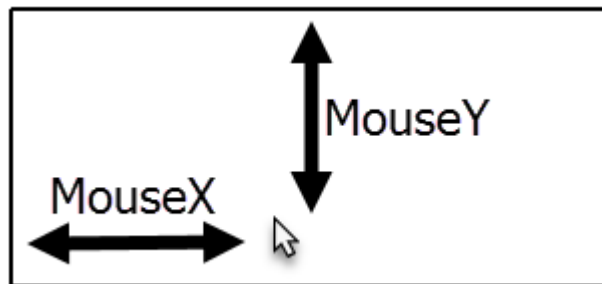
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.24.3.21. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.24.3.22. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.24.3.23. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



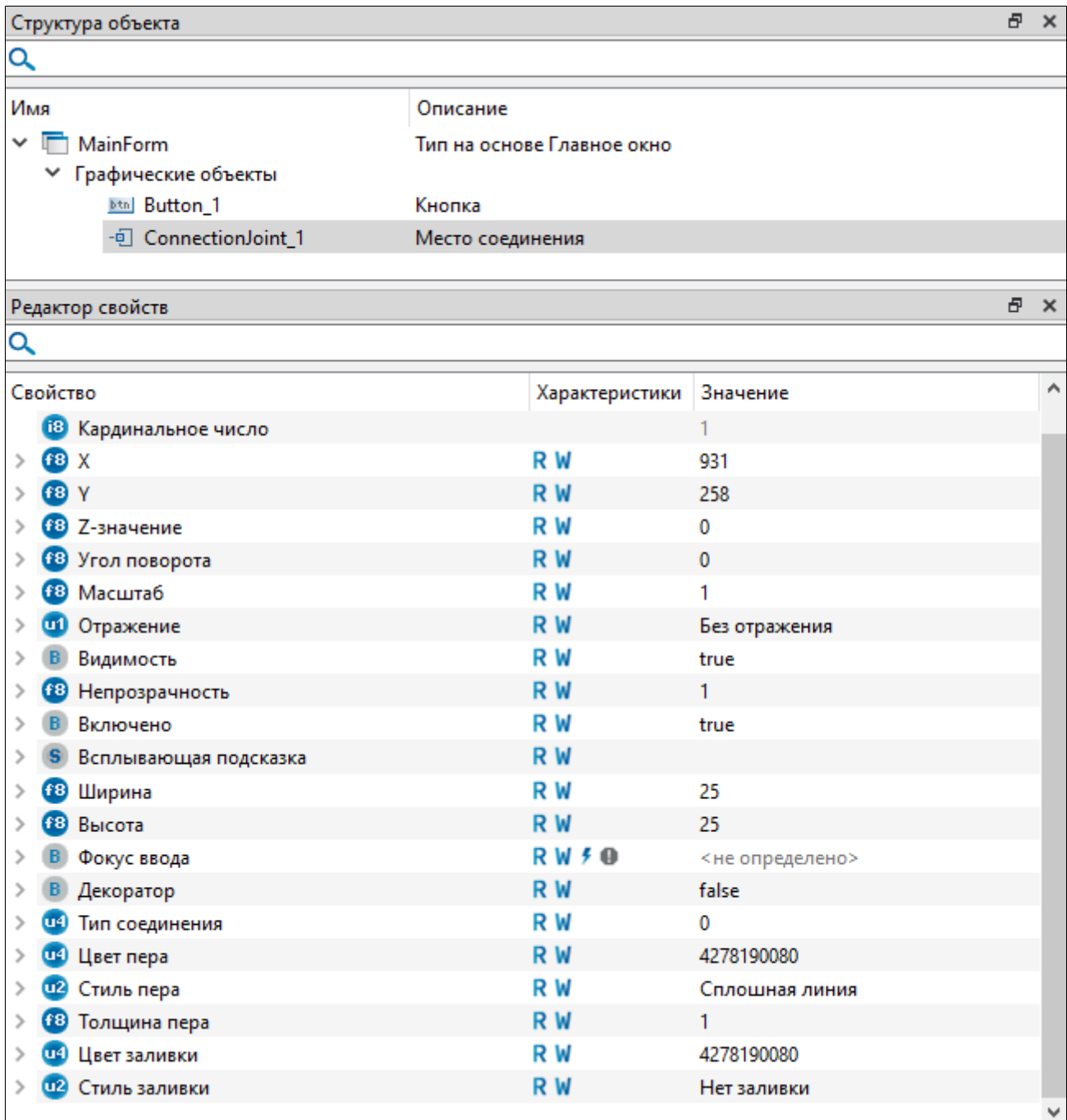
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.25. Место соединения

Чтобы нарисовать сегменты трубопровода и соединители труб, используйте элементы Соединительная линия и Место соединения. Основное отличие элемента Соединительная линия от элемента Линия – возможность перемещения отдельных отрезков линии. Такая возможность позволяет построить сложный участок трубопровода из одной линии.



Структура объекта

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Button_1	Кнопка
ConnectionJoint_1	Место соединения

Редактор свойств

Свойство	Характеристики	Значение
Кардинальное число		1
X	R W	931
Y	R W	258
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	25
Высота	R W	25
Фокус ввода	R W ⚡ ⓘ	<не определено>
Декоратор	R W	false
Тип соединения	R W	0
Цвет пера	R W	4278190080
Стиль пера	R W	Сплошная линия
Толщина пера	R W	1
Цвет заливки	R W	4278190080
Стиль заливки	R W	Нет заливки

Чтобы выбрать тип и цвет места соединения, используйте свойства элемента Место соединения:

1. Включите декоратор в свойстве Декоратор. Декоратор позволяет изобразить соединитель в фигурном виде и добавить объемность в изображение.

2. Укажите тип места соединения в свойстве Тип соединения:

0	
1	
2	
3	
4	

3. Задайте цвет соединителя в свойстве Цвет пера. Соединитель окрашивается в градиентный цвет при выборе любого цвета, кроме черного.

Чтобы менять величину соединителя, используйте свойство Масштаб.

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана



## 1.1.4.2.4.10.25.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.25.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.25.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.25.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.25.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.25.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

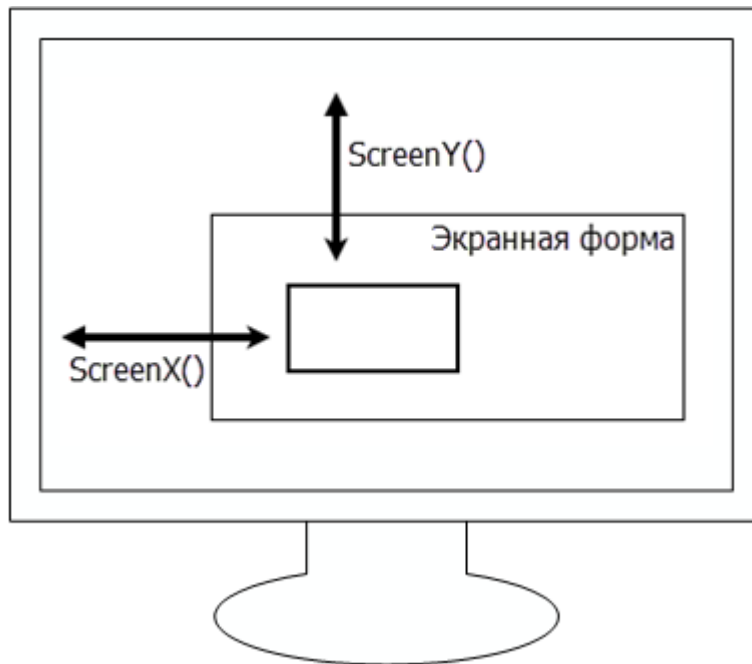
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.25.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



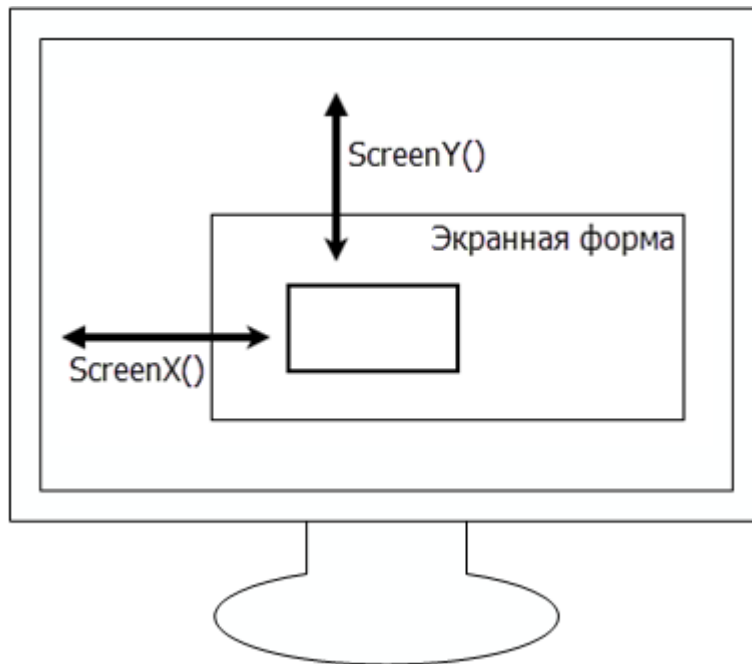
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.25.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

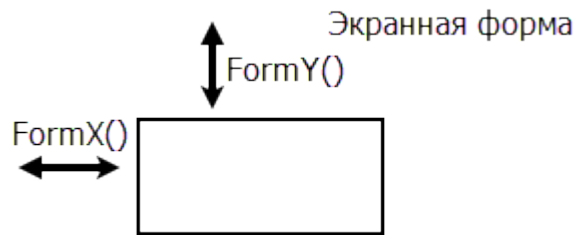
```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.25.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



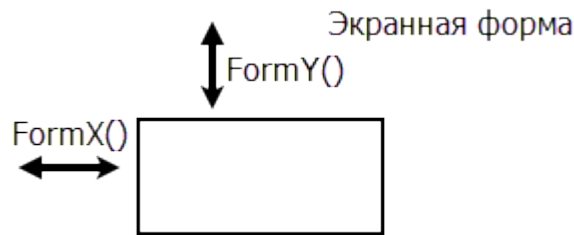
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.25.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.25.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Декоратор</a>	Декоратор линии
<a href="#">Отражение</a>	Отражение
<a href="#">Тип соединения</a>	Тип соединения
<a href="#">Цвет пера</a>	Цвет внешней границы объекта

<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки

## 1.1.4.2.4.10.25.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.25.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

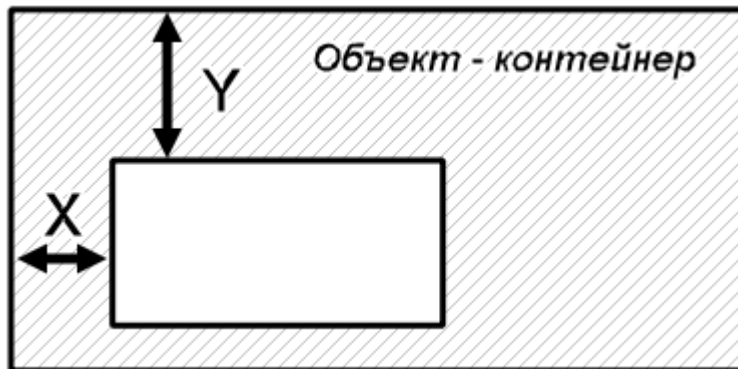
## 1.1.4.2.4.10.25.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

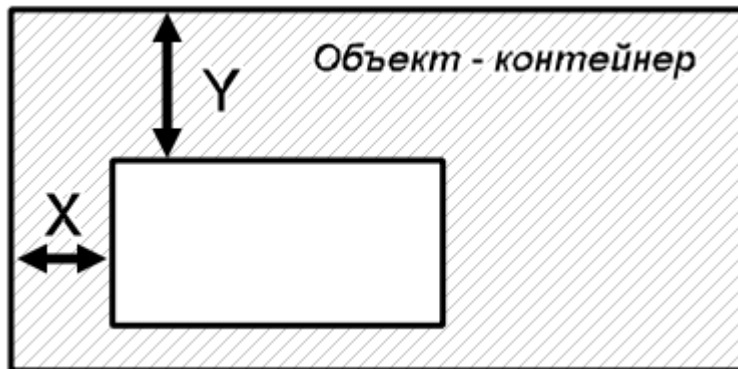
## 1.1.4.2.4.10.25.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.2.4.10.25.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.25.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.25.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.25.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.25.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.25.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.25.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.25.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

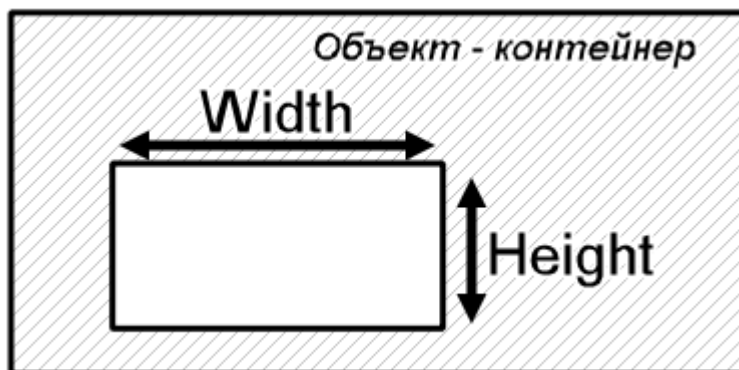


## 1.1.4.2.4.10.25.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

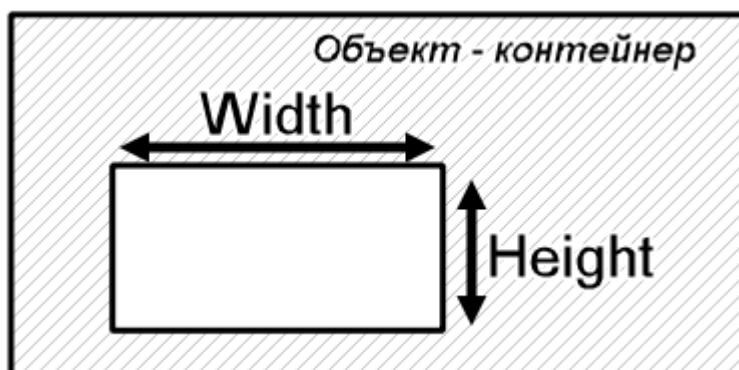


## 1.1.4.2.4.10.25.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.25.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.25.3.16. Декоратор

Декоратор линии.



bool EnableDecoration

## 1.1.4.2.4.10.25.3.17. Тип соединения

Тип соединения.



uint4 JoinType

## 1.1.4.2.4.10.25.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```


## 1.1.4.2.4.10.25.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;
```

## 1.1.4.2.4.10.25.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```



## 1.1.4.2.4.10.25.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





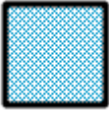

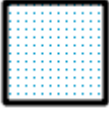
## 1.1.4.2.4.10.25.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



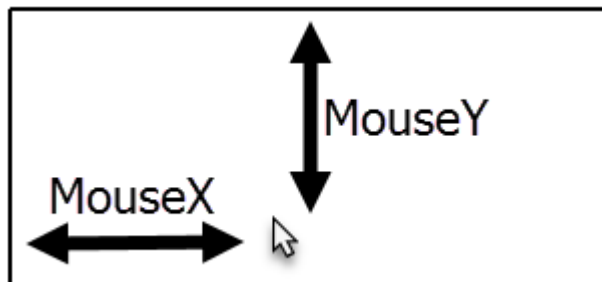
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.4.10.25.3.23. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

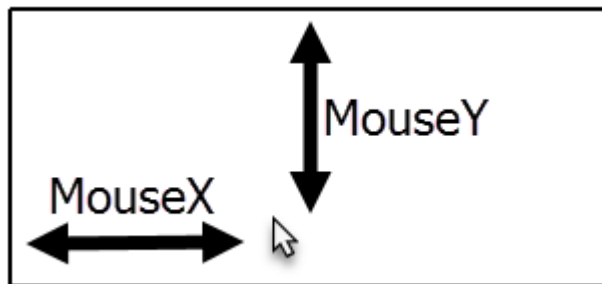
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.25.3.24. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.25.3.25. Lenght

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Lenght



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.25.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.26. Кривая линия

Элемент для создания криволинейных прямых.

Структура объекта

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Button_1	Кнопка
Spline_1	Кривая линия
Точки	
Point_1	Точка
Point_2	Точка
Point_3	Точка

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		Spline_1
Кардинальное число		1
X	R W	839
Y	R W	277
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	185.4495951417004
Высота	R W	184.15294117647056
Фокус ввода	R W	<не определено>
Цвет пера	R W	4278190080
Стиль пера	R W	Сплошная линия
Толщина пера	R W	1
Цвет заливки	R W	4278190080



# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.26.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.26.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.26.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.26.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.26.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.26.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

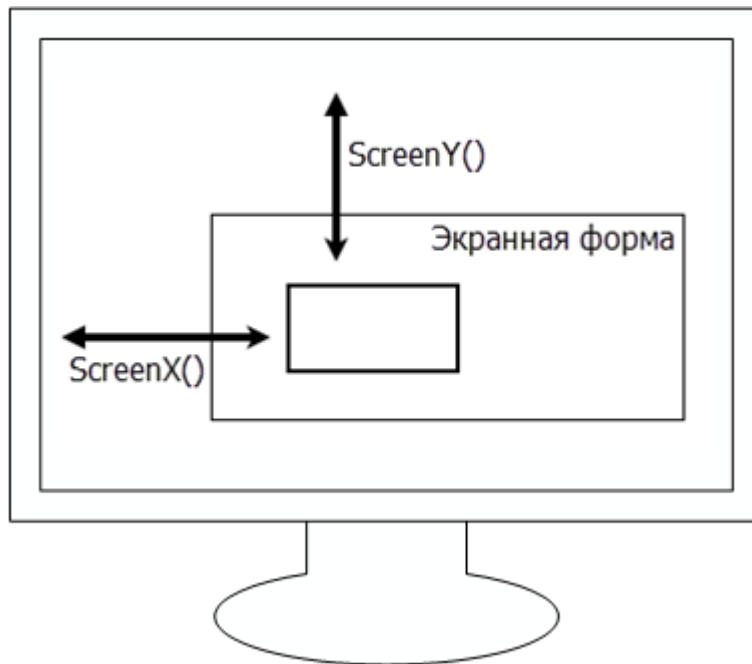
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.26.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

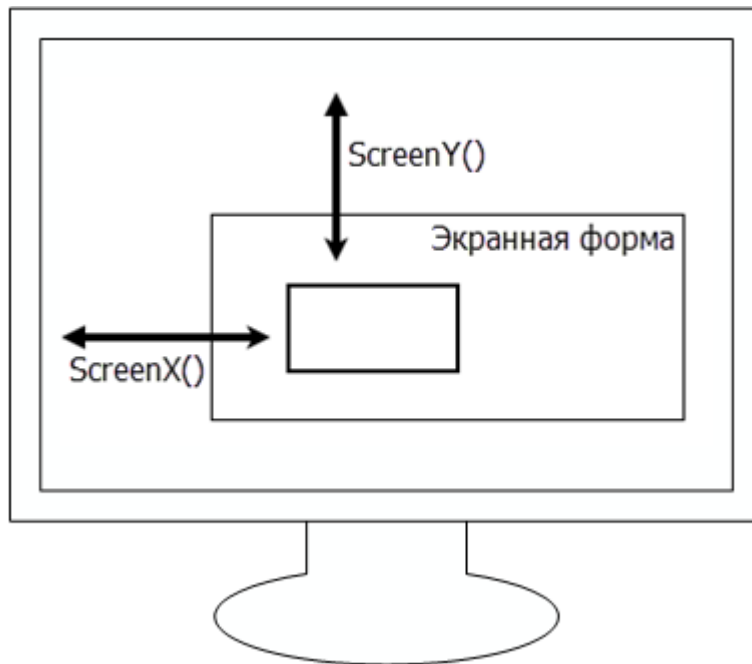
```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.26.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



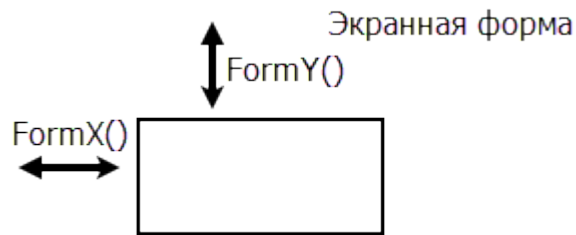
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.26.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



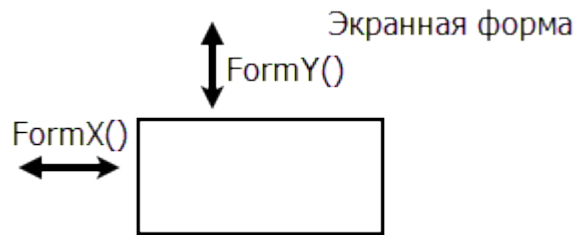
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.26.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.26.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<u>Стиль заливки</u>	Заливает объект одним из стилей заливки
<u>Отражение</u>	Отражение

## 1.1.4.2.4.10.26.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.26.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

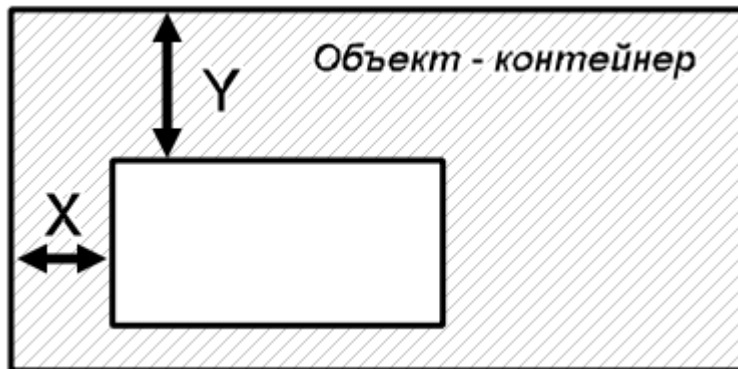
## 1.1.4.2.4.10.26.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```



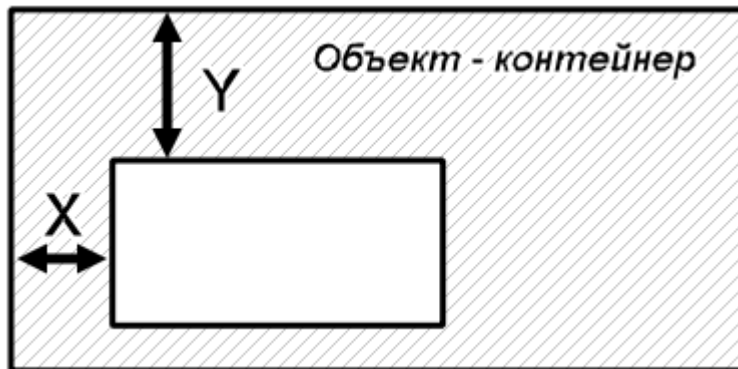
## 1.1.4.2.4.10.26.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.26.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.26.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.26.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.26.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.26.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.10.26.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.26.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.2.4.10.26.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

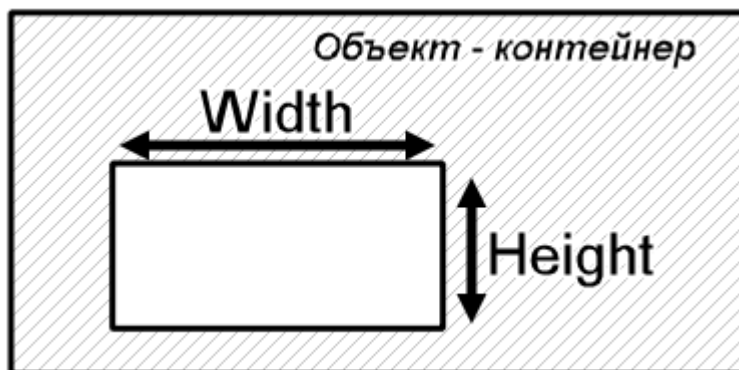
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.26.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

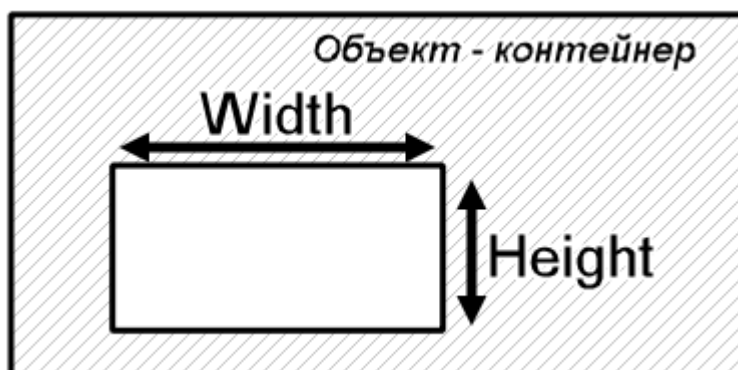


## 1.1.4.2.4.10.26.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.26.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.26.3.16. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.2.4.10.26.3.17. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.26.3.18. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.2.4.10.26.3.19. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.BrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.BrushColor = 4294901760;
```




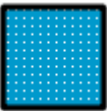
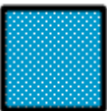
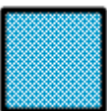

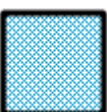
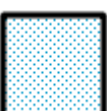
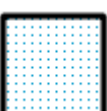
## 1.1.4.2.4.10.26.3.20. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



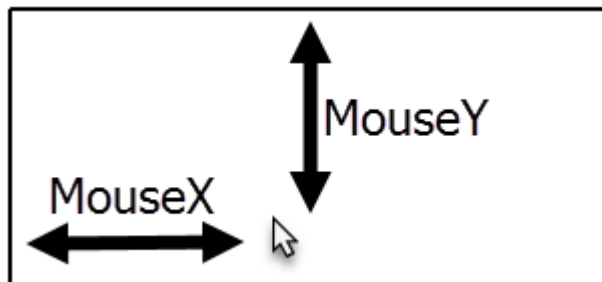
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.4.10.26.3.21. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

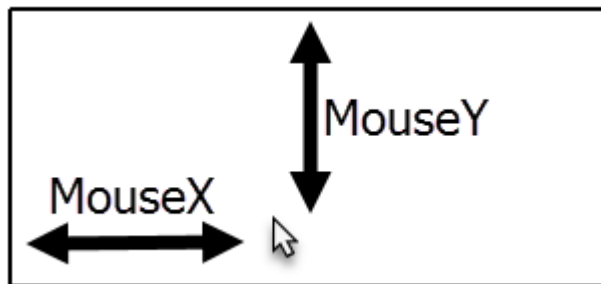
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.26.3.22. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.26.3.23. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.26.3.24. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.27. Системная кнопка

Компонент применяется на мнемосхемах для взаимодействия с пользователем в диалоговых окнах.

The screenshot displays two windows from the Visual Studio IDE. The top window, titled "Структура объекта" (Object Structure), shows a tree view of the project's components. Under "Графические объекты" (Visual Objects), "SystemButton\_1" is selected, with a description of "Системная кнопка" (System button). The bottom window, titled "Редактор свойств" (Properties Editor), shows a list of properties for the selected control. The properties are organized into columns: "Свойство" (Property), "Характеристики" (Characteristics), and "Значение" (Value).

Свойство	Характеристики	Значение
Отображаемое имя		SystemButton_1
Кардинальное число		1
X	R W	800
Y	R W	327
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	101
Высота	R W	23
Фокус ввода	R W ⚡	<не определено>
Удерживаемая	R W	false
Нажата	R W	false
Текст	R W	Системная кнопка
Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0
Цвет шрифта	R W	4278190080

## 1.1.4.2.4.10.27.1. События

Событие	Описание
<a href="#">ButtonPressed</a>	Нажатие на кнопку
<a href="#">FocusChanged</a>	Изменение фокуса у объекта



## 1.1.4.2.4.10.27.1.1. ButtonPressed

Нажатие на кнопку.

### Параметры

Параметр	Тип	Описание
checked	bool	Указывает была ли нажата (true) или отпущена (false) кнопка

## 1.1.4.2.4.10.27.1.2. FocusChanged

Изменение фокуса у объекта.

### Параметры

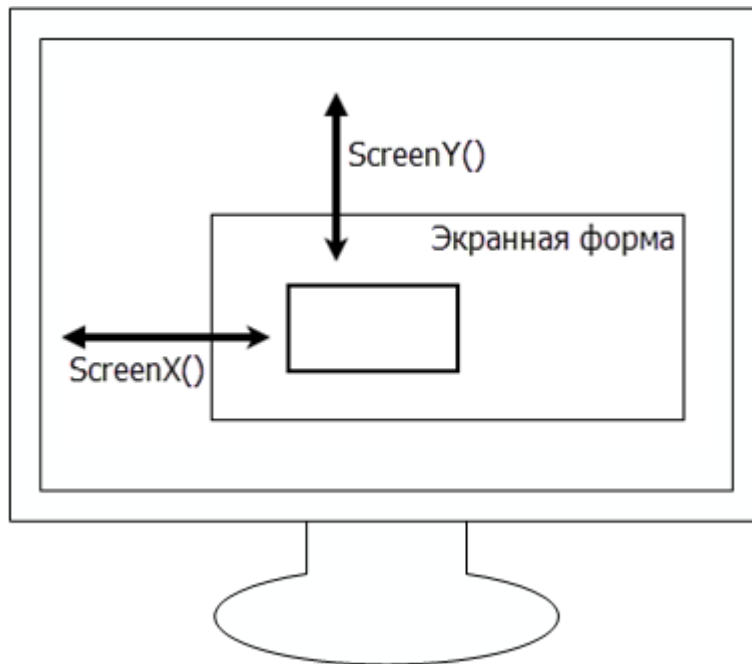
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">Click</a>	Имитировать нажатие кнопки

## 1.1.4.2.4.10.27.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



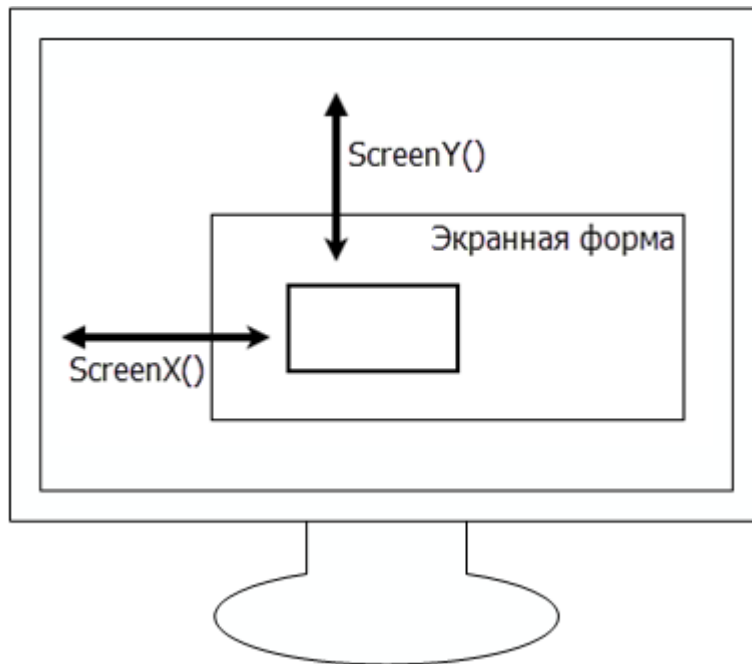
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.27.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



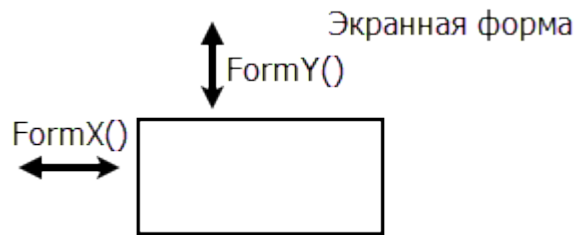
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.27.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



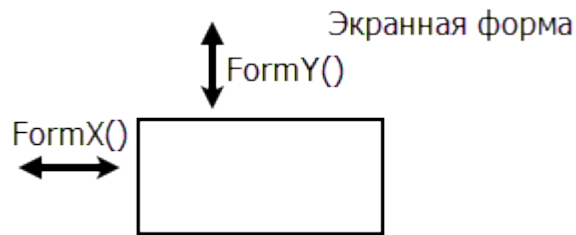
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.27.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.27.2.5. Click

Имитировать нажатие кнопки.



void Click()



## 1.1.4.2.4.10.27.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Удерживаемая</a>	Оставляет кнопку удерживаемой/отпускаемой после нажатия
<a href="#">Z-значение</a>	Z-значение

<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Нажата</a>	Нажата ли кнопка
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.27.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.27.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

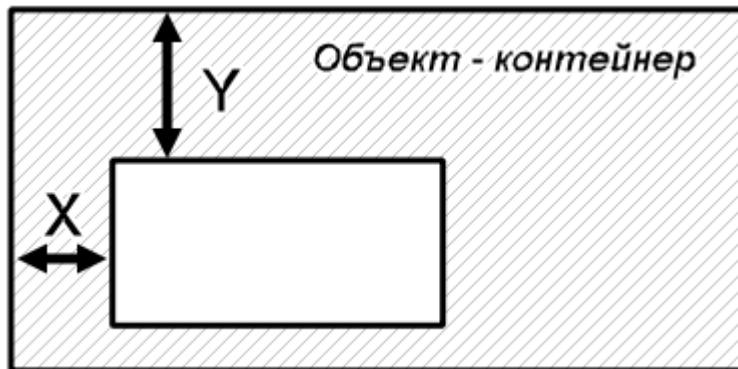
## 1.1.4.2.4.10.27.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

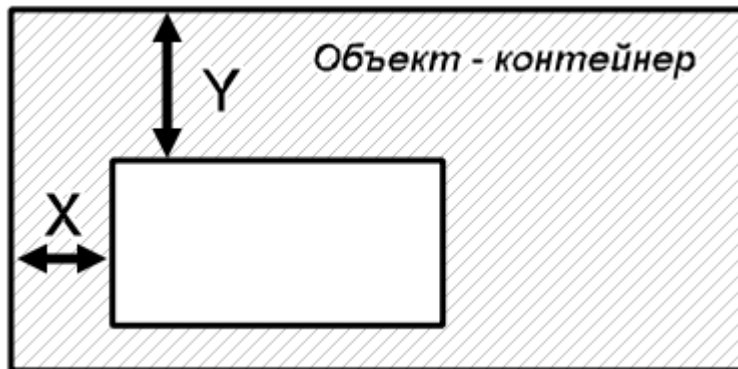
## 1.1.4.2.4.10.27.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.27.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.27.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```



## 1.1.4.2.4.10.27.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.27.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.27.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.27.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.27.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.27.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

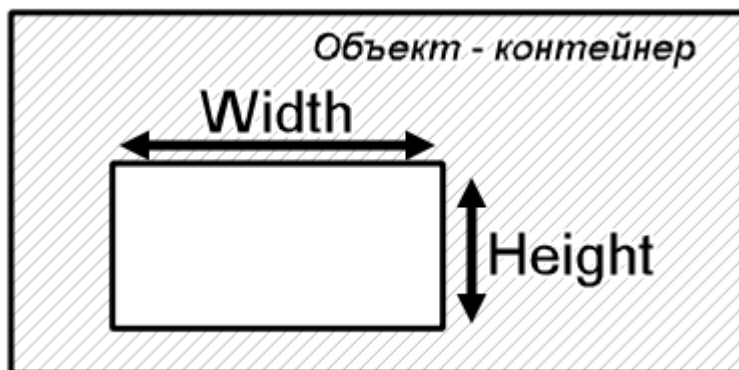
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.27.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

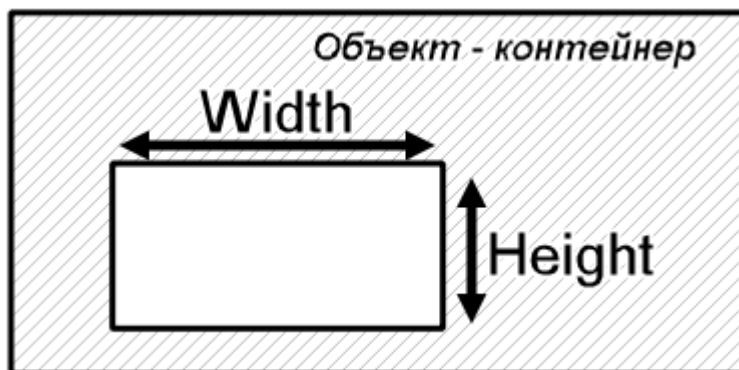


## 1.1.4.2.4.10.27.3.14. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.2.4.10.27.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.27.3.16. Удерживаемая

Оставляет кнопку удерживаемой/отпускаемой после нажатия.



bool Checkable

### Значение

Значение	Описание
true	Удерживать кнопку после нажатия
false	Отпускать кнопку после нажатия

## 1.1.4.2.4.10.27.3.17. Нажата

Нажата ли кнопка.



bool Checked

## 1.1.4.2.4.10.27.3.18. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.2.4.10.27.3.19. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.27.3.20. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

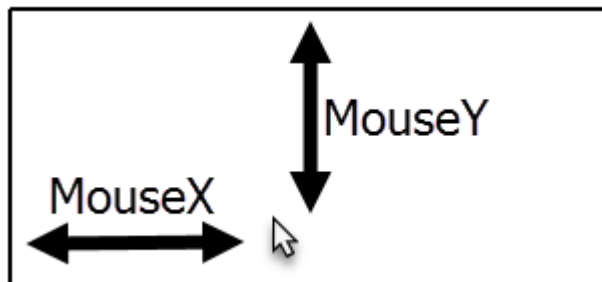
```
Object.FontColor = 4294901760;
```

## 1.1.4.2.4.10.27.3.21. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

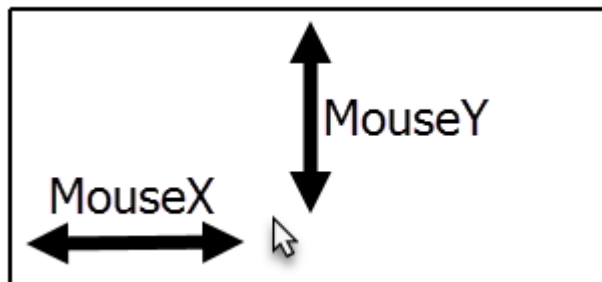
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.27.3.22. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```



## 1.1.4.2.4.10.27.3.23. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.27.3.24. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.28. Флажок

Компонент Флажок применяется на мнемосхемах для организации опциональных возможностей.

The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project's components. Under 'MainForm', the 'Графические объекты' (Visual Objects) folder is expanded, showing 'Button\_1' and 'Checkbox\_1'. 'Checkbox\_1' is selected and highlighted. The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties of the selected 'Checkbox\_1' control. The properties are listed in a table with columns for 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		Checkbox_1
Кардинальное число		1
>  X	R W	925
>  Y	R W	203
>  Z-значение	R W	0
>  Угол поворота	R W	0
>  Масштаб	R W	1
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Ширина	R W	63
>  Высота	R W	17
>  Фокус ввода	R W ⚡	<не определено>
>  Текст	R W	Флажок
>  Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0
>  Цвет шрифта	R W	4278190080
>  Состояние	R W ⚡	false

## 1.1.4.2.4.10.28.1. События

Событие	Описание
<a href="#">StateChanged</a>	Изменение состояния флажка
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">Clicked</a>	Событие клика

## 1.1.4.2.4.10.28.1.1. StateChanged

Изменение состояния флажка.

### Параметры

Параметр	Тип	Описание
state	bool	Указывает был ли установлен флаг (true) или снят (false)

## 1.1.4.2.4.10.28.1.2. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.28.1.3. Clicked

Событие клика.

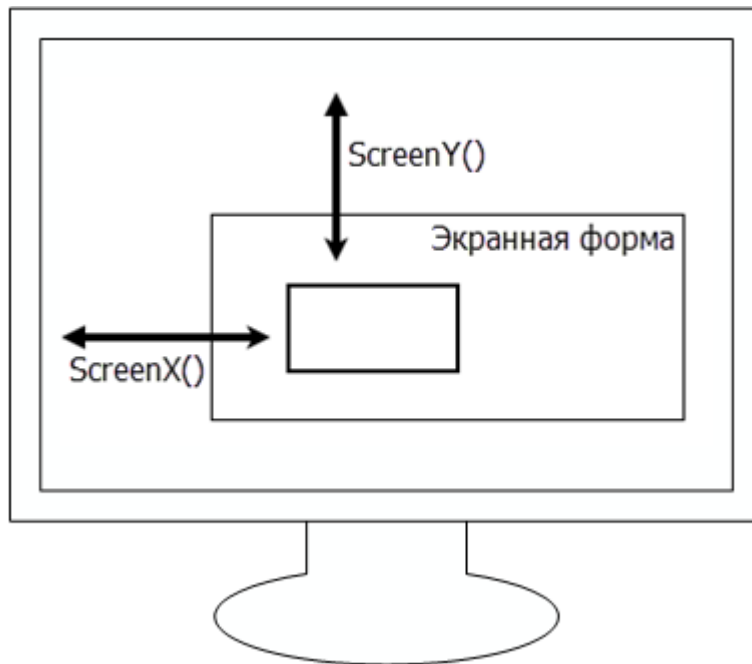
## Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме



## 1.1.4.2.4.10.28.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



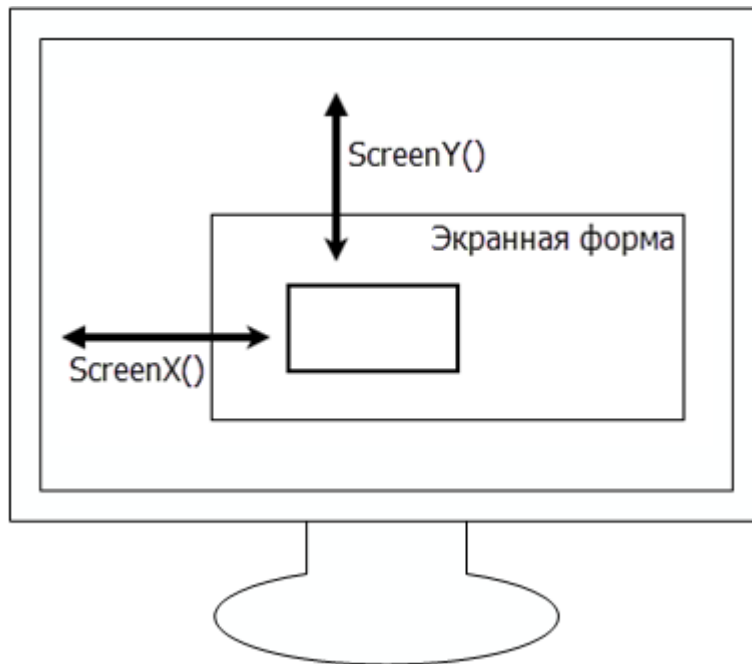
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.28.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



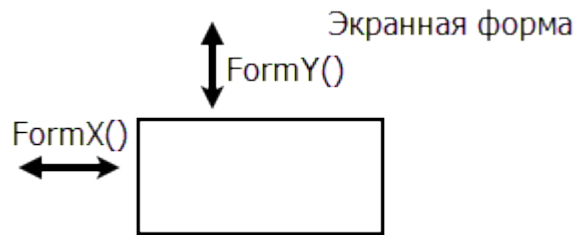
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.28.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



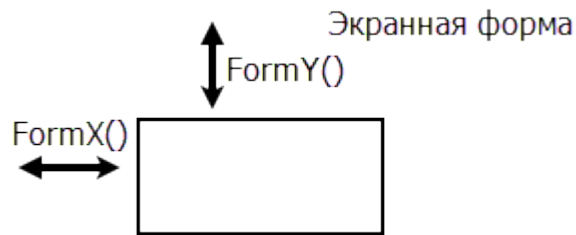
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.28.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.28.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Состояние</a>	Состояние флажка
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность

<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.28.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.28.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



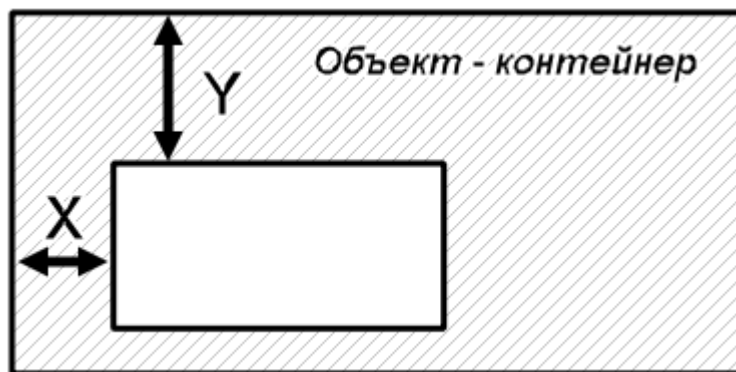
## 1.1.4.2.4.10.28.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.2.4.10.28.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.28.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.28.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.28.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.28.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.28.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.10.28.3.10. Непрозрачность

Непрозрачность.



float8 Opacity



## 1.1.4.2.4.10.28.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.28.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

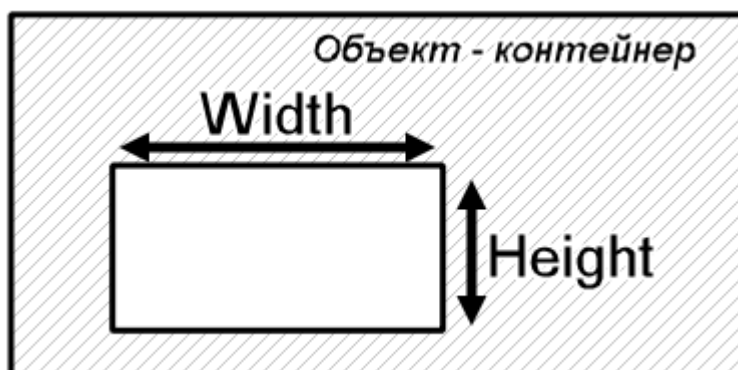
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.28.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

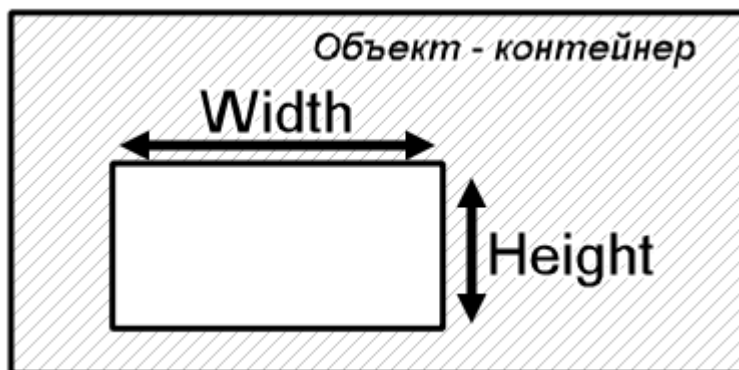


## 1.1.4.2.4.10.28.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.28.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.28.3.16. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.2.4.10.28.3.17. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.28.3.18. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```



## 1.1.4.2.4.10.28.3.19. Состояние

Состояние флажка.



bool State

### Значение

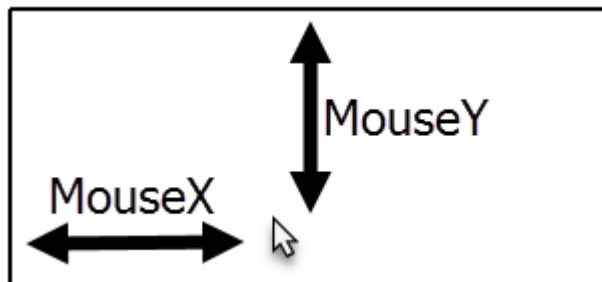
Значение	Описание
true	Флажок включен
false	Флажок выключен

## 1.1.4.2.4.10.28.3.20. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

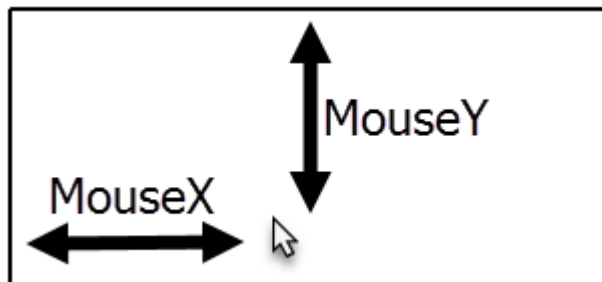
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.28.3.21. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.28.3.22. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.28.3.23. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
else  
    {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.29. Поле ввода

Компонент Поле ввода применяется на мнемосхемах в местах, где требуется ручной ввод неких значений.

The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project. Under 'Графические объекты' (Visual Objects), 'LineEdit\_1' is selected, with a description of 'Поле ввода' (Input field). The bottom window, titled 'Редактор свойств' (Properties Editor), shows a list of properties for the selected control. The properties are organized into columns: 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		LineEdit_1
Кардинальное число		1
X	R W	822
Y	R W	257
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	133
Высота	R W	20
Фокус ввода	R W ⚡	< не определено >
Текст	R W ⚡ ⚙	Поле ввода
Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0
Цвет шрифта	R W	4278190080
Выравнивание текста	R W	По центру слева
Скрывать ввод	R W	false
Текст заполнителя	R W	
Только для чтения	R W	false



## 1.1.4.2.4.10.29.1. События

Событие	Описание
<a href="#">KeyPress</a>	Нажатие клавиши или сочетания клавиш в поле ввода
<a href="#">TextChanged</a>	Редактирование текста
<a href="#">EditFinished</a>	Завершение редактирования текста снятием фокуса с поля ввода или нажатием Enter
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.2.4.10.29.1.1. KeyPress

Нажатие клавиши или сочетания клавиш в поле ввода. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код нажатой клавиши   //При нажатии клавиши Enter записать текст из поля ввода в текстовое поле: <pre>if (event.KeyCode == 82)     {Text_1.Text = TextEdit_1.Text;}</pre>
Modifiers	int4	Код нажатой клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Alt, Shift   //При удержании клавиши F5 записать текст из поля ввода в текстовое поле: <pre>if (event.KeyCode == 55 &amp;&amp; event.Repeated == true)     {Text_1.Text = TextEdit_1.Text;}</pre>
Text	string	Текст нажатой клавиши



## 1.1.4.2.4.10.29.1.2. TextChanged

Редактирование текста.

### Параметры

Параметр	Тип	Описание
text	string	Новый текст в поле ввода

## 1.1.4.2.4.10.29.1.3. EditFinished

Завершение редактирования текста снятием фокуса с поля ввода или нажатием Enter.

### Параметры

Параметр	Тип	Описание
text	string	Новый текст в поле ввода

## 1.1.4.2.4.10.29.1.4. FocusChanged

Изменение фокуса у объекта.

### Параметры

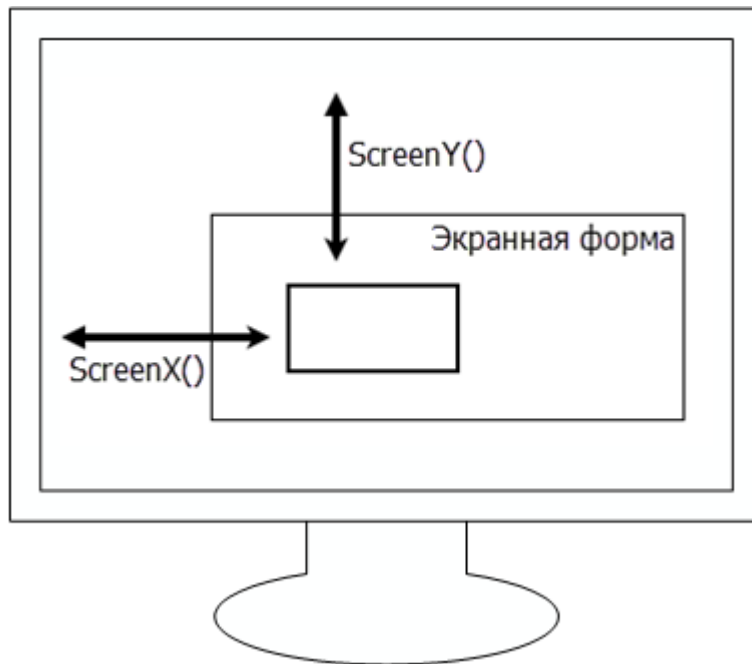
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.29.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



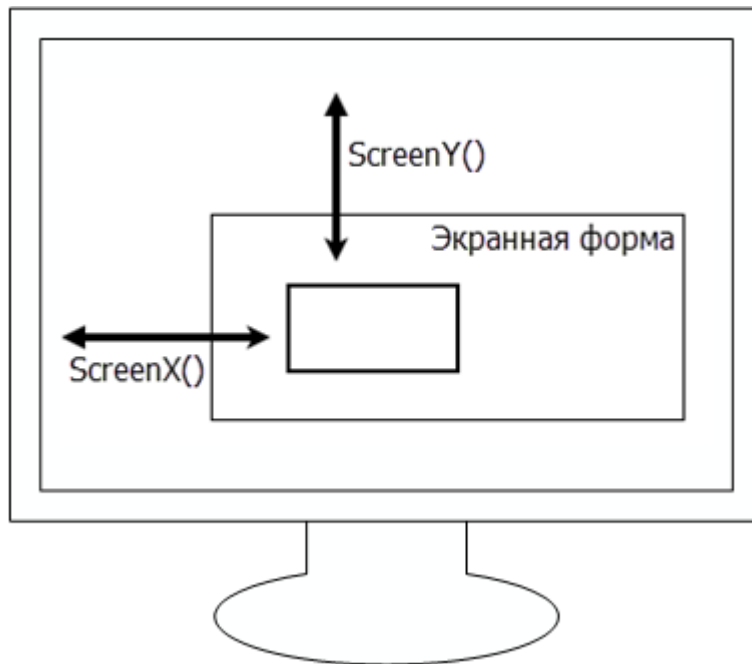
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.29.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



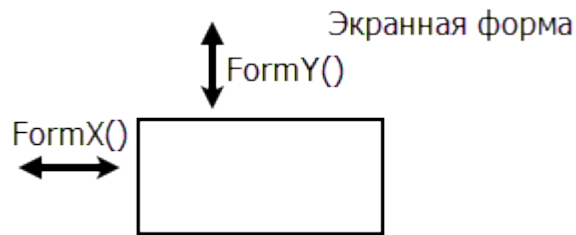
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.29.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



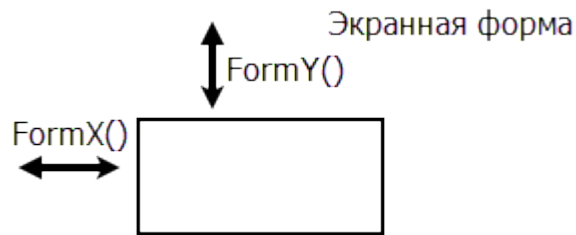
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.29.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.29.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Текст заполнителя</a>	Текст, который отображается в пустом текстовом поле
<a href="#">Скрывать ввод</a>	Режим маскировки вводимого текста

<a href="#">Выравнивание текста</a>	Выравнивание текста
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Только для чтения</a>	Доступно для чтения
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.29.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.29.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.29.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.2.4.10.29.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.29.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.29.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```



## 1.1.4.2.4.10.29.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.29.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.29.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.2.4.10.29.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.29.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.29.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

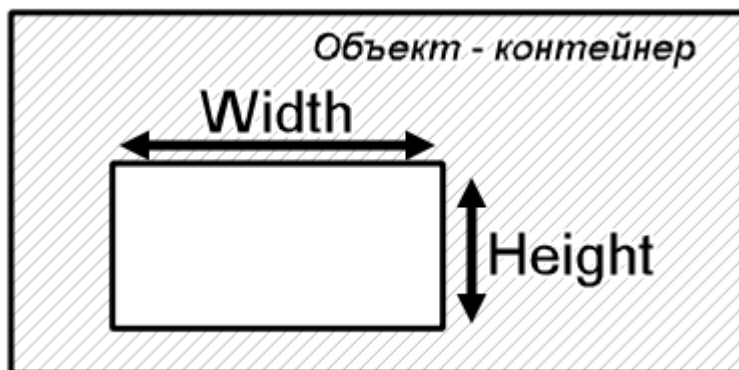
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.29.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

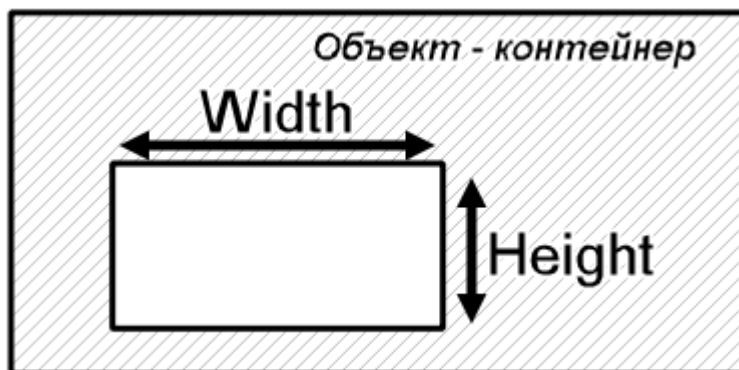


## 1.1.4.2.4.10.29.3.14. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.2.4.10.29.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.29.3.16. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.2.4.10.29.3.17. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.29.3.18. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

## 1.1.4.2.4.10.29.3.19. Выравнивание текста



int4 TextAlignment

### Значение

Значение	Описание
33	Слева сверху
65	Слева снизу
129	Слева по центру
34	Справа сверху
66	Справа снизу
130	Справа по центру
36	По центру сверху
68	По центру снизу
132	По центру объекта
40	По всей ширине сверху
72	По всей ширине снизу
136	По всей ширине по центру

## 1.1.4.2.4.10.29.3.20. Скрывать ввод

Режим маскировки вводимого текста.



bool HideInput

### Значение

Значение	Описание
true	Вводимые символы маскируются
false	Вводимые символы не маскируются

## 1.1.4.2.4.10.29.3.21. Текст заполнителя

Текст, который отображается в пустом текстовом поле. Чтобы отображался текст заполнителя, свойство Текст должно быть незаполненным.



string PlaceholderText

Поле ввода

Введите пароль

## 1.1.4.2.4.10.29.3.22. Только для чтения

Доступно для чтения.



bool ReadOnly

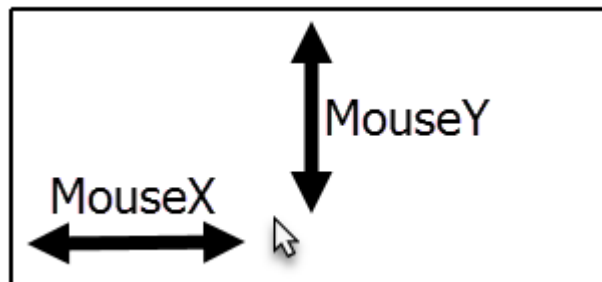


## 1.1.4.2.4.10.29.3.23. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

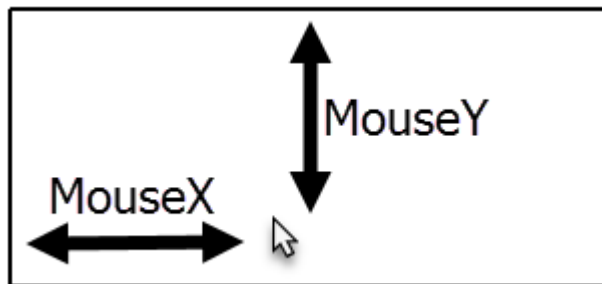
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.29.3.24. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.29.3.25. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.29.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.30. Редактор текста

Компонент для редактирования текста.

The image shows two windows from the Visual Studio IDE. The top window, titled "Структура объекта" (Object Structure), displays a tree view of the UI components. Under "MainForm", there is a "Графические объекты" (Visual Objects) folder containing "Button\_1" and "TextEdit\_1". "TextEdit\_1" is selected and highlighted in blue, with its description "Редактор текста" (Text Editor) shown to the right. The bottom window, titled "Редактор свойств" (Properties Editor), shows a list of properties for the selected "TextEdit\_1" control. The properties are organized into columns: "Свойство" (Property), "Характеристики" (Characteristics), and "Значение" (Value).

Свойство	Характеристики	Значение
Отображаемое имя		TextEdit_1
Кардинальное число		1
>  X	R W	784
>  Y	R W	220
>  Z-значение	R W	0
>  Угол поворота	R W	0
>  Масштаб	R W	1
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Ширина	R W	256
>  Высота	R W	192
>  Фокус ввода	R W	< не определено >
>  Текст	R W	Редактор текста
>  Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0,0
>  Цвет шрифта	R W	4278190080
>  Выравнивание текста	R W	По левому краю
>  Текст заполнителя	R W	
>  Только для чтения	R W	false
>  Режим переноса по словам	R W	По границам слов или в любом месте



## 1.1.4.2.4.10.30.1. События

Событие	Описание
<a href="#">KeyPress</a>	Нажатие клавиши или сочетания клавиш в поле ввода
<a href="#">TextChanged</a>	Редактирование текста
<a href="#">EditFinished</a>	Завершение редактирования текста снятием фокуса с поля ввода или нажатием Enter
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.2.4.10.30.1.1. KeyPress

Нажатие клавиши или сочетания клавиш в поле ввода. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код нажатой клавиши   //При нажатии клавиши Enter записать текст из поля ввода в текстовое поле: <pre>if (event.KeyCode == 82)     {Text_1.Text = TextEdit_1.Text;}</pre>
Modifiers	int4	Код нажатой клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Alt, Shift   //При удержании клавиши F5 записать текст из поля ввода в текстовое поле: <pre>if (event.KeyCode == 55 &amp;&amp; event.Repeated == true)     {Text_1.Text = TextEdit_1.Text;}</pre>
Text	string	Текст нажатой клавиши

## 1.1.4.2.4.10.30.1.2. TextChanged

Редактирование текста.

### Параметры

Параметр	Тип	Описание
text	string	Новый текст в поле ввода



## 1.1.4.2.4.10.30.1.3. EditFinished

Завершение редактирования текста снятием фокуса с поля ввода или нажатием Enter.

### Параметры

Параметр	Тип	Описание
text	string	Новый текст в поле ввода

## 1.1.4.2.4.10.30.1.4. FocusChanged

Изменение фокуса у объекта.

### Параметры

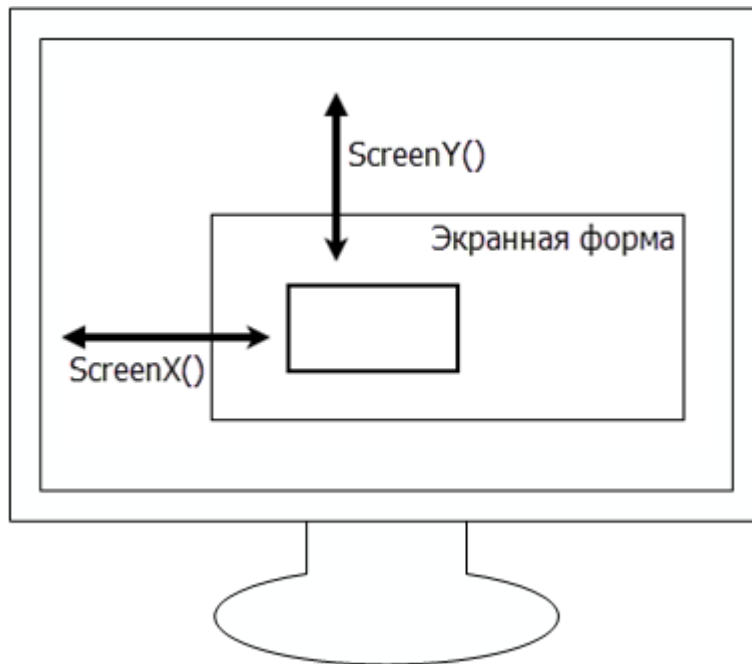
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">Apped</a>	Добавляет строку в конце

## 1.1.4.2.4.10.30.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



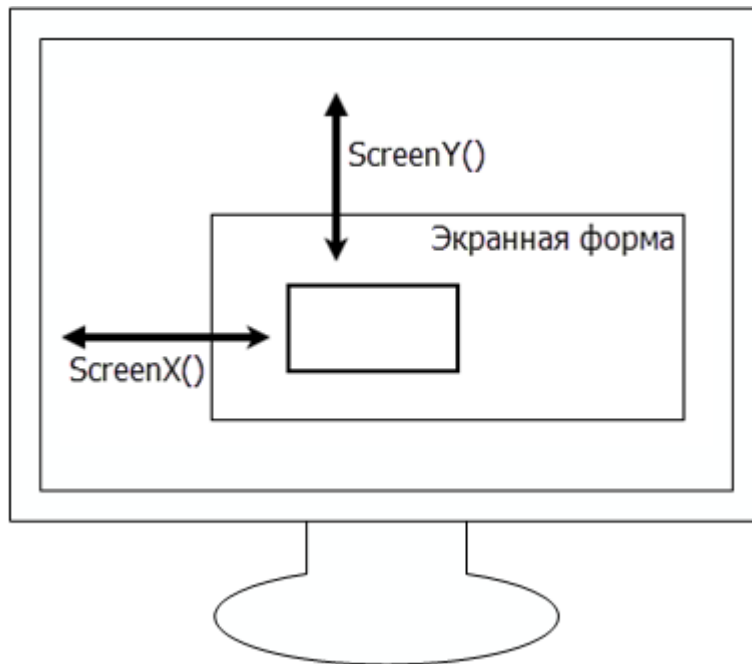
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.30.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



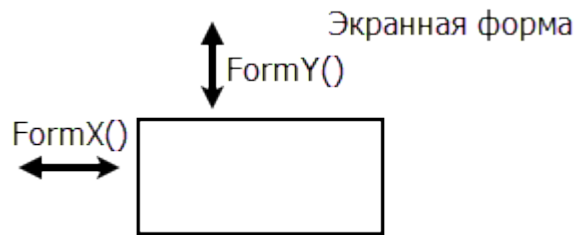
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.30.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



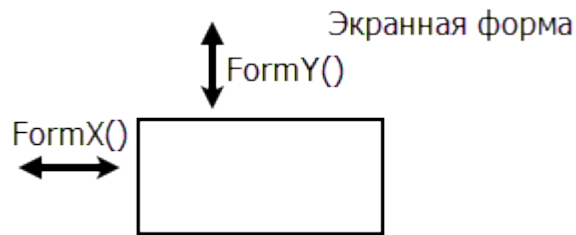
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.30.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.30.2.5. Apped

Добавить строку в конце.



void Append(string Text)

### Параметры

Параметр	Тип	Описание
Text	string	Строка для добавления



## 1.1.4.2.4.10.30.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Текст заполнителя</a>	Текст, который отображается в пустом текстовом поле
<a href="#">Выравнивание текста</a>	Выравнивание текста
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность
<a href="#">Только для чтения</a>	Доступно для чтения

<a href="#">Режим переноса по словам</a>	Режим переноса по слогам
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.30.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.30.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.30.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.2.4.10.30.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.30.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.30.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```



## 1.1.4.2.4.10.30.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.30.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.30.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	объект виден
false	объект скрыт

## 1.1.4.2.4.10.30.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.30.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	объект активен
false	объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.30.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

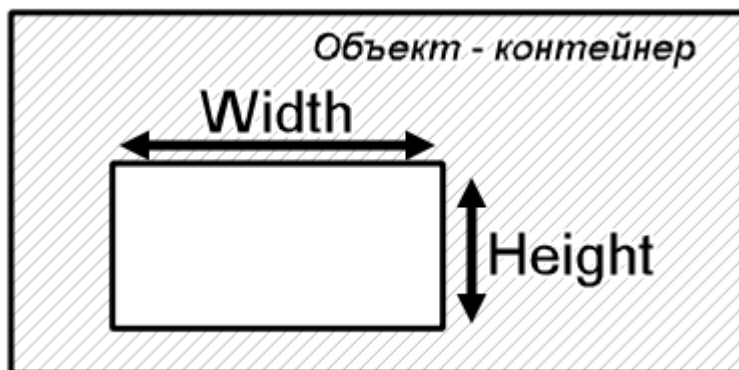
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.30.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

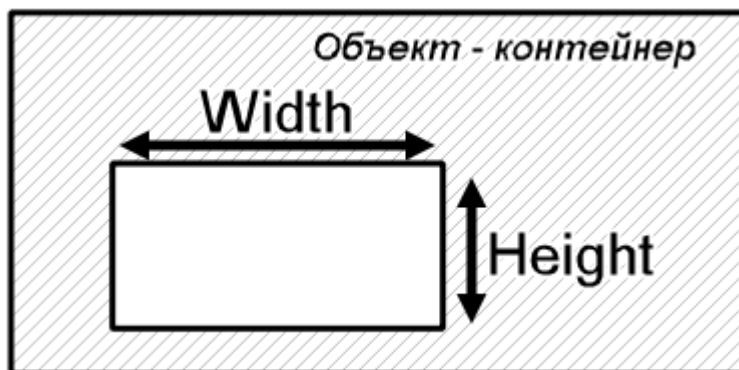


## 1.1.4.2.4.10.30.3.14. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.2.4.10.30.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	фокус на объекте
false	фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.30.3.16. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.2.4.10.30.3.17. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.30.3.18. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

## 1.1.4.2.4.10.30.3.19. Выравнивание текста



int4 TextAlignment

### Значение

Значение	Описание
33	слева сверху
65	слева снизу
129	слева по центру
34	справа сверху
66	справа снизу
130	справа по центру
36	по центру сверху
68	по центру снизу
132	по центру объекта
40	по всей ширине сверху
72	по всей ширине снизу
136	по всей ширине по центру

## 1.1.4.2.4.10.30.3.20. Текст заполнителя

Текст, который отображается в пустом текстовом поле. Чтобы отображался текст заполнителя, свойство Текст должно быть незаполненным.



string PlaceholderText

Поле ввода

Введите пароль

## 1.1.4.2.4.10.30.3.21. Только для чтения

Доступно для чтения.



bool ReadOnly

## 1.1.4.2.4.10.30.3.22. Режим переноса по словам

Режим переноса по словам.



int4 WordWrapMode



## 1.1.4.2.4.10.30.3.23. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.30.3.24. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



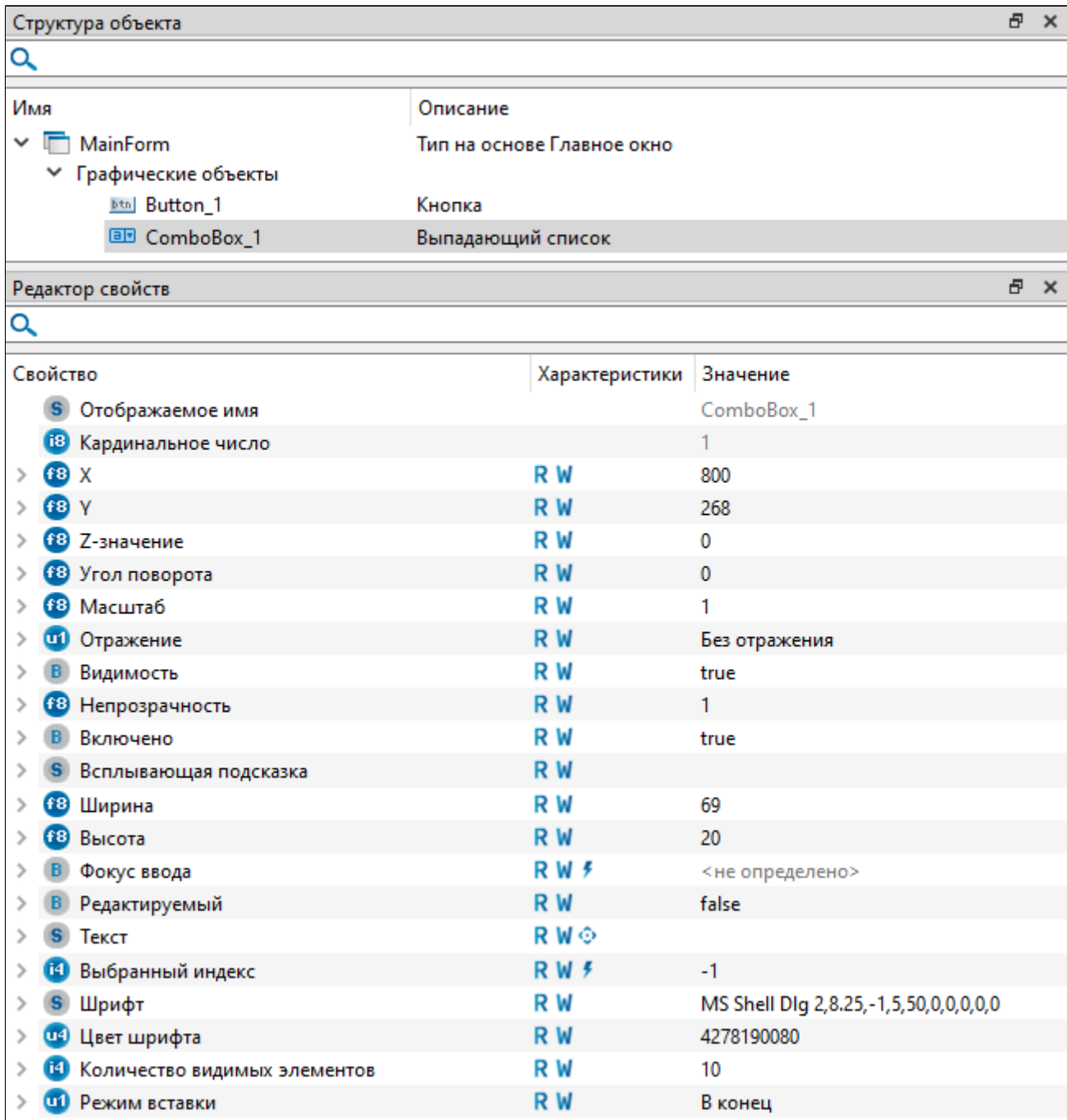
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.31. Выпадающий список

Компонент Выпадающий список служит для выбора пользователем некоторых предустановленных значений.



The image shows two windows from the Visual Studio IDE. The top window, titled "Структура объекта" (Object Structure), displays a tree view of the project's components. Under "Графические объекты" (Visual Objects), "ComboBox\_1" is selected, with the description "Выпадающий список" (Dropdown list). The bottom window, titled "Редактор свойств" (Properties Editor), shows the properties of the selected ComboBox\_1 control. The properties are listed in a table with columns for the property name, its characteristics (Read/Write), and its current value.

Свойство	Характеристики	Значение
Отображаемое имя		ComboBox_1
Кардинальное число		1
X	R W	800
Y	R W	268
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	69
Высота	R W	20
Фокус ввода	R W ⚡	<не определено>
Редактируемый	R W	false
Текст	R W ↻	
Выбранный индекс	R W ⚡	-1
Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0,0
Цвет шрифта	R W	4278190080
Количество видимых элементов	R W	10
Режим вставки	R W	В конец

## 1.1.4.2.4.10.31.1. События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">TextChanged</a>	Текст изменен
<a href="#">SelectedChanged</a>	Смена выбранного элемента
<a href="#">Activated</a>	Активация элемента пользователем

## 1.1.4.2.4.10.31.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.31.1.2. TextChanged

Событие изменения текста.

## 1.1.4.2.4.10.31.1.3. SelectedChanged

Событие смены выбранного элемента.

## 1.1.4.2.4.10.31.1.4. Activated

Активация элемента пользователем.

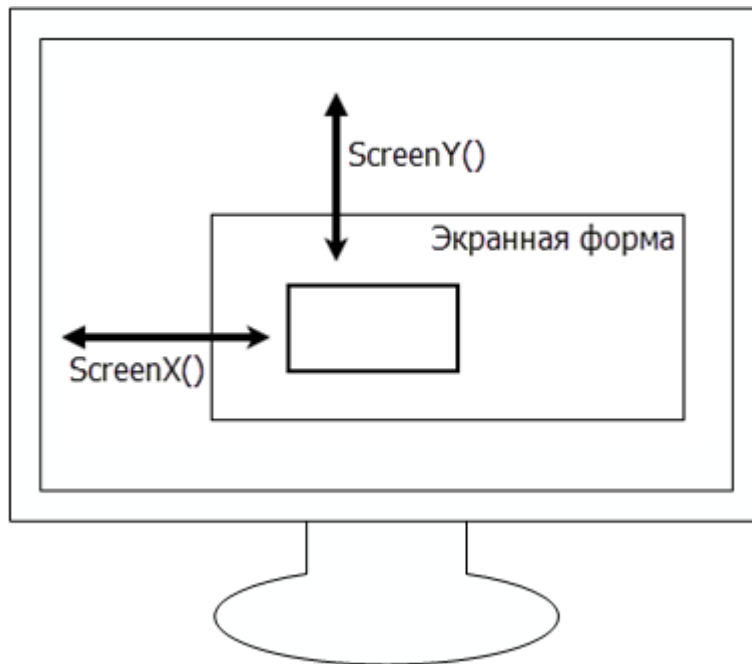


# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">Clear</a>	Очистить список
<a href="#">GetItem</a>	Получить текст элемента по индексу
<a href="#">IndexOf</a>	Получить индекс элемента по его тексту
<a href="#">SetItem</a>	Установить текст элемента по индексу
<a href="#">AddItem</a>	Добавить новый элемент в список
<a href="#">RemoveItem</a>	Удалить элемент из списка

## 1.1.4.2.4.10.31.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



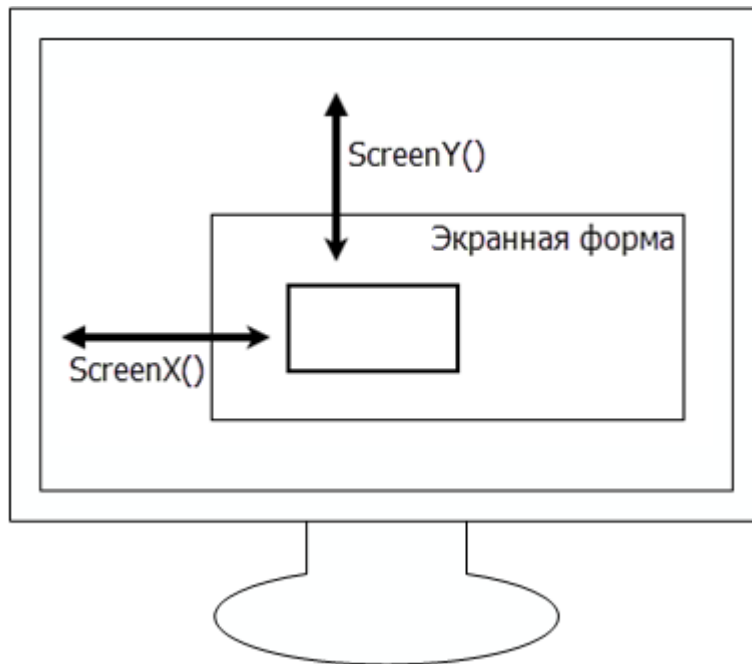
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.31.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



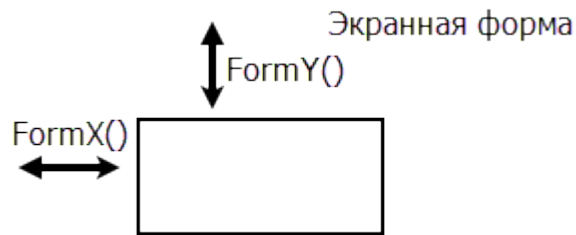
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.31.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



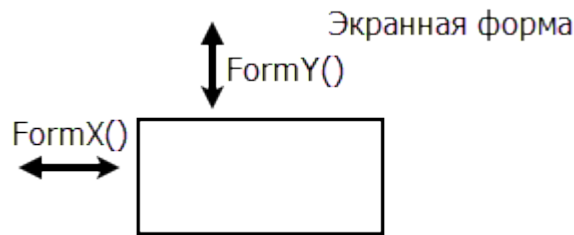
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.31.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.31.2.5. Clear

Очистить список.



`void Clear()`

## 1.1.4.2.4.10.31.2.6. GetItem

Получить текст элемента по индексу.



```
string GetItem(int4 itemIndex)
```

### Параметры

Параметр	Тип	Описание
itemIndex	int4	Индекс элемента

## 1.1.4.2.4.10.31.2.7. IndexOf

Получить индекс элемента по его тексту.



int4 IndexOf(string itemText)

### Параметры

Параметр	Тип	Описание
itemText	string	Текст элемента



## 1.1.4.2.4.10.31.2.8. SetItem

Установить текст элемента по индексу.



```
void SetItem(int4 itemIndex, string itemText)
```

### Параметры

Параметр	Тип	Описание
itemIndex	int4	Индекс элемента
itemText	string	Текст элемента

## 1.1.4.2.4.10.31.2.9. AddItem

Добавить новый элемент в список.



```
void AddItem(string itemText)
```

### Параметры

Параметр	Тип	Описание
itemText	string	Новый элемент

## 1.1.4.2.4.10.31.2.10. RemoveItem

Удалить элемент из списка.



```
void RemoveItem(int4 itemIndex)
```

### Параметры

Параметр	Тип	Описание
itemIndex	int4	Удаляемый компонент

## 1.1.4.2.4.10.31.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Редактируемый</a>	Доступность поля для редактирования
<a href="#">Z-значение</a>	Z-значение

<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Выбранный индекс</a>	Индекс выбранного элемента
<a href="#">Количество видимых элементов</a>	Количество видимых элементов
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.31.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.31.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.31.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```



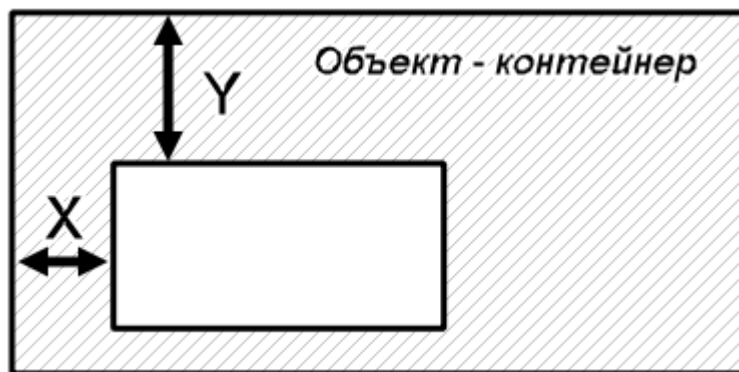
## 1.1.4.2.4.10.31.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.31.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.31.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.31.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.31.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.31.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.31.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.31.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.2.4.10.31.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

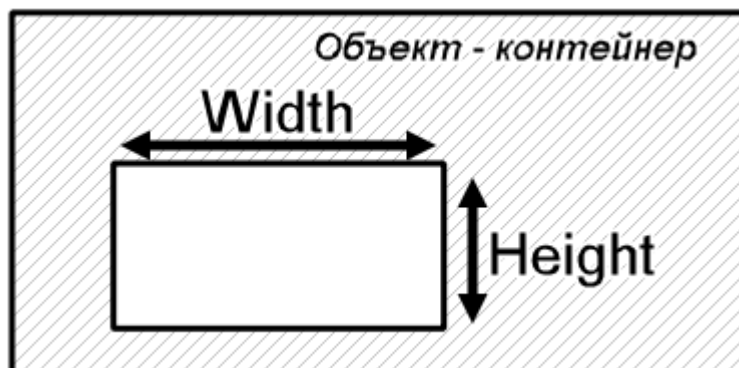
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.31.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

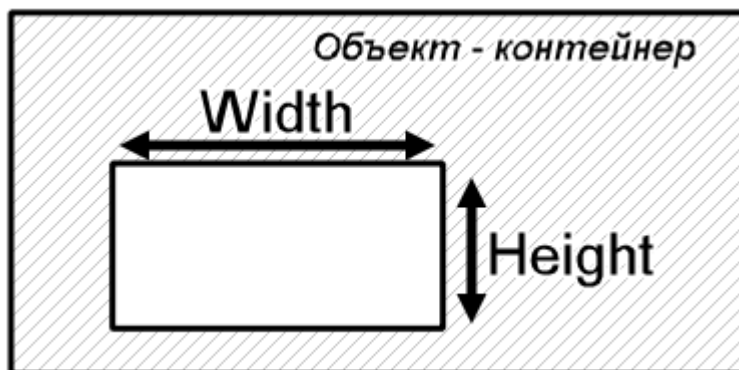


## 1.1.4.2.4.10.31.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.31.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.31.3.16. Редактируемый

Доступность поля для редактирования.



bool Editable

### Значение

Значение	Описание
true	В поле выпадающего списка можно ввести текст
false	В поле выпадающего списка нельзя ввести текст

## 1.1.4.2.4.10.31.3.17. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.2.4.10.31.3.18. Выбранный индекс

Индекс выбранного элемента.



int4 SelectedIndex

## 1.1.4.2.4.10.31.3.19. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font



## 1.1.4.2.4.10.31.3.20. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

# 1.1.4.2.4.10.31.3.21. Количество видимых элементов

Количество видимых элементов.



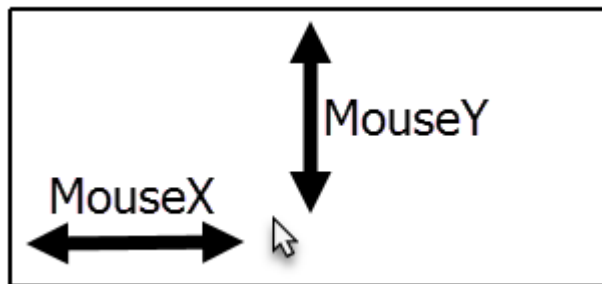
int4 VisibleItems

## 1.1.4.2.4.10.31.3.22. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

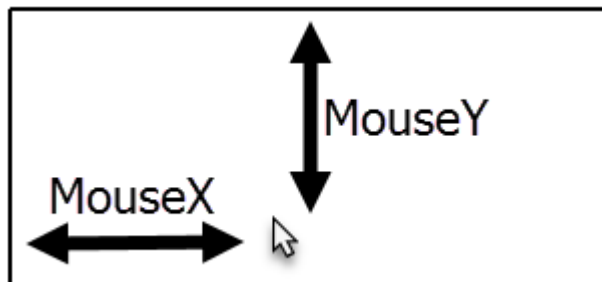
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.31.3.23. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.31.3.24. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.31.3.25. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



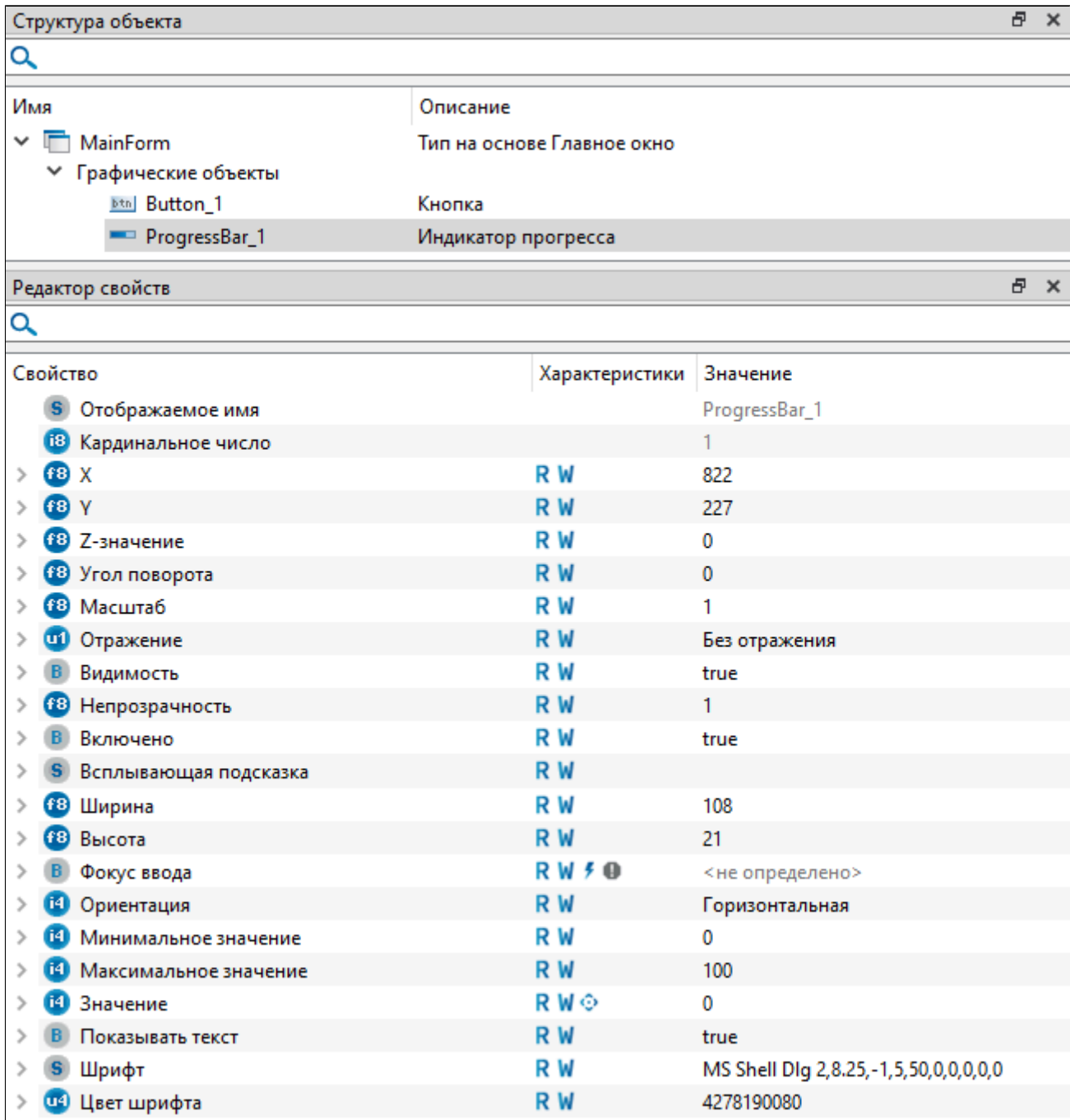
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.32. Индикатор прогресса

Компонент Индикатор прогресса используется для отображения процентной доли завершения процесса.



The image shows two windows from a development environment. The top window, titled "Структура объекта" (Object Structure), displays a tree view of the object hierarchy. Under "Графические объекты" (Visual Objects), "ProgressBar\_1" is selected. The bottom window, titled "Редактор свойств" (Properties Editor), shows the properties of the selected "ProgressBar\_1" component.

Имя	Описание
MainForm	Тип на основе Главное окно
Графические объекты	
Button_1	Кнопка
<b>ProgressBar_1</b>	Индикатор прогресса

Свойство	Характеристики	Значение
Отображаемое имя		ProgressBar_1
Кардинальное число		1
X	R W	822
Y	R W	227
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	108
Высота	R W	21
Фокус ввода	R W ⚡ ⓘ	<не определено>
Ориентация	R W	Горизонтальная
Минимальное значение	R W	0
Максимальное значение	R W	100
Значение	R W ⚙	0
Показывать текст	R W	true
Шрифт	R W	MS Shell Dlg 2,8.25,-1,5,50,0,0,0,0,0
Цвет шрифта	R W	4278190080

## 1.1.4.2.4.10.32.1. События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта



## 1.1.4.2.4.10.32.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

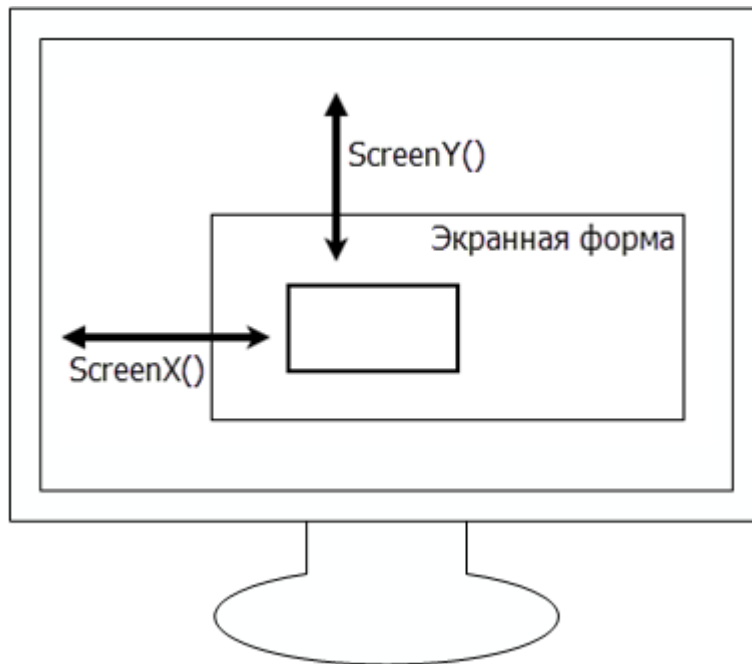
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.32.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



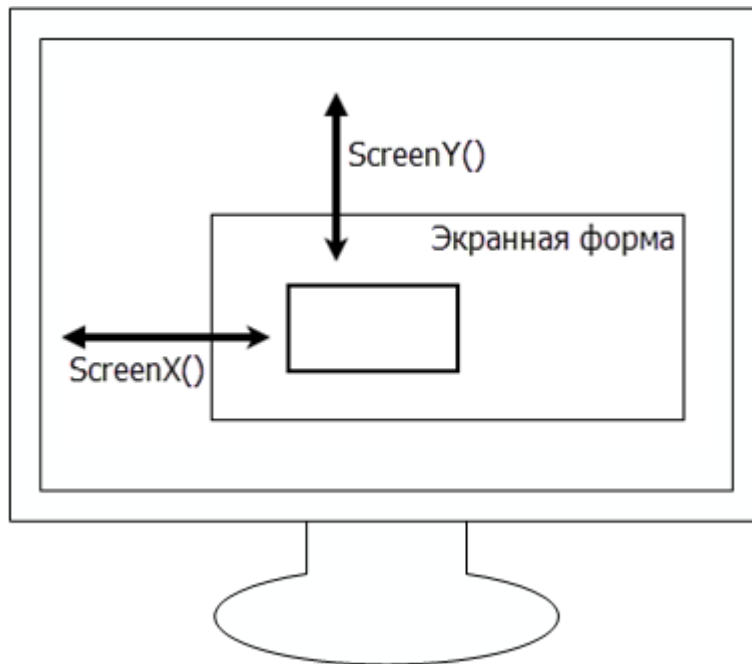
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.32.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



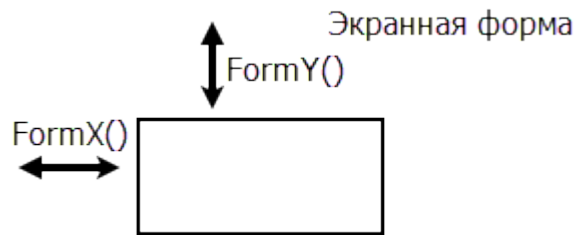
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.32.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



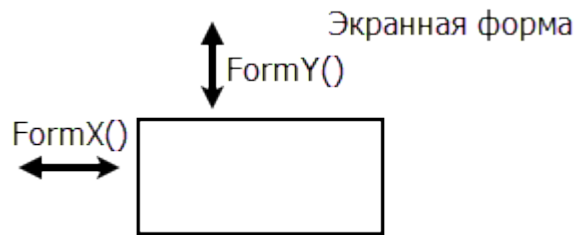
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.32.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.32.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Показывать текст</a>	Позволяет отобразить/скрыть текущее значение прогресса справа от индикатора
<a href="#">Значение</a>	Текущее значение индикатора прогресса

<u>Максимальное значение</u>	Значение, которое будет соответствовать 100% заполненности шкалы прогресса
<u>Минимальное значение</u>	Значение, которое будет соответствовать 0% заполненности шкалы прогресса
<u>Ориентация</u>	Расположение индикатора прогресса
<u>Z-значение</u>	Z-значение
<u>Непрозрачность</u>	Непрозрачность объекта
<u>Отражение</u>	Отражение



## 1.1.4.2.4.10.32.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.32.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.32.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

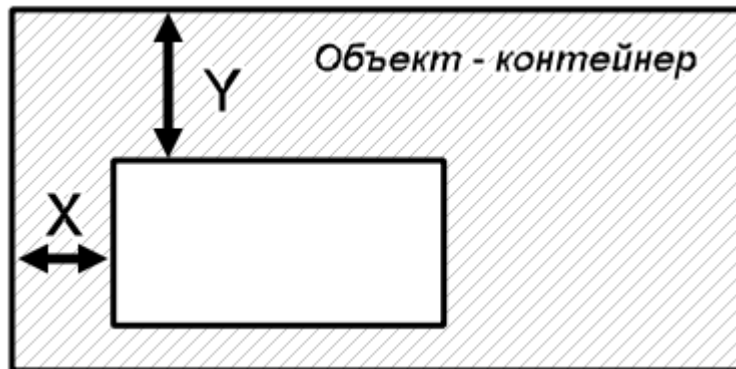
## 1.1.4.2.4.10.32.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.32.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.32.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.32.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.2.4.10.32.3.8. Отражение

Отражение объекта.



uint1 Flip



## 1.1.4.2.4.10.32.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.32.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.32.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.32.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

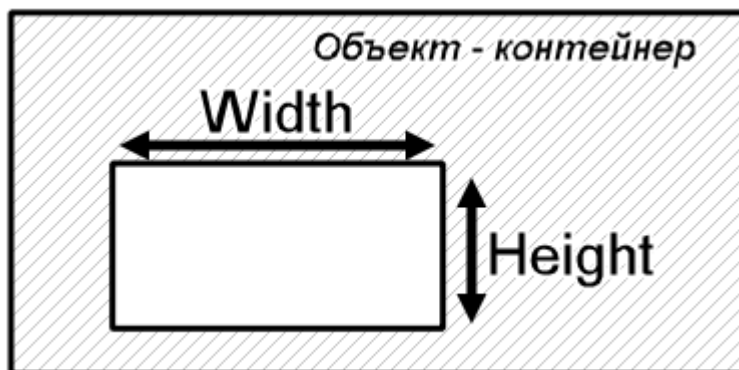
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.32.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

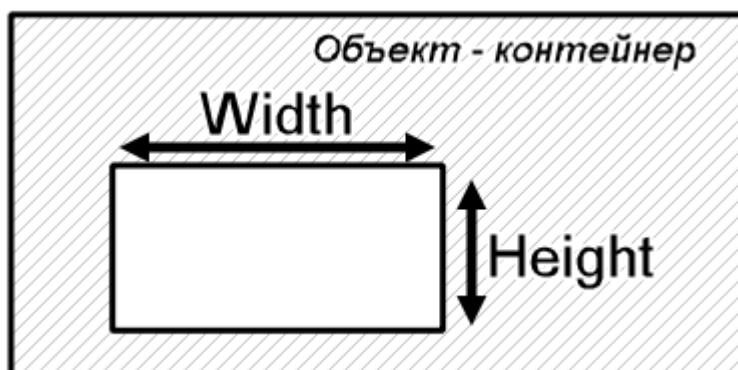


## 1.1.4.2.4.10.32.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.32.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры




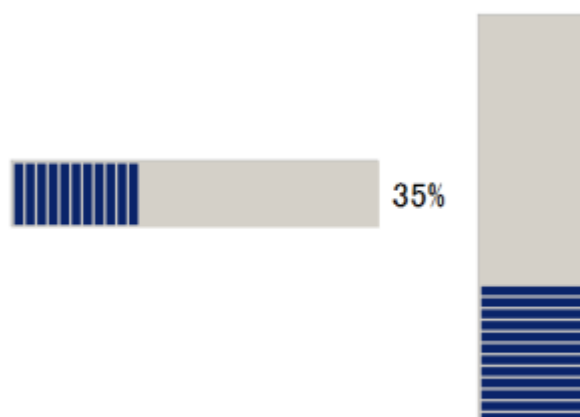
//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.32.3.16. Ориентация

Расположение индикатора прогресса.

 int4 Orientation



### Значение

Значение	Описание
0	Индикатор прогресса ориентирован горизонтально и у него есть цифровое поле
2	Индикатор прогресса ориентирован вертикально, без цифрового поля



## 1.1.4.2.4.10.32.3.17. Минимальное значение

Значение, которое будет соответствовать 0% заполненности шкалы прогресса.



int4 MinLimit

## 1.1.4.2.4.10.32.3.18. Максимальное значение

Значение, которое будет соответствовать 100% заполненности шкалы прогресса.



int4 MaxLimit

## 1.1.4.2.4.10.32.3.19. Значение

Текущее значение индикатора прогресса.



int4 Value

## 1.1.4.2.4.10.32.3.20. Показывать текст

Позволяет отобразить/скрыть текущее значение прогресса справа от индикатора.



bool ShowText

### Значение

Значение	Описание
true	Текущее значение прогресса отображается
false	Текущее значение прогресса не отображается

## 1.1.4.2.4.10.32.3.21. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.32.3.22. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

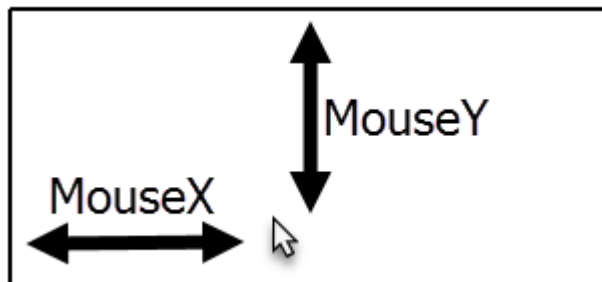
```
Object.FontColor = 4294901760;
```

## 1.1.4.2.4.10.32.3.23. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

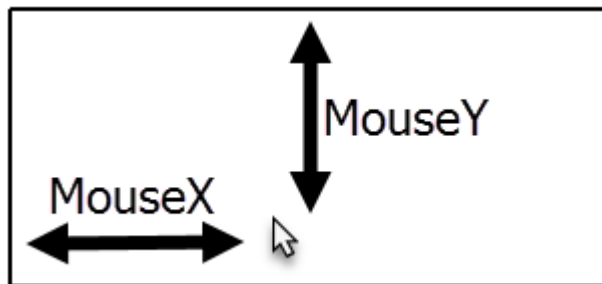
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.32.3.24. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```



## 1.1.4.2.4.10.32.3.25. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.32.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.4.10.33. Ползунок

Компонент Ползунок применяется для выставления некоторых фиксированных значений для параметров.

The image shows two windows from a development environment. The top window, titled "Структура объекта", displays a tree view of the object's structure. It shows a hierarchy starting with "MainForm", followed by "Графические объекты", which contains "Button\_1" and "Slider\_1". The "Slider\_1" component is selected and highlighted. The bottom window, titled "Редактор свойств", shows the properties of the selected "Slider\_1" component. It is a table with three columns: "Свойство" (Property), "Характеристики" (Characteristics), and "Значение" (Value). The properties listed include "Отображаемое имя", "Кардинальное число", "X", "Y", "Z-значение", "Угол поворота", "Масштаб", "Отражение", "Видимость", "Непрозрачность", "Включено", "Всплывающая подсказка", "Ширина", "Высота", "Фокус ввода", "Ориентация", "Минимальное значение", "Максимальное значение", and "Значение".

Свойство	Характеристики	Значение
Отображаемое имя		Slider_1
Кардинальное число		1
>  X	R W	699
>  Y	R W	170
>  Z-значение	R W	0
>  Угол поворота	R W	0
>  Масштаб	R W	1
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Ширина	R W	84
>  Высота	R W	22
>  Фокус ввода	R W ⚡	<не определено>
>  Ориентация	R W	Горизонтальная
>  Минимальное значение	R W	0
>  Максимальное значение	R W	99
>  Значение	R W ⚡ ⚙	0

## 1.1.4.2.4.10.33.1. События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">ValueChanged</a>	Событие при изменении положения ползунка

## 1.1.4.2.4.10.33.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.33.1.2. ValueChanged

Событие при изменении положения ползунка.

### Параметры

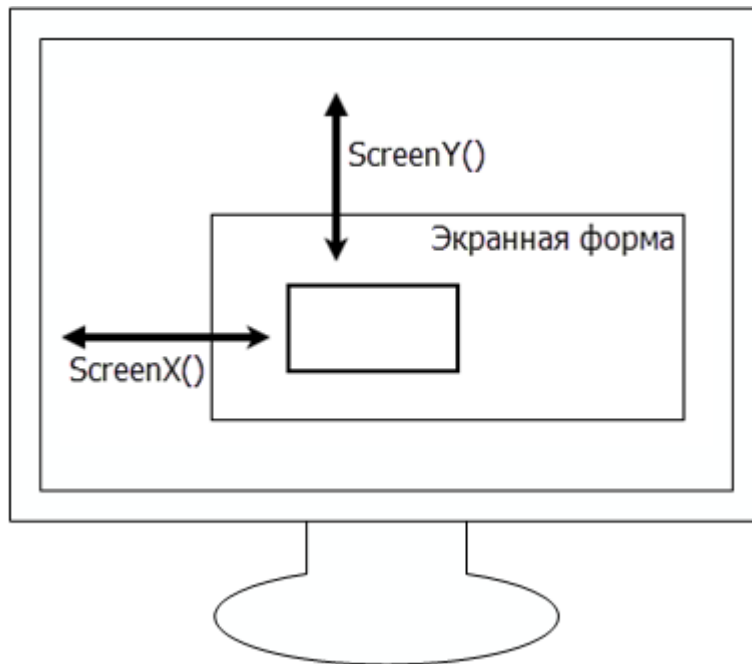
Параметр	Тип	Описание
value	int4	Новое значение положения ползунка

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме

## 1.1.4.2.4.10.33.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

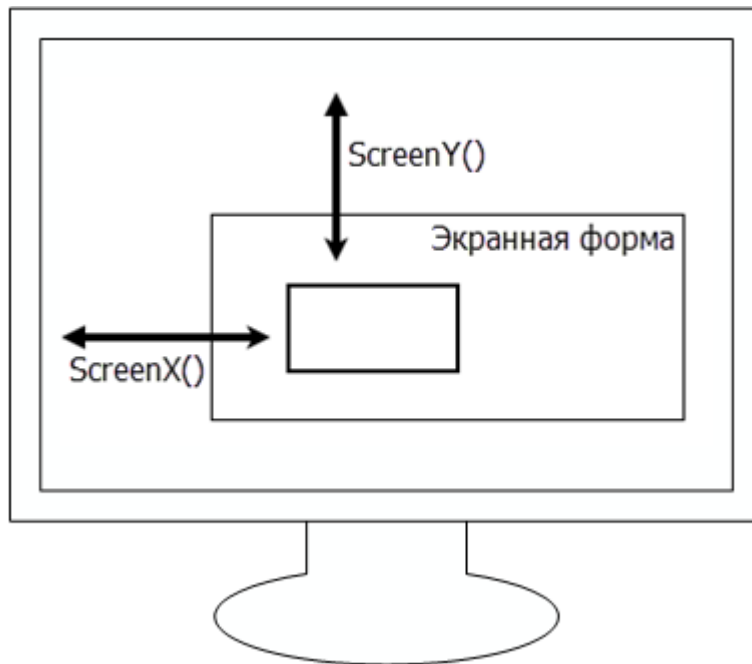
```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.33.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



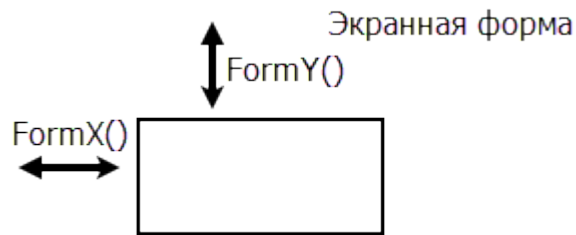
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.33.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



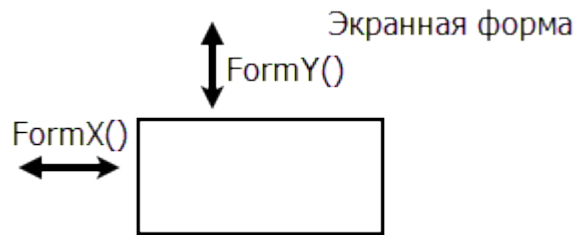
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.33.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.33.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Значение</a>	Текущее значение индикатора прогресса
<a href="#">Максимальное значение</a>	Значение, которое будет соответствовать 100% заполненности шкалы прогресса
<a href="#">Минимальное значение</a>	Значение, которое будет соответствовать 0% заполненности шкалы прогресса
<a href="#">Ориентация</a>	Расположение индикатора прогресса

<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.33.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.33.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

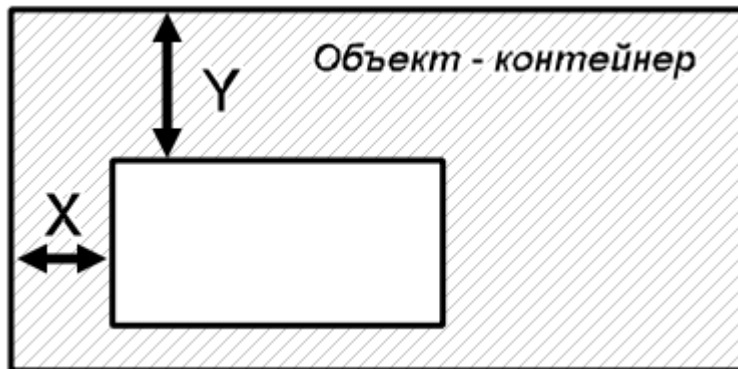
## 1.1.4.2.4.10.33.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```



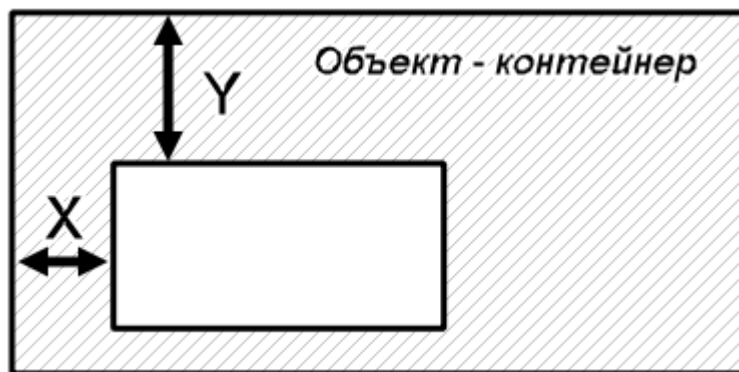
## 1.1.4.2.4.10.33.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.33.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.33.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.33.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.2.4.10.33.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.33.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.33.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.33.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.2.4.10.33.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true


### Примеры

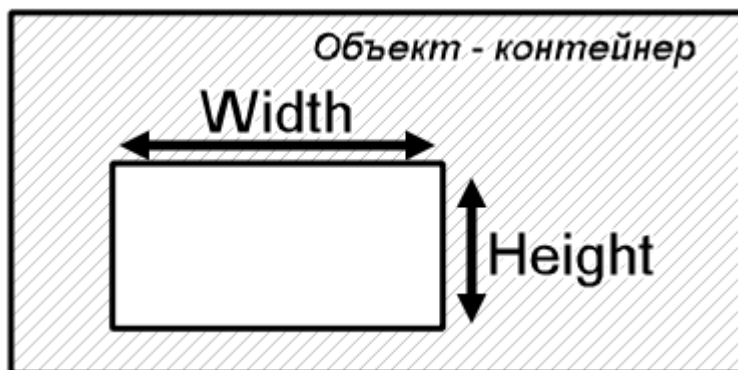


```
//Задать всплывающую подсказку для объекта:  
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.2.4.10.33.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

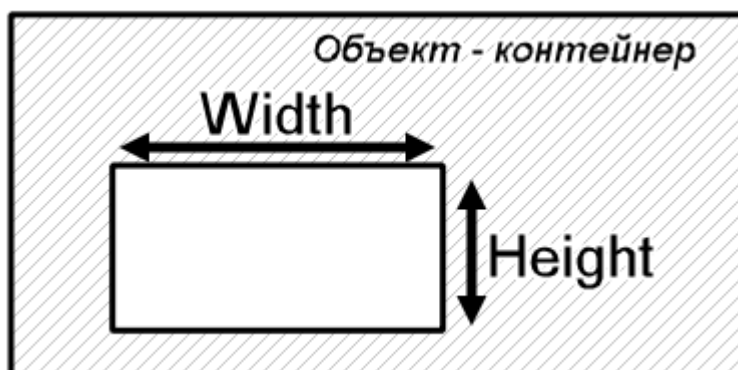


## 1.1.4.2.4.10.33.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.33.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры

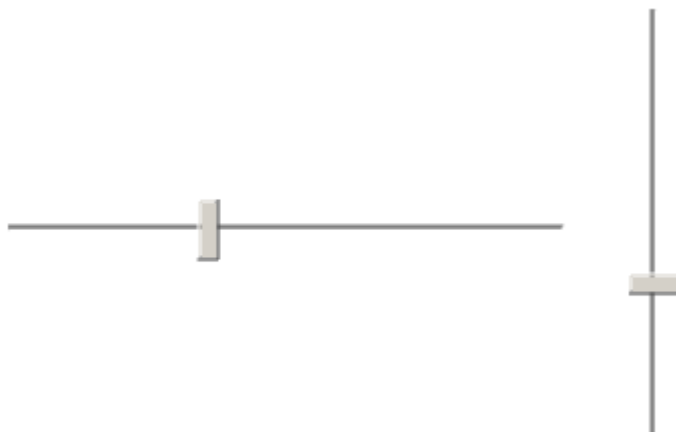
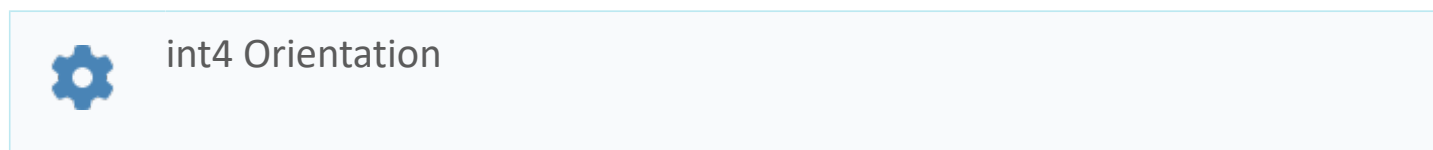


//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.33.3.16. Ориентация

Расположение ползунка.



### Значение

Значение	Описание
1	Ползунок ориентирован горизонтально
2	Ползунок ориентирован вертикально

## 1.1.4.2.4.10.33.3.17. Минимальное значение

Значение, которое будет соответствовать 0% заполненности шкалы прогресса.



int4 MinLimit

## 1.1.4.2.4.10.33.3.18. Максимальное значение

Значение, которое будет соответствовать 100% заполненности шкалы прогресса.



int4 MaxLimit

## 1.1.4.2.4.10.33.3.19. Значение

Текущее значение индикатора прогресса.



int4 Value

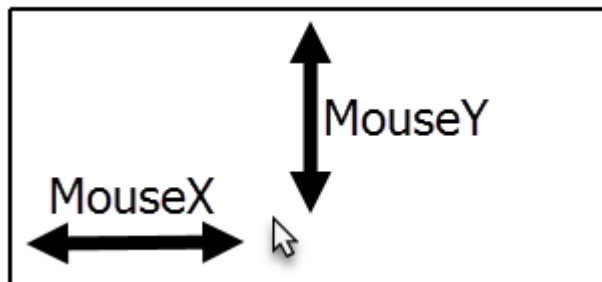


## 1.1.4.2.4.10.33.3.20. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

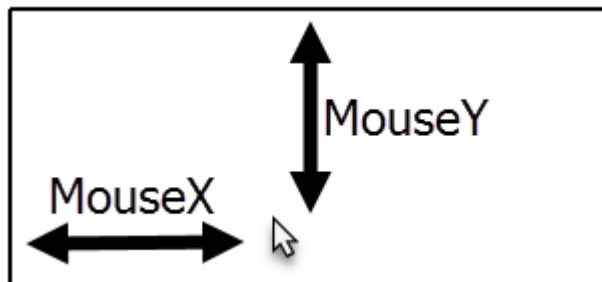
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.33.3.21. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.33.3.22. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.33.3.23. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.34. Дерево

Компонент для создания древовидной структуры.

The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the UI components. The 'MainForm' is expanded to show 'Графические объекты' (Visual Objects), which includes 'Button\_1' and 'Tree\_1'. The 'Tree\_1' component is selected and highlighted. The bottom window, titled 'Редактор свойств' (Properties), shows the properties for the selected 'Tree\_1' component. The properties are organized into columns: 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
Отображаемое имя		Tree_1
Кардинальное число		1
X	R W	771
Y	R W	203
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	256
Высота	R W	192
Фокус ввода	R W ⚡	< не определено >
Шрифт	R W	Segoe UI, 9, -1, 5, 50, 0, 0, 0, 0
Цвет шрифта	R W	4278190080
Цвет фона	R W	4294967295
Видимость рамки	R W	true
Число столбцов	R W	1
Показывать заголовки	R W	true
Текущий элемент	R W ⚡	

# События

Событие	Описание
<a href="#">ItemCollapsed</a>	Сворачивание узлов элемента дерева
<a href="#">ItemExpanded</a>	Разворачивание узлов элемента дерева
<a href="#">ItemDoubleClicked</a>	Двойной щелчок левой кнопкой мыши по элементу дерева
<a href="#">ItemClicked</a>	Щелчок по элементу дерева
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">ItemPressed</a>	Нажатие левой кнопки мыши на элемент дерева
<a href="#">CurrentItemChanged</a>	Сигнал смены текущего элемента
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseDown</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">ItemCheckStateChanged</a>	Смена состояния флажка выбора



## 1.1.4.2.4.10.34.1.1. ItemCollapsed

Сворачивание узлов элемента дерева.

### Параметры

Параметр	Тип	Описание
itemid	string	Идентификатор выбранного элемента дерева



## 1.1.4.2.4.10.34.1.2. ItemExpanded

Разворачивание узлов элемента дерева.

### Параметры

Параметр	Тип	Описание
itemid	string	Идентификатор выбранного элемента дерева

## 1.1.4.2.4.10.34.1.3. ItemDoubleClicked

Двойной щелчок левой кнопкой мыши по элементу дерева.

### Параметры

Параметр	Тип	Описание
itemid	string	Идентификатор выбранного элемента дерева
column	int4	Номер столбца выбранного элемента дерева

## 1.1.4.2.4.10.34.1.4. ItemClicked

Щелчок по элементу дерева.

### Параметры

Параметр	Тип	Описание
itemid	string	Идентификатор выбранного элемента дерева
column	int4	Номер столбца выбранного элемента дерева

## 1.1.4.2.4.10.34.1.5. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.34.1.6. ItemPressed

Нажатие левой кнопки мыши на элемент дерева.

### Параметры

Параметр	Тип	Описание
itemid	string	Идентификатор выбранного элемента дерева
column	int4	Номер столбца выбранного элемента дерева

## 1.1.4.2.4.10.34.1.7. CurrentItemChanged

Сигнал смены текущего элемента.

## 1.1.4.2.4.10.34.1.8. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.10.34.1.9. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.34.1.10. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.34.1.11. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.34.1.12. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.10.34.1.13. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.34.1.14. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";}
```

```
else
```

```
{TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.34.1.15. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.4.10.34.1.16. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.34.1.17. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.2.4.10.34.1.18. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.34.1.19. ItemCheckStateChanged

Смена состояния флажка выбора.

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">SetItemIconPath</a>	Устанавливает иконку для элемента в указанном столбце
<a href="#">SetItemTextColor</a>	Задаёт цвет фона элемента в указанном столбце
<a href="#">SetItemBackgroundColor</a>	Задаёт цвет фона элемента в указанном столбце
<a href="#">SetItemText</a>	Задаёт имя элемента в указанном столбце
<a href="#">GetItemText</a>	Возвращает строковое имя элемента указанного столбца
<a href="#">GetItemData</a>	Возвращает данные (тип variant), хранящиеся в элементе указанного столбца
<a href="#">SetItemData</a>	Добавляет данные элементу в указанном столбце
<a href="#">SetColumnHidden</a>	Устанавливает видимость столбца
<a href="#">SetColumnWidth</a>	Устанавливает ширину столбца
<a href="#">SetColumnName</a>	Устанавливает имя столбца
<a href="#">RemoveChildren</a>	Удаляет дочерние элементы
<a href="#">RemoveItem</a>	Удаляет элемент
<a href="#">MoveItem</a>	Перемещает элемент в указанное место
<a href="#">ChildCount</a>	Возвращает число дочерних элементов
<a href="#">InsertItem</a>	Добавляет новый дочерний элемент в указанное место
<a href="#">AddItem</a>	Добавляет новый элемент в список форм
<a href="#">EnumChildrenJSON</a>	Список дочерних элементов
<a href="#">SetCurrentItem</a>	Выделить элемент

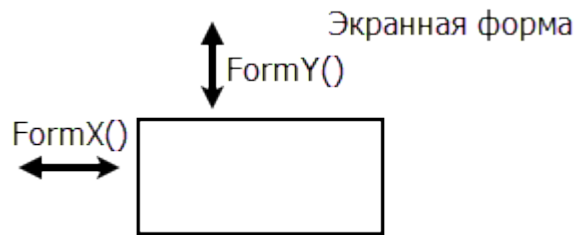
<a href="#">ScrollToItem</a>	Переместиться к элементу
<a href="#">ExpandItem</a>	Развернуть элемент
<a href="#">CollapseItem</a>	Свернуть элемент
<a href="#">GetColumnName</a>	Имя столбца
<a href="#">GetColumnWidth</a>	Ширина столбца
<a href="#">GetColumnHidden</a>	Видимость столбца
<a href="#">IsItemExpanded</a>	Развернут ли элемент
<a href="#">IsItemCollapsed</a>	Свернут ли элемент
<a href="#">ExpandAll</a>	Развернуть все узлы дерева
<a href="#">CollapseAll</a>	Свернуть все узлы дерева
<a href="#">GetItemCheckState</a>	Получить состояние флажка выбора
<a href="#">SetItemCheckState</a>	Установить состояние флажка выбора
<a href="#">IsItemUserCheckable</a>	Возможна ли смена состояния флажка выбора пользователем
<a href="#">SetItemUserCheckable</a>	Установить возможность смены состояния флажка выбора пользователем
<a href="#">IsItemAutoTristate</a>	Автоматическая синхронизация при изменении любого состояния
<a href="#">SetItemAutoTristate</a>	Включение автоматической синхронизации при изменении любого состояния
<a href="#">GetItemId</a>	Получить идентификатор дочернего элемента, расположенного по указанному индексу строки
<a href="#">GetParentId</a>	Получить идентификатор родительского элемента
<a href="#">IsItemIdValid</a>	Является ли указанный идентификатор элемента корректным
<a href="#">SetColumnFilter</a>	Установить фильтр по тексту элемента в столбце
<a href="#">SetItemExpanded</a>	Установить состояние раскрытости элемента.
<a href="#">SetItemExpandPolicy</a>	Установить политику разворачиваемости элемента
<a href="#">GetItemExpandPolicy</a>	Получить политику разворачиваемости элемента

[IsItemExpandable](#)

Возможно ли развернуть элемент

## 1.1.4.2.4.10.34.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



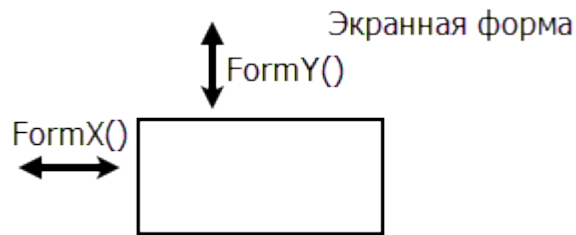
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.34.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



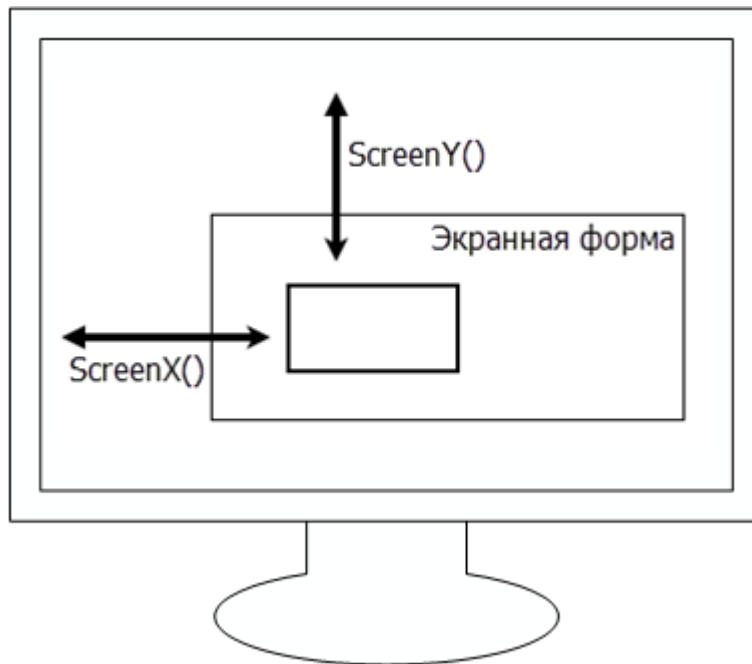
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.34.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

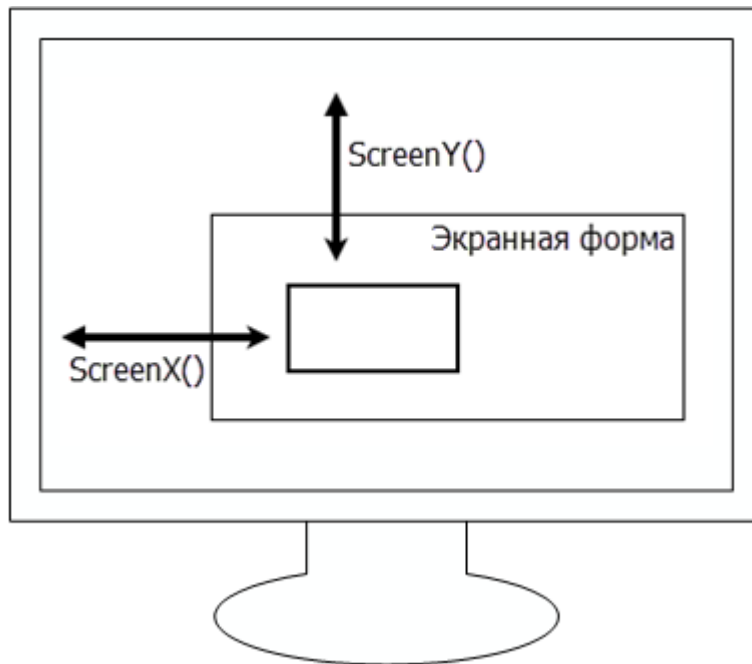
```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.2.4.10.34.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.34.2.5. AddItem

Добавляет новый элемент в список форм.



string AddItem (ItemID, ItemText)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор родительского элемента. Для вставки корневого элемента списка оставьте значение пустым. Элемент будет вставлен в конце списка дочерних/корневых элементов
ItemText	string	Имя элемента

### Примеры



```
//Добавить новый элемент в состав дочерних:  
item_1: var = Tree_1.AddItem(item_0, "Задвижка 1");
```

## 1.1.4.2.4.10.34.2.6. InsertItem

Добавляет новый дочерний элемент в указанное место.



string InsertItem (ItemID, ItemText, n)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор родительского элемента. Для вставки корневого элемента списка оставьте значение пустым. Элемент будет вставлен в конце списка дочерних/корневых элементов
ItemText	string	Имя элемента
n	int4	Порядковый номер дочернего элемента внутри родительского элемента, после которого будет вставлен новый элемент

### Примеры



```
//Добавить новый элемент в состав дочерних:  
Tree_1.InsertItem(item_0, "Задвижка 1", 3);
```

## 1.1.4.2.4.10.34.2.7. ChildCount

Возвращает число дочерних элементов.



string ChildCount (ItemID)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента

### Примеры



```
//Вывести число дочерних элементов в текстовом поле:  
ch_count = Tree_1.ChildCount(item_1);  
TextEdit_2.Text = String.ToString(ch_count);
```

## 1.1.4.2.4.10.34.2.8. EnumChildrenJSON

Список идентификаторов дочерних элементов.



string EnumChildrenJSON(string \_1)

### Параметры

Параметр	Тип	Описание
_1	string	

## 1.1.4.2.4.10.34.2.9. MoveItem

Перемещает элемент в указанное место.



```
void MoveItem (ItemID, ItemID, n)
```

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор родительского элемента
ItemID	string	Идентификатор перемещаемого элемента
n	int4	Порядковый номер дочернего элемента внутри родительского элемента, после которого будет вставлен перемещаемый элемент. Если указанное число превышает количество дочерних элементов, перемещаемый элемент вставится в конец списка, если меньше 0 - в начало списка.

### Примеры



```
//Переместить элемент на второе место в списке:
```

```
Tree_1.MoveItem(item1, item5, 1);
```

## 1.1.4.2.4.10.34.2.10. RemoveItem

Удаляет элемент.



void RemoveItem (ItemID)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента

### Примеры



```
//Удалить выделенный элемент:  
Tree_1.RemoveItem(SelectedItem);
```

## 1.1.4.2.4.10.34.2.11. RemoveChildren

Удаляет дочерние элементы.



void RemoveChildren (ItemID)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор родительского элемента

### Примеры



```
//Удалить дочерние элементы выделенного элемента:  
Tree_1.RemoveChildren(SelectedItem);
```



## 1.1.4.2.4.10.34.2.12. SetCurrentItem

Выделить элемент.



void SetCurrentItem(string\_1)

### Параметры

Параметр	Тип	Описание
string_1	string	

## 1.1.4.2.4.10.34.2.13. ScrollToItem

Переместиться к элементу.



```
void ScrollToItem(string_1)
```

### Параметры

Параметр	Тип	Описание
string_1	string	

## 1.1.4.2.4.10.34.2.14. ExpandItem

Развернуть элемент.



void ExpandItem(string\_1)

### Параметры

Параметр	Тип	Описание
string_1	string	

## 1.1.4.2.4.10.34.2.15. Collapseltem

Свернуть элемент.



void Collapseltem(string\_1)

### Параметры

Параметр	Тип	Описание
string_1	string	

## 1.1.4.2.4.10.34.2.16. SetColumnName

Устанавливает имя столбца.



```
void SetColumnName (ColumnName, ColumnNumber)
```

### Параметры

Параметр	Тип	Описание
ColumnName	string	Имя столбца
ColumnNumber	int4	Номер столбца

### Примеры



```
//Установить имя столбца:  
Tree_1.SetColumnName("ID", 1);
```

## 1.1.4.2.4.10.34.2.17. GetColumnName

Имя столбца.



string GetColumnName(int4\_1)

### Параметры

Параметр	Тип	Описание
int4_1	int4	

## 1.1.4.2.4.10.34.2.18. SetColumnWidth

Устанавливает ширину столбца.



```
void SetColumnWidth (ColumnNumber, w)
```

### Параметры

Параметр	Тип	Описание
ColumnNumber	int4	Номер столбца
w	int4	Ширина (в пикселях)

### Примеры



```
//Установить ширину столбца:  
Tree_1.SetColumnWidth(1, 300);
```

## 1.1.4.2.4.10.34.2.19. GetColumnWidth

Ширина столбца.



int4 GetColumnWidth(int4\_1)

### Параметры

Параметр	Тип	Описание
int4_1	int4	



## 1.1.4.2.4.10.34.2.20. SetColumnHidden

Устанавливает видимость столбца.



```
void SetColumnHidden (ColumnNumber, f)
```

### Параметры

Параметр	Тип	Описание
ColumnNumber	int4	Номер столбца
f	bool	Флаг включения/отключения видимости столбца

### Примеры



```
//Скрыть столбец:  
Tree_1.SetColumnHidden(1, true);
```

## 1.1.4.2.4.10.34.2.21. GetColumnHidden

Видимость столбца.



bool GetColumnHidden(int4\_1)

### Параметры

Параметр	Тип	Описание
int4_1	int4	

## 1.1.4.2.4.10.34.2.22. SetItemData

Добавляет данные элементу в указанном столбце.



```
void SetItemData (ItemID, Data, ColumnNumber)
```

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
Data	variant	Добавляемое элементу значение
ColumnNumber	int4	Номер столбца

### Примеры



```
//Добавить строковый идентификатор формы в элемент:  
Tree_1.SetItemData(item_2, "f6d8d7a5-a2fc-438b-8e15-4858cada6df1",  
0);
```

## 1.1.4.2.4.10.34.2.23. GetItemData

Возвращает данные (тип variant), хранящиеся в элементе указанного столбца.



variant GetItemData (ItemID, ColumnNumber)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
ColumnNumber	int4	Номер столбца

### Примеры



```
//Вывести данные элемента в текстовом поле:  
itemdata = Tree_1.GetItemData(item_2, 0);  
TextEdit_2.Text = Variant.ToString(itemdata);
```

## 1.1.4.2.4.10.34.2.24. GetItemText

Возвращает строковое имя элемента указанного столбца.



string GetItemText (ItemID, ColumnNumber)

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
ColumnNumber	int4	Номер столбца

### Примеры



```
//Вывести имя элемента в текстовом поле:  
itemtext = Tree_1.GetItemText(item_1, 0);  
TextEdit_2.Text = itemtext;
```

## 1.1.4.2.4.10.34.2.25. SetItemText

Задаёт имя элемента в указанном столбце.



```
void SetItemText (ItemID, ItemText, ColumnNumber)
```

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
ItemText	string	Имя элемента
ColumnNumber	int4	Номер столбца

### Примеры



```
//Задать имя элемента:  
Name: string = "Третья форма";  
Tree_1.SetItemText(item_16, Name, 1);
```

## 1.1.4.2.4.10.34.2.26. SetItemBackgroundColor

Задает цвет фона элемента в указанном столбце.



```
void SetItemBackgroundColor (ItemID, color, ColumnNumber)
```

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
color	uint4	Код цвета (в шестнадцатеричной или десятичной системе счисления)
ColumnNumber	int4	Номер столбца

### Примеры



```
//Задать цвет фона элемента:  
Tree_1.SetItemBackgroundColor (item_2, 0x9CA2A3, 0);
```

## 1.1.4.2.4.10.34.2.27. SetItemTextColor

Задаёт цвет фона элемента в указанном столбце.



```
void SetItemTextColor (ItemID, color, ColumnNumber)
```

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
color	uint4	Код цвета (в шестнадцатеричной или десятичной системе счисления)
ColumnNumber	int4	Номер столбца

### Примеры



```
//Задать цвет текста элемента:  
Tree_1.SetItemTextColor (item_2, 0x9CA2A3, 0);
```



## 1.1.4.2.4.10.34.2.28. SetItemIconPath

Устанавливает иконку для элемента в указанном столбце.



```
void SetItemIconPath (ItemID, pathToIcon, ColumnNumber)
```

### Параметры

Параметр	Тип	Описание
ItemID	string	Идентификатор элемента
pathToIcon	string	Путь к файлу с иконкой относительно папки resources (вложенная папка проекта)
ColumnNumber	int4	Номер столбца

### Примеры



//Задать иконку элемента:

```
Tree_1.SetItemIconPath(item_2, "icons/alarms/n_3.png", 0);
```

## 1.1.4.2.4.10.34.2.29. SetColumnFilter

Устанавливает фильтр по тексту элемента в столбце.



```
void SetColumnFilter (int4 Column, string Filter)
```

### Параметры

Параметр	Тип	Описание
Column	int4	Номер столбца
Filter	string	Условие фильтрации

## 1.1.4.2.4.10.34.2.30. SetItemExpanded

Устанавливает состояние раскрытости элемента.



```
void SetItemExpanded (string ItemId, bool Value)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	Идентификатор элемента
Value	bool	Состояние

## 1.1.4.2.4.10.34.2.31. IsItemExpanded

Развернут ли элемент.



bool IsItemExpanded(string ItemId)

### Параметры

Параметр	Тип	Описание
ItemId	string	

## 1.1.4.2.4.10.34.2.32. IsItemCollapsed

Свернут ли элемент.



bool IsItemCollapsed(string ItemId)

### Параметры

Параметр	Тип	Описание
ItemId	string	

## 1.1.4.2.4.10.34.2.33. ExpandAll

Развернуть все узлы дерева.



```
void ExpandAll()
```

## 1.1.4.2.4.10.34.2.34. CollapseAll

Свернуть все узлы дерева.



```
void CollapseAll()
```

## 1.1.4.2.4.10.34.2.35. GetItemCheckState

Получить состояние флажка выбора.



```
uint1 GetItemCheckState(string ItemId, int4 Column)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	
Column	int4	



## 1.1.4.2.4.10.34.2.36. SetItemCheckState

Установить состояние флажка выбора.



```
void SetItemCheckState(string ItemId, uint1 CheckState, int4 Column)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	
CheckState	uint1	
Column	int4	

## 1.1.4.2.4.10.34.2.37. IsItemUserCheckable

Возможна ли смена состояния флажка выбора пользователем.



```
bool IsItemUserCheckable(string ItemId, int4 Column)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	
Column	int4	

## 1.1.4.2.4.10.34.2.38. SetItemUserCheckable

Установить возможность смены состояния флажка выбора пользователем.



```
void SetItemUserCheckable(string ItemId, bool IsUserCheckable, int4  
Column)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	
IsUserCheckable	bool	
Column	int4	

## 1.1.4.2.4.10.34.2.39. IsItemAutoTristate

В режиме автоматического вывода состояний флажков, состояния флажка элемента и флажков его непосредственных дочерних элементов будут автоматически синхронизироваться при изменении состояния любого из флажков.



```
bool IsItemAutoTristate(string ItemId, int4 Column)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	
Column	int4	

## 1.1.4.2.4.10.34.2.40. SetItemAutoTristate

Позволяет включить или выключить автоматическую синхронизацию состояний флажка элемента и флажков его непосредственных дочерних элементов при изменении состояния любого из флажков.



```
void SetItemAutoTristate(string ItemId, bool IsAutoTrictate, int4 Column)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	
IsAutoTrictate	bool	
Column	int4	

## 1.1.4.2.4.10.34.2.41. SetItemExpandPolicy

Устанавливает политику разворачиваемости элемента.



```
void SetItemExpandPolicy (string ItemId, uint1 ExpandPolicy)
```

### Параметры

Параметр	Тип	Описание
ItemId	string	Идентификатор элемента
ExpandPolicy	uint1	

## 1.1.4.2.4.10.34.2.42. GetItemExpandPolicy

Получает политику разворачиваемости элемента.



uint1 GetItemExpandPolicy (string ItemId)

### Параметры

Параметр	Тип	Описание
ItemId	string	Идентификатор элемента

## 1.1.4.2.4.10.34.2.43. IsItemExpandable

Возможно ли развернуть элемент.



bool GetItemExpandPolicy (string ItemId)

### Параметры

Параметр	Тип	Описание
ItemId	string	Идентификатор элемента



## 1.1.4.2.4.10.34.2.44. GetItemId

Получить идентификатор дочернего элемента, расположенного по указанному индексу строки.



```
string GetItemId(string ParentId, int4 Index)
```

### Параметры

Параметр	Тип	Описание
ParentId	string	
Index	int4	

## 1.1.4.2.4.10.34.2.45. GetParentId

Получить идентификатор родительского элемента.



string GetParentId(string ItemId)

### Параметры

Параметр	Тип	Описание
ItemId	string	

## 1.1.4.2.4.10.34.2.46. IsItemIdValid

Является ли указанный идентификатор элемента корректным.



bool IsItemIdValid(string ItemId)

### Параметры

Параметр	Тип	Описание
int4_1	int4	

## 1.1.4.2.4.10.34.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Число столбцов</a>	Отображает/скрывает полосы прокрутки для формы, открытой во фрейме
<a href="#">Видимость рамки</a>	Видимость рамки, обрамляющей список форм
<a href="#">Цвет фона</a>	Указывается через системное окно Выбор цвета
<a href="#">Цвет шрифта</a>	Указывается через системное окно Выбор цвета
<a href="#">Шрифт</a>	Стиль шрифта

<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Показывать заголовок</a>	Показывать ли заголовок
<a href="#">Текущий элемент</a>	Индекс текущего элемента
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.34.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.34.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

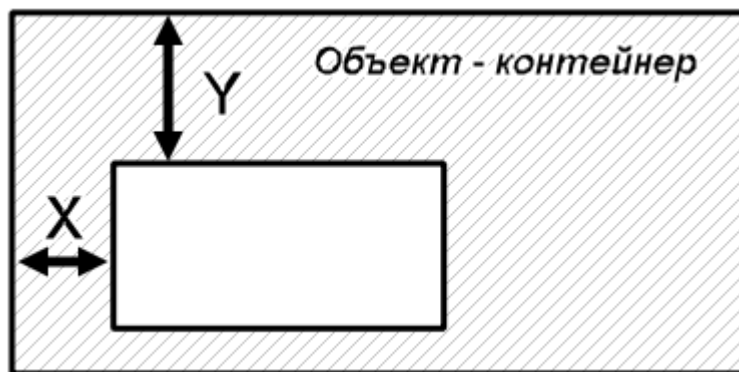
## 1.1.4.2.4.10.34.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```



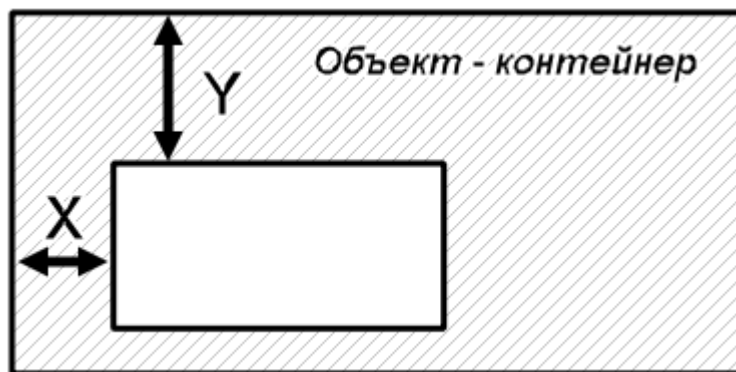
## 1.1.4.2.4.10.34.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.34.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.34.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.34.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.2.4.10.34.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.34.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.34.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.2.4.10.34.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.2.4.10.34.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

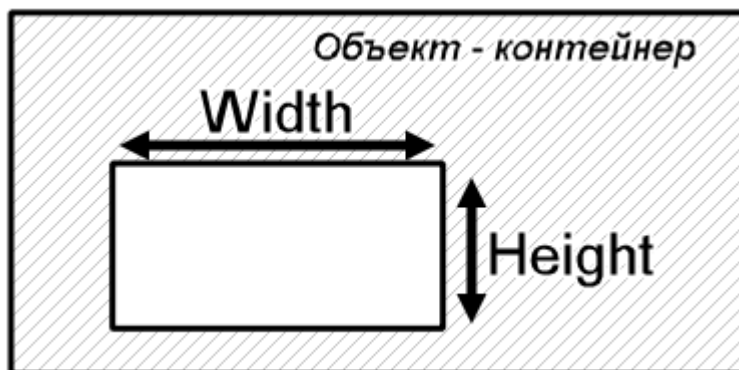
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.34.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

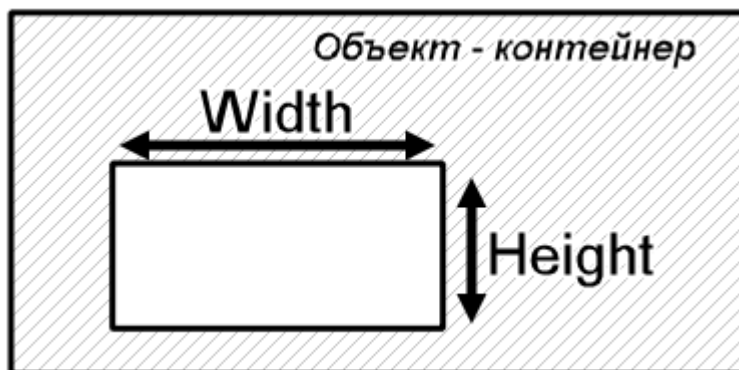


## 1.1.4.2.4.10.34.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.34.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.34.3.16. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку нужного формата.



string Font

## 1.1.4.2.4.10.34.3.17. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

## 1.1.4.2.4.10.34.3.18. Цвет фона

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BackgroundColor

### Примеры



```
//Задать серый цвет фона в шестнадцатеричной системе счисления:  
Object.BackgroundColor = 0x9CA2A3;
```



```
//Задать серый цвет фона в десятичной системе счисления:  
Object.BackgroundColor = 10265251;
```

## 1.1.4.2.4.10.34.3.19. Видимость рамки

Видимость рамки, обрамляющей список форм.



bool FrameVisible

### Значение

Значение	Описание
true	Рамка видна
false	Рамка скрыта



## 1.1.4.2.4.10.34.3.20. Число столбцов

Отображает/скрывает полосы прокрутки для формы, открытой во фрейме.



uint4 ColumnCount

## 1.1.4.2.4.10.34.3.21. Показывать заголовок

Показывать ли заголовок.



bool HeaderVisible

## 1.1.4.2.4.10.34.3.22. Текущий элемент

Индекс текущего элемента.



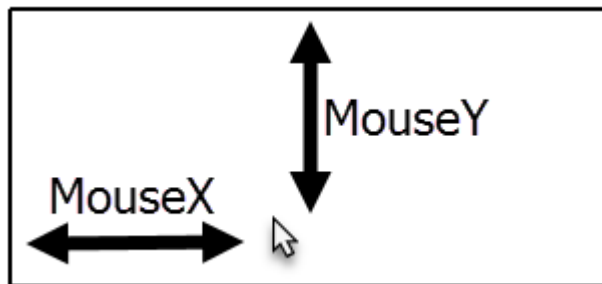
string CurrentItem

## 1.1.4.2.4.10.34.3.23. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

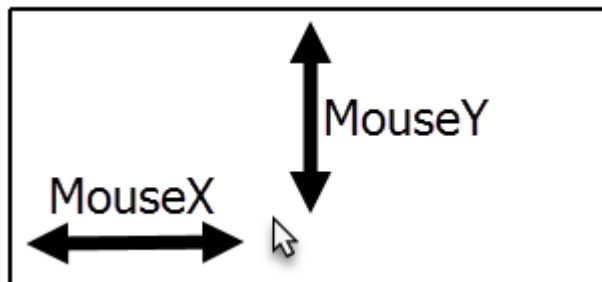
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.34.3.24. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.34.3.25. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.34.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

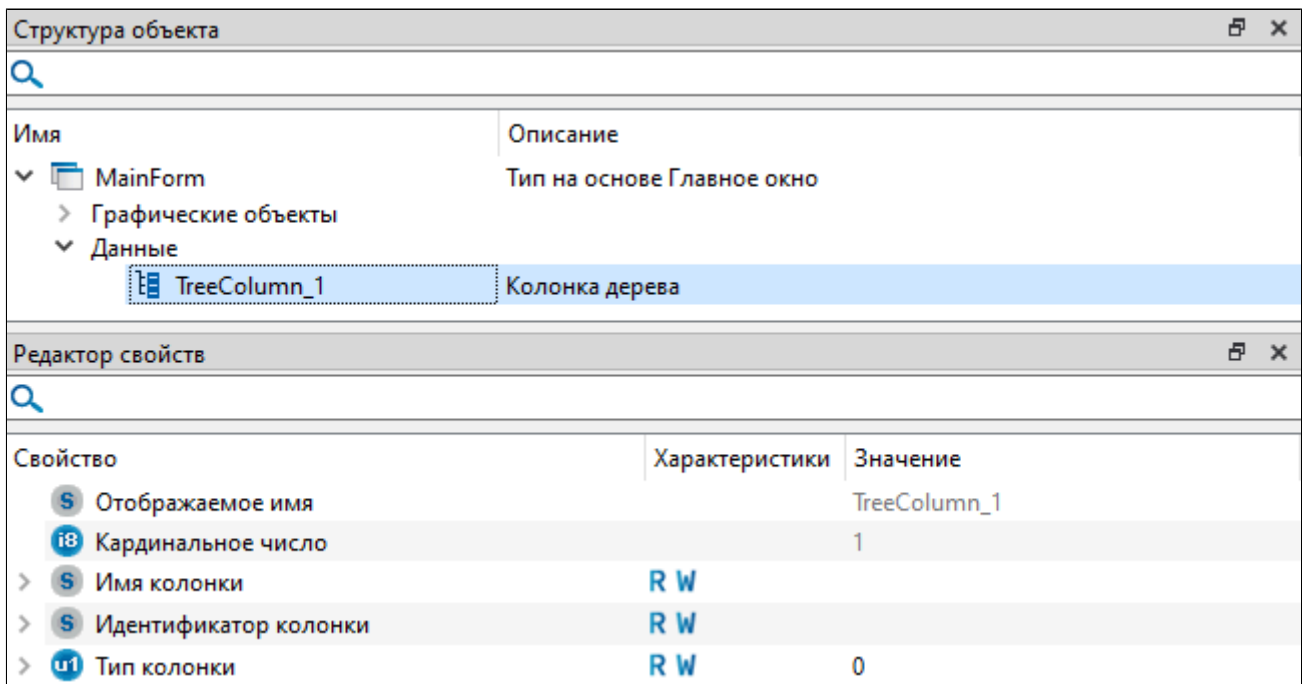


//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:






```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.35. Колонка дерева

Компонент редактирования колонок дерева.



The screenshot displays two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), shows a tree view of the project's components. The 'MainForm' folder is expanded, showing 'Графические объекты' (Visual Objects) and 'Данные' (Data). Under 'Данные', the 'TreeColumn\_1' control is selected and highlighted in blue. The bottom window, titled 'Редактор свойств' (Properties Editor), shows the properties of the selected 'TreeColumn\_1' control. The properties are listed in a table with columns for 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value).

Свойство	Характеристики	Значение
 Отображаемое имя		TreeColumn_1
 Кардинальное число		1
>  Имя колонки	R W	
>  Идентификатор колонки	R W	
>  Тип колонки	R W	0



## 1.1.4.2.4.10.35.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Имя колонки</a>	Отображаемое имя колонки
<a href="#">Идентификатор колонки</a>	Идентификатор колонки
<a href="#">Тип колонки</a>	Тип колонки

## 1.1.4.2.4.10.35.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.35.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.35.1.3. Имя колонки

Отображаемое имя колонки.



string Name

## 1.1.4.2.4.10.35.1.4. Идентификатор колонки

Идентификатор колонки.



string Id

## 1.1.4.2.4.10.35.1.5. Тип колонки

Тип колонки.



uint1 Type

## 1.1.4.2.4.10.35.1.6. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.35.1.7. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.2.4.10.36. Источник данных дерева

Ссылка на источник данных для компонента Дерево.

# События

Событие	Описание
<a href="#">OnNodeReceived</a>	Сигнал о получении узлов

## 1.1.4.2.4.10.36.1.1. OnNodeReceived

Данный сигнал уведомляет о получении информации об узлах или о невозможности ее получения, или об отмене операции.

## Функции

Компонент	Описание
<a href="#">GetFullName</a>	Получить полное имя элемента
<a href="#">Cancel</a>	Отменить загрузку данных

## 1.1.4.2.4.10.36.2.1. GetFullName

Получить полное имя элемента.



string GetFullName (string\_1)

### Параметры

Параметр	Тип	Описание
string_1	string	

## 1.1.4.2.4.10.36.2.2. Cancel

Отменить загрузку данных.



string Cancel()

## 1.1.4.2.4.10.36.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.2.4.10.36.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.2.4.10.36.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.2.4.10.36.3.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.36.3.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.2.4.10.37. Фрейм

Область, в которой можно динамически открывать формы. Чтобы открыть форму во фрейме следует в обработчике событий выбрать команду Открыть во фрейме, указать фрейм, в котором откроется форма, и форму.

# События

Событие	Описание
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseDown</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClick</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана

## 1.1.4.2.4.10.37.1.1. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.4.10.37.1.2. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.4.10.37.1.3. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.4. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.6. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.7. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.8. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.9. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.4.10.37.1.10. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.2.4.10.37.1.11. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.2.4.10.37.1.12. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

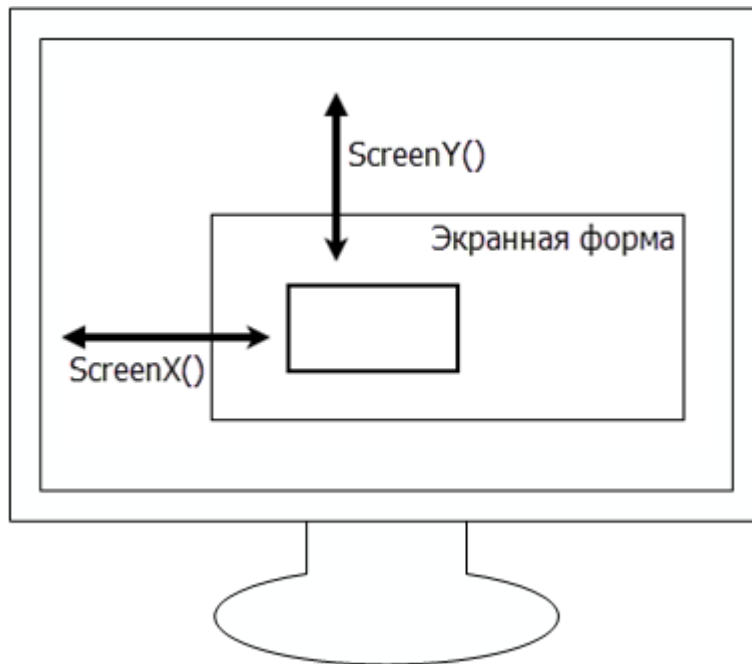
```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

# Функции

Компонент	Описание
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScaleContentTo</a>	Отмасштабировать к размеру
<a href="#">SetContentScale</a>	Установить масштаб содержимого
<a href="#">MoveViewport</a>	Переместить окно просмотра

## 1.1.4.2.4.10.37.2.1. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



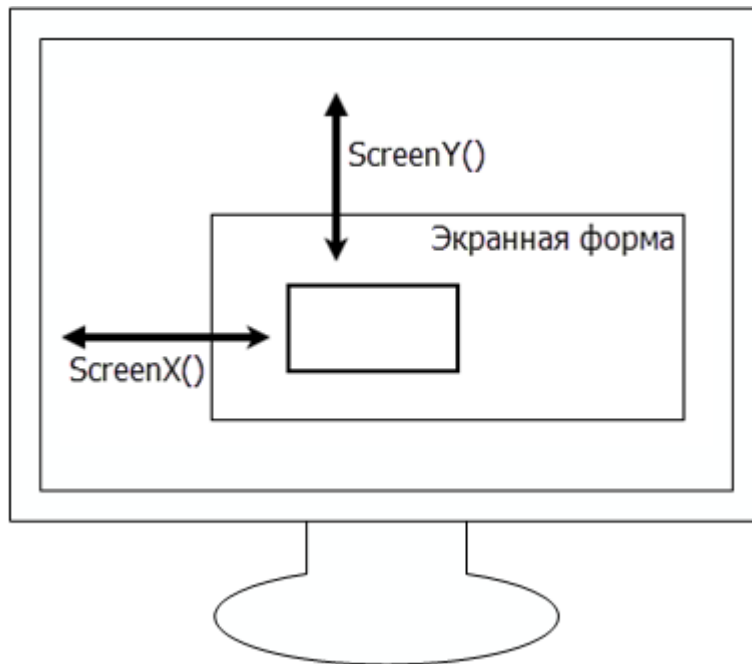
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.37.2.2. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



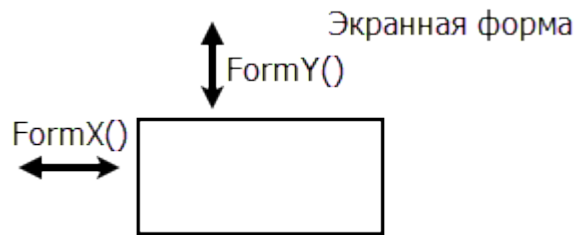
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.37.2.3. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



FormX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



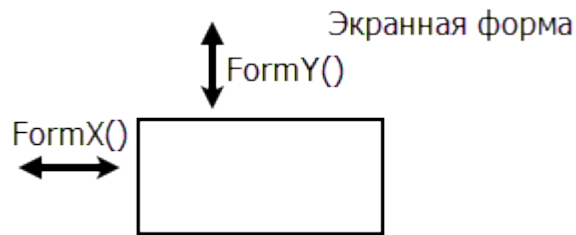
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.37.2.4. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



FormY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.2.4.10.37.2.5. ScaleContentTo

Отмасштабировать к размеру.



```
void ScaleContentTo(float8_1, float8_2)
```

### Параметры

Параметр	Тип	Описание
float8_1	float	
float8_2	float	



## 1.1.4.2.4.10.37.2.6. SetContentScale

Установить масштаб содержимого.



```
void SetContentScale(float8_1)
```

### Параметры

Параметр	Тип	Описание
float8_1	float	

## 1.1.4.2.4.10.37.2.7. MoveViewport

Переместить окно просмотра.



```
void MoveViewport(float8_1, float8_2)
```

### Параметры

Параметр	Тип	Описание
float8_1	float	
float8_2	float	

## 1.1.4.2.4.10.37.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Включено</a>	Активность объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта

<a href="#">Перемещение мышью</a>	Позволяет/запрещает перетаскивать форму, открытую во фрейме, мышью
<a href="#">Отображать полосы прокрутки</a>	Отображает/скрывает полосы прокрутки для формы, открытой во фрейме
<a href="#">Ручное управление масштабom</a>	Разрешает/запрещает использование функций ScaleContentTo, SetContentScale и MoveViewport фрейма
<a href="#">Z-значение</a>	Z-значение
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">Отражение</a>	Отражение

## 1.1.4.2.4.10.37.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.2.4.10.37.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.2.4.10.37.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.2.4.10.37.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.



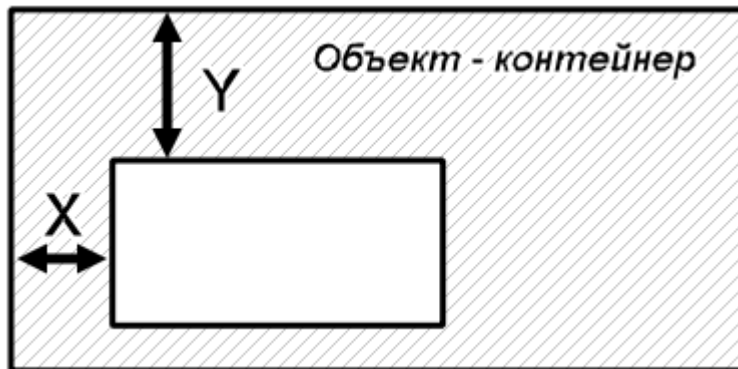
## 1.1.4.2.4.10.37.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



### Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

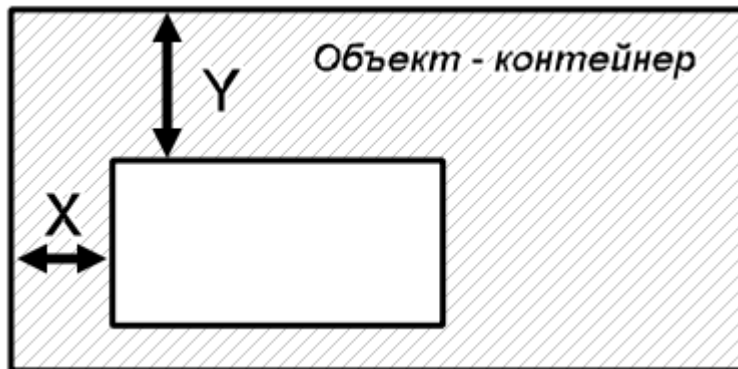
## 1.1.4.2.4.10.37.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



### Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.2.4.10.37.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.4.10.37.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.2.4.10.37.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.4.10.37.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.4.10.37.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.2.4.10.37.3.12. Непрозрачность

Непрозрачность.



float8 Opacity



# 1.1.4.2.4.10.37.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.4.10.37.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

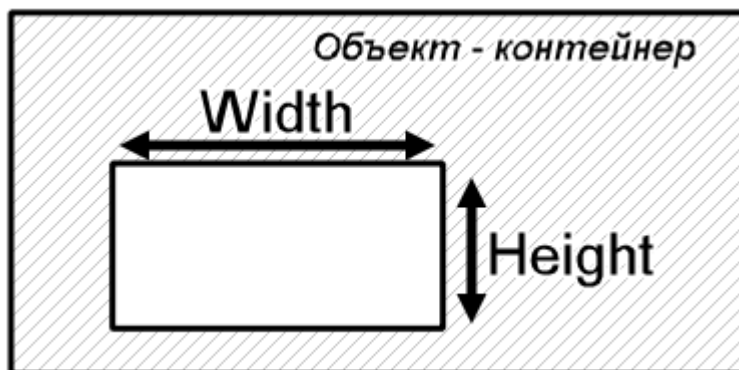
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.4.10.37.3.15. Ширина

Ширина объекта в пикселях.



float8 Width

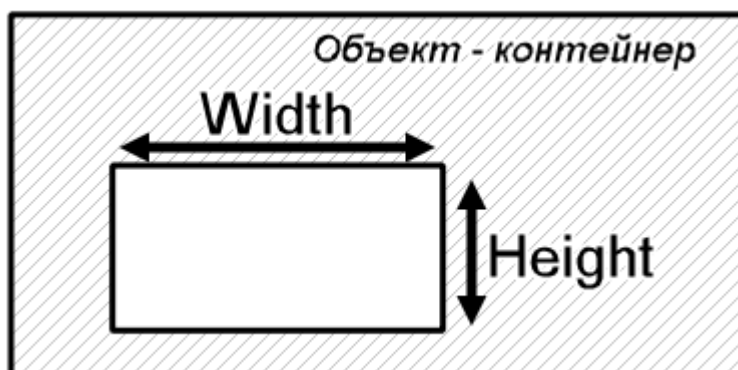


## 1.1.4.2.4.10.37.3.16. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.2.4.10.37.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.4.10.37.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.2.4.10.37.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.2.4.10.37.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```



## 1.1.4.2.4.10.37.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.BrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.BrushColor = 4294901760;
```

## 1.1.4.2.4.10.37.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.4.10.37.3.23. Ручное управление масштабом

Разрешает/запрещает использование функций ScaleContentTo, SetContentScale и MoveViewport фрейма.



bool OverrideScaling

### Значение

Значение	Описание
true	Разрешено ручное управление масштабом
false	Запрещено ручное управление масштабом

### Примеры



```
//При установке флажка включить режим ручного  
масштабирования:  
if (Checkbox_1.State == true)  
{Frame_1.OverrideScaling = true;}
```

## 1.1.4.2.4.10.37.3.24.      Отображать      полосы прокрутки

Отображает/скрывает полосы прокрутки для формы, открытой во фрейме.



bool ShowScrollBars

### Значение

Значение	Описание
true	Полосы прокрутки отображаются
false	Полосы прокрутки скрыты

### Примеры



```
//Отображать/скрывать полосы прокрутки при нажатии пробела:  
if (event.KeyCode == 84)  
Frame_1.ShowScrollBars = !Frame_1.ShowScrollBars;
```

## 1.1.4.2.4.10.37.3.25. Перемещение мышью

Позволяет/запрещает перетаскивать форму, открытую во фрейме, мышью.



bool MoveByMouse

### Значение

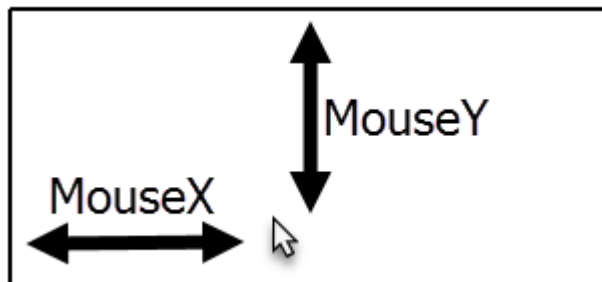
Значение	Описание
true	Перемещение формы мышью разрешено
false	Перемещение формы мышью запрещено

## 1.1.4.2.4.10.37.3.26. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

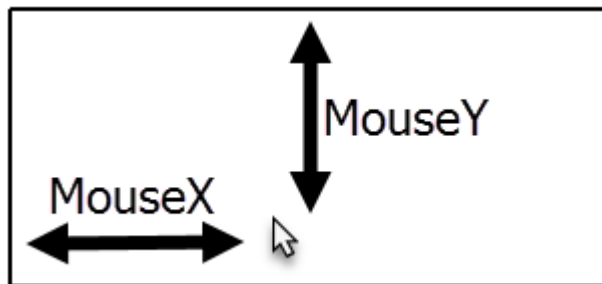
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.4.10.37.3.27. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

```
TextEdit.Text = String.ToString(Object.MouseY);
```



## 1.1.4.2.4.10.37.3.28. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.2.4.10.37.3.29. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.2.5. Горячие клавиши

Команда	Сочетание клавиш
Управление проектом	
Создать проект	Ctrl+N
Открыть проект	Ctrl+O
Сохранить изменения проекта на активной вкладке	Ctrl+S
Сохранить все изменения проекта	Ctrl+Shift+S
Общие команды	
Отменить последнее действие	Ctrl+Z
Повторить последнее действие	Ctrl+Y
Вырезать	Ctrl+X
Копировать	Ctrl+C
Вставить	Ctrl+V
Вставить строку со сдвигом в редакторе формул по условию	Ctrl+Shift++
Вставка символа табуляции (сдвиг выделенного текста вправо)	Tab
Удаление символа табуляции (сдвиг выделенного текста влево)	Shift+Tab
Найти	Ctrl+F
Выделить всё	Ctrl+A
Закрыть текущую вкладку	Ctrl+W
Компиляция проекта и запуск в рантайме	
Запустить компиляцию проекта	F5

Запустить компиляцию активной вкладки	Ctrl+F5
Отменить компиляцию	Ctrl+Break
Запустить активную вкладку проекта в рантайме	Ctrl+F9
Запустить главную форму проекта в рантайме	F9
Порядок объектов на рабочей области	
Переместить объект на передний план	Ctrl+L
Переместить объект на задний план	Ctrl+K
Типы	
Перейти к типу	F12
Переименовать объект или тип	F2
Прочие	
Вызвать справку	F1

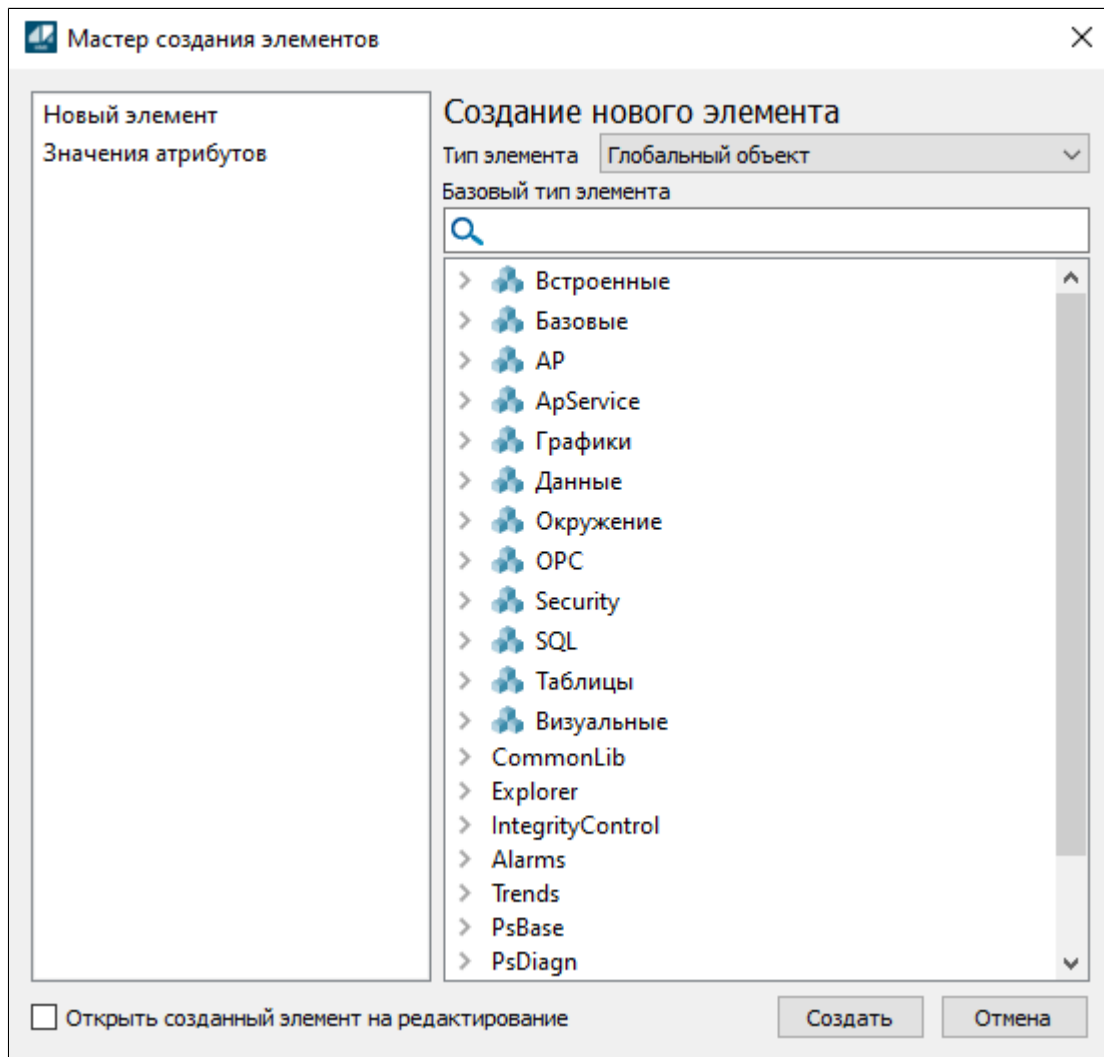
## 1.1.4.2.6. Разработка проекта

Astra.HMI позволяет разрабатывать многооконные проекты. Чтобы начать разрабатывать проект в Astra.HMI, добавьте в проект экранную форму.

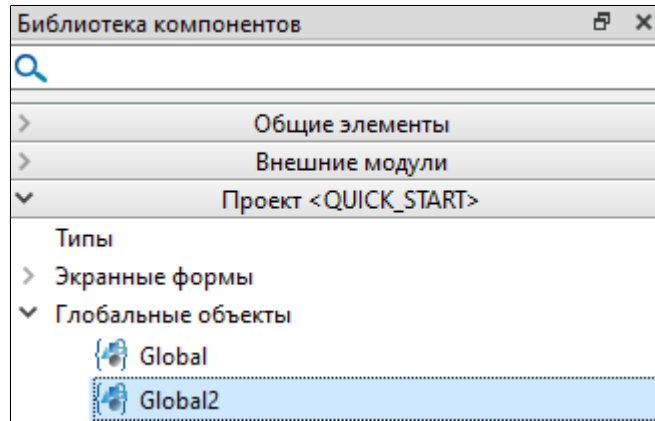
## 1.1.4.2.6.1. Глобальные объекты

### Создание глобального объекта

Чтобы создать новый глобальный объект, выберите команду контекстного меню Создать и воспользуйтесь мастером создания элементов.



Создастся новый глобальный объект со стандартным названием. Для редактирования дважды щелкните по нему.

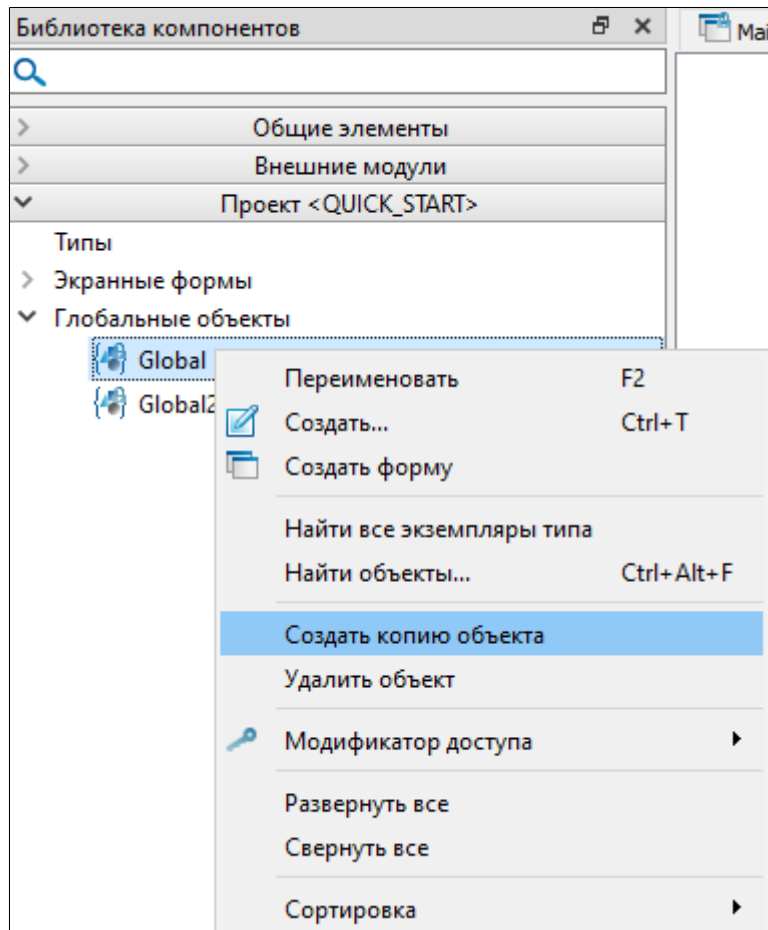


При первичном редактировании глобального объекта появится чистая рабочая область. Возможности работы в рабочей области глобального объекта аналогичны возможностям работы в рабочей области экранной формы.

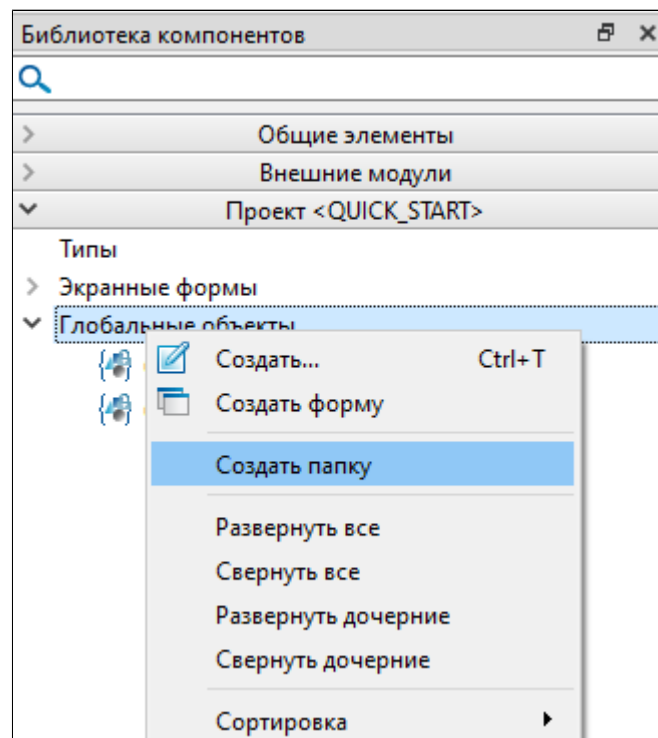


Созданные глобальные объекты сохраняются в папке проекта objects с расширением файлов \*.omobj.

Чтобы создать копию имеющегося глобального объекта, в Библиотеке компонентов выберите глобальный объект и в контекстном меню объекта выберите Создать копию объекта.



Чтобы группировать глобальные объекты, используйте папки. Для создания папки перейдите в контекстное меню и выберите команду Создать папку.







Избегайте совпадения имён объектов на разных уровнях иерархии. Например, если имя объекта в папке совпадёт с именем объекта вне папки, то вызываться будет объект размещённый вне папки.

## Редактирование глобального объекта

Для редактирования глобального объекта дважды щелкните по нему.

При первичном редактировании глобального объекта появится чистая рабочая область. Возможности работы в рабочей области глобального объекта аналогичны возможностям работы в рабочей области экранной формы.

## Обращение к глобальному объекту

К глобальному объекту и его содержимому (переменные, функции, элементы) можно обратиться из любого объекта проекта.



Чтобы обратиться к глобальному объекту из кода, используйте ключевое слово – `unit`:  
`unit.Global.Function_1();`

Чтобы обратиться к глобальному объекту, расположенному в папке, в коде указывайте только имя объекта без имени папки.

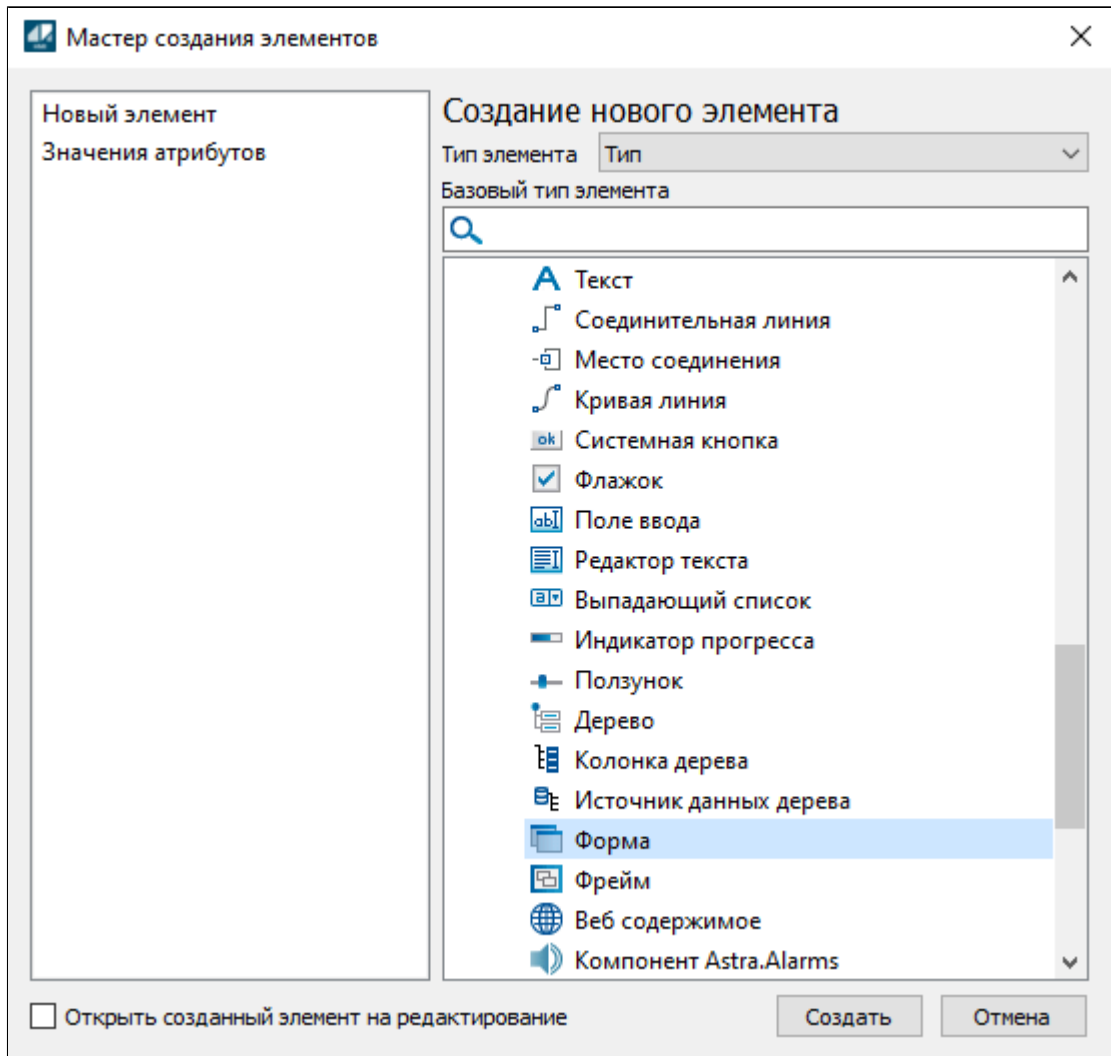
Чтобы получать оповещения при изменении значения уведомляющего поля глобального объекта, обратитесь к полю, используя точку.



`unit.Global.notifying_bool_1;`

## 1.1.4.2.6.2. Экранные формы

Чтобы создать новую экранную форму, выберите команду контекстного меню Создать и воспользуйтесь мастером создания элементов.

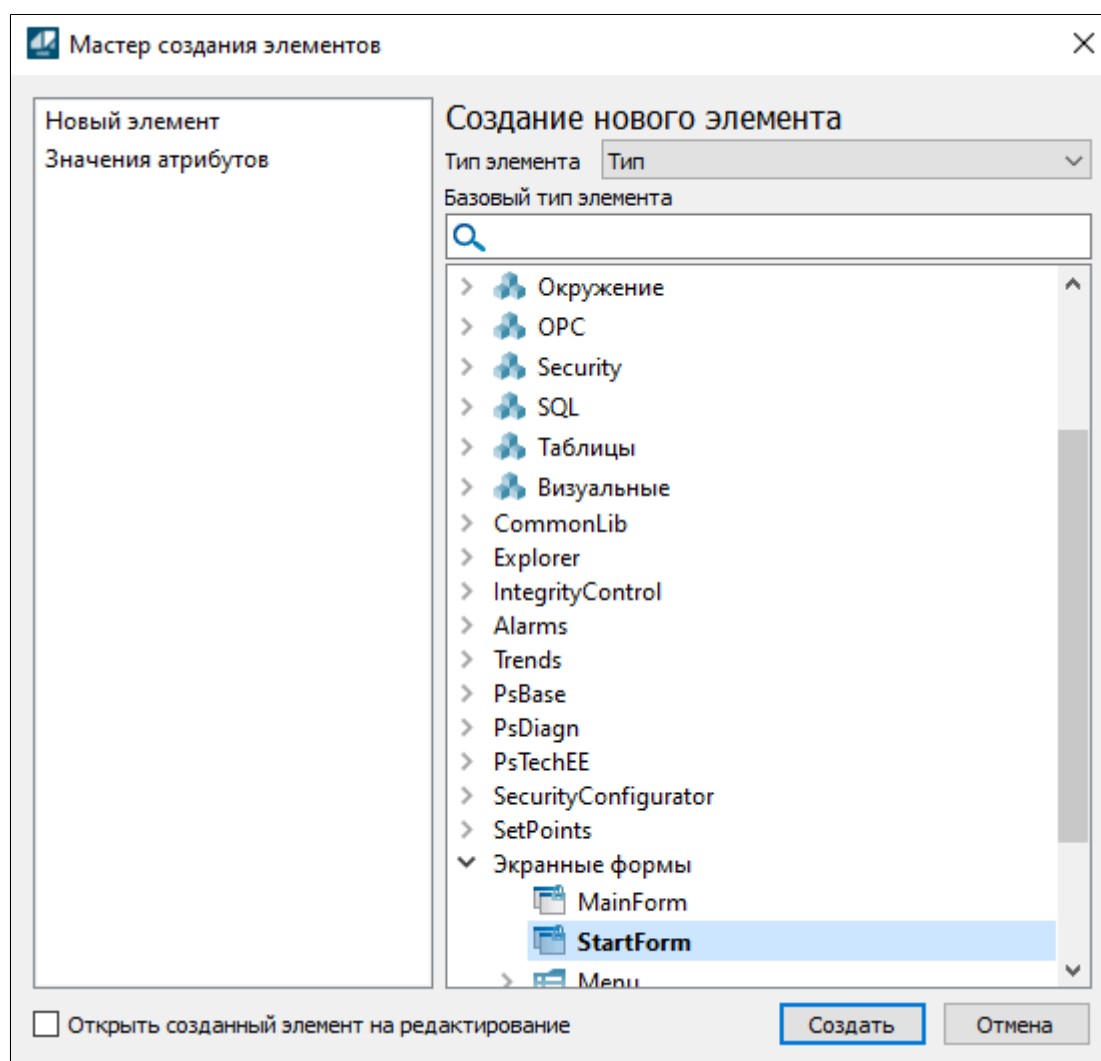


Создастся новая экранная форма со стандартным названием. Для каждой экранной формы создается новая вкладка рабочей области. Чтобы быстро переключаться между открытыми вкладками рабочей области, используйте сочетание клавиш Ctrl+Tab.

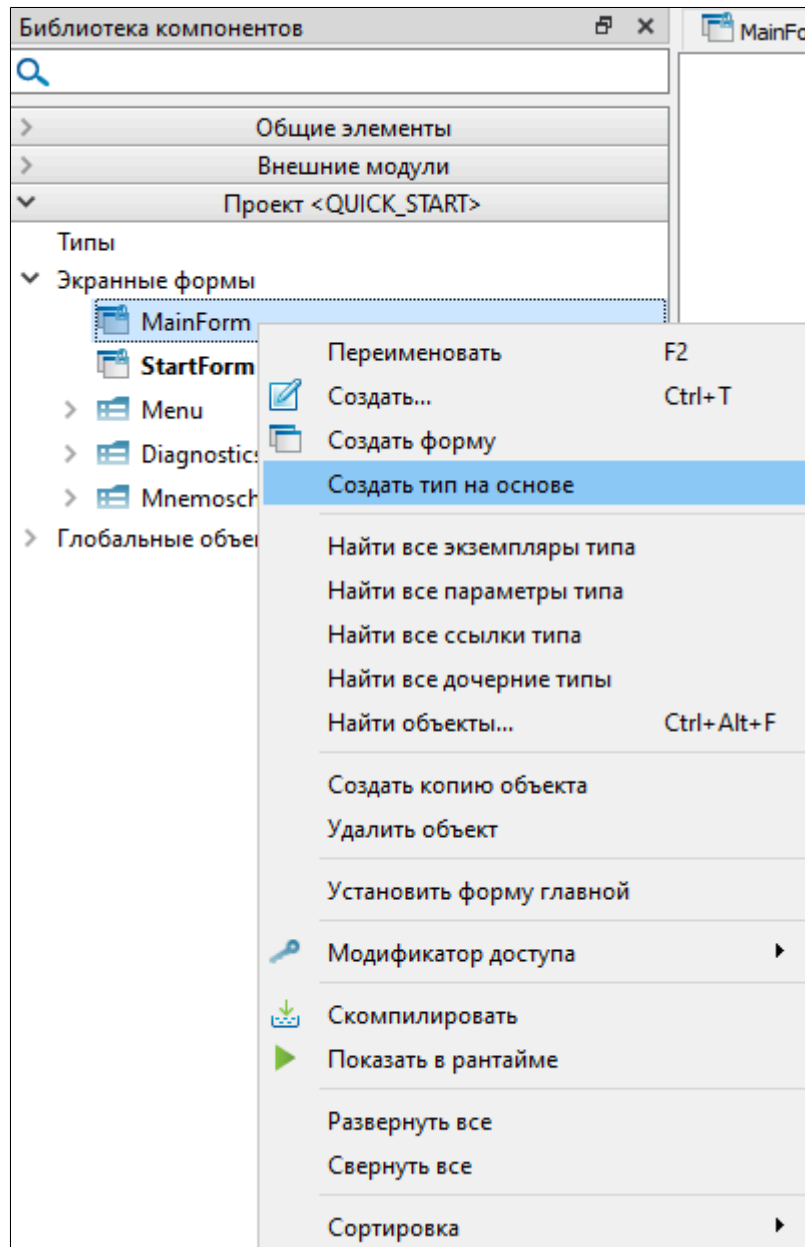
Чтобы открыть экранную форму для редактирования, дважды щелкните по ней в библиотеке компонентов.

При первичном редактировании формы появится чистая рабочая область.

Чтобы создать экранную форму на основе уже имеющейся экранной формы, в мастере создания элементов выберите форму, на основе которой, хотите создать новую форму и нажмите кнопку Создать.



Также создать экранную форму на основе существующей можно, выбрав команду Создать тип на основе в контекстном меню формы, на основе которой должна быть создана форма.



Чтобы скопировать экранную форму вместе с содержимым, в Библиотеке компонентов выберите экранную форму и команду Создать копию объекта в контекстном меню формы.

Чтобы группировать экранные формы, используйте папки. Для создания папки перейдите в контекстное меню и выберите команду Новая папка.

Созданные экранные формы сохраняются в папке проекта objects с расширением файлов \*.omobj. Созданные экранные формы можно экспортировать из проекта или импортировать в проект.

Чтобы настроить переходы между экранными формами, настройте обработчики событий:

- › открыть в текущем окне;
- › открыть в новом окне;
- › открыть в диалоговом окне;
- › открыть во фрейме.



Чтобы обратиться к свойствам экранной формы из кода, используйте служебное слово `here`.

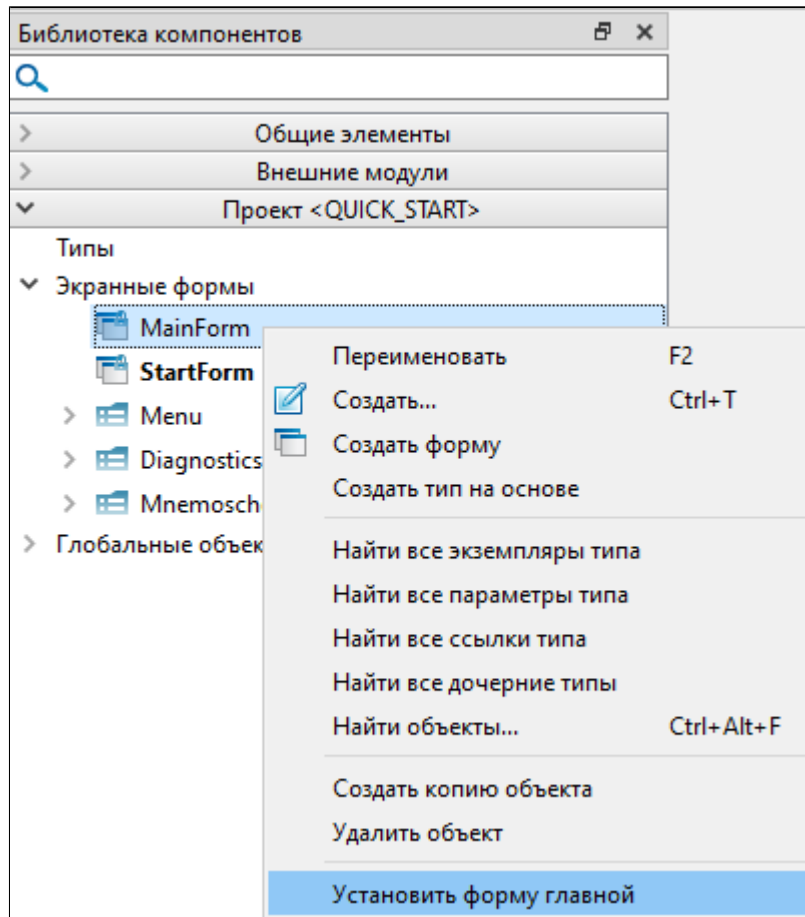
Чтобы скрыть/отобразить кнопки управления окном (Свернуть, Развернуть, Закреть), укажите значения соответствующим свойствам Показать кнопку "свернуть", Показать кнопку "развернуть", Показать кнопку "закреть".



Взаимодействие между основным и вспомогательными окнами осуществляется через компоненты окружения: группа окон и окно.

## 1.1.4.2.6.2.1. Указание главной формы

Чтобы указать главную форму в проекте, вызовите контекстное меню нужной экранной формы и выберите команду Установить форму главной.



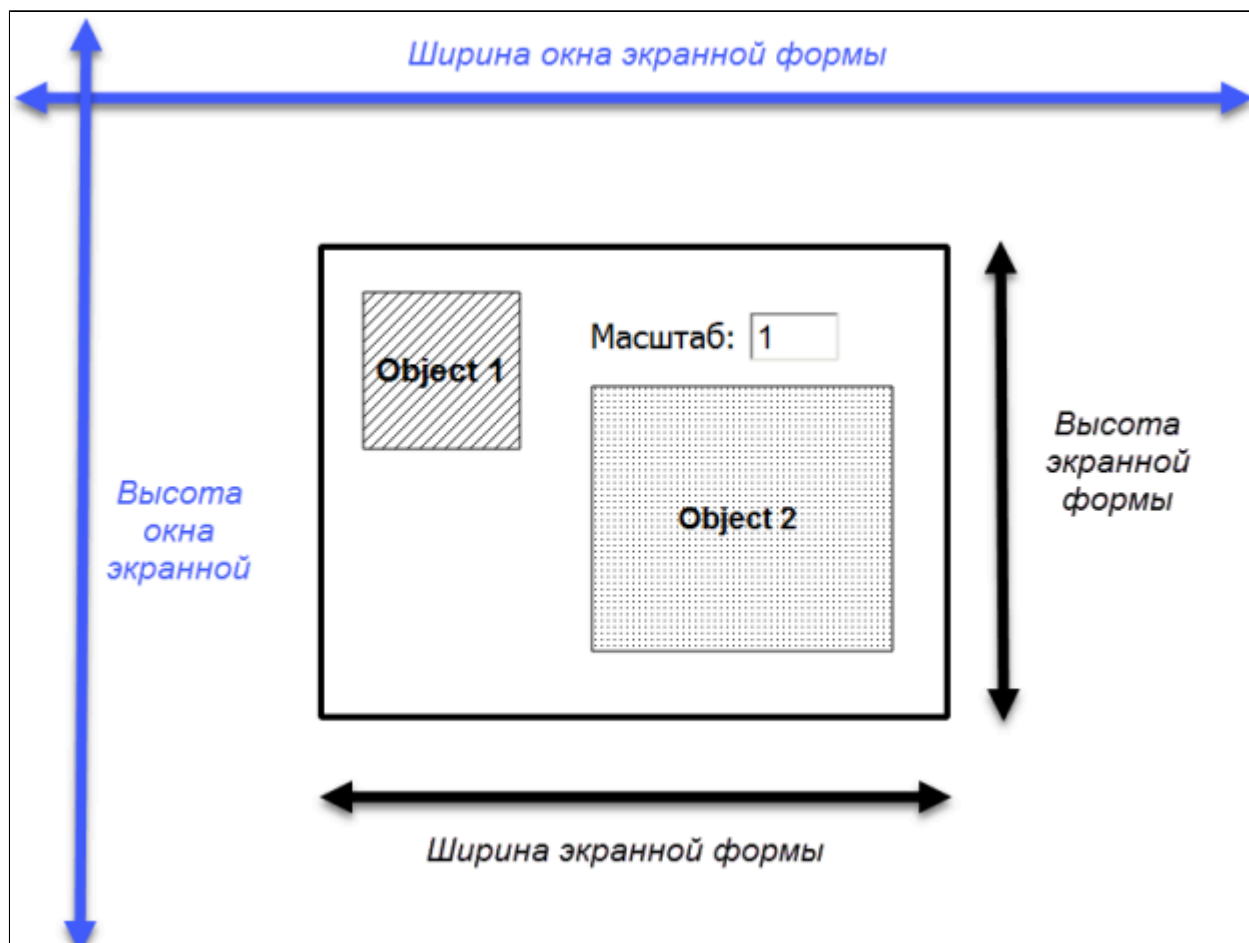
Главная форма запускается в рантайм при:

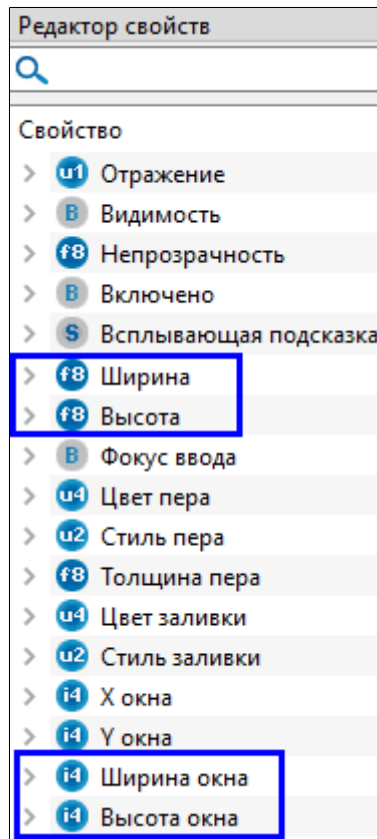
- › запуске проекта из проводника Windows;
- › выборе команды Показать главную форму в рантайме (F9).

Если ни одна форма не назначена главной, новая форма при добавлении автоматически устанавливается главной.

## 1.1.4.2.6.2.2. Размер и формат экранных форм

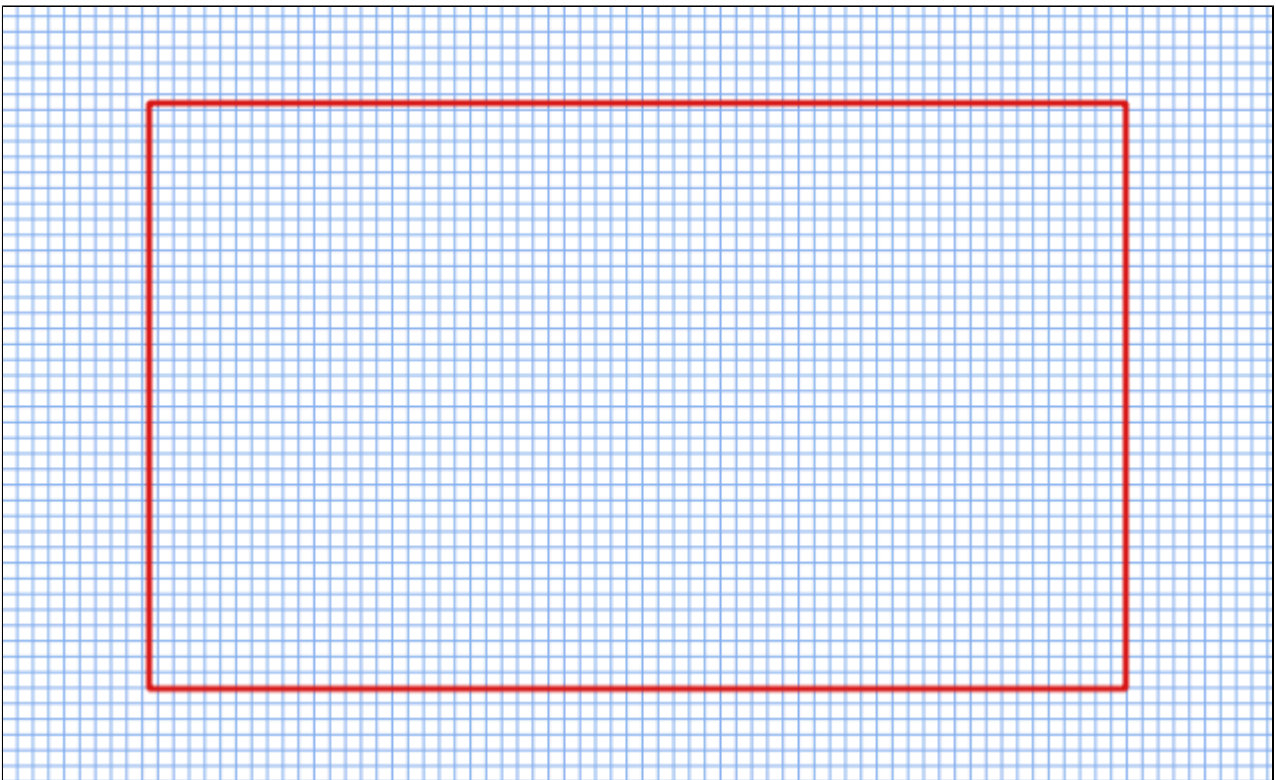
Любая экранная форма при запуске в рантайме будет расположена внутри окна-контейнера.





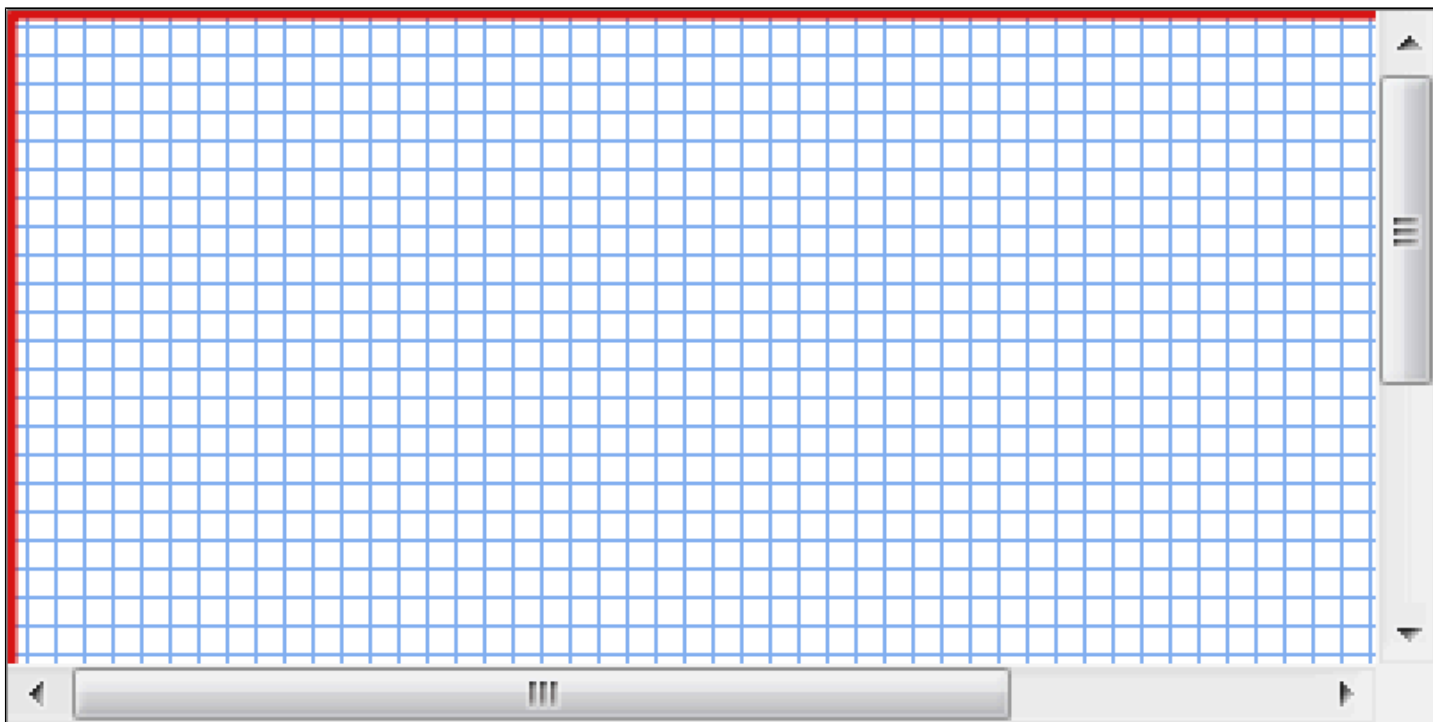
Размеры окна могут совпадать или не совпадать с размерами экранной формы:

- › если размер окна больше экранной формы, то будут видны пустые области.





› если размер окна меньше экранной формы, то появятся полосы прокрутки.



Варианты установки размеров окна:

- › Чтобы указать точный размер окна вручную (в пикселях), установите свойство экранной формы Размеры окна в состояние Вручную и укажите значения свойств Ширина окна и Высота окна.
- › Чтобы размер окна при открытии автоматически подстроился под размер экранной формы (и в дальнейшем не менялся), установите свойство Размеры окна в положение Автоподбор при открытии.
- › Чтобы размер окна динамически подстраивался под размер экранной формы на протяжении всего времени работы с окном, установите свойство Размеры окна в положение Автоподбор.



Режим Автоподбор может быть полезен, если размеры экранной формы динамически меняются в процессе исполнения мнемосхемы.

- › Чтобы размер окна автоматически подстроился под размеры экранной формы в режиме исполнения мнемосхемы, воспользуйтесь функцией экранной формы [DoAutoSize](#).

- Чтобы заблокировать возможность менять размеры окна, выставьте свойству Стиль рамки окна значение Фиксированный размер.
- Чтобы заблокировать возможность перемещать окно и менять размеры окна, выставите свойству Стиль рамки окна значение Без рамки.

Варианты установки размеров и масштаба экранной формы:

- Чтобы указать вручную точный размер экранной формы (в пикселях), установите свойство Режим масштабирования в состояние Не масштабировать и укажите значения свойств Ширина и Высота.
- Чтобы указать вручную коэффициент масштабирования экранной формы, установите свойство Масштаб. Значения меньше 1 уменьшают масштаб, значения больше 1 - увеличивают масштаб.
- Чтобы автоматически уменьшать масштаб экранной формы до размеров окна, установите свойство Режим масштабирования в положение Только уменьшение.



Использование опций авто-масштабирования экранной формы максимально впишет экранную форму в окно, но может привести к плохой прорисовке графических элементов или, напротив, к излишней величине графических элементов.

- Чтобы автоматически подгонять масштаб экранной формы до размеров окна, установите свойство Режим масштабирования в положение Всегда масштабировать.

Варианты установки состояния окна при открытии:

- Чтобы окно открывалось в одном из системных состояний (свернуто, развёрнуто, на весь экран), выберите нужное состояние для свойства Состояние окна.

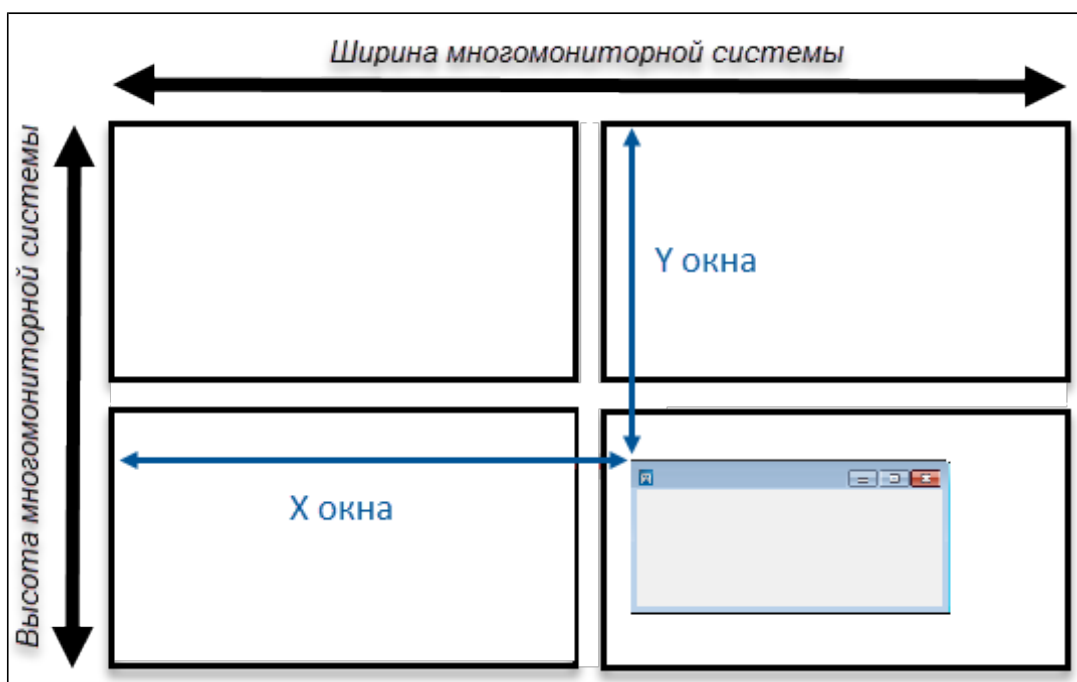
## 1.1.4.2.6.2.3. Расположение экранных форм

При работе с окнами экранных форм можно гибко задавать координаты их открытия. Координаты могут высчитываться относительно верхнего-левого угла единственного монитора или по абсолютным координатам многомониторной системы отображения.

### Координаты открытия экранной формы

Чтобы указать точное место открытия окна в абсолютных координатах многомониторной системы:

- › выставите свойство Положение окна в состояние Вручную;
- › укажите свойства X окна и Y окна в абсолютных координатах многомониторной системы.

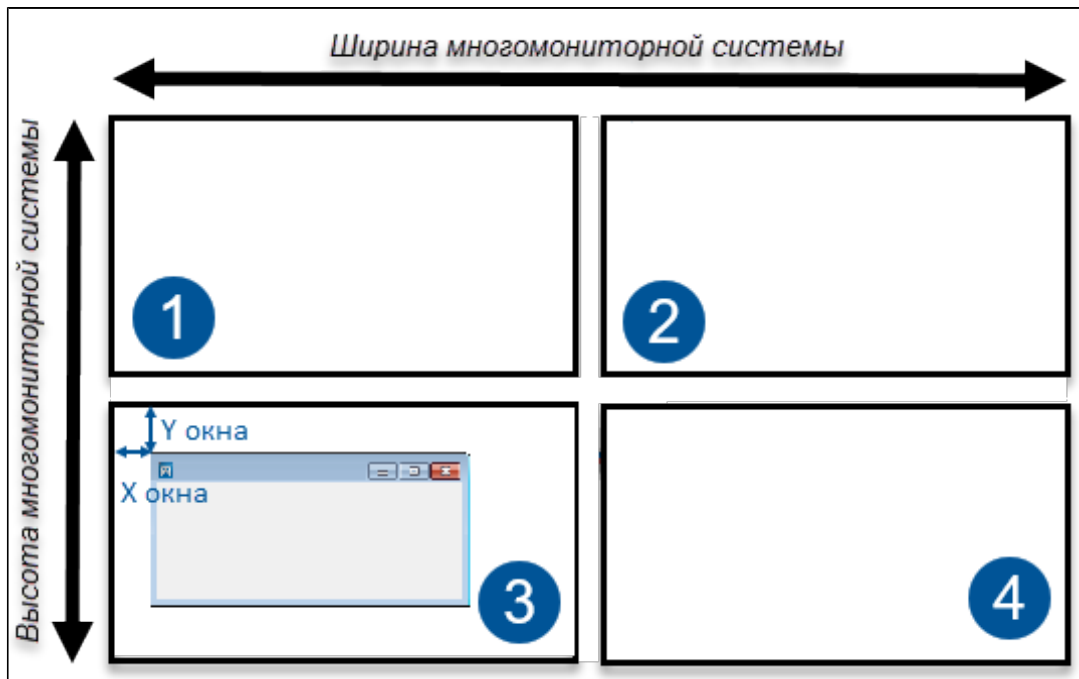


### Монитор открытия экранной формы

Чтобы указать точное место открытия окна экранной формы на конкретном мониторе в многомониторной системе:

- › выставите свойство Положение окна в состояние Вручную относительно монитора;

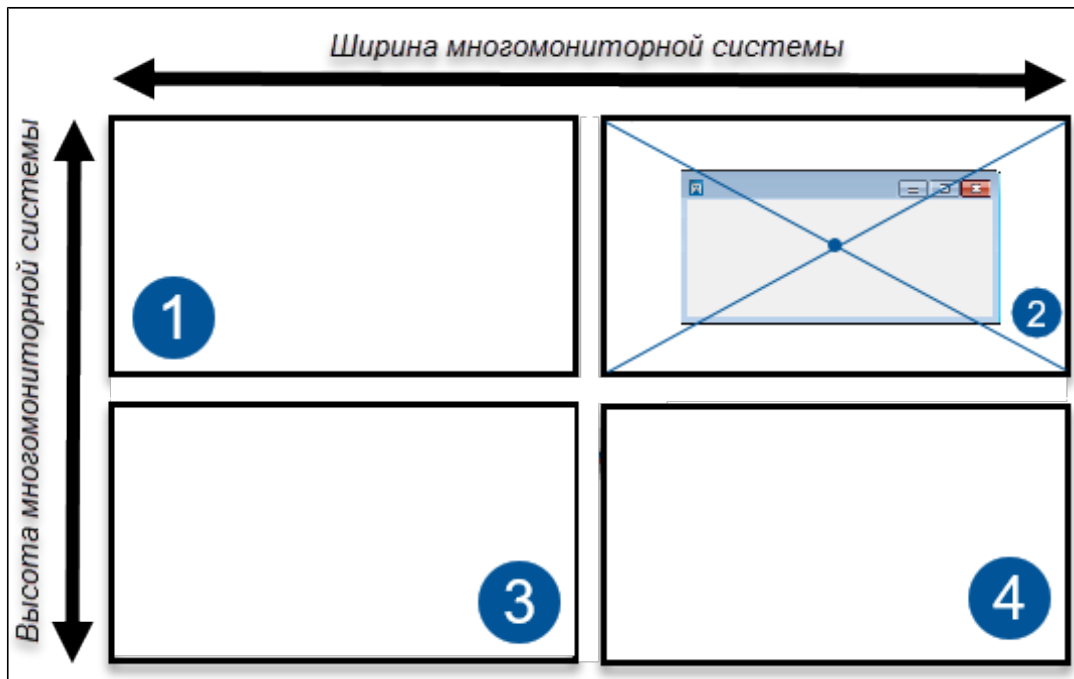
- › укажите номер монитора в свойстве Монитор;
- › укажите свойства X окна и Y окна относительно координат указанного монитора.



## Позиция открытия экранной формы

Чтобы окно экранной формы открывалось по центру указанного монитора, родительского окна или родительской формы:

- › выставите свойство экранной формы Положение окна в состояние По центру монитора / По центру родительского окна / По центру родительской формы;
- › укажите номер монитора в свойстве Монитор, если был выбран режим По центру монитора.



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства `Монитор` нужно оставлять равным 0.



Чтобы окно экранной формы открывалось поверх остальных окон мнемосхемы, активируйте свойство `Поверх всех окон`.

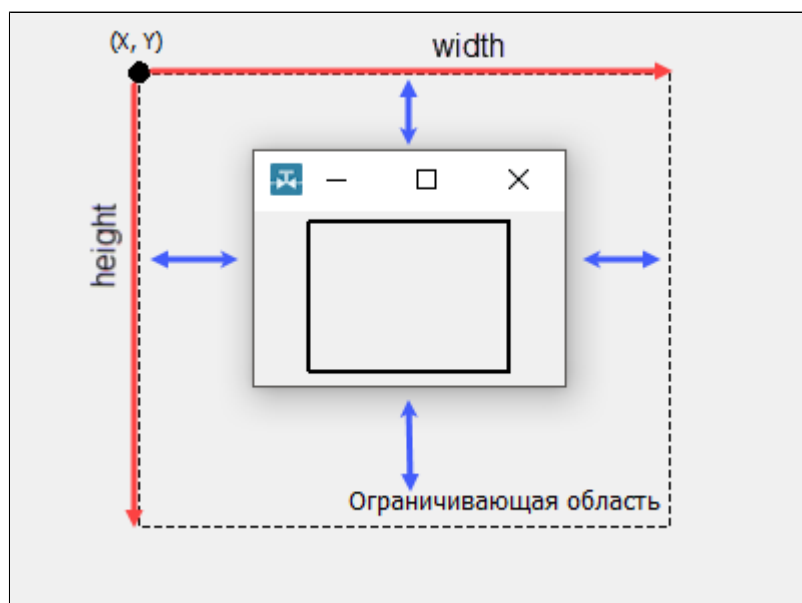
## Ограничение области перемещения экранной формы

Чтобы ограничить область перемещения экранной формы, воспользуйтесь соответствующей функцией [SetBoundingRegion](#).

Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.

Входные параметры:

- › `x` и `y` - координаты начала области ограничения;
- › `width` и `height` - ширина и высота области ограничения, отсчитываются от точки начала области.



## Примеры

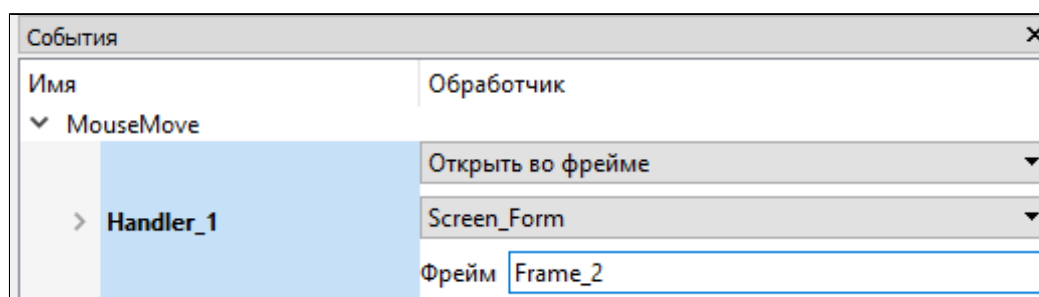


```
//Установить ограничения перемещения окна:  
here.SetBoundingRegion(500, 200, 400, 400);
```

## 1.1.4.2.6.2.4. Открытие экранной формы во фрейме

Чтобы содержимое экранной формы динамически открылось внутри другой формы, выполните следующие действия:

1. Добавьте на форму компонент Фрейм из вкладки Визуальные общих элементов.
2. Создайте обработчик события, открывающий экранную форму во фреймовой области:
  - Тип обработчика – Открыть во фрейме;
  - Тип формы – из выпадающего списка выберите имя экранной формы, которая должна открыться во фрейме;
  - В текстовом поле введите имя фреймовой области, в котором будет открыта экранная форма.



3. При срабатывании события в указанной фреймовой области Frame\_2 откроется экранная форма Screen\_Form.

# Масштабирование и перемещение экранной формы во фрейме

Чтобы в режиме рантайма менять размер и двигать экранную форму во фрейме, используйте функции Astra.Om: [ScaleContentTo](#), [SetContentScale](#) и [MoveViewport](#). Функции работают при включенном свойстве фрейма [Ручное управление масштабом](#).

## Масштабирование формы во фрейме

Укажите размеры формы во входных аргументах функции [ScaleContentTo](#) или масштаб в аргументе функции [SetContentScale](#). Если размер формы превысил размер фрейма, появляются полосы прокрутки. Убрать полосы прокрутки можно с помощью свойства фрейма [Отображать полосы прокрутки](#).

## Перемещение формы во фрейме

Перетаскивайте форму с помощью мыши или задайте координаты формы во входных аргументах функции [MoveViewport](#). Координаты высчитываются относительно верхнего левого угла фрейма. Перетаскивать форму мышью можно при включенном свойстве фрейма [Перемещение мышью](#).



## 1.1.4.2.6.2.6. События

Событие	Описание
<a href="#">KeyRelease</a>	Отпускание определённой клавиши или сочетания клавиш на форме
<a href="#">KeyPress</a>	Нажатие определённой клавиши или сочетания клавиш на форме
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">Closed</a>	Закрытие формы
<a href="#">Opened</a>	Открытие формы
<a href="#">AboutToOpen</a>	Событие перед открытием окна
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseDown</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана

## 1.1.4.2.6.2.6.1. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.2.6.2.6.2. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



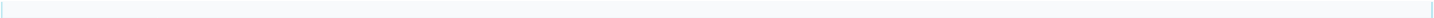
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.2.6.2.6.3. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.2.6.2.6.4. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.2.6.2.6.5. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)



## 1.1.4.2.6.2.6.6. Closed

Закрытие формы.

## 1.1.4.2.6.2.6.7. Opened

Открытие формы.

## 1.1.4.2.6.2.6.8. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.2.6.2.6.9. CloseRequested

Запрос на закрытие формы.

## 1.1.4.2.6.2.6.10. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.11. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.12. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.2.6.2.6.13. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.14. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.15. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.16. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.17. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.2.6.2.6.18. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.2.6.2.6.19. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.2.6.2.6.20. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```



## 1.1.4.2.6.2.6.21. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.2.6.2.7. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы



## 1.1.4.2.6.2.7.1. ResetBoundingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoundingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoundingRegion();
```

## 1.1.4.2.6.2.7.2. SetBoundingRegion

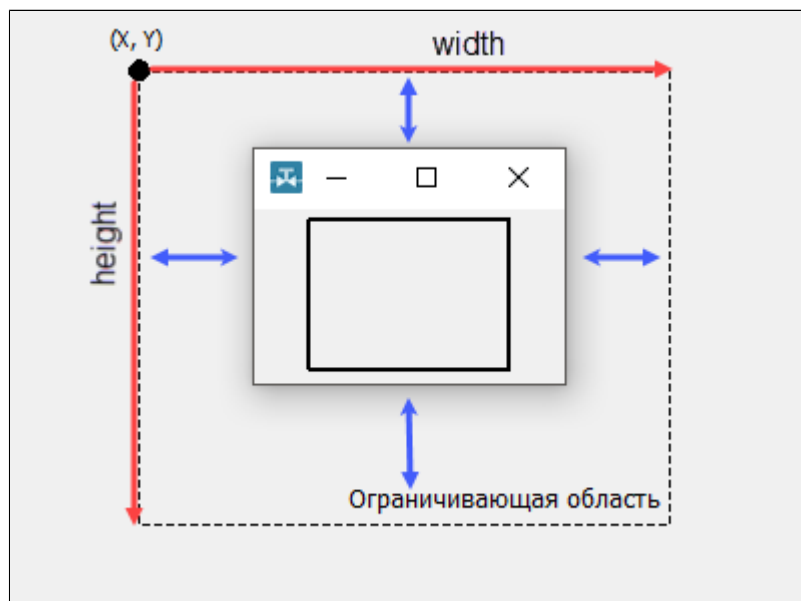
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
here.SetBoundingRegion(500, 200, 400, 400);

## 1.1.4.2.6.2.7.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.2.6.2.7.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла



## 1.1.4.2.6.2.7.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.2.6.2.7.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.2.6.2.7.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.2.6.2.7.8. Activate

Сделать окно формы активным.



```
void Activate()
```

## 1.1.4.2.6.2.7.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.2.6.2.7.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

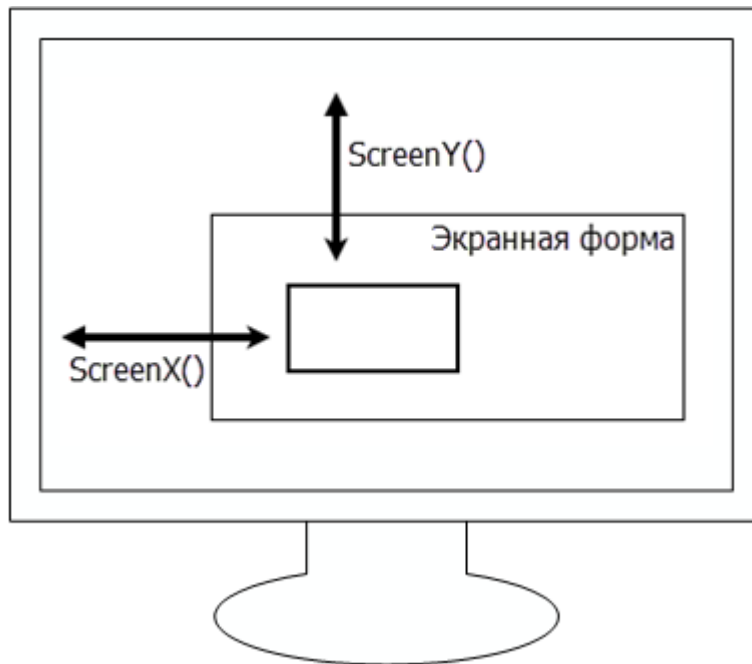
### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.2.6.2.7.11. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



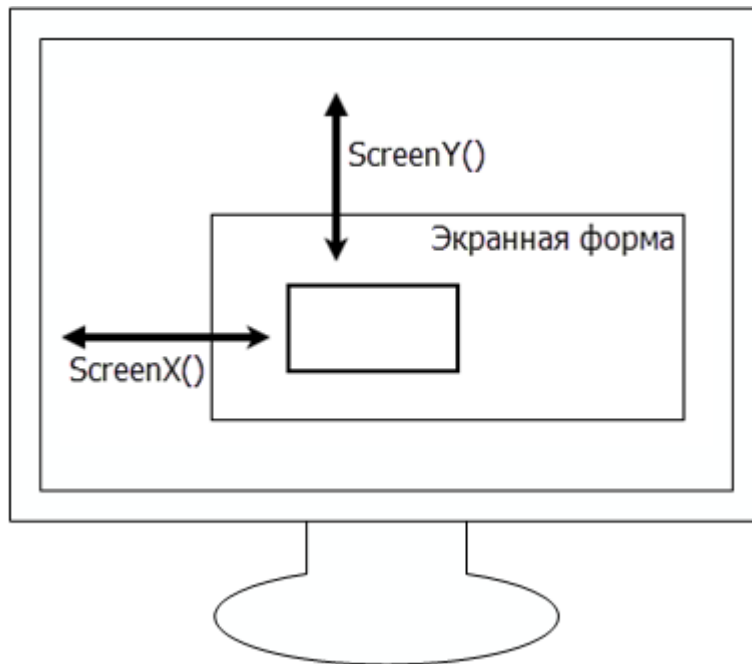
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.2.6.2.7.12. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.2.6.2.7.13. Show

Показать окно формы.



void Show()

## 1.1.4.2.6.2.7.14. Hide



Скрыть окно формы.



void Hide()

## 1.1.4.2.6.2.8. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования контейнера экранной формы
<a href="#">Отражение</a>	Отражение
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на свободную область контейнера экранной формы
<a href="#">Ширина</a>	Ширина контейнера экранной формы в пикселях
<a href="#">Высота</a>	Высота контейнера экранной формы в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы контейнера экранной формы
<a href="#">Стиль пера</a>	Стиль внешней границы контейнера экранной формы
<a href="#">Толщина пера</a>	Толщина внешней границы контейнера экранной формы
<a href="#">Цвет заливки</a>	Цвет заливки окна экранной формы
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки

<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закрыть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна
<a href="#">Активность окна</a>	Текущее состояние активности окна экранной формы

<a href="#">MouseX</a>	Позиция курсора мыши по оси X в локальных координатах объекта
<a href="#">MouseY</a>	Позиция курсора мыши по оси Y в локальных координатах объекта
<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки

## 1.1.4.2.6.2.8.1. Отображаемое имя

Произвольное описание объекта (допускается русскоязычное).

## 1.1.4.2.6.2.8.2. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

## 1.1.4.2.6.2.8.3. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.



## 1.1.4.2.6.2.8.4. X

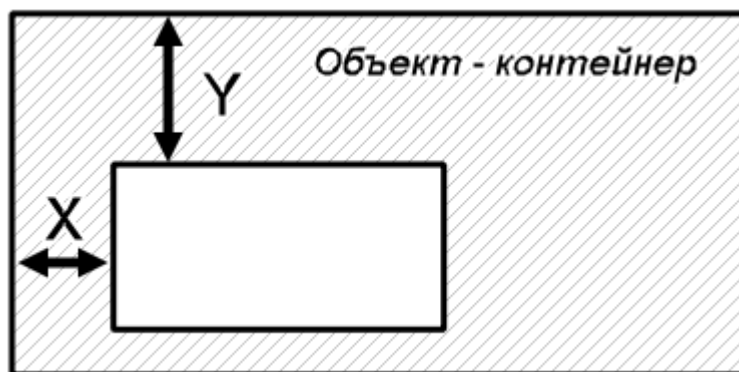
Позиция объекта по горизонтали в пикселях (сдвиг относительно левого верхнего угла родительского объекта-контейнера).



float8 X



Свойства X и Y заблокированы в редакторе свойств. Чтобы изменить значения данных свойств, воспользуйтесь скриптом обработчика событий.



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.2.6.2.8.5. Y

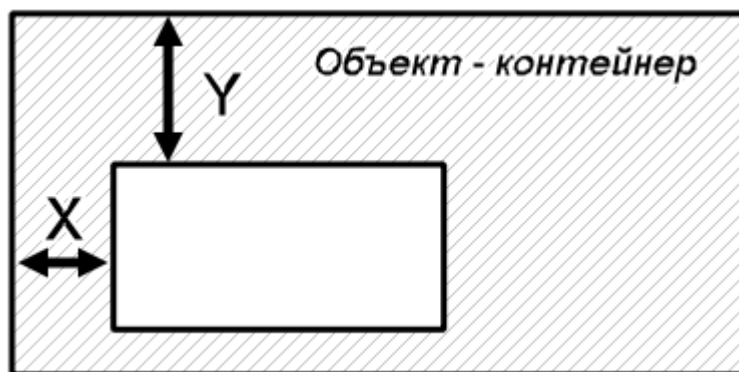
Позиция объекта по вертикали в пикселях (сдвиг относительно левого верхнего угла родительского объекта-контейнера).



float8 Y



Свойства X и Y заблокированы в редакторе свойств. Чтобы изменить значения данных свойств, воспользуйтесь скриптом обработчика событий.



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

## 1.1.4.2.6.2.8.6. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.2.6.2.8.7. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation



Свойство Угол поворота заблокировано в редакторе свойств. Чтобы изменить значения данного свойства, воспользуйтесь скриптом обработчика событий.

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;
```

## 1.1.4.2.6.2.8.8. Масштаб

Коэффициент масштабирования контейнера экранной формы.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.2.6.2.8.9. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.2.6.2.8.10. Видимость

Видимость объекта на форме.



bool Visible

### Примеры



```
//Отображение по условию  
here.Visible = signal1.value < signal2.value;
```

## 1.1.4.2.6.2.8.11. Непрозрачность

Непрозрачность.



float8 Opacity



## 1.1.4.2.6.2.8.12. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.2.6.2.8.13. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на свободную область контейнера экранной формы.



string Tooltip

### Примеры



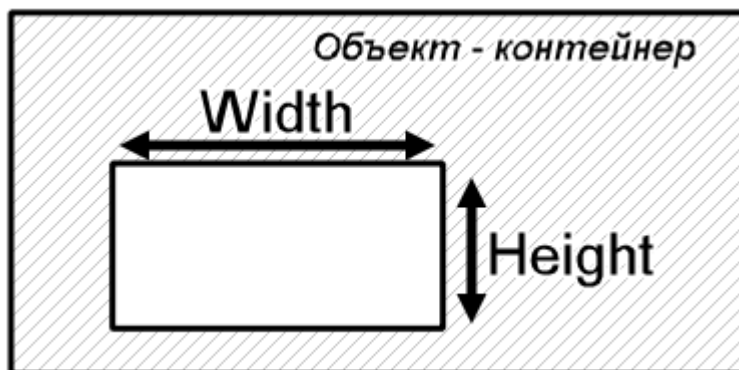
```
//Задать всплывающую подсказку для объекта:  
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.2.6.2.8.14. Ширина

Ширина контейнера экранной формы в пикселях.



float8 Width

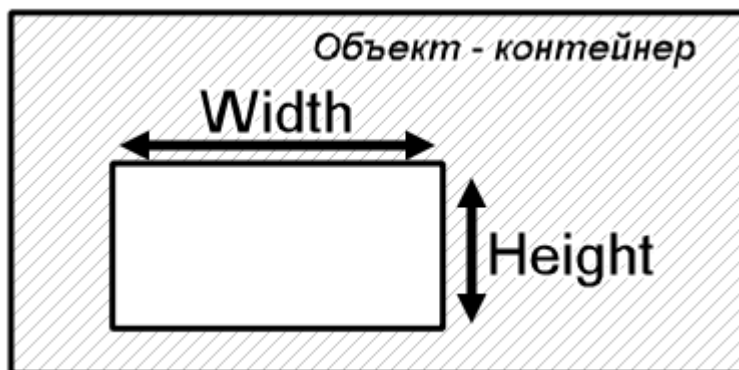


## 1.1.4.2.6.2.8.15. Высота

Высота контейнера экранной формы в пикселях.



float8 Height



## 1.1.4.2.6.2.8.16. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта


### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
{
    pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
}
```

## 1.1.4.2.6.2.8.17. Цвет пера

Цвет внешней границы контейнера экранной формы. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления  .



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;






## 1.1.4.2.6.2.8.18. Стиль пера

Стиль внешней границы контейнера экранной формы.



uint2 PenStyle

### Значение

Значение	Граница	Внешний вид
0	Без границы	
1	Сплошная линия	
2	Длинный пунктир	
3	Короткий пунктир	
4	Смешанный пунктир	
5	Смешанный пунктир	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;
```

## 1.1.4.2.6.2.8.19. Толщина пера

Толщина внешней границы контейнера экранной формы. Измеряется в пикселях.



float8 PenWidth


### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```



## 1.1.4.2.6.2.8.20. Цвет заливки

Цвет заливки окна экранной формы. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления  .



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.BrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.BrushColor = 4294901760;
```

## 1.1.4.2.6.2.8.21. Стиль заливки


Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнере	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.2.6.2.8.22. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.2.6.2.8.23. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.2.6.2.8.24. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.2.6.2.8.25. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.2.6.2.8.26. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```



## 1.1.4.2.6.2.8.27. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

## 1.1.4.2.6.2.8.28. Показывать кнопку "свернуть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

### Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

## 1.1.4.2.6.2.8.29. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

### Примеры



```
//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;
```

## 1.1.4.2.6.2.8.30. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.2.6.2.8.31. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.2.6.2.8.32. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.2.6.2.8.33. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.2.6.2.8.34. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```



## 1.1.4.2.6.2.8.35. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.2.6.2.8.36. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.2.6.2.8.37. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.2.6.2.8.38. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна – Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.2.6.2.8.39. Активность окна

Текущее состояние активности окна экранной формы.



bool WindowActive



Доступно только для чтения в режиме рантайма.

### Значение

Значение	Описание
true	Окно активно
false	Окно неактивно

### Примеры



```
//Вывести состояние активности окна в поле ввода:  
TextEdit.Text = String.ToString(here.WindowActive);
```

## 1.1.4.2.6.2.8.40. Путь к файлу иконки

Путь к файлу иконки.



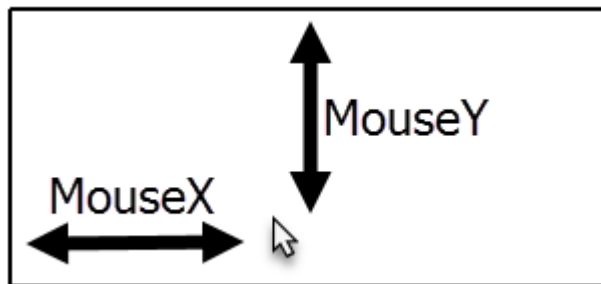
string WindowIconPath

## 1.1.4.2.6.2.8.41. MouseX

Позиция курсора мыши по оси X в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseX



Доступно только для чтения в режиме рантайма.

## Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

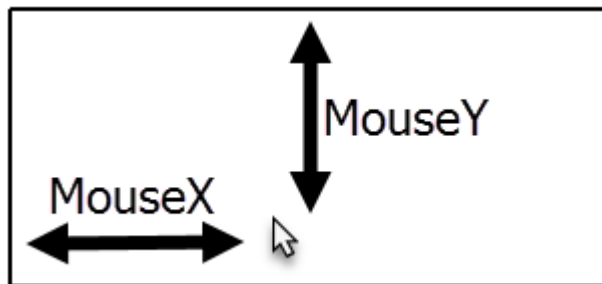
```
TextEdit.Text = String.ToString(Object.MouseY);
```

## 1.1.4.2.6.2.8.42. MouseY

Позиция курсора мыши по оси Y в локальных координатах объекта. Измеряется в пикселях, как сдвиг относительно левого верхнего угла объекта.



float8 MouseY



Доступно только для чтения в режиме рантайма.

### Примеры



//Вывести в текстовую область координаты курсора мыши внутри объекта:

```
TextEdit.Text = String.ToString(Object.MouseX);
```

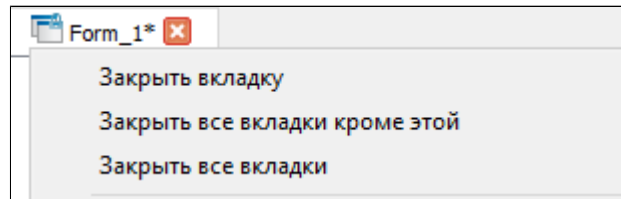
```
TextEdit.Text = String.ToString(Object.MouseY);
```



## 1.1.4.2.6.3. Объекты

Любой объект проекта характеризуется свойствами, функциями и событиями.

Чтобы завершить просмотр или редактирование объектов, типов или экранных форм, вызовите контекстное меню редактируемой вкладки и выберите нужную команду.



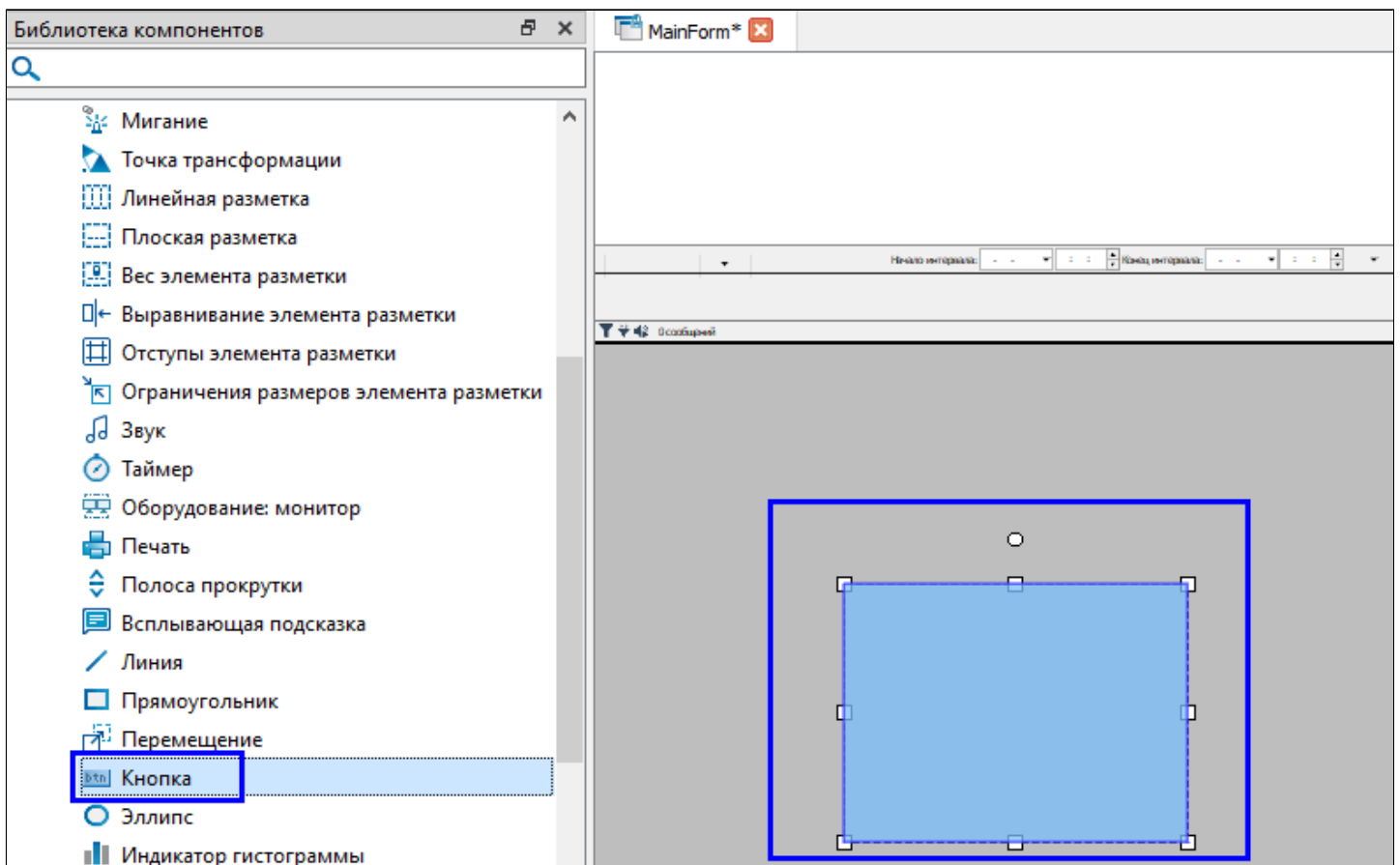
Для быстрого закрытия текущей вкладки используйте сочетание клавиш Ctrl+W.

После запуска в режиме исполнения в окне экранной формы отобразятся только те объекты, которые были расположены в ее пределах.

## 1.1.4.2.6.3.1. Добавление объектов на форму

Чтобы добавить объекты:

- › перетащите нужные компоненты из библиотеки на рабочую область;
- › в Структуре объекта выберите объект, которому нужно добавить дочерний, откройте контекстное меню и выберите Создать. Также добавить дочерний элемент можно перетащив нужный элемент из вкладки Общие элементы на родительский элемент в рабочей области.



Для объектов, размещенных на рабочей области, действуют стандартные операции форматирования:

- › вырезать (Ctrl+X);
- › копировать (Ctrl+C);
- › вставить (Ctrl+V).

Также эти команды можно вызвать из меню Правка или с панели инструментов.



После запуска в режиме исполнения в окне экранной формы отобразятся только те объекты, которые были расположены в пределах экранной формы.

## 1.1.4.2.6.3.2. Именованние объектов

Чтобы переименовать добавленный объект, перейдите в область Структура объекта и дважды щелкните по имени объекта либо вызовите контекстное меню объекта и выберите команду Переименовать.



Не используйте в названиях объектов пробелы.



Следите, чтобы имена объектов в папках и вне папок не совпадали. При обращении к объектам приоритет всегда отдается объектам вне папок.

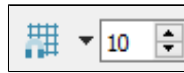


Чтобы обратиться к объекту, расположенному в папке, в коде указывайте только имя объекта без имени папки.

Для любого объекта или элемента вы можете указать краткое русскоязычное описание в свойстве Отображаемое имя. При этом сохраняется возможность обращения к объекту/элементу в скриптах по англоязычному имени, которое указано в области Структура объекта. Удобно использовать отображаемое имя для форм, типов, экземпляров типов и полей.

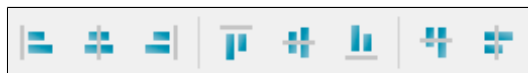
## 1.1.4.2.6.3.3. Выравнивание и распределение

Чтобы выровнять объекты вручную, включите привязку к сетке на Панели инструментов. В поле ввода можно установить ширину шага (в пикселях).



Чтобы выровнять группу объектов автоматически, используйте инструменты выравнивания Панели инструментов.

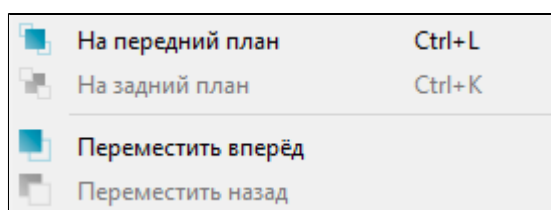
Чтобы выровнять объекты по краю (левый, правый, верхний, нижний) или центру, выделите нужную группу объектов и примените один из инструментов выравнивания Панели инструментов.



Чтобы равномерно распределить объекты на рабочей области (с одинаковыми расстояниями между ними), выделите нужную группу объектов и используйте инструменты распределения (по горизонтали, по вертикали) на Панели инструментов.



Чтобы изменить порядок объектов, перекрывающих друг друга, выделите нужный объект и укажите нужный порядок для объекта в контекстном меню, либо используя кнопки Панели инструментов.



Комбинации Ctrl+L, Ctrl+K используются для быстрого перемещения объекта на передний или задний план.

## 1.1.4.2.6.3.4. Масштабирование и поворот объектов

Объекты, добавленные на рабочую область, можно растягивать или поворачивать, используя маркеры захвата. Менять размер объекта можно также с помощью сочетаний клавиш Ctrl+↓, Ctrl+↑, Ctrl+←, Ctrl+→. Чтобы изменить размер объекта без нарушения пропорций, растягивайте его с зажатой клавишей SHIFT.

Чтобы изменить точку, относительно которой происходит масштабирование или поворот объекта, добавьте на объект компонент библиотеки Точка трансформации и укажите координаты точки трансформации (свойства X и Y). Перемещать точку трансформации можно также с помощью мыши, либо клавиш ↓, ↑, ←, →.



Чтобы масштабировать рабочую область и все объекты рабочей области, используйте инструменты панели масштабирования или поворачивайте колесико мыши с зажатой клавишей CTRL.



Чтобы включить/отключить сглаживание изображения при работе в дизайнере Astra.HMI, используйте функцию Сглаживание изображений, которая находится в меню Вид.

## 1.1.4.2.6.3.5. Видимость и активность объекта

Чтобы выключить видимость объекта в режимах разработки и исполнения, установите свойство Видимость в значение false.

Чтобы деактивировать объект (элемент будет виден, но не будет реагировать на действия пользователя), установите свойство Включено в значение false.

Чтобы управлять видимостью/активностью всех объектов экранной формы, меняйте значения свойств Видимость/Включено экранной формы.

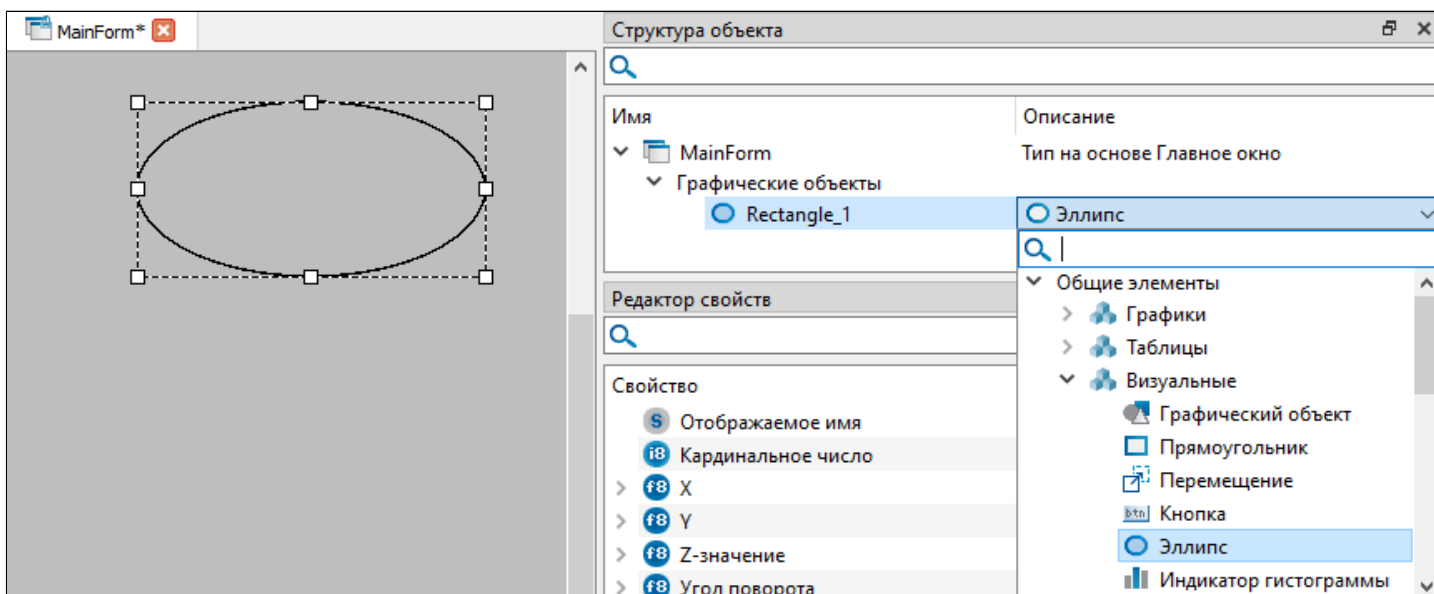
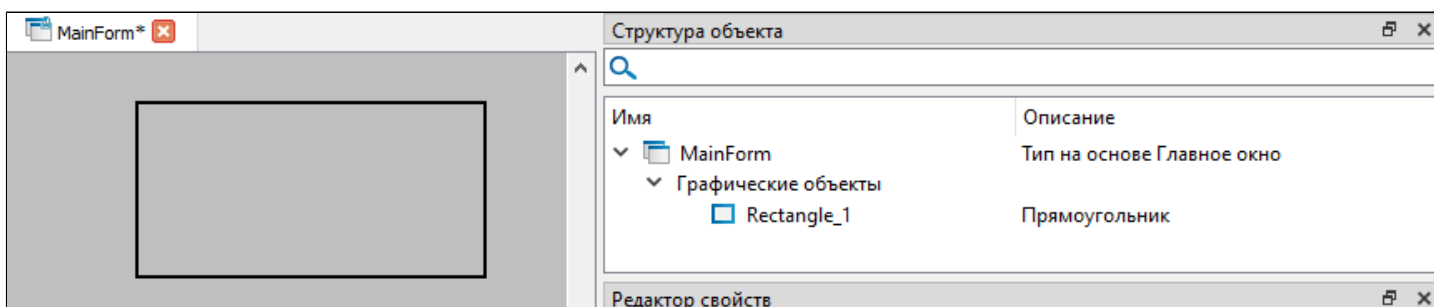
При деактивации объекта деактивируются все его дочерние объекты. При активации родительского объекта активируются только те дочерние объекты, которые были активны до выключения родителя.



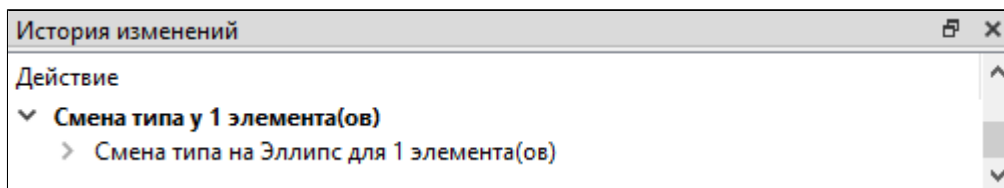
## 1.1.4.2.6.3.6. Изменение типа

Через структуру объекта можно быстро менять тип объекта (смена допустима в рамках схожих типов).

Чтобы изменить тип, в структуре выберите объект и нажмите на тип. В результате появится выпадающий список, из которого выберите новый тип для объекта.



Информация об изменении типа объекта отражается в истории изменений.



## 1.1.4.2.6.3.7. Поиск объектов

Чтобы найти объекты на рабочей области, выберите команду Правка → Найти (CTRL+F).

В появившемся окне Поиск по проекту введите название объекта и нажмите кнопку Найти.

Чтобы при поиске учитывался регистр символов искомой фразы, установите флаг С учетом регистра.

Для поиска могут применяться символы подстановки:

- › \* – заменяют любую строку символов;
- › ? – заменяет любой одиночный символ.

Перечень найденных объектов появляется в области Результаты поиска.

### Примеры



Найти в проекте объекты, в названиях которых есть сочетание символов "tu1":

\*tu1\*

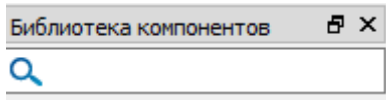


Найти в проекте объекты, в названиях которых заканчивается на ".txt", а начало содержит любые 3 символа:

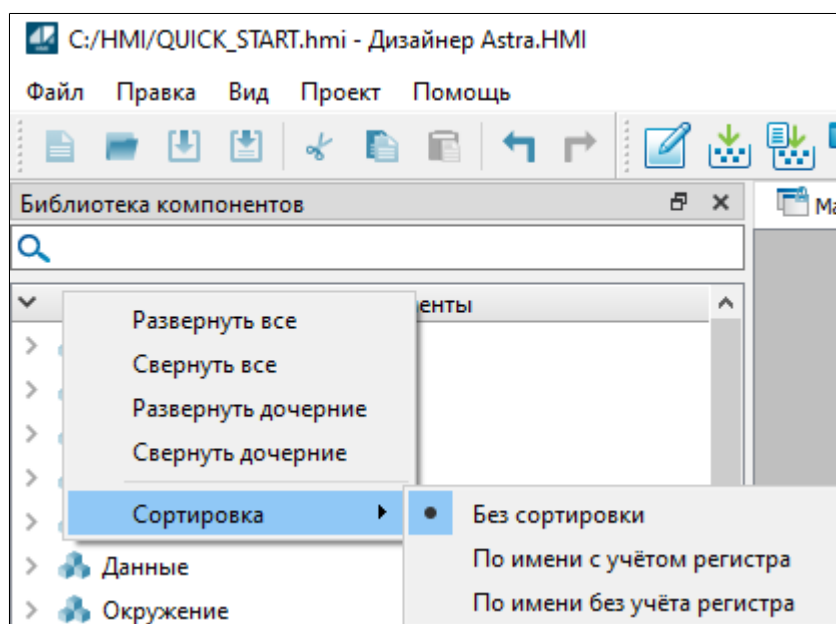
???.txt

## 1.1.4.2.6.3.8. Фильтрация и сортировка

Чтобы отфильтровать компоненты в библиотеке, введите нужное сочетание

символов в поле . После фильтрации отобразятся только вложенные элементы групп и папок (компоненты, экранные формы, типы, глобальные объекты). Родительские элементы (группы, папки) отобразятся только если один или несколько их вложенных компонентов соответствуют фильтру.

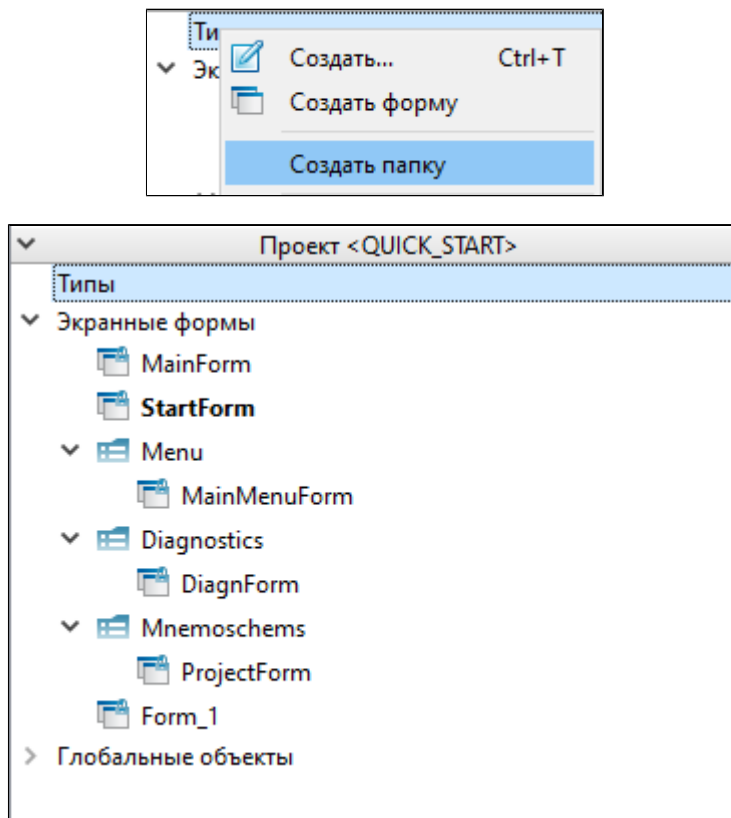
Чтобы отсортировать элементы в библиотеке компонентов, вызовите контекстное меню, выберите команду Сортировка и укажите тип сортировки. Сортировка применяется для всех групп библиотеки одновременно.



- › Без сортировки – элементы располагаются в произвольном порядке;
- › По имени с учетом регистра – элементы располагаются по алфавиту, в начало списка помещаются элементы, названия которых начинаются с заглавной буквы;
- › По имени без учета регистра – элементы располагаются по алфавиту.

## 1.1.4.2.6.3.9. Группировка и вложенность объектов

Чтобы сгруппировать экранные формы, типы или глобальные объекты, поместите их в папку. Чтобы создать папку, перейдите в контекстное меню элементов Типы/Экранные формы/Глобальные объекты и выберите команду Создать папку.



Вложенность объектов удобно использовать для распространения изменения свойств объекта-родителя на параметры дочерних объектов. Чтобы вложить объекты друг в друга, перетаскивайте их мышью в области Структура объекта. Визуальные графические объекты автоматически перемещаются на рабочей области в соответствии с иерархией в Структуре объектов. Привязку невидимых графических объектов к объекту-родителю можно отследить только в Структуре объектов.

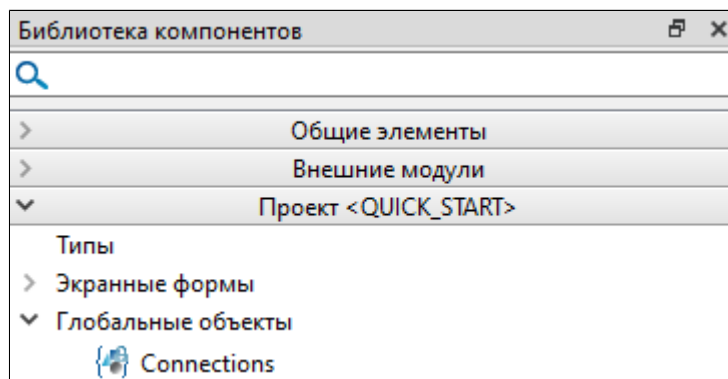


Для быстрой навигации в Структуре объектов используйте кнопки клавиатуры ←↑→↓.

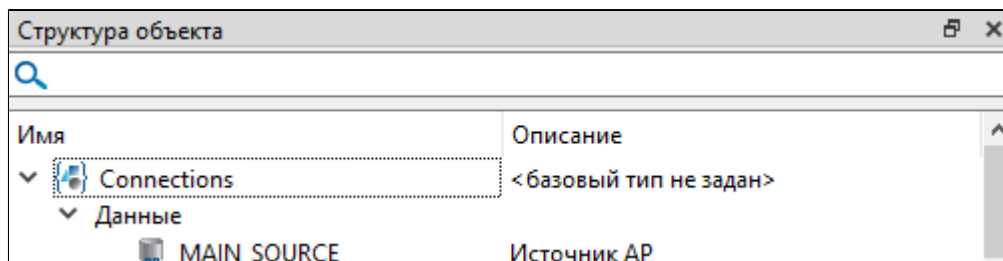
## 1.1.4.2.6.4. Создание источника данных

Взаимодействие с источником данных осуществляется по проприетарному протоколу на базе TCP.

1. Добавьте в проект глобальный объект "Connections".



2. Добавьте в глобальный объект "Connections" элемент "Источник AP" и задайте имя MAIN\_SOURCE.



Компонент Источник AP не визуальный и виден только в области Структура объекта.

3. Настройте свойства добавленного источника.

Свойство	Описание
Хост	Сетевой адрес источника данных по протоколу TCP. Для локального компьютера используется IP-адрес: 127.0.0.1
Порт	Номер порта для подключения к источнику данных.

	По умолчанию: 4388
Порт истории	Номер порта для доступа к историческим данным. По умолчанию: 4950
Путь	Путь до целевой ветки дерева сигналов источника данных по протоколу TCP. Если путь не указан, то по умолчанию целевой веткой будет считаться корень дерева сигналов источника данных
Активность	Активность источника AP. С неактивными источниками не происходит обмен данными

Редактор свойств

🔍

Свойство	Характеристики	Значение
🅈 Отображаемое имя		MAIN_SOURCE
🅈 Кардинальное число		1
🖨 Родительский источник	R W → 0	< не определено >
🖨 Источник учетных данных	R W → 0	< не определено >
🅈 Таймаут операций с БД	R ≤ ✓	1000
🅈 Таймаут установления соединения с БД	R ≤ ✓	1000
🅈 Максимальный размер пула буферов	R ≤ ✓	1000000
🅈 <b>Хост</b>	R W	
Начальное значение	✓	127.0.0.1
Вычисляемое значение		< не определено >
> 🅈 Порт	R W	4388
> 🅈 Порт истории	R W	4950
> 🅈 Путь	R W	
> 🅈 <b>Активность</b>	R W	true
> 🅈 Период переподписки	R W	0
> 🅈 Имя клиента	R W	HMI
> 🅈 Идентификатор клиента	R W	HMI

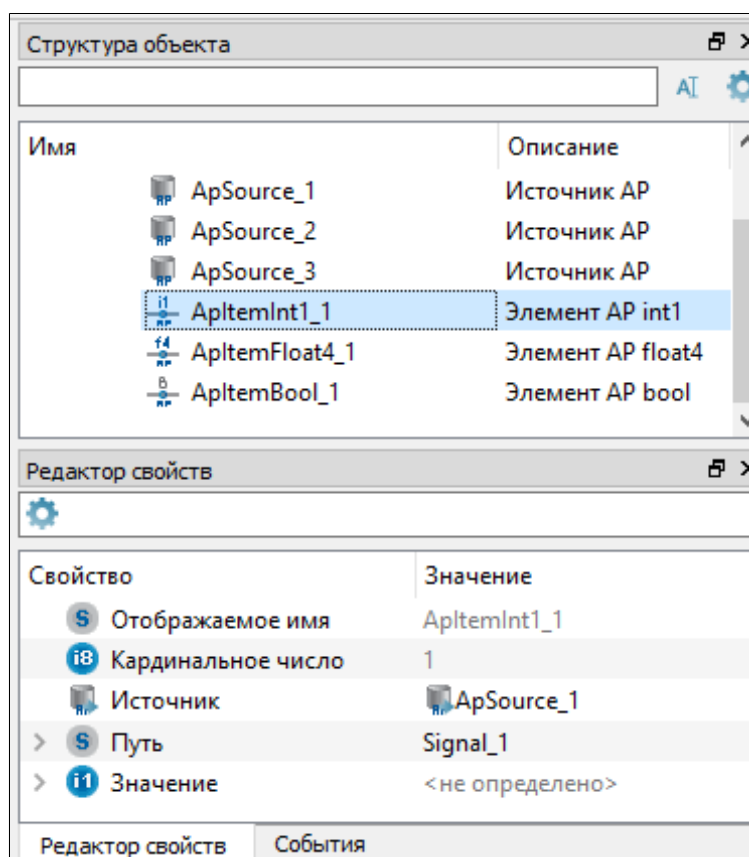
## 1.1.4.2.6.5. Работа с сигналами

Каждому типу сигнала на сервере соответствует отдельный компонент Элемент AP <T>, где <T> - тип сигнала на сервере.

Если тип сигнала на сервере не известен можно использовать универсальный компонент Элемент AP, который оперирует универсальным типом данных variant.

После определения типов сигналов добавьте нужное количество компонентов Элемент AP <T> на экранную форму (один сигнал - один элемент AP).

Компонент не визуальный и виден только в области Структура объекта.



### Свойства

Свойство	Описание
----------	----------



Источник	Ссылка на компонент Источник AP, к дереву которого относится данный Элемент AP
Путь	Путь до сигнала относительно корня Источника

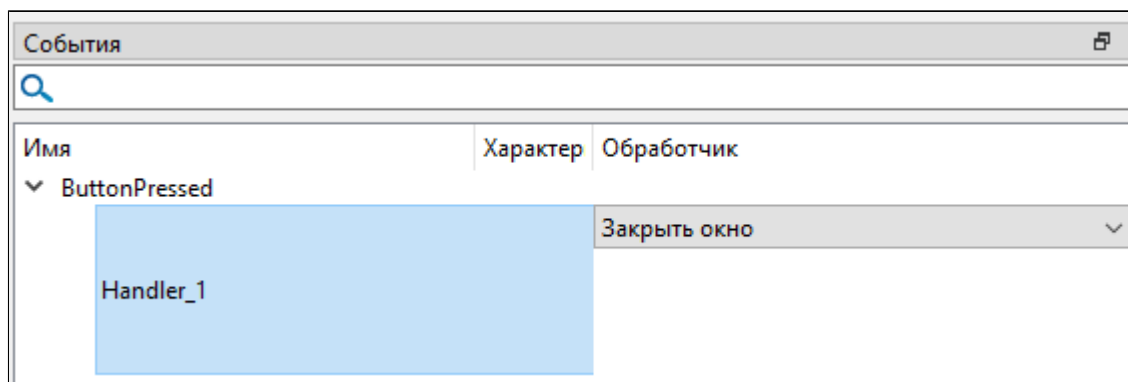
## 1.1.4.2.6.6. Обработчики событий

Чтобы реагировать на возникновение различных ситуаций используйте События. Чтобы приложение выполняло некоторую работу в ответ на некое событие, разработчик должен определить реакцию на возникновение события – обработчик события (например, обработчик на событие двойного клика мыши по объекту).

Любое событие может иметь параметры, которые его характеризуют. Все события выделенного объекта можно посмотреть в области События.

# Заккрытие окна

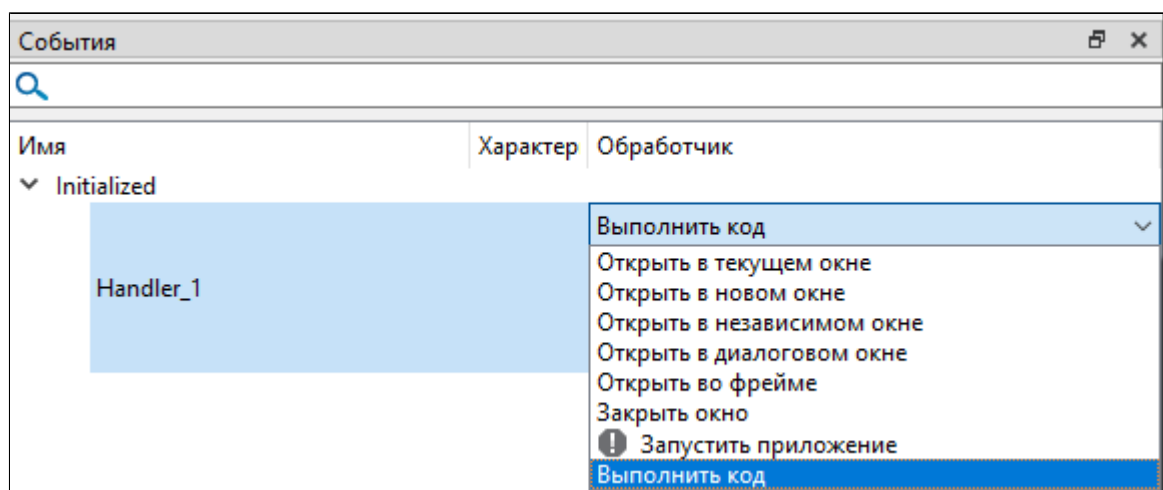
Тип обработчика "Закреть окно" применяется для закрытия текущей экранной формы.



# Событие при создании объекта

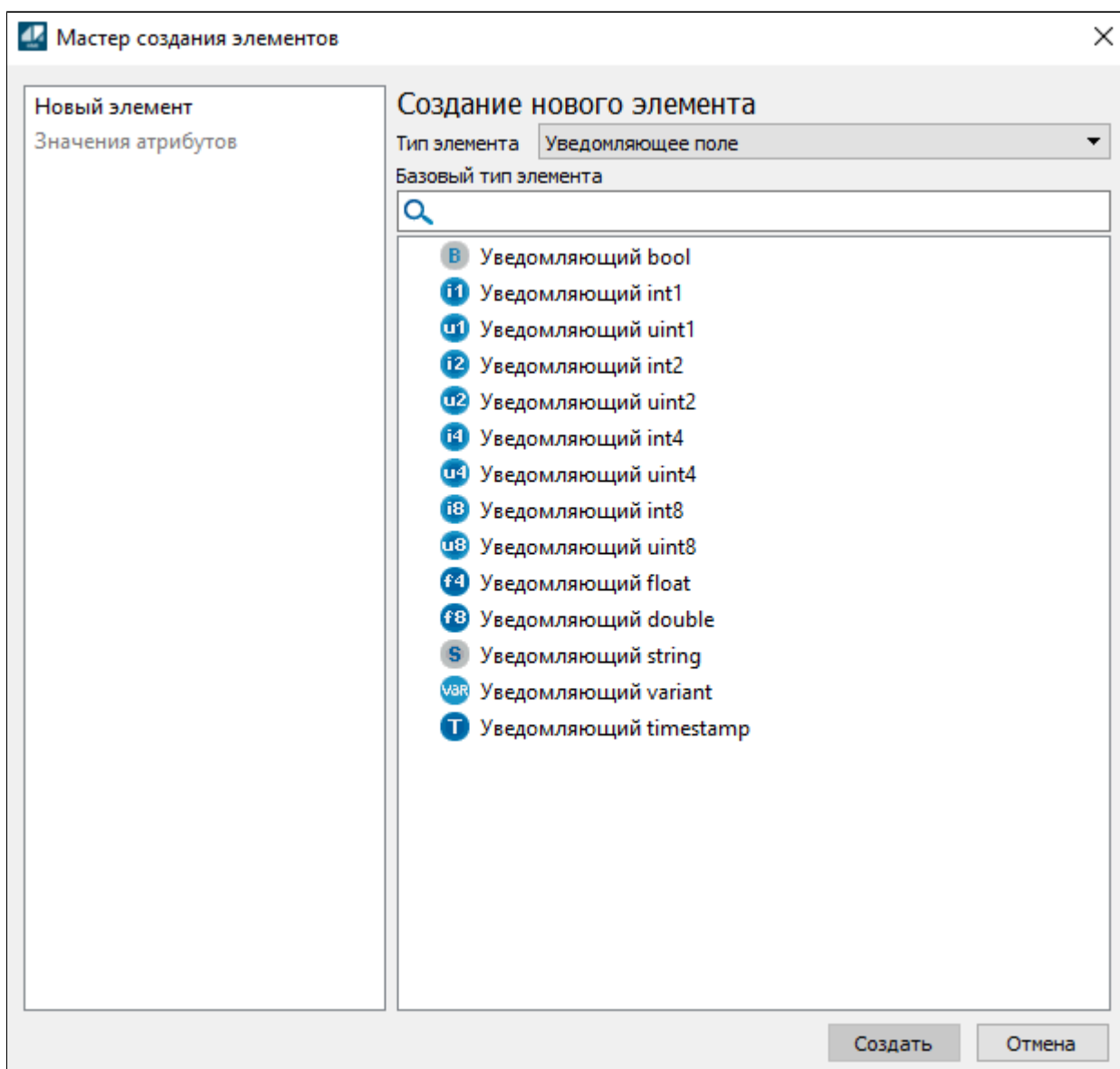
Чтобы определить события, которые будут происходить сразу после создания объекта или экранной формы в режиме исполнения, добавьте компонент ["Инициализатор"](#) в объект или на экранную форму. Компонент невидимый и виден только в области Структура объекта.

После добавления "Инициализатора", определите любой тип обработчика события.



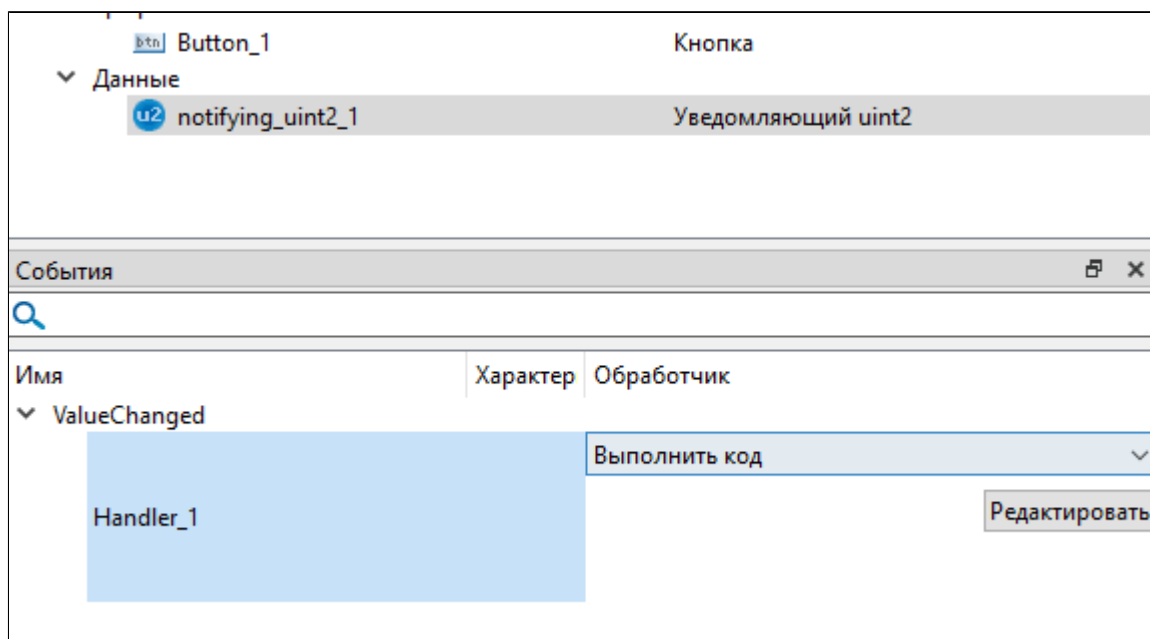
## 1.1.4.2.6.6.3. Событие при изменении свойства объекта

Чтобы каждое изменение свойства объекта приводило к возникновению события, применяйте уведомляющие поля. Уведомляющие поля удобны, когда требуется оперативная реакция на любое изменение важного параметра. Чтобы создать уведомляющее поле для объекта, выберите команду контекстного меню в "Структуре объекта" "Создать" и воспользуйтесь мастером создания элементов.



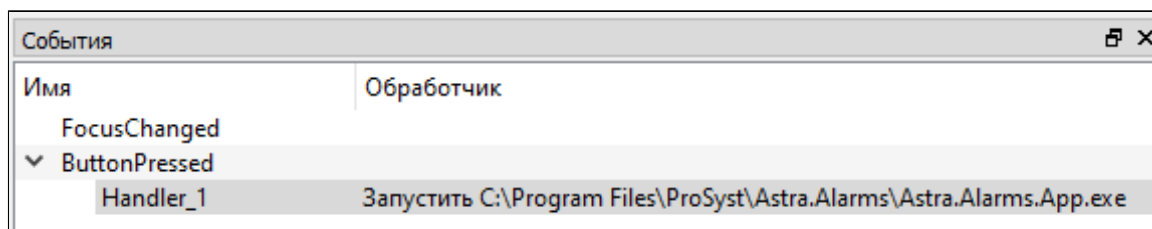
Уведомляющее поле выбранного типа добавится в "Структуру объекта" (группа Данные) и в список свойств объекта. Чтобы определить

событие, которое будет срабатывать каждый раз при изменении уведомляющего поля, создайте обработчик для события типа "ValueChanged".



# Запуск приложения

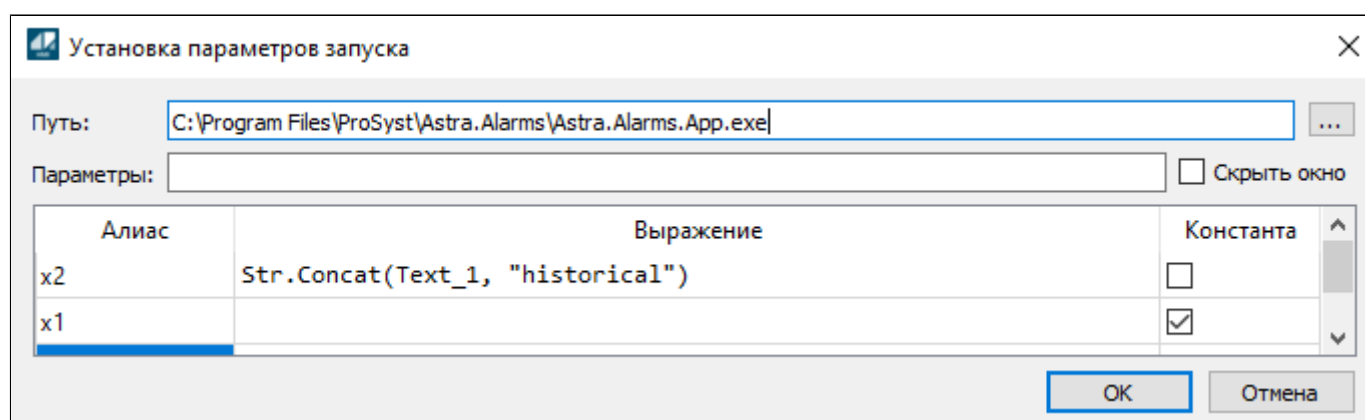
Тип обработчика события Запустить приложение применяется для запуска сторонних файлов формата \*.exe, \*.bat, \*.com.



Имя	Обработчик
FocusChanged	
▼ ButtonPressed	
Handler_1	Запустить C:\Program Files\ProSyst\Astra.Alarms\Astra.Alarms.App.exe

Чтобы настроить обработчик события, нажмите кнопку "...":

1. В окне Установка параметров запуска укажите полный путь до файла.
2. При запуске приложения с параметрами, заполните таблицу, расположенную ниже.
3. В столбце Выражения указываются параметры запуска приложения.
4. Флагом Константа отмечаются выражения, содержимое которых в неизменном виде должно попасть в параметры запуска приложения. Если в выражении используется формула или переменная, то флаг не устанавливается.
5. В поле Параметры укажите созданные переменные, в которых записаны параметры на запуск приложения с параметрами в фигурных скобках, несколько параметров разделяются пробелом.



Установка параметров запуска

Путь: C:\Program Files\ProSyst\Astra.Alarms\Astra.Alarms.App.exe

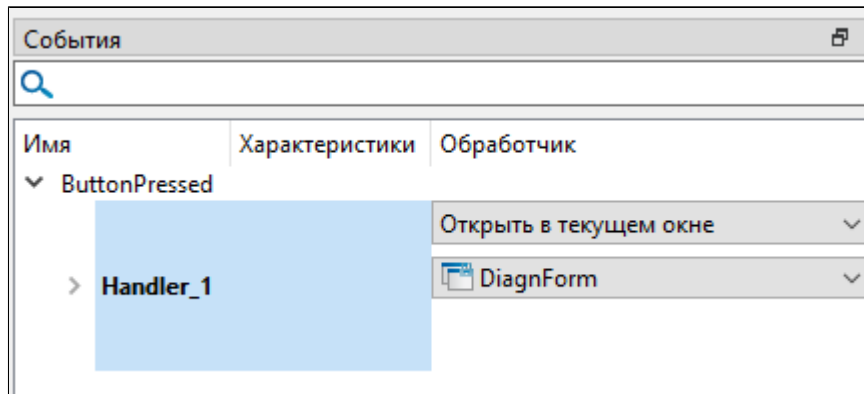
Параметры:   Скрыть окно

Алиас	Выражение	Константа
x2	Str.Concat(Text_1, "historical")	<input type="checkbox"/>
x1		<input checked="" type="checkbox"/>

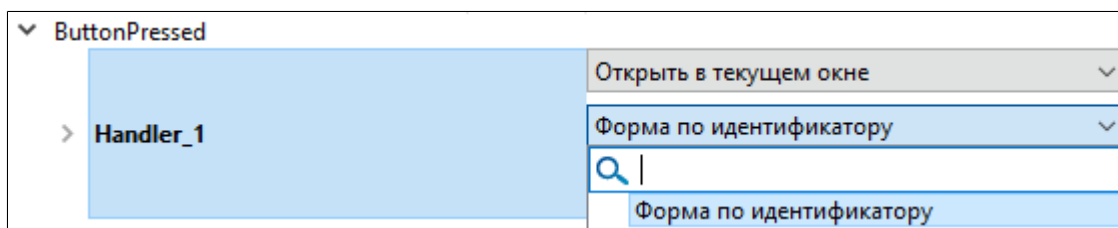
OK Отмена

## 1.1.4.2.6.6.5. Открыть в текущем окне

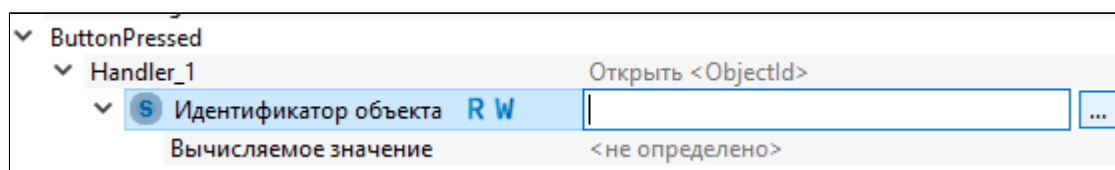
Тип обработчика "Открыть в текущем окне" применяется для открытия экранной формы в текущем окне.



Также есть возможность выбрать форму по идентификатору:



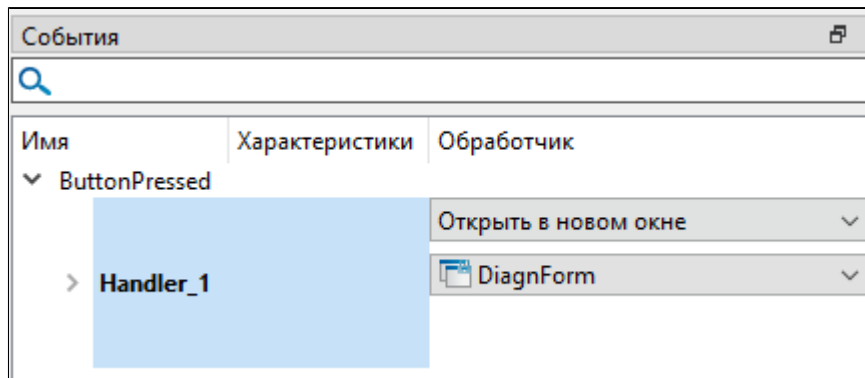
Для настройки идентификатора раскройте обработчик и задайте необходимое условие:



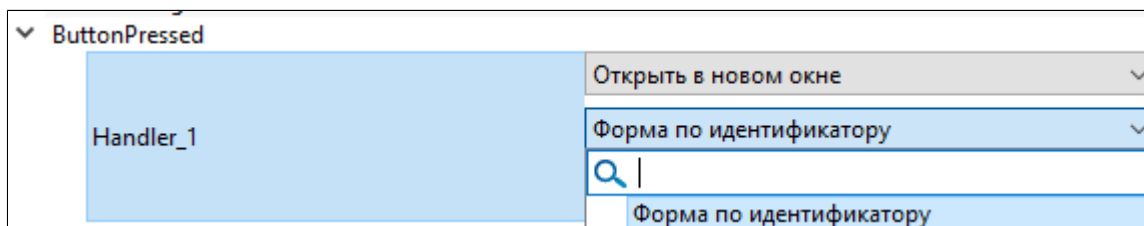


## 1.1.4.2.6.6.6. Открыть в новом окне

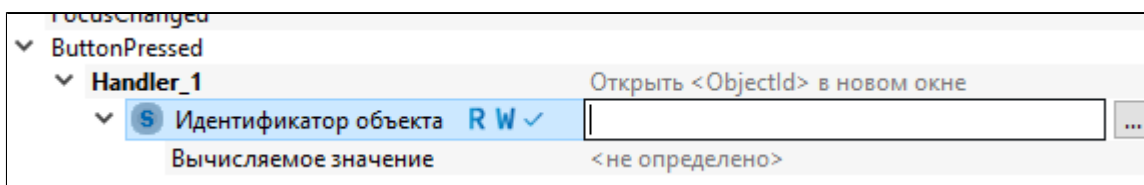
Тип обработчика "Открыть в новом окне" применяется для открытия экранной формы в новом окне. После открытия новой экранной формы, обе формы остаются активными, между ними можно переключаться.



Также есть возможность выбрать форму по идентификатору:

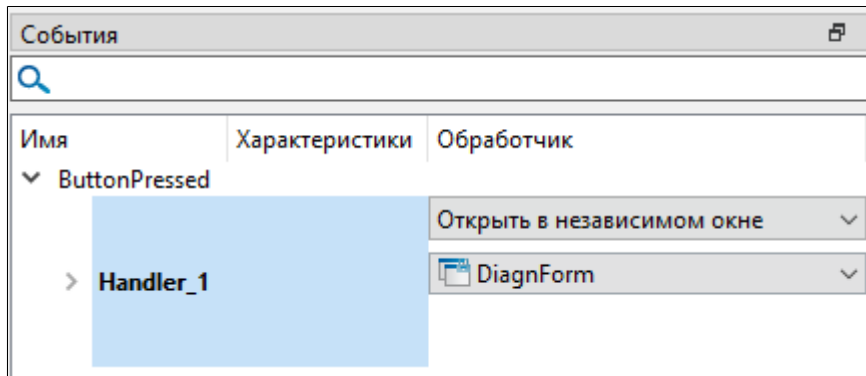


Для настройки идентификатора раскройте обработчик и задайте необходимое условие:

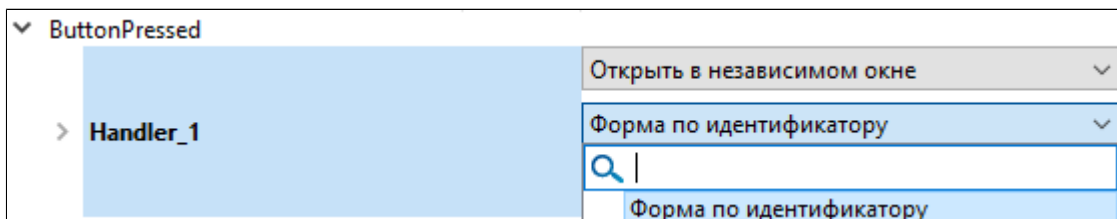


## 1.1.4.2.6.6.7. Открыть в независимом окне

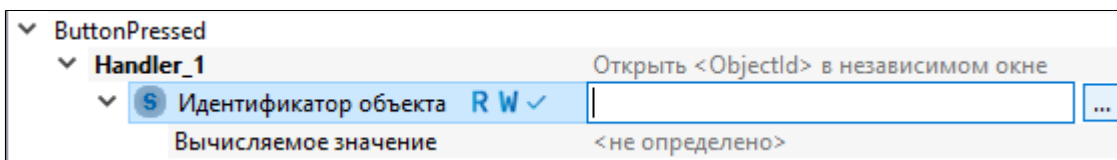
Тип обработчика "Открыть в независимом окне" применяется для открытия дочернего окна, независящего от родительского окна. После открытия независимой экранной формы, обе формы остаются активными, между ними можно переключаться.



Также есть возможность выбрать форму по идентификатору:

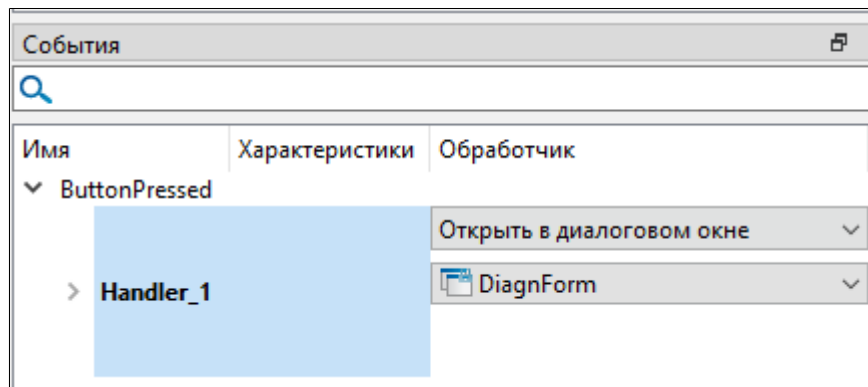


Для настройки идентификатора раскройте обработчик и задайте необходимое условие:

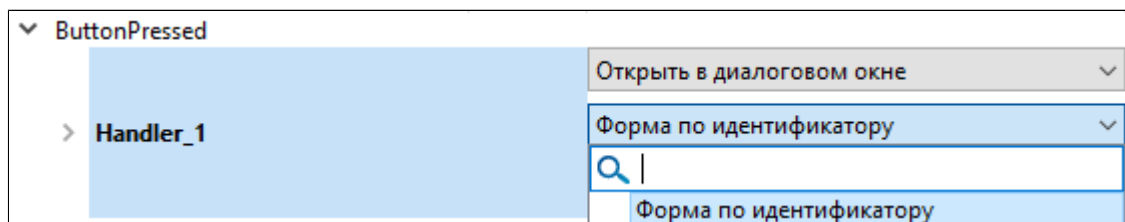


## 1.1.4.2.6.6.8. Открыть в диалоговом окне

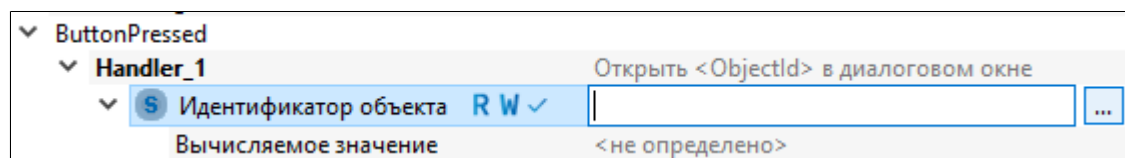
Тип обработчика "Открыть в диалоговом окне" применяется для открытия поверх основного окна другой экранной формы. Пока открыто диалоговое окно, у пользователя отсутствует возможность взаимодействовать с родительским окном.



Также есть возможность выбрать форму по идентификатору:



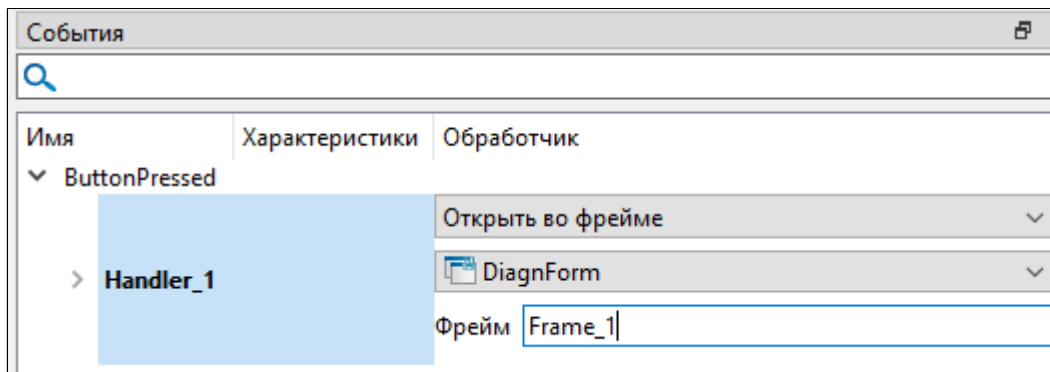
Для настройки идентификатора раскройте обработчик и задайте необходимое условие:



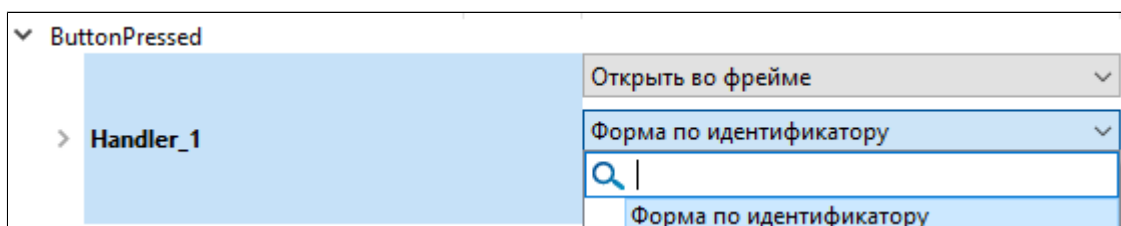
## 1.1.4.2.6.6.9. Открыть во фрейме

Тип обработчика "Открыть во фрейме" применяется для динамического открытия экранных форм внутри области фрейма.

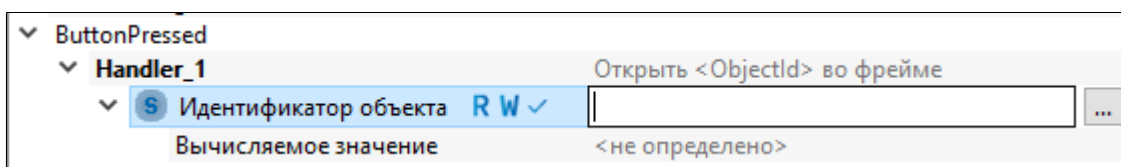
В обработчике укажите экранную форму и имя фрейма, в котором откроется выбранная форма.



Также есть возможность выбрать форму по идентификатору:



Для настройки идентификатора раскройте обработчик и задайте необходимое условие:



# Функции

Функции – вызываемые процедуры, позволяющие влиять на состояние объектов или их свойств. Как правило, у функций есть набор входных параметров. К примеру, у объекта стандартной библиотеки компонентов Выпадающий список есть предустановленная функция `RemoveItem(Index)`, предназначенная для удаления элемента списка.

В Astra.HMI можно определять свои собственные функции для любых объектов, используя язык Astra.Om или JavaScript. Любая созданная функция может возвращать значение, обращаться к входным аргументам в функции, а также вызывать другие функции.

## Предустановленные функции

Некоторые объекты, созданные с помощью стандартной библиотеки компонентов Astra.HMI, имеют предустановленные функции. К примеру, выпадающий список, добавленный на экранную форму, будет иметь набор предустановленных функций для работы с ним:

- › очистка списка – `Clear()`;
- › добавление элемента в список – `AddItem(Text)`;
- › удаление элемента из списка – `RemoveItem(Index)`;
- › модификация элемента списка – `SetItem(ItemIndex,Text)` и т.д.

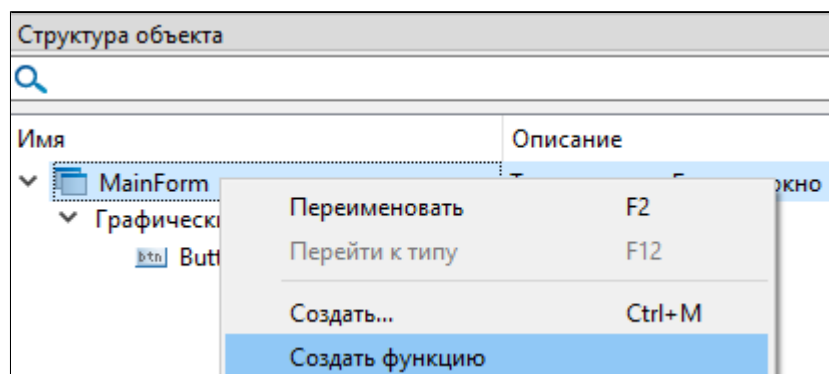
## Пользовательские функции

В Astra.HMI можно определять свои собственные функции для любых объектов, используя языки Astra.Om или JavaScript.

Любая созданная функция может возвращать значение, обращаться к входным аргументам в функции, а также вызывать другие функции.

Чтобы создать собственную функцию для какого-либо элемента, вызовите контекстное меню в области Структура объекта и выберите Создать функцию.

Собственные функции появляются в разделе Функции области Структура объекта.



В окне Редактор функции определите имена входных аргументов и их типы, а также укажите тип действия, которое будет выполнять функция (типы действий аналогичны обработчикам событий). При необходимости, можно сменить тип возвращаемого значения или оставить его пустым (void).

# Вызов внешних функций

Чтобы вызывать функции, определенные в сторонних динамических библиотеках (DLL-файлах), используйте тип действия Вызывать внешнюю функцию в Редакторе функций.

**fx Редактор функции**

Тип возвращаемого значения : **S string**

Имя аргумента	Ссылка	Тип аргумента
x	<input type="checkbox"/>	float8
y	<input type="checkbox"/>	float8
r	<input type="checkbox"/>	float8

Действие : **Вызывать внешнюю функцию**

Имя модуля :

Имя функции :

Тип вызова : **CDed**

Кодировка : **Utf8**

Опции маршallingа :

**OK**

## Параметры вызова внешней функции

Параметр	Описание
Имя модуля	Абсолютный путь до DLL-файла. Если DLL- файл расположен в корне каталога с установленным Astra.HMI, то достаточно указать его имя без пути
Имя функции	Имя вызываемой функции, которая определена в файле динамической библиотеки
Тип вызова	Способ вызова функции CDecl или StdCall (Win API). Зависит от спецификации вызываемой внешней функции

Кодировка	Кодировка символов: <ul style="list-style-type: none"> <li>&gt; UTF8</li> <li>&gt; UTF16</li> <li>&gt; ANSI</li> </ul>
Опции маршаллинга	Настройки направления работы аргументов (входной, выходной или с применением буфера). Все аргументы по умолчанию - входные. Если аргумент выходной, то это нужно указать в опциях маршаллинга: ["out"], если входной/выходной - ["in", "out"]. Если выходной аргумент типа string, то опции маршаллинга: ["out", "buffer"].

## Пример

Разберем детально процесс вызова внешней функции ExtLibCalcNoErrorDesc, которая определена в файле extlib.dll.

Ниже показан фрагмент исходного кода динамической библиотеки:



```
std::int32_t __stdcall ExtLibCalcNoErrorDesc(
    wchar_t const *inputString,
    double a1,
    double x1,
    double a2,
    double x2,
    double *y1,
    double *y2)
{
    try
    {
        //Проверяем входные параметры
        If( !inputString ) throw Error( L"Не указана входная строка" );
        // Считаем выражения
        *y1 = a1 * x1 + a2 * x2;
        *y2 = ( 1 / a1 ) * x1 - ( 1 / a2 ) * x2;
```



```
    // Всё прошло успешно, вернуть 1
    return 1;
}
// Ловим ошибку и возвращаем 0
catch( Error const &e )
{
    return 0;
}
}
```

Создайте новую функцию ExtLibCalcNoErrorDesc и настройте параметры:

- Тип возвращаемого значения выберите в соответствии с типом возвращаемого значения самой функции - это int4;
- Перечень входных аргументов настройте в соответствии с типами входных аргументов самой функции: первый аргумент строковый, все остальные - тип float8;
- Два последних аргумента функции являются ссылочными - отметьте их соответствующими флагами в колонке Ссылка;
- Выберите в выпадающем списке действие Вызывать внешнюю функцию;
- Укажите Имя модуля - название файла, из которого будет вызываться внешняя функция - extlib.dll;
- Укажите Имя функции, которую требуется вызывать - ExtLibCalcNoErrorDesc;
- Укажите Тип вызова. Для вызываемой функции - StdCall(Win API);
- Укажите кодировку - Utf16;
- Укажите Опции маршалинга в формате: {"имя параметра": ["in","out","buffer"]}

*fx* Редактор функции

Тип возвращаемого значения : i4 int4

Имя аргумента	Ссылка	Тип аргумента
1	<input type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">S</span> string
2	<input type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">f8</span> float8
3	<input type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">f8</span> float8
4	<input type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">f8</span> float8
5	<input type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">f8</span> float8
6	<input checked="" type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">f8</span> float8
7	<input checked="" type="checkbox"/>	<span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px;">f8</span> float8

Действие : Вызывать внешнюю функцию

Имя модуля :

Имя функции :

Тип вызова : StdCall(Win API)

Кодировка : Utf16

Опции маршallingа :

OK

Ниже показан участок кода, вызывающий внешнюю функцию из скриптов:



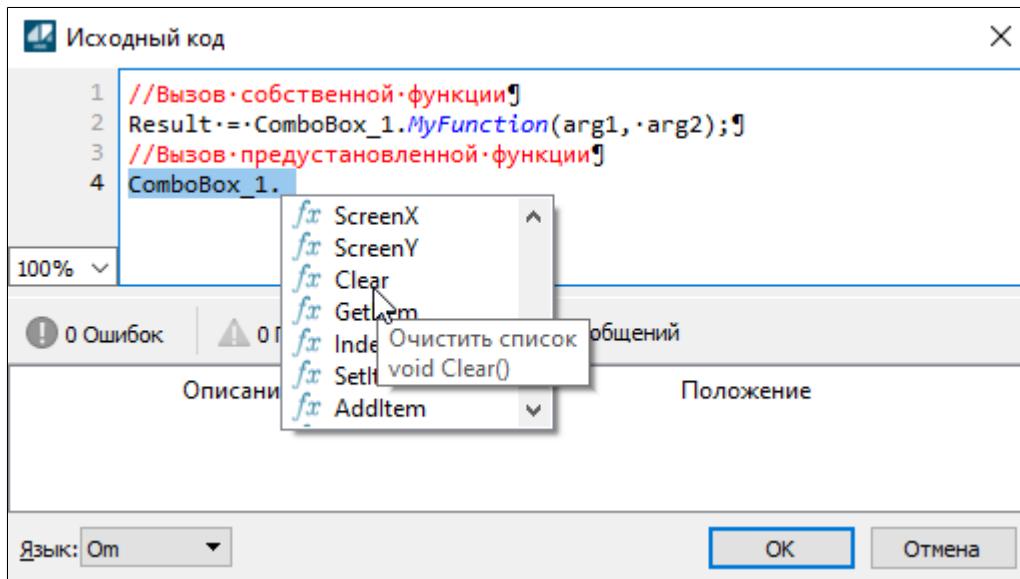
```
//Готовим входные параметры для внешней функции
inp_str: string = "входная строка";
a1: double = 0.5;
a2: double = 0.75;
x1: double = 0.25;
x2: double = 0.01;
_y1: double = 0;
_y2: double = 0;
//Вызвать внешнюю функцию и присвоить результат выполнения
переменной Result
Result: ExtLibCalcNoErrorDesc (inp_str, a1, x1, a2, x2, &_y1, &_y2);
//Обработка результатов выполнения функций
if(Result == 0) TextField.Text = "Произошла ошибка";
```



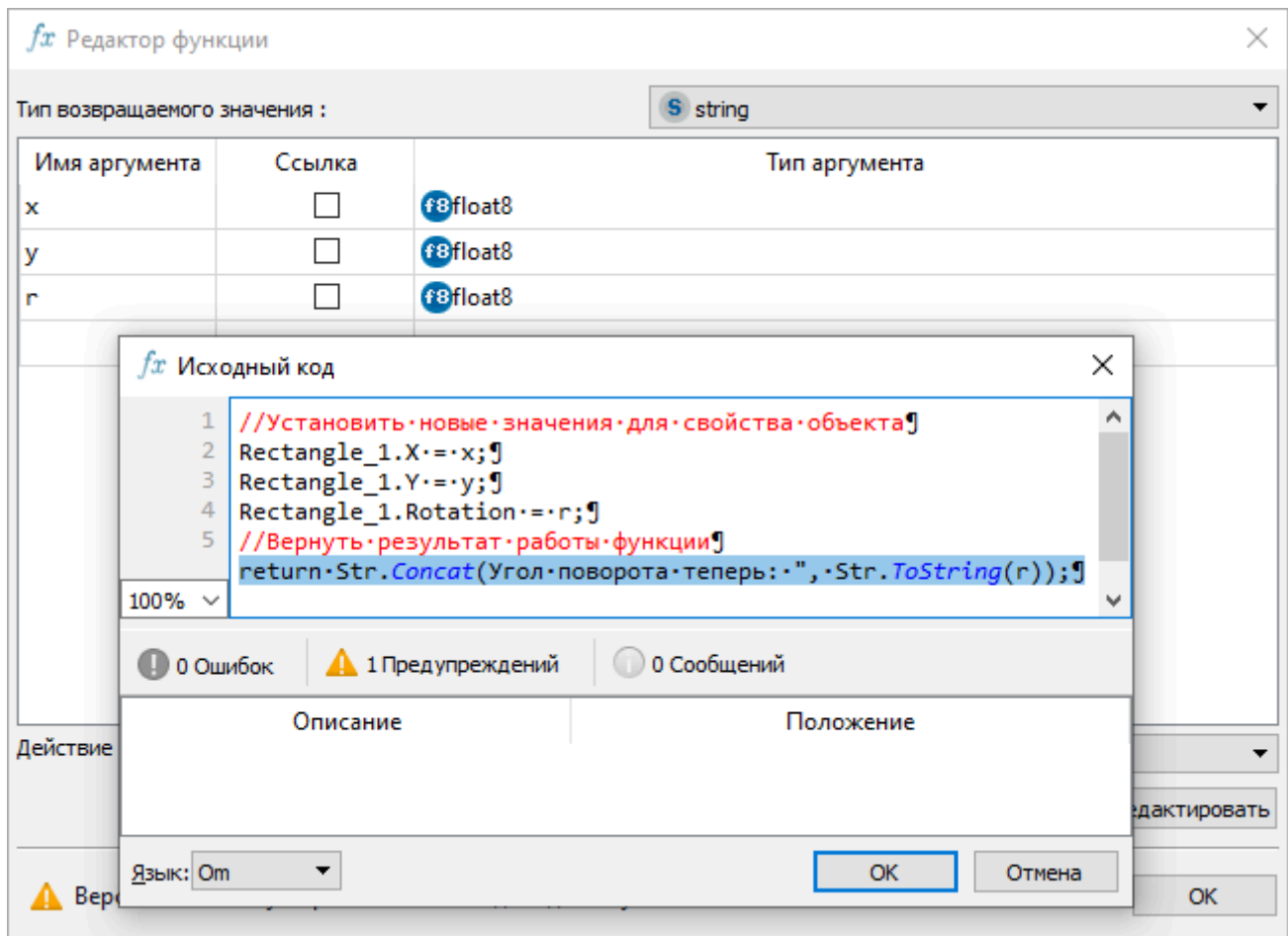
Передача параметров по ссылке выполняется через символ &.

# Вызов функций и возврат значений

Чтобы обращаться к функциям объекта из скриптов и формул, используйте точку в качестве разделителя между именем объекта и именем функции. Возможные варианты функций объекта можно выбирать из выпадающего списка, который появляется после ввода точки. Просмотреть назначение функции и входные параметры можно во всплывающей подсказке.

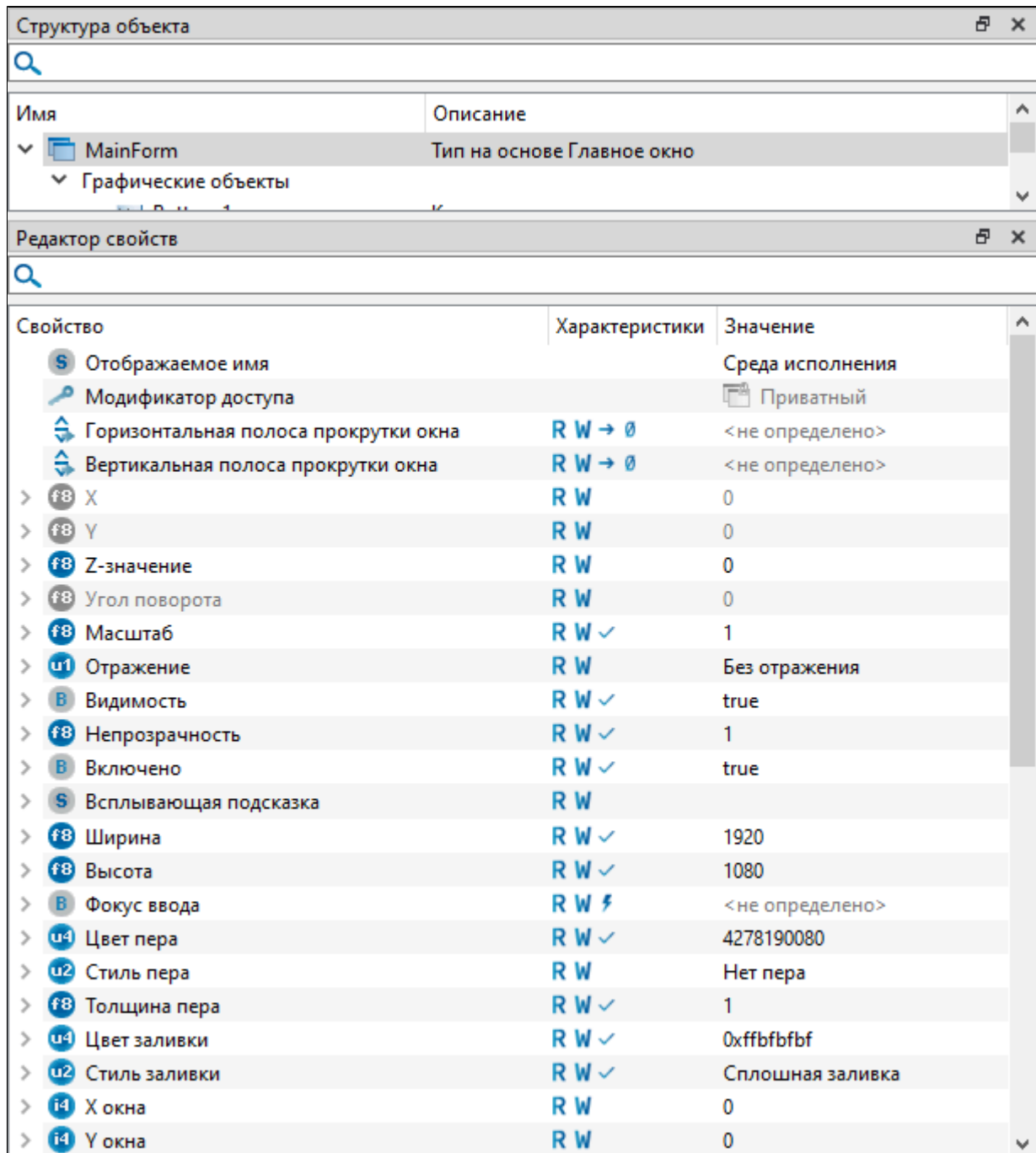


Если функция возвращает какой-либо результат, то для возврата значения используйте оператор `return`. В примере ниже показана функция, которая изменяет свойства прямоугольника (ширину, высоту и угол поворота), а результат работы функции - строка Угол поворота теперь: N.



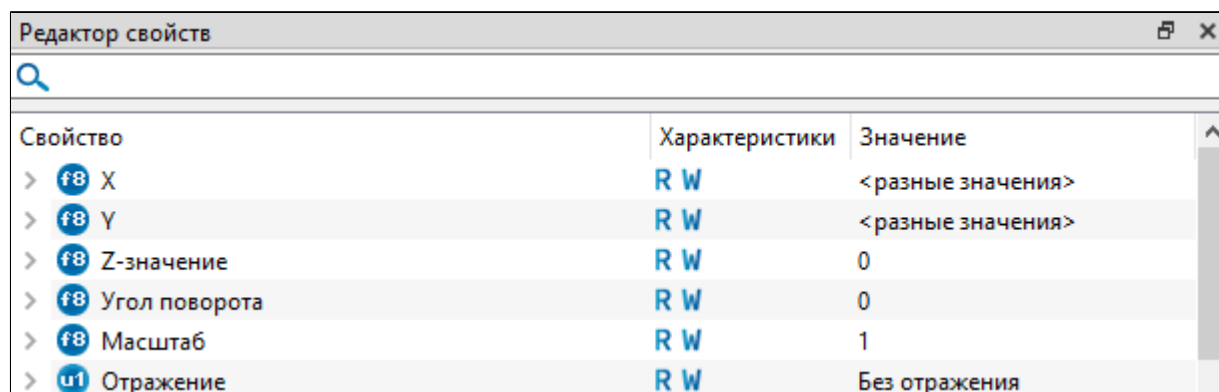
## 1.1.4.2.6.8. Свойства

Свойства – это атрибуты, которые определяют особенности поведения или отображения объекта на экране. Свойства выделенного объекта отображаются в области Редактор свойств.



Чтобы массово поменять схожие свойства у множества объектов, выделите их на рабочей области или в области Структура объектов. После выделения нескольких объектов в области Редактор свойств останутся лишь схожие

свойства. Различающиеся значения свойств отмечены фразой <разные значения>.

















The screenshot shows a window titled 'Редактор свойств' (Properties Editor) with a search bar and a table of properties. The table has three columns: 'Свойство' (Property), 'Характеристики' (Characteristics), and 'Значение' (Value). The properties listed are X, Y, Z-значение, Угол поворота, Масштаб, and Отражение. The values for X and Y are '<разные значения>', while Z-значение, Угол поворота, and Масштаб have numerical values (0, 0, and 1 respectively). Отражение has the value 'Без отражения'.

Свойство	Характеристики	Значение
> f8 X	R W	<разные значения>
> f8 Y	R W	<разные значения>
> f8 Z-значение	R W	0
> f8 Угол поворота	R W	0
> f8 Масштаб	R W	1
> u1 Отражение	R W	Без отражения

Некоторые свойства объектов нередактируемые (только для чтения). К таким свойствам можно обращаться в обработчиках событий или функциях. Если попытаться скомпилировать проект, где есть попытка установить значения нередактируемым свойствам, то компилятор выдаст ошибку.

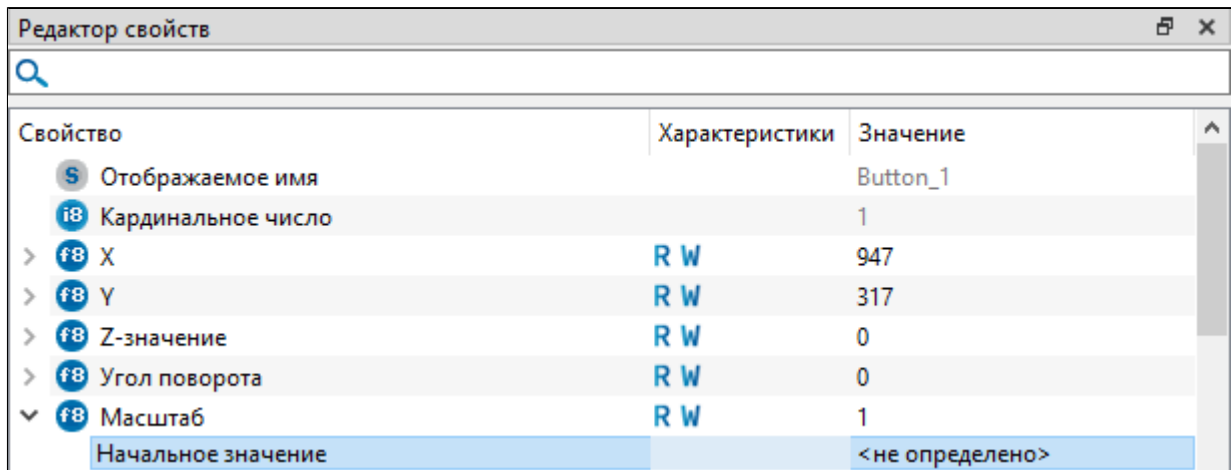
## Типы данных свойств

Тип	Иконка	Описание и диапазон значений
bool		Логическое значение. Принимает значения True или False
int1		Знаковое целое размером 1 байт. Принимает значения от -128 до 127
uint1		Беззнаковое целое размером 1 байт. Принимает значения от 0 до 255
int2		Знаковое целое размером 2 байта. Принимает значения от -32 768 до 32 767
uint2		Беззнаковое целое размером 2 байта. Принимает значения от 0 до 65535
int4		Знаковое целое размером 4 байта. Принимает значения от -2 147 483 648 до 2 147 483 647
uint4		Беззнаковое целое размером 4 байта. Принимает значения от 0 до 4 294 967 295
int8		Знаковое целое размером 8 байт. Принимает значения от $-9 \cdot 10^{18}$ до $9 \cdot 10^{18}$
uint8		Беззнаковое целое размером 8 байт. Принимает значения от 0 до 18 446 744 073 709 551 615
float		Значение с плавающей запятой размером 4 байта
double		Значение с плавающей запятой размером 8 байт
string		Строковое значение
timestamp		Дата, время
variant		Универсальный тип данных. Свойство этого типа может хранить любые значения. Используется, когда заранее не известен род данных, с которыми предстоит работать.



## 1.1.4.2.6.8.1. Начальные значения свойств

Начальные значения свойств объектов применяются в момент инициализации экранной формы в рантайме. Чтобы указать начальное значение свойства для объекта, введите параметр в поле Начальное значение.

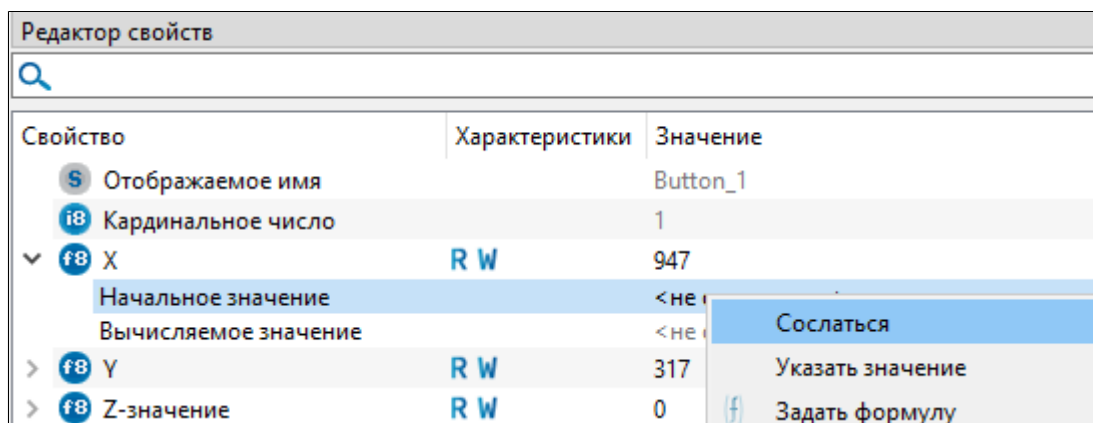


Свойство	Характеристики	Значение
Отображаемое имя		Button_1
Кардинальное число		1
>  X	R W	947
>  Y	R W	317
>  Z-значение	R W	0
>  Угол поворота	R W	0
▼  Масштаб	R W	1
Начальное значение		<не определено>

Если Начальное значение для свойства не указано, то для целей инициализации в рантайме берется значение свойства в режиме проектирования.

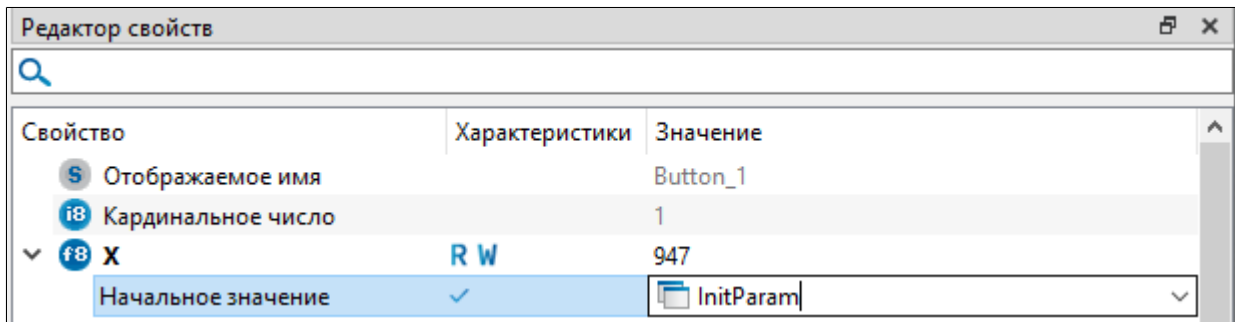
### Сослаться на значение

Начальное значение свойства может ссылаться на значение ранее инициализированного параметра. Для этого в контекстном меню выберите команду Сослаться и укажите нужный параметр инициализации.



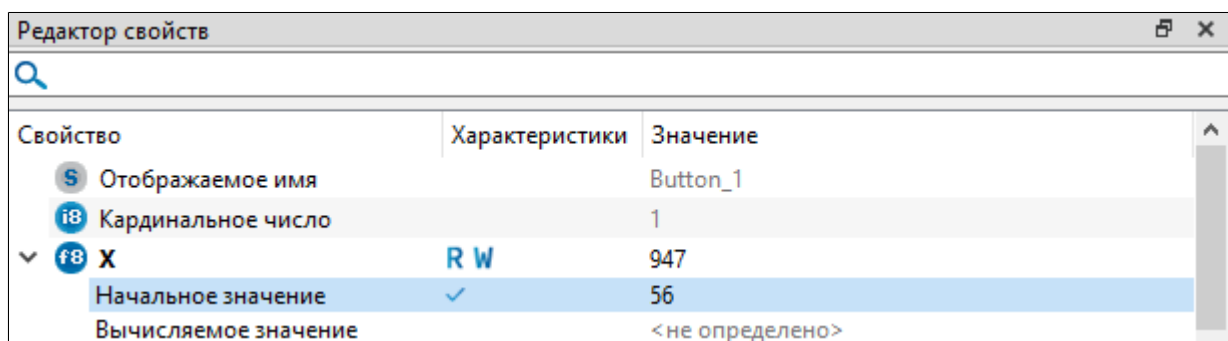
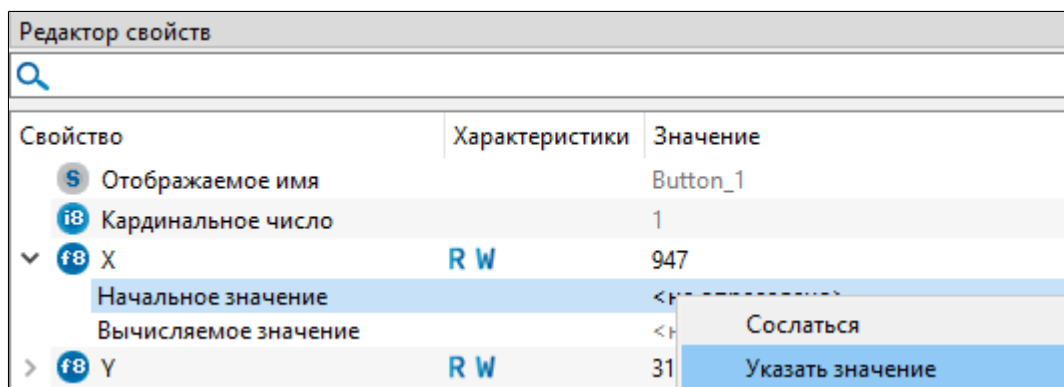
Свойство	Характеристики	Значение
Отображаемое имя		Button_1
Кардинальное число		1
▼  X	R W	947
Начальное значение		<не определено>
Вычисляемое значение		<не определено>
>  Y	R W	317
>  Z-значение	R W	0

- Сослаться
- Указать значение
- Задать формулу



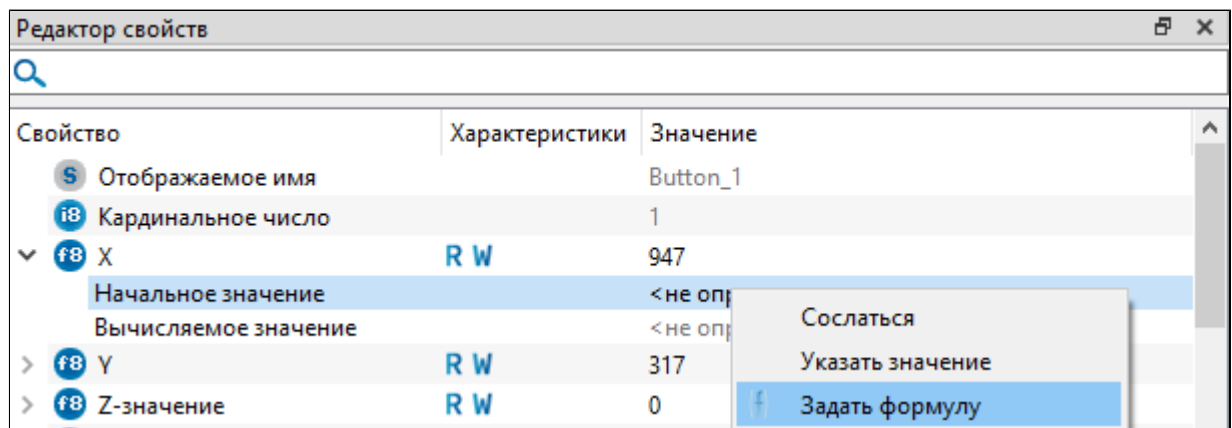
## Указать значение

Для указания начального значения свойства в контекстном меню выберите команду Указать значение и введите нужное значение.



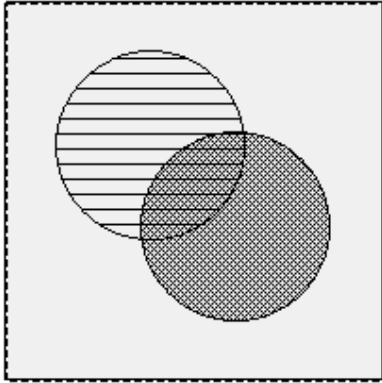
## Задать формулу

Начальное значение свойства может быть указано в виде формулы. Формула может состоять из одного или нескольких параметров инициализации объектов и/или строк. Чтобы начальное значение задать в виде формулы, в контекстном меню выберите команду Задать формулу и укажите нужную формулу.



В примере ниже показано, как в заголовок окна формы выводится формула, состоящая из параметра инициализации типа string и строкового значения  
Заголовок окна:

**i** string\_1 + " - Заголовок окна"



Свойство	Значение
▼ <b>S</b> Заголовок окна	MainForm
Начальное значение	<code>string_1 + " - Заголовок окна"</code>
Вычисляемое значение	<не определено>
> <b>B</b> Показывать заголовок	true
> <b>B</b> Показывать кнопку "свернуть"	true
> <b>B</b> Показывать кнопку "развернуть"	true
> <b>B</b> Показывать кнопку "закрыть"	true
> <b>B</b> Поверх всех окон	false
> <b>u4</b> Размеры окна	Вручную
> <b>u4</b> Стиль рамки окна	Изменяемый размер
> <b>u4</b> Состояние окна	По умолчанию
> <b>u4</b> Режим масштабирования	Не масштабировать
> <b>i4</b> Монитор	0
> <b>u4</b> Положение окна	Вручную
> <b>u2</b> Режим обработки закрытия окна	Закрыть
<b>S</b> string_1	Window caption

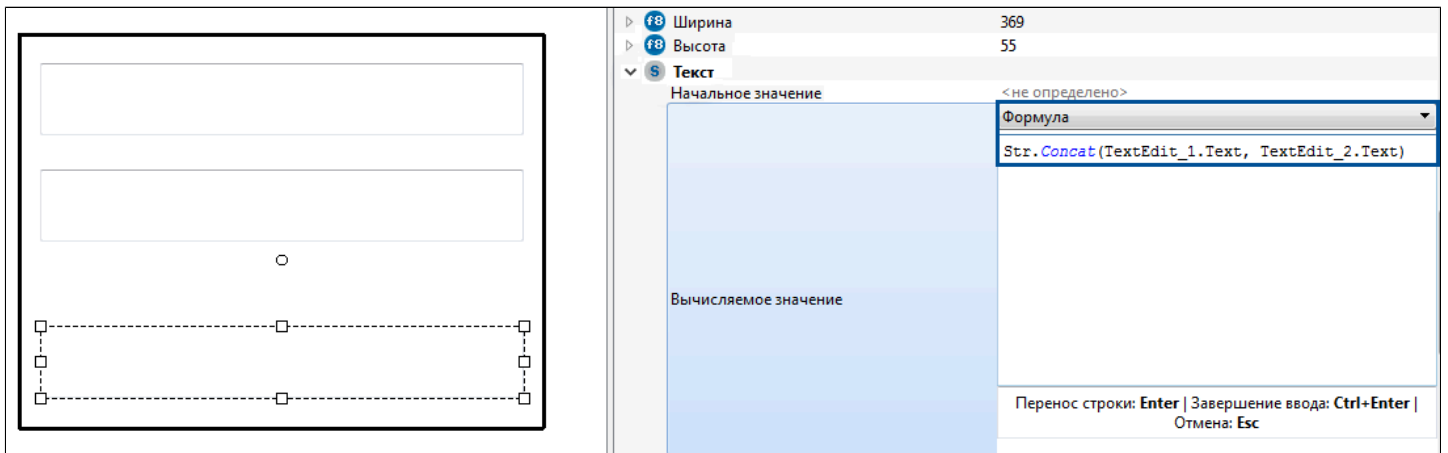
## 1.1.4.2.6.8.2. Вычисляемые значения свойств

Значения свойств могут вычисляться в процессе работы приложения по формуле, при наступлении определенных условий.

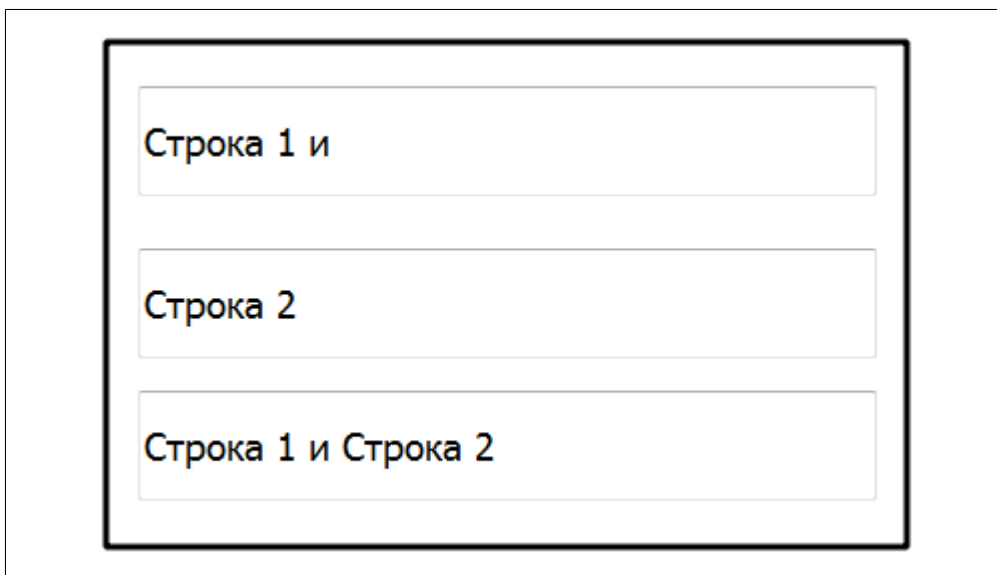
### Формула

Значения свойств могут динамически вычисляться в рантайме по логике, заданной в формуле - выражении на языке Astra.Оm.

В формулу могут передаваться значения свойств собственных объектов формы, а также объектов других форм. В Astra.HMI возможна передача значений только из главной формы в дочерние.



По логике, заданной в формуле, текст нижнего поля будет отображать результат слияния текста из верхних полей. Результат будет пересчитываться динамически при изменении любого аргумента формулы.

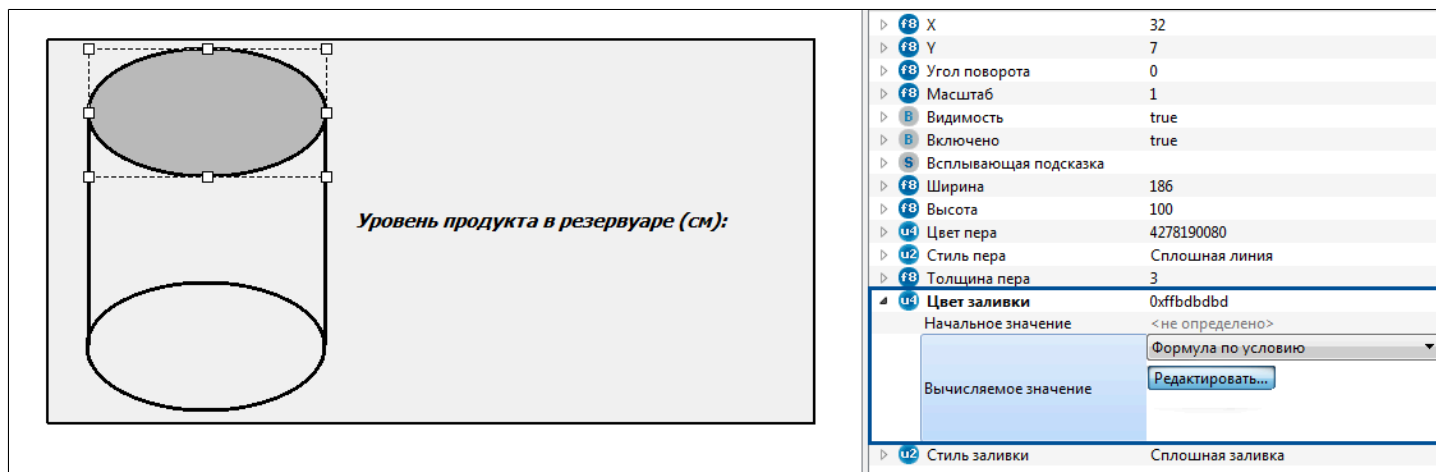


Для подписки на свойства объектов других форм, обращайтесь к объектам через ссылку на родительскую форму. На рисунке ниже показано обращение к значению элемента Number, который принадлежит родительской форме, через ссылку на родительскую форму Main\_Form\_1.







Текст	****
Начальное значение	<не определено>
Вычисляемое значение	Str.ToString(Main_Form_1.SW_1.Number.Value)

## Формула по условию

Формулы по условию применяются для выставления значений свойствам в рантайме в зависимости от выполнения некоторых условий. На рисунках ниже показан резервуар, верхушка которого окрашивается в определенный цвет в зависимости от уровня нефти. Для установки формулы по условию, выберите для свойства Вычисляемое значение типа Формула по условию и нажмите кнопку Редактировать.



В появившемся окне Редактор формул по условию установите нужные условия и результирующее значение при выполнении каждого условия. По умолчанию содержит значение, которое будет выставляться, если ни одно условие не было удовлетворено.

Условие	Значение
Level >= 0 && Level <= 100	0xffcb8733 
Level > 100 && Level <= 200	0xffd8da33 
Level > 200 && Level <= 300	0xff2b37e6 
Level > 300 && Level <= 350	0xffb80b1c 
Нет условия	Нет действия 
По умолчанию	0xffd0d0d0 

После запуска в рантайме, заливка эллипса будет динамически меняться в зависимости от уровня продукта в резервуаре (значение Level).



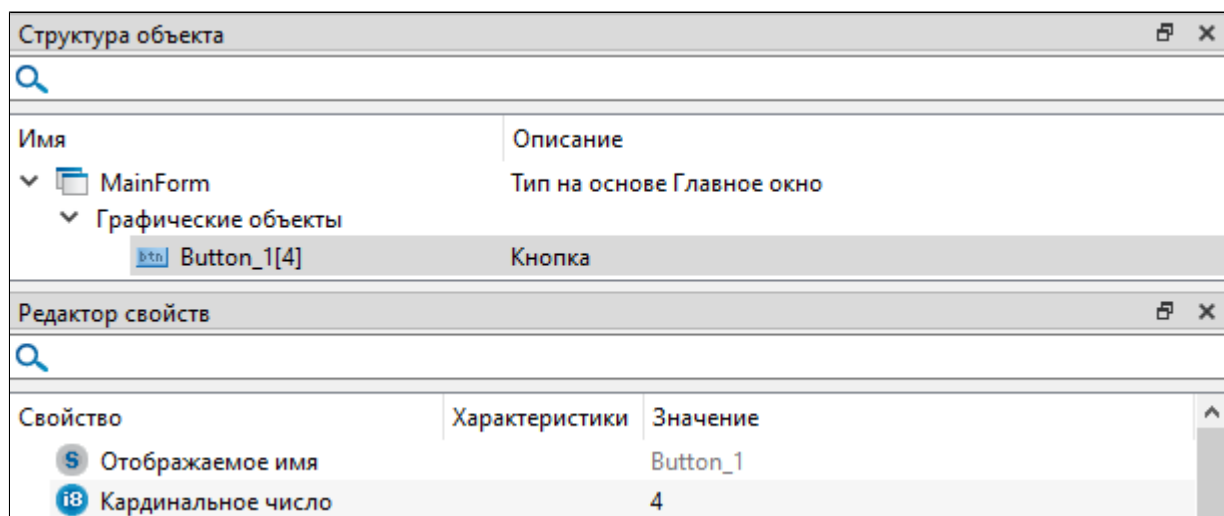
Чтобы быстро заполнять таблицу редактора формул по условию, используйте горячие клавиши:

- Ctrl+C (скопировать строку);
- Ctrl+V (вставить строку);
- Ctrl+Shift+(+) (вставить строку со сдвигом).

# Массивы

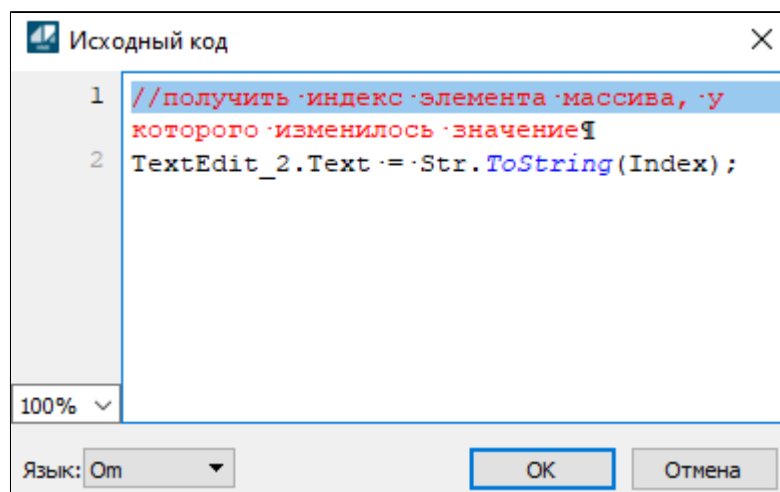
Любой объект в Astra.HMI можно преобразовать в массив объектов. Массивы позволяют хранить несколько объектов внутри одного объекта, добавленного на форму. Все элементы массива имеют один тип значений.

Чтобы преобразовать объект в массив, укажите размерность массива в свойстве Кардинальное число объекта.



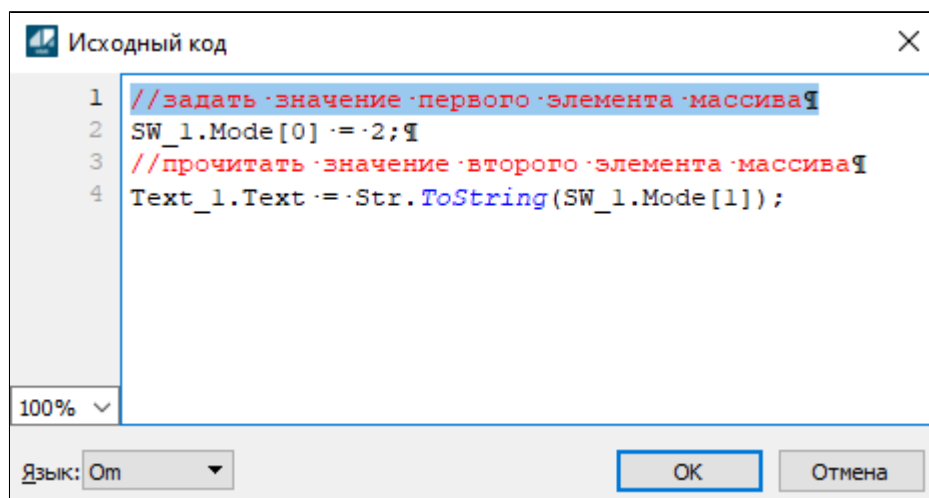
Каждый элемент в массиве имеет свой индекс. Нумерация элементов массива начинается с 0.

Чтобы получить индекс элемента, у которого сработало какое-то событие, используйте свойство Index объекта-массива.





Чтобы обращаться к элементам массива, указывайте индекс элемента в квадратных скобках после имени объекта-массива.



```
1 //задать значение первого элемента массива
2 SW_1.Mode[0] = 2;
3 //прочитать значение второго элемента массива
4 Text_1.Text = Str.ToString(SW_1.Mode[1]);
```

100% ▾

Язык: C# ▾

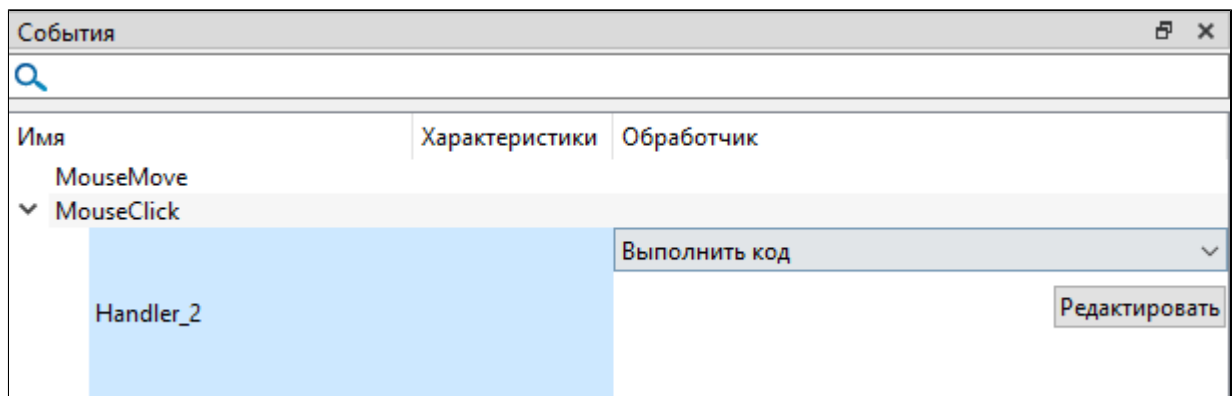
OK Отмена

## 1.1.4.2.6.10. Скрипты

В Astra.HMI доступны два языка для выполнения пользовательских скриптов:

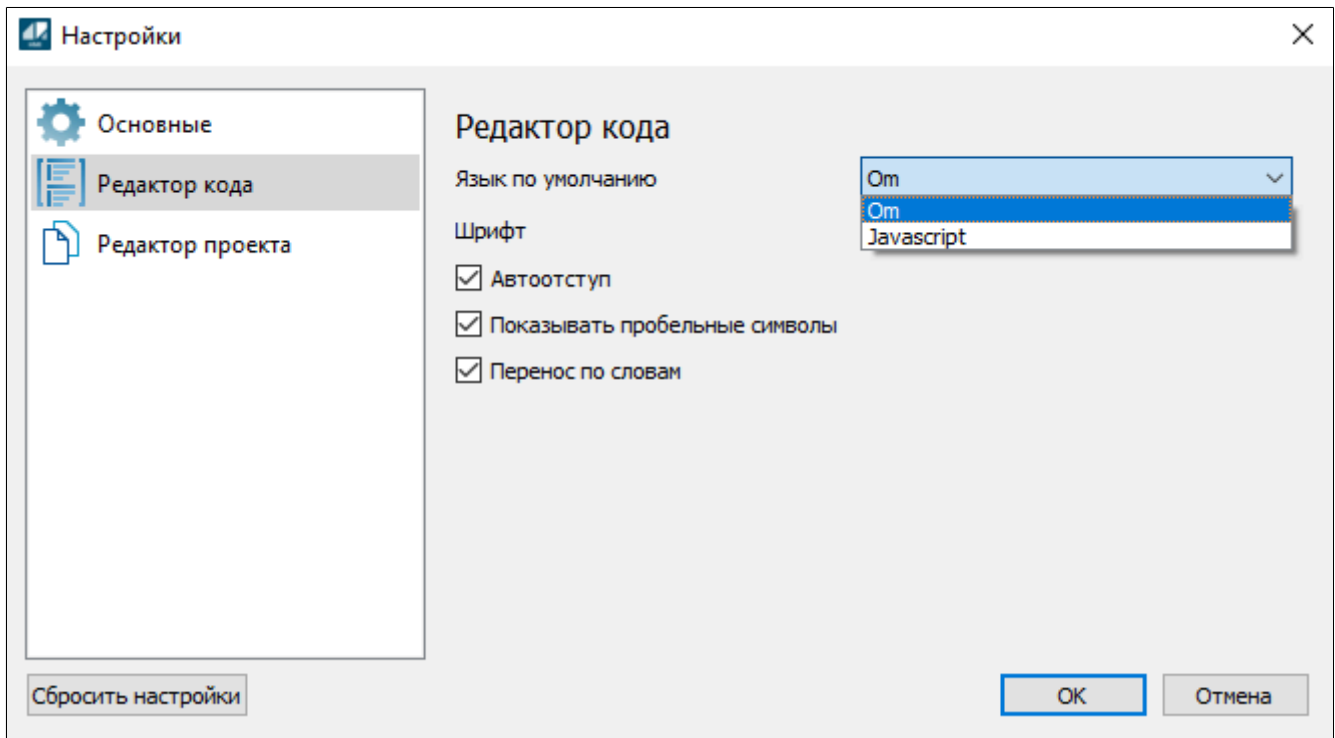
- › Astra.Om
- › JavaScript.

Чтобы написать скрипт, выполняемый при наступлении какого-либо события, используйте обработчик типа Выполнить код. Для открытия редактора кода нажмите кнопку Редактировать.

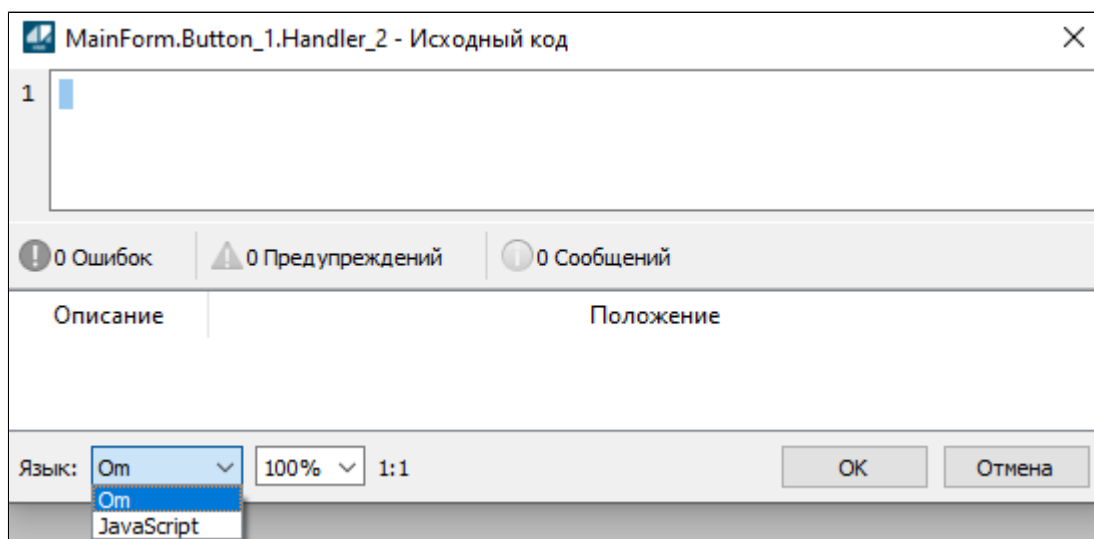


### Выбор языка

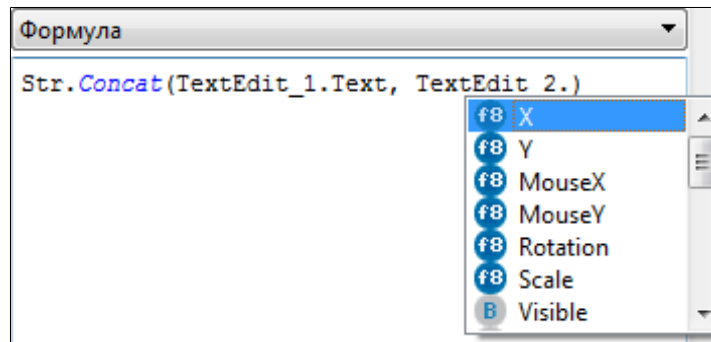
Чтобы выбрать язык, по умолчанию используемый в каждом редакторе исходного кода, перейдите в меню Файл, выберите Настройки и укажите язык на вкладке Редактор кода.



Чтобы выбрать язык для конкретного обработчика, используйте выпадающий список окна Исходный код.



В процессе написания скриптов или формул возможные варианты свойств объекта можно выбирать из выпадающего списка, который появляется после ввода точки.



При написании кода используйте горячие клавиши Tab, Shift+Tab для ввода табуляции.

Чтобы настроить внешний вид кода по умолчанию, перейдите в меню Файл, выберите Настройки и включите нужные опции на вкладке Редактор кода.



Чтобы посмотреть полный текст кода, не открывая редактор обработчика, наведите на код курсор мыши.

## Astra.Om

Astra.Om является единым скриптовым языком для различных программных продуктов АстраРегул:

- сервер ввода/вывода Astra.Server (сигнальная модель данных);
- среда разработки проектов автоматизации AStudio (объектная модель данных);
- среда разработки и исполнения визуальной части проектов автоматизации Astra.HMI (объектная модель данных).

В Astra.HMI язык применяется для исполнения формул, обработчиков функций, обработчиков событий и т.д.



Если вам недостаточно возможностей языка Astra.Om или использование языка вызвало какие-либо трудности, следует переключиться на язык JavaScript.

Чтобы обращаться к объектам на различных уровнях иерархии , используйте при написании скриптов специальные ключевые слова:

- › me или this - обращение к текущему объекту;
- › parent - обращение к родителю текущего объекта;
- › here - обращение к ограничивающему типу (корню иерархии в рамках типа).

Чтобы обращаться к свойствам объектов из скриптов и формул, используйте точку в качестве разделителя между именем объекта и именем свойства:



```
//Установить значение 1 свойству Property для объекта Object  
Object.Property = 1;  
//Присвоить переменной Variable значение свойства Property  
объекта Object  
Variable = Object.Property;
```

## JavaScript

Язык JavaScript применяется для расширения стандартных возможностей языка Astra.Ом. Большинство возможностей языка JavaScript (в соответствии с ECMAScript Language Specification) доступно для применения в Astra.HMI. Исключение составляют браузерно-ориентированные возможности языка. Используемые в Astra.HMI встроенные функции JavaScript, методы встроенных объектов или собственные функции могут возвращать значение, обращаться к входным аргументам в функции, а также вызывать другие функции.

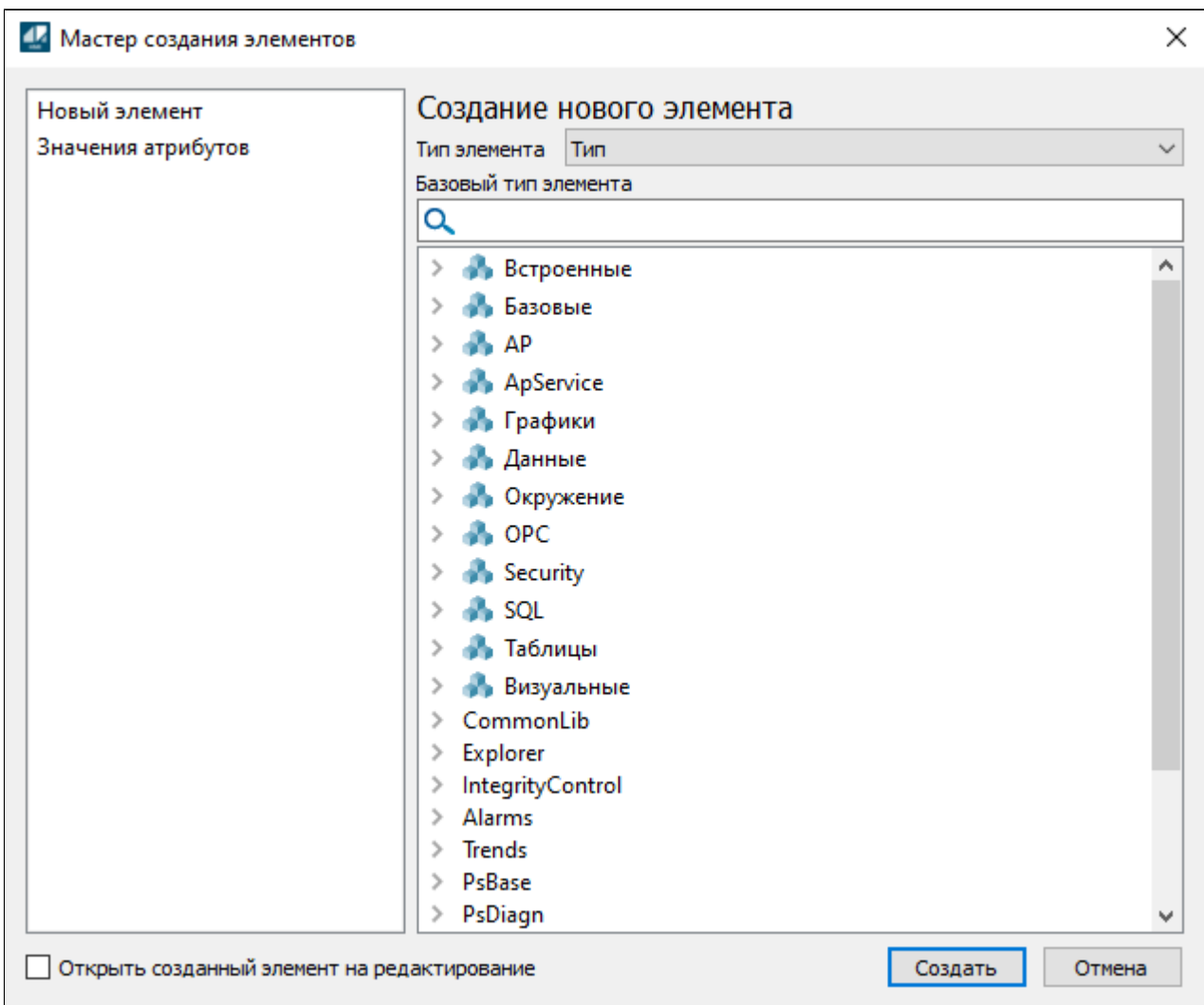


Следует помнить, что из-за технологии динамической компиляции, скрипты JavaScript не проверяются на наличие ошибок в момент запуска режима исполнения. Все ошибки и предупреждения, возникающие в ходе исполнения скриптов JavaScript, будут логироваться в области Журнал времени исполнения.

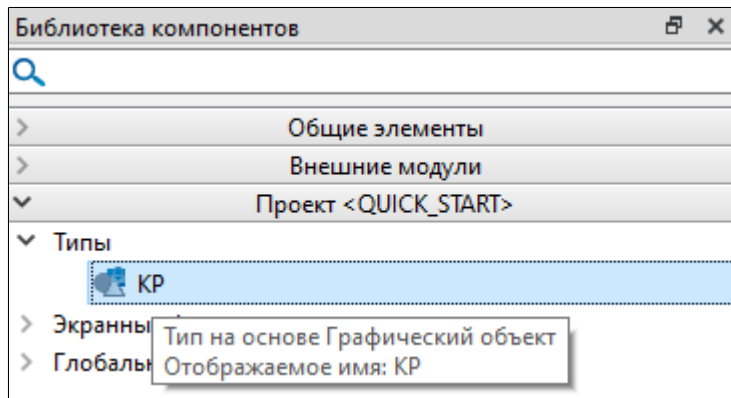
# Создание типов графических объектов

Для многократного применения одноподобных объектов на мнемосхемах создаются типы графических объектов. Разработанный единожды тип графического объекта может многократно применяться на разных экранных формах проекта.

Чтобы создать новый тип, выберите команду контекстного меню Создать и воспользуйтесь мастером создания элементов.



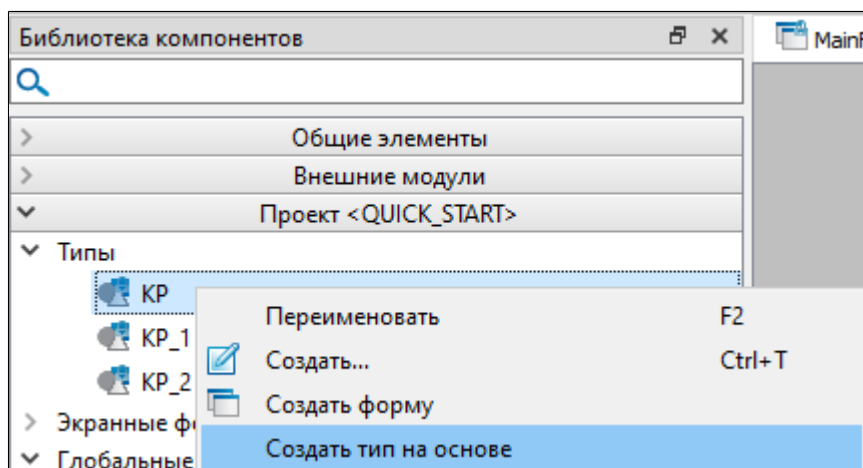
Создастся новый тип со стандартным названием. Прототип, на основе которого создан тип, отображается во всплывающей подсказке. Для редактирования типа дважды щелкните по нему.



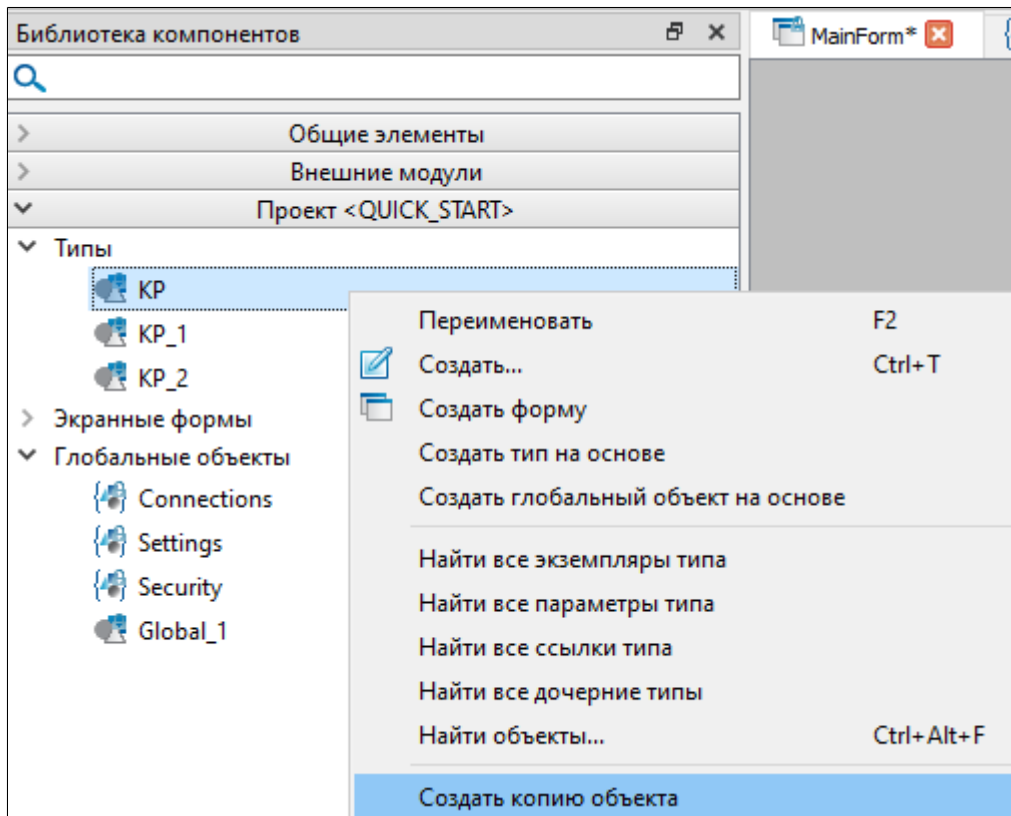
При первичном редактировании типа появится чистая рабочая область. Возможности работы в рабочей области типа аналогичны возможностям работы в рабочей области экранной формы.

Созданные типы сохраняются в папке проекта objects с расширением файлов \*.omobj.

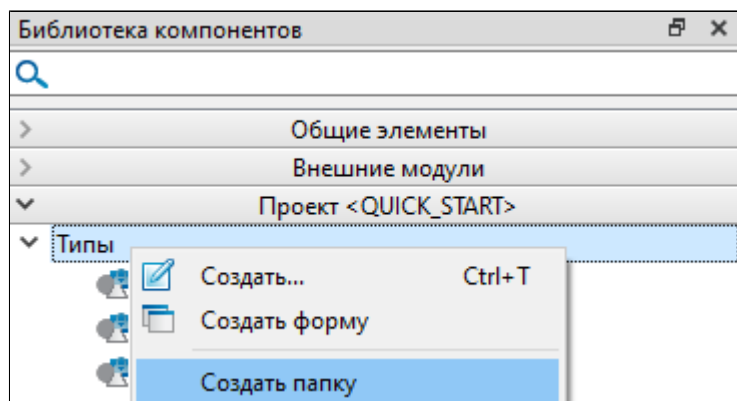
Чтобы создать тип на основе компонентов добавленных ранее, выберите команду контекстного меню элемента – Создать тип на основе.



Чтобы скопировать тип графического объекта, в Библиотеке компонентов выберите тип и команду Создать копию объекта в контекстном меню. Копия добавится рядом с копируемым типом.



Чтобы группировать типы графических объектов, используйте папки. Для создания папки перейдите в контекстное меню и выберите команду Создать папку.



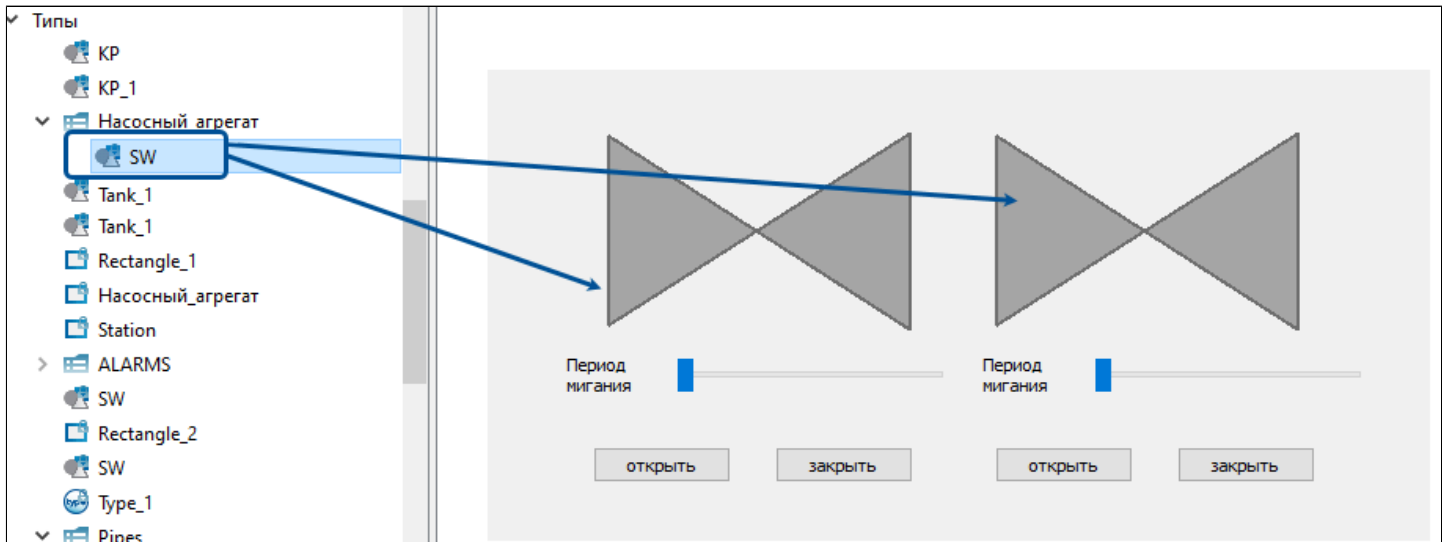
Нельзя допускать совпадения имён объектов, находящихся в папках и вне папок. Например, если имя объекта в папке совпадёт с именем объекта вне папки, то ссылка будет произведена на объект размещённый вне папки.





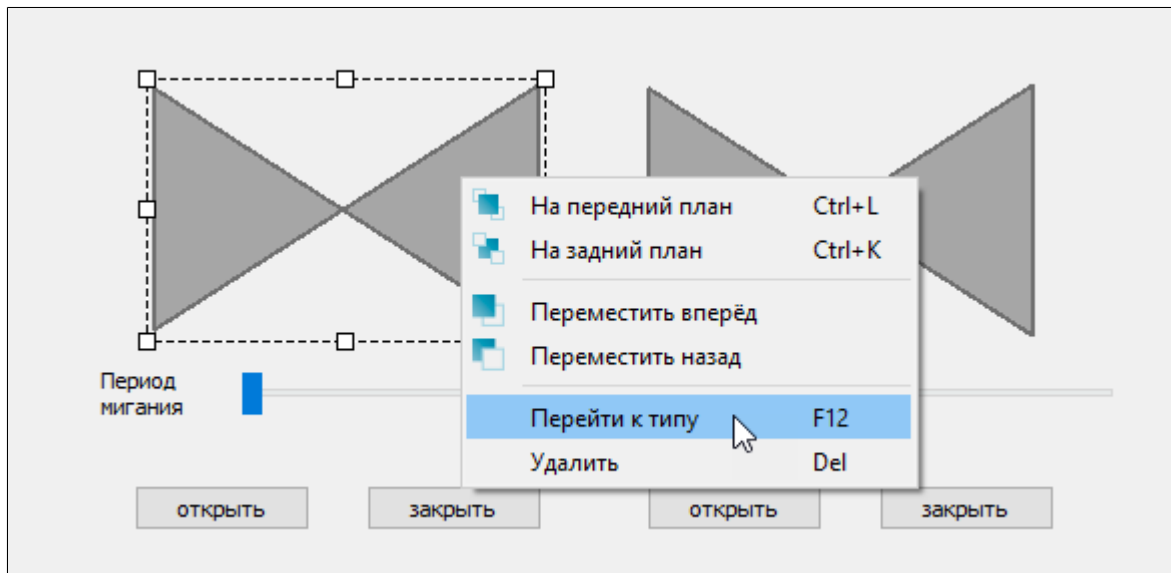
Чтобы обратиться к графическому объекту, расположенному в папке, в коде указывайте только имя объекта без имени папки.

Чтобы добавить экземпляр объекта созданного типа, перетащите нужный тип на рабочую область экранной формы или рабочую область другого типа.



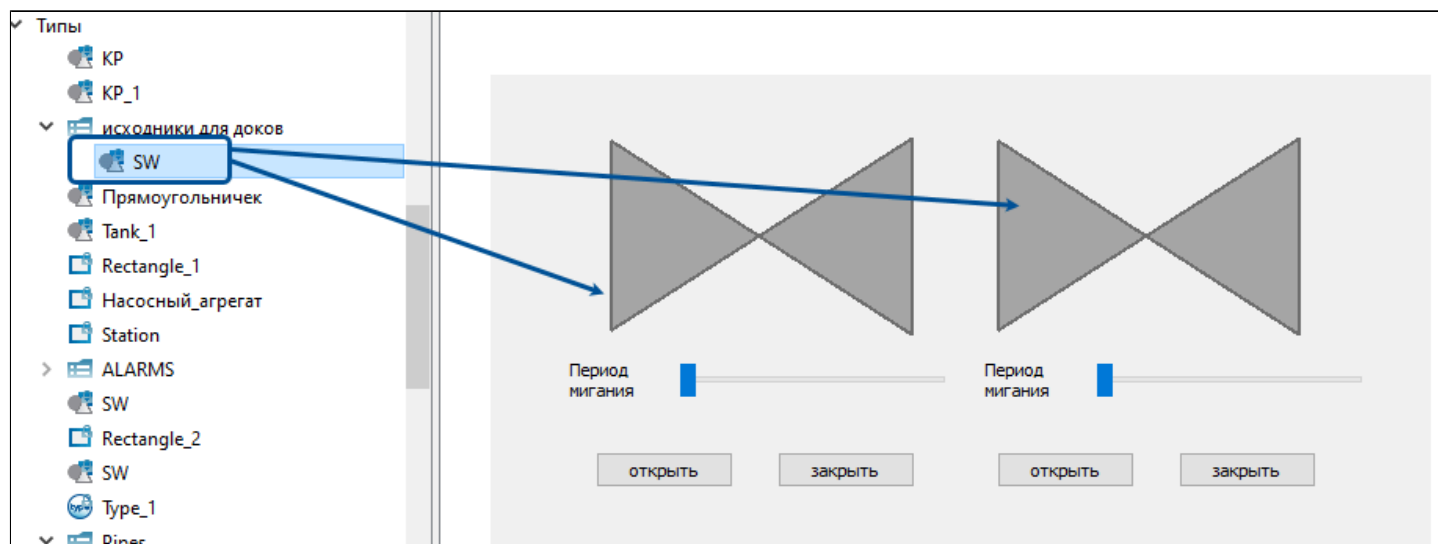
Чтобы добавить произвольное дополнительное описание экземпляру типа, используйте свойство Отображаемое имя.

Чтобы перейти к редактированию типа графического объекта, находящегося на форме, выберите объект и в контекстном меню объекта выберите Перейти к типу (F12). В результате тип графического объекта будет открыт для редактирования и выделен в Библиотеке компонентов.



# Добавление в проект экземпляров типа

Чтобы добавить на форму экземпляр созданного типа, перетащите нужный тип на рабочую область экранной формы или рабочую область другого типа.



Для каждого добавленного экземпляра вы можете задать свои значения свойств в Редакторе свойств.

## Работа с параметрами инициализации

Экземпляры объектов графических типов и экранные формы как правило имеют параметры инициализации, которые нужны для определения специфичных параметров каждого отдельно взятого экземпляра.

Параметрами инициализации объектов могут быть:

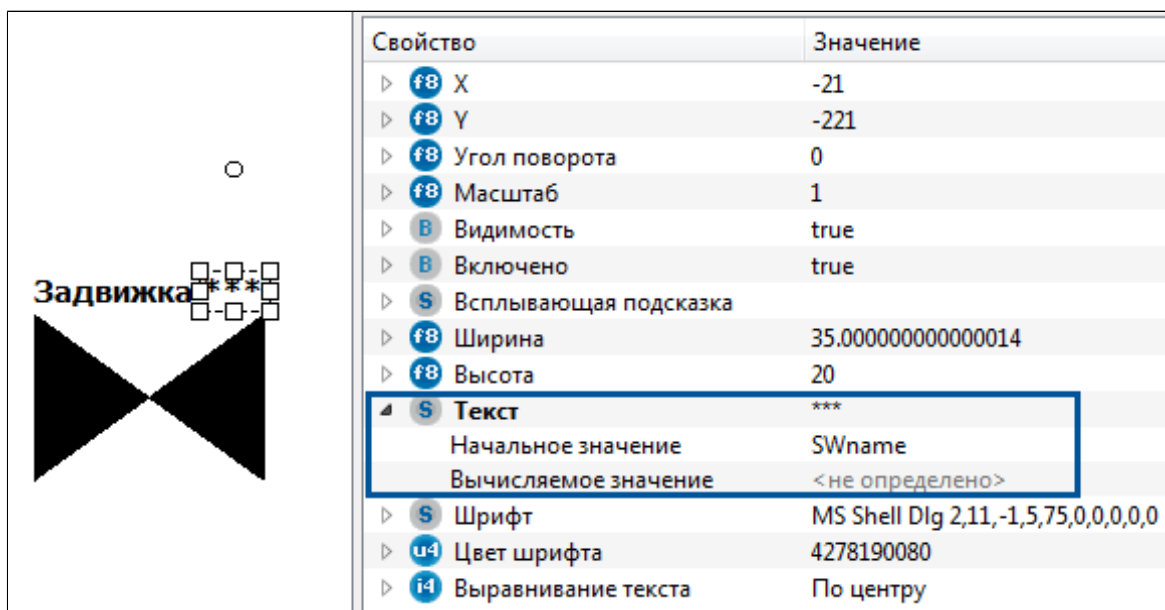
- › Значения примитивных типов (Создать → Параметр инициализации → Примитивный тип → <Имя типа>);
- › Ссылки на объекты стандартной библиотеки компонентов;
- › Ссылки на объекты графических типов.

Параметры инициализации объектов позволяют указывать начальные значения свойствам. После запуска экранной формы в рантайме свойства объектов, для которых было указано инициализирующее значение, примут данные значения.

## Пример работы с параметрами инициализации

В проекте автоматизации многократно используется объект технологического процесса – задвижка. Определен тип графического объекта для задвижки. Имя задвижки должно быть разным для каждого экземпляра объекта данного типа.

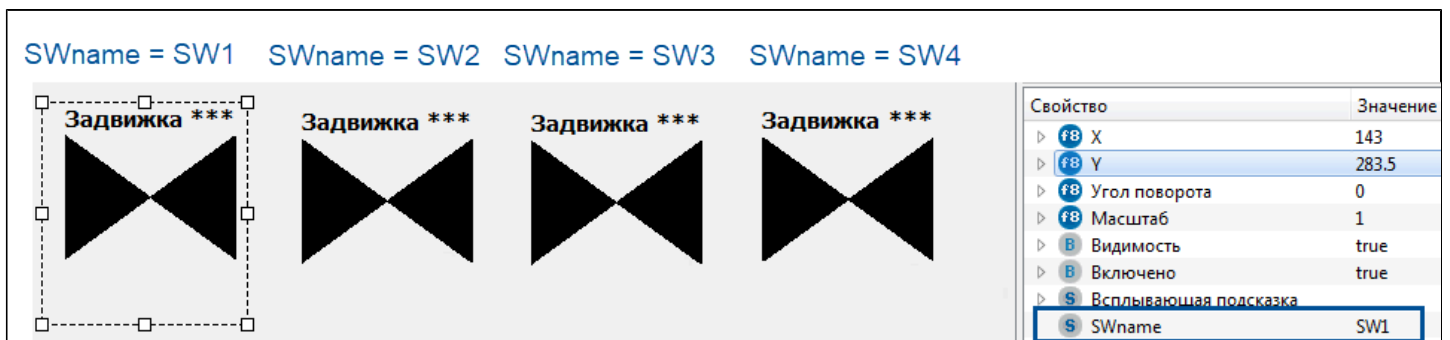
В структуре типа был определен параметр инициализации SWname (примитивный тип string). В режиме исполнения значение параметра инициализации должно выводиться вместо звездочек в текстовом блоке.



The screenshot shows a graphical object representing a valve, labeled "Задвижка \*\*\*". To its right is a table of properties:

Свойство	Значение
▷ f8 X	-21
▷ f8 Y	-221
▷ f8 Угол поворота	0
▷ f8 Масштаб	1
▷ B Видимость	true
▷ B Включено	true
▷ S Всплывающая подсказка	
▷ f8 Ширина	35.000000000000014
▷ f8 Высота	20
▲ S Текст	***
Начальное значение	SWname
Вычисляемое значение	<не определено>
▷ S Шрифт	MS Shell Dlg 2,11,-1,5,75,0,0,0,0
▷ u4 Цвет шрифта	4278190080
▷ i4 Выравнивание текста	По центру

При добавлении экземпляров типа на форму, значение параметра инициализации указывается для каждого экземпляра типа в редакторе свойств.



The screenshot shows four instances of the valve symbol, each with a different initialization parameter: SWname = SW1, SWname = SW2, SWname = SW3, and SWname = SW4. The first instance is highlighted with a dashed border. To the right is a table of properties:

Свойство	Значение
▷ f8 X	143
▷ f8 Y	283.5
▷ f8 Угол поворота	0
▷ f8 Масштаб	1
▷ B Видимость	true
▷ B Включено	true
▷ S Всплывающая подсказка	
▷ S SWname	SW1

В результате, при запуске в рантайме каждая задвижка будет называться в соответствии со своим параметром инициализации.

**Задвижка SW1**



**Задвижка SW2**



**Задвижка SW3**

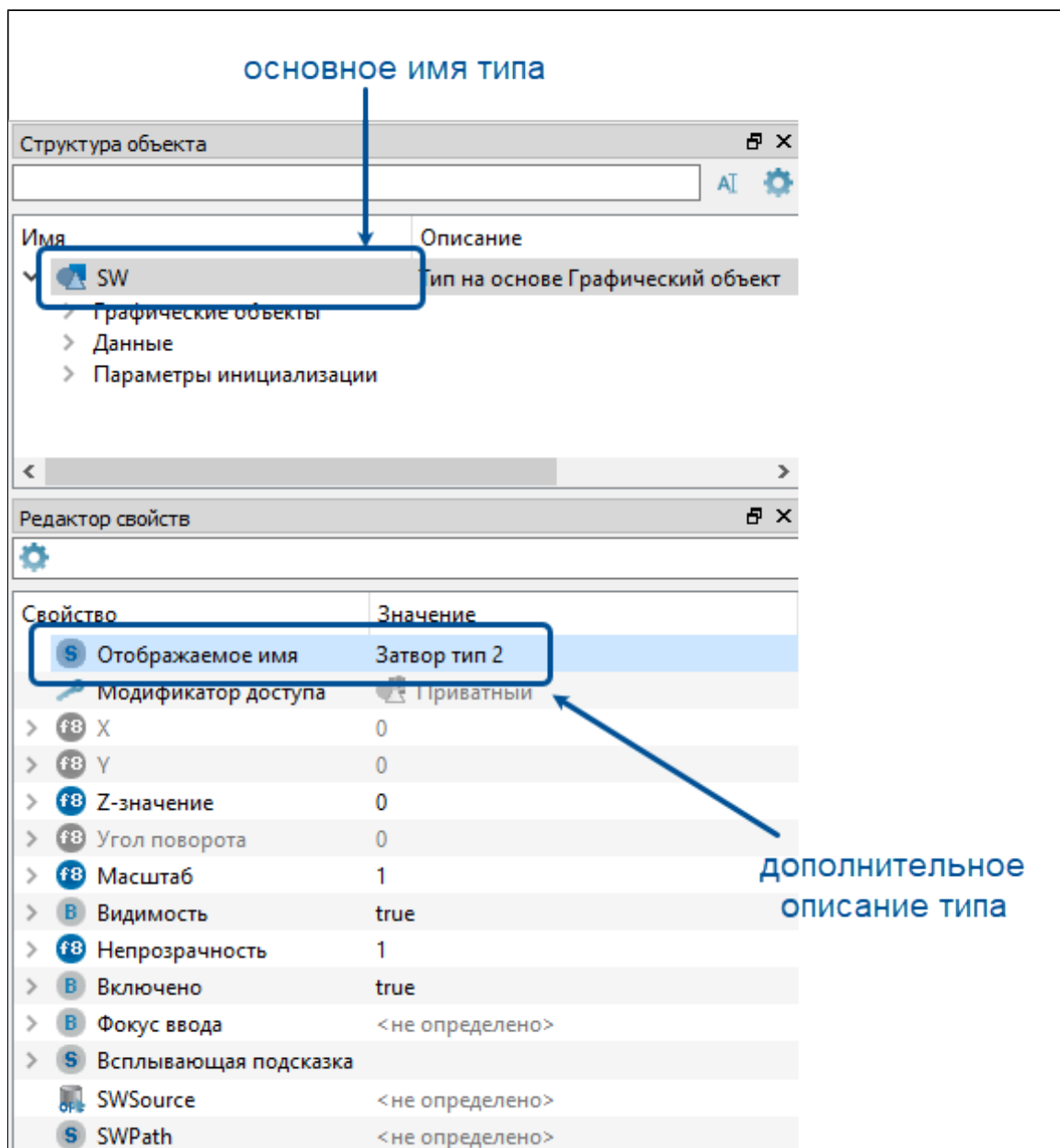


**Задвижка SW4**

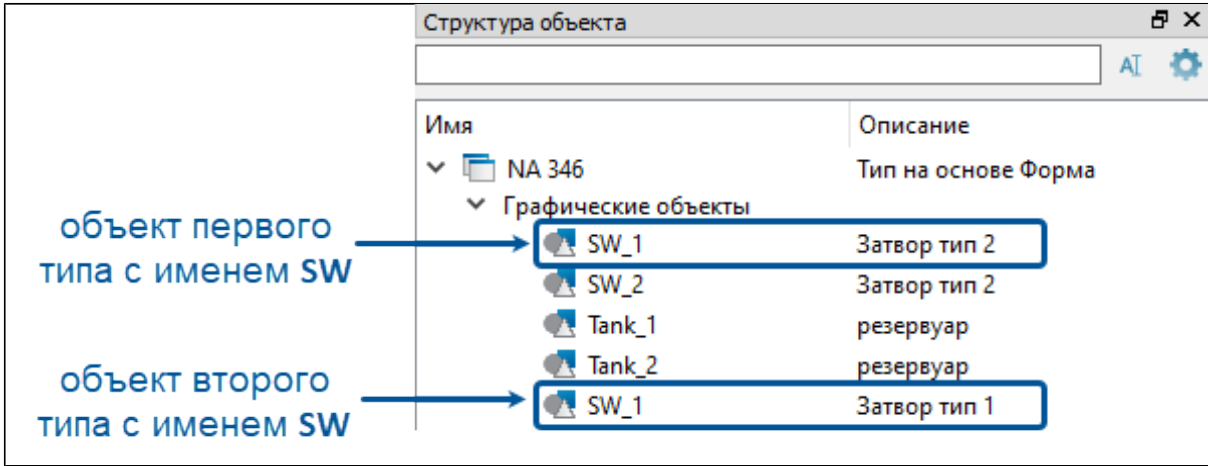


# Добавление описаний

Чтобы добавить произвольное дополнительное описание каждому созданному типу, используйте свойство Отображаемое имя.



После добавления экземпляра (объекта) типа на форму, описание будет отображаться рядом с именем экземпляра в области Структура объекта. Благодаря наличию отображаемого имени можно ориентироваться среди объектов с одинаковыми именами, но разными типами.

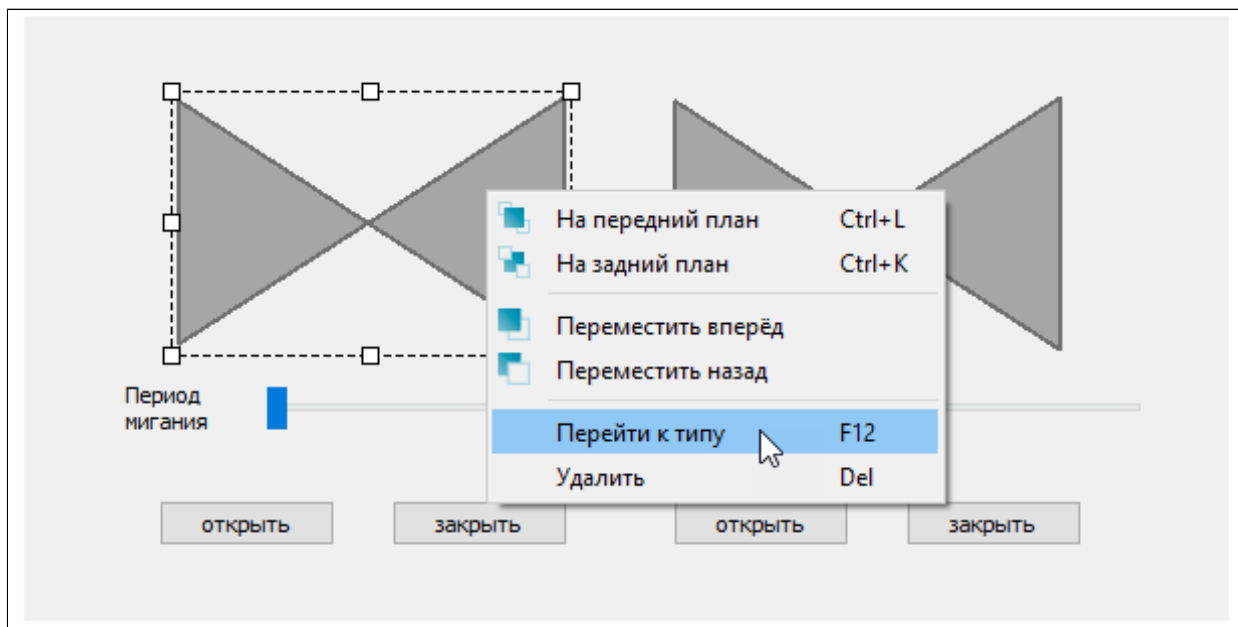


# Редактирование типов графических объектов

Для редактирования типа дважды щелкните по нему.

При первом редактировании типа появится чистая рабочая область. Действия в рабочей области типа аналогичны действиям в рабочей области экранной формы.

Чтобы перейти к редактированию типа конкретного экземпляра объекта данного типа выделите экземпляр и в контекстном меню выберите Перейти к типу (клавиша F12). В результате тип графического объекта будет открыт для редактирования и выделен в Библиотеке компонентов.





## 1.1.4.2.6.12. Встраивание внешних библиотек .NET

Astra.HMI позволяет добавлять сторонние .NET-компоненты (.NET Framework, .NET Core и QtPlugin).



Встраиваемые библиотеки .NET Framework недоступны для использования в ОС Linux.

С помощью библиотек на основе .NET Framework, .NET и QtPlugin Core можно добавить дополнительные компоненты в библиотеку компонентов и расширить стандартные возможности Astra.HMI внутри отдельного проекта.



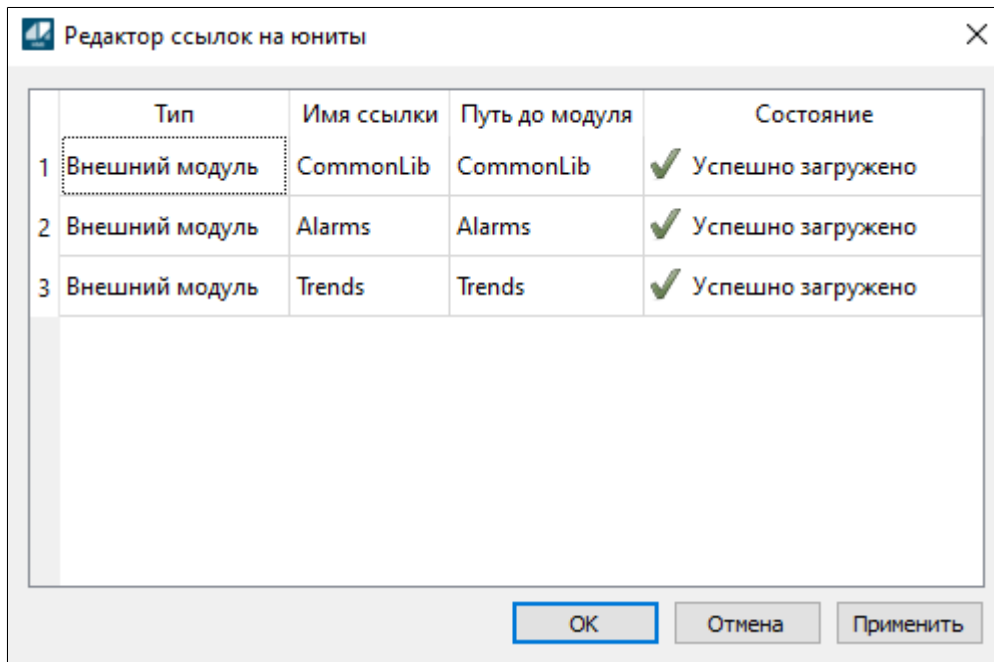
Для корректной работы библиотек .NET Core в ОС Linux установите пакеты среды выполнения .NET Core.

Установка .Net Core с использованием сети интернет описана [здесь](#).

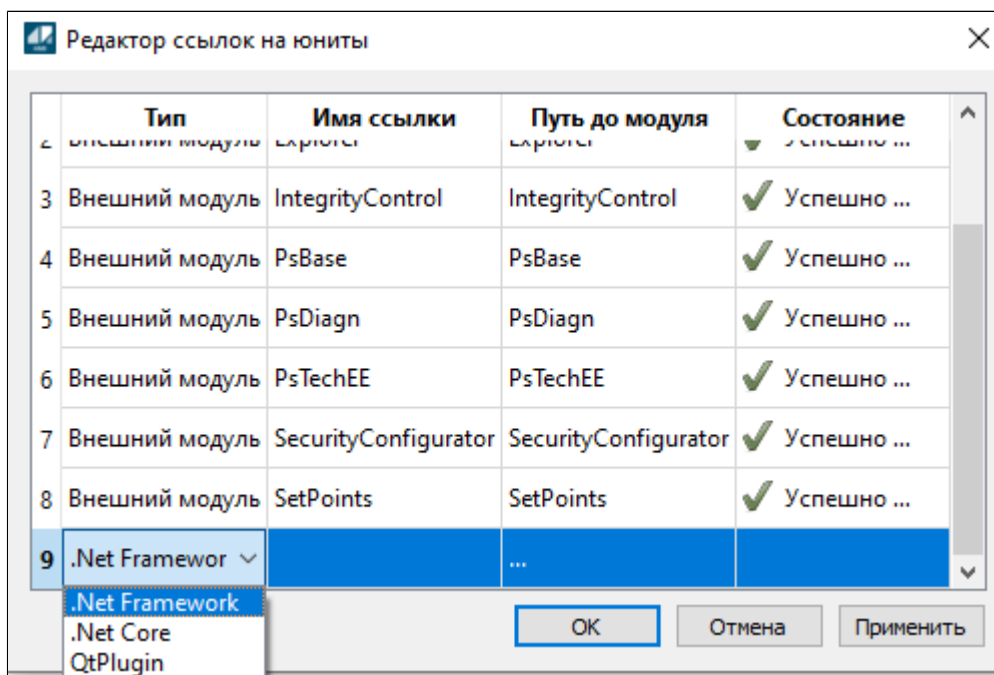
Уточнить актуальную версию можно [по ссылке](#).

Чтобы добавить библиотеки в проект:

1. Выполните команду Проект → Ссылки на юниты. В открывшемся окне используйте команды контекстного меню для добавления или удаления библиотек.



2. Укажите платформу для каждой библиотеки, в столбце Тип.




3. Чтобы просмотреть статус загрузки библиотеки, нажмите ОК в окне Редактор ссылок на юниты и повторно откройте окно. Загруженные библиотеки добавляют в библиотеку компонентов Astra.HMI новые юниты с компонентами.


4. Используйте новые компоненты аналогично встроенным компонентам Astra.HMI.

## 1.1.4.2.6.13. Сохранение

Чтобы сохранить изменения на активной вкладке, выполните одно из действий:

- › перейдите в меню Файл и выберите команду Сохранить;
- › нажмите кнопку  на Панели инструментов;
- › нажмите сочетание клавиш Ctrl+S.

Чтобы сохранить все изменения проекта, выполните одно из действий:

- › перейдите в меню Файл и выберите команду Сохранить проект,
- › нажмите кнопку  на Панели инструментов;
- › нажмите сочетание клавиш Ctrl+Shift+S.

Чтобы сохранить проект по указанному пути, выберите команду Сохранить проект как... в меню Файл.



Каждые 30 секунд в папку проекта objects автоматически сохраняется резервная копия проекта с расширением \*.bak.

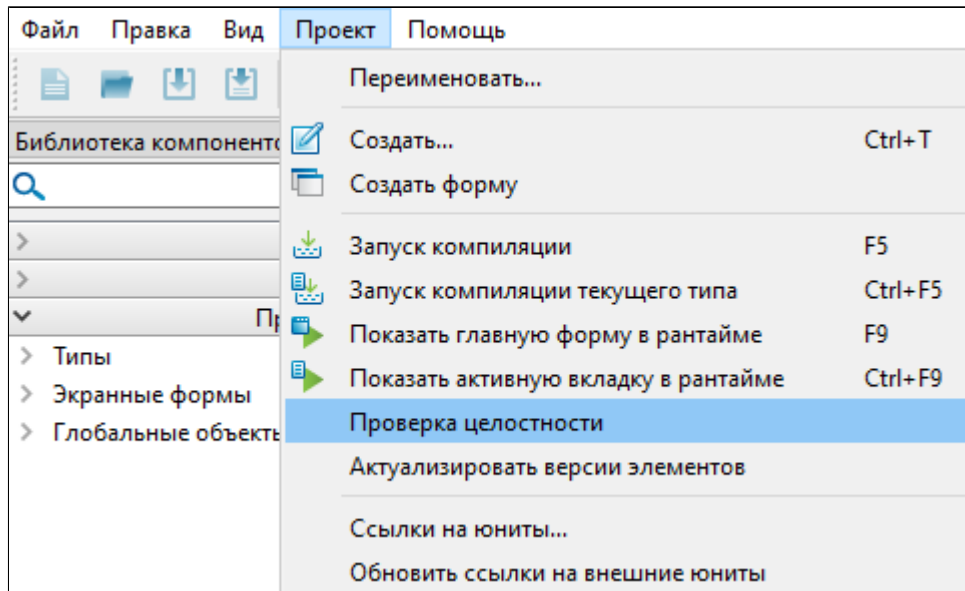
Резервная копия удаляется при сохранении проекта или отклонении сохранения во время закрытия дизайнера Astra.HMI.

## 1.1.4.3. Отладочные средства

- › [Проверка целостности](#)
- › [История изменений](#)
- › [Журнал времени исполнения](#)
- › [Контрольные значения](#)

## 1.1.4.3.1. Проверка целостности

Чтобы исправить ошибки некорректных ссылок или пустых присвоений, используйте функцию Проверка целостности в меню Проект.

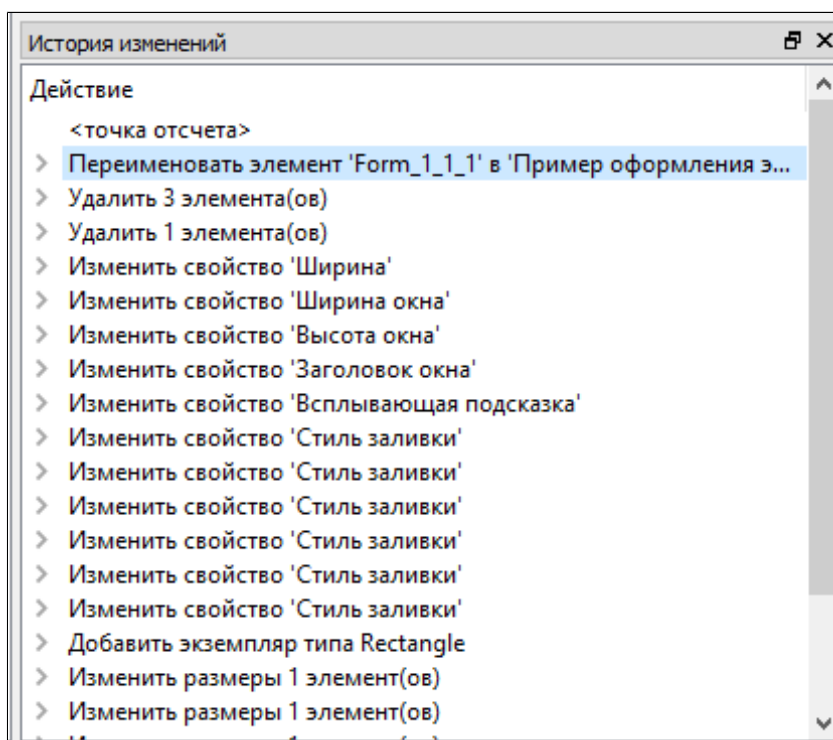


## 1.1.4.3.2. История изменений

История изменений – это область, отражающая в хронологическом порядке все изменения, произведенные пользователем с объектами на рабочей области.

В области История изменений представлена информация о добавлении, удалении, переименовании, перемещении, изменении размера элемента и т.д. Каждый пункт истории можно раскрыть и детально просмотреть иерархию изменений.

Жирным шрифтом выделяется последнее или активное изменение в истории.

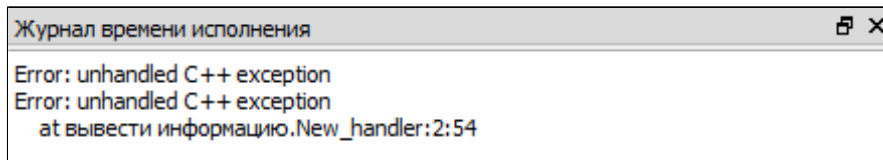


Чтобы отменить последнее действие, на панели инструментов нажмите кнопку "Отменить действие" (Ctrl+Z).

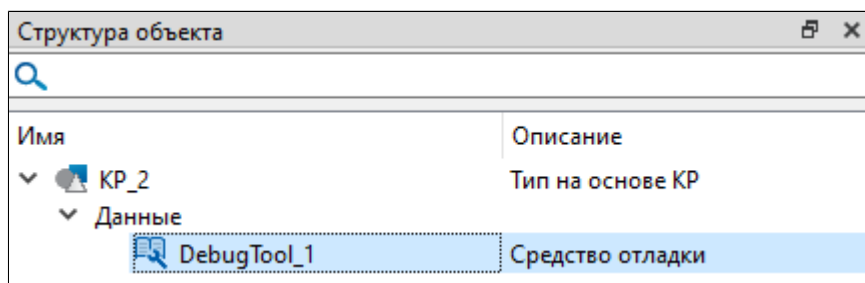
Чтобы вернуться к последнему отмененному действию, на панели инструментов нажмите кнопку "Повторить действие" (Ctrl+Y).

## 1.1.4.3.3. Журнал времени исполнения

Просмотреть ошибки, обнаруженные во время исполнения проекта, можно в области Журнал времени исполнения.



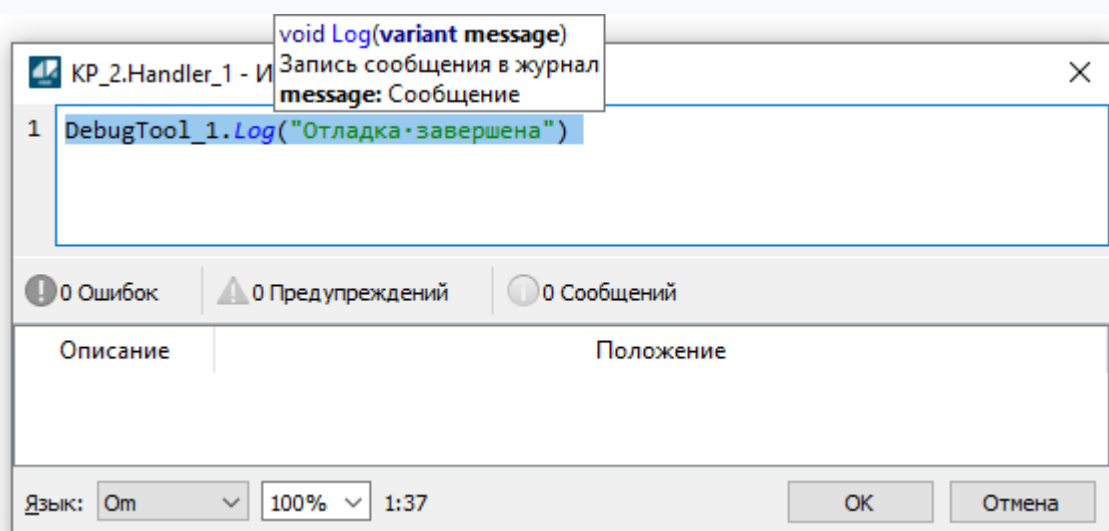
Чтобы в области Журнал времени исполнения выводились отладочные замечания во время исполнения кода, используйте элемент Средство отладки.



### Пример



```
//Отобразить сообщение в Журнале времени исполнения после  
окончания отладки  
DebugTool_1.Log("Отладка завершена");
```



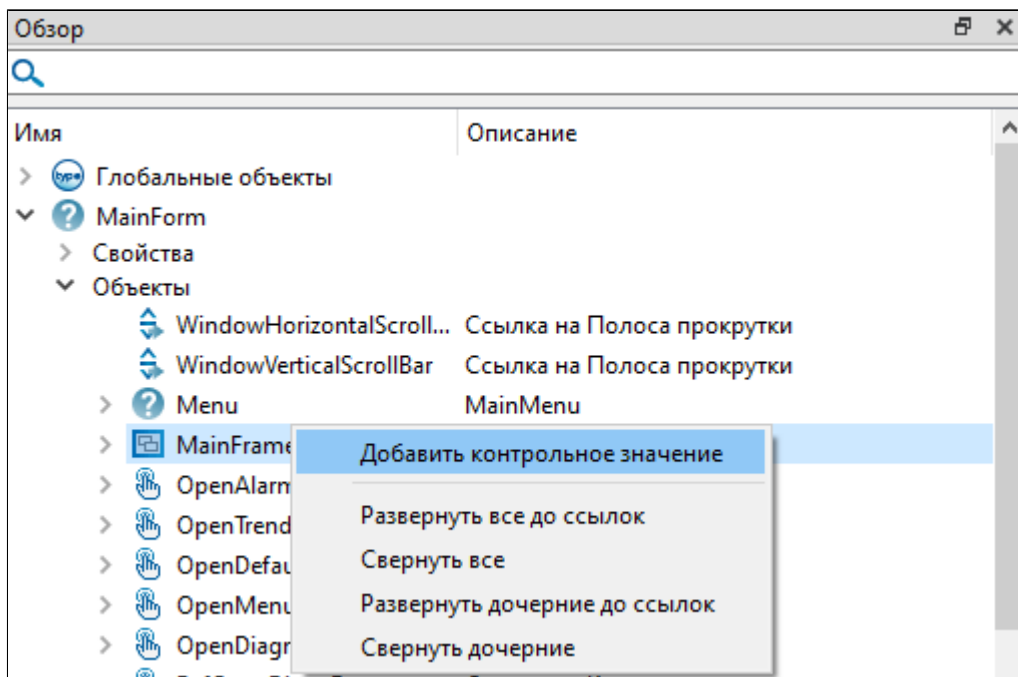




## 1.1.4.3.4. Контрольные значения

Чтобы просмотреть и изменить текущие значения свойств объектов:

1. Запустите проект или форму в режим рантайма.
2. Перейдите в окно Обзор (появляется только при наличии запущенной на исполнение формы), выберите контролируемый объект или свойство, вызовите контекстное меню и выполните команду Добавить контрольное значение.



3. В окне Контрольные значения (появляется только при наличии запущенной на исполнение формы) отображаются текущие значения свойств выбранного объекта.

Имя	Значение	Описание	Путь
MainFrame		Фрейм	MainForm.MainFrame
Свойства			
f8 Координата курсора X	2289	double	
f8 Координата курсора Y	575	double	
f8 X	10	double	
f8 Y	112	double	
f8 Z-значение	0	double	
f8 Угол поворота	0	double	
f8 Масштаб	1	double	
u1 Отражение	Без отражения	uint1	
b Видимость	true	bool	

Чтобы изменить значения свойств, редактируйте значения в окне Контрольные значения. Свойства, доступные только для чтения, изменять нельзя.



Значения свойств, которые были изменены в окне Контрольные значения, после выхода из режима рантайм будут сброшены.

## 1.1.4.4. Библиотеки и расширения

Библиотека	Версия	Описание
<a href="#">Astra.HMI.CommonLib</a>	1.7.3.1	Библиотека дополнительных компонентов Astra.HMI
<a href="#">Astra.HMI.Charts</a>	2.0.3.8	Расширение Astra.HMI
<a href="#">Astra.HMI.Tables</a>	2.0.5.1	Расширение Astra.HMI

## 1.1.4.4.1. Astra.HMI.Charts

**Astra.HMI.Charts** – библиотека компонентов Astra.HMI, позволяющих представлять данные, принятые от источника, в виде графиков.

Возможности:

- › просмотр графиков оперативных данных по значениям, полученным с локальных и удаленных серверов;
- › просмотр графиков исторических данных.



После установки доступна в виде вкладки [Графики](#) в окне Общие элементы Astra.HMI.

## 1.1.4.4.2. Astra.HMI.Tables

**Astra.HMI.Tables** – библиотека компонентов Astra.HMI, позволяющих помещать в таблицу данные об оперативных событиях с источника либо формировать таблицы на основе любых собственных данных.

Возможности:

- › заполнение невизуальной таблицы модели данными с источника;
- › заполнение невизуальной таблицы модели данными, введёнными вручную, для последующего хранения;
- › визуализация таблицы, хранящейся в модели данных;
- › декорирование визуальной таблицы.



После установки доступна в виде вкладки [Таблицы](#) в окне Общие элементы Astra.HMI.

## 1.1.4.4.3. Astra.HMI.CommonLib

**Astra.HMI.CommonLib** – библиотека дополнительных компонентов Astra.HMI.

Содержит:

- › диалоговые окна;
- › контекстное меню;
- › календарь;
- › файловый менеджер;
- › конвертер значений;
- › дерево сигналов;
- › журнал сообщений;
- › компоненты безопасности;
- › различные кнопки и индикаторы.

Внешний вид и функции компонентов можно настроить с помощью [API](#).

## Требования к окружению

Для использования средств Astra.HMI.CommonLib должны быть установлены:

- › среда разработки и исполнения Astra.HMI;
- › подсистема безопасности Astra.Security;
- › расширения Astra.HMI.Tables и Astra.HMI.Security среды Astra.HMI.

Если работаете с веб-версией проекта автоматизации, установите дополнительно:

- › Astra.HMI.WebViewer – для использования проектов Astra.HMI в веб-интерфейсе;
- › Astra.HMI.Security.WebViewer – для работы компонентов Astra.HMI.Security в веб-интерфейсе;
- › Astra.HMI.Tables.WebViewer – для работы компонентов Astra.HMI.Tables в веб-интерфейсе.

## 1.1.4.4.3.1. Встраивание компонента в проект

Расширение Astra.HMI.CommonLib выполнено в виде проекта Astra.HMI. Чтобы использовать компоненты расширения, необходимо подключить проект расширения в свой проект Astra.HMI в виде внешнего модуля.

### Подключение внешнего модуля Astra.HMI.CommonLib

Чтобы подключить Astra.HMI.CommonLib как внешний модуль, выполните следующие действия:

1. Создайте в папке своего проекта папку externals, в которой нужно размещать файлы всех подключаемых внешних модулей.
2. В созданную папку скопируйте папку расширения CommonLib, расположенную в:

› ОС Windows:



C:\Program Files\AstraRegul\Astra.HMI.Extensions

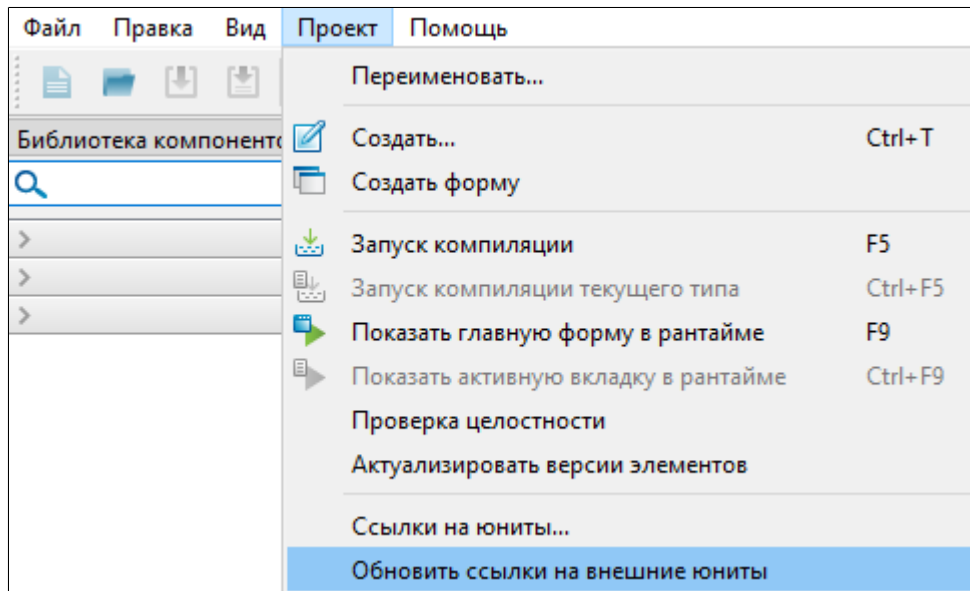
› ОС Linux:



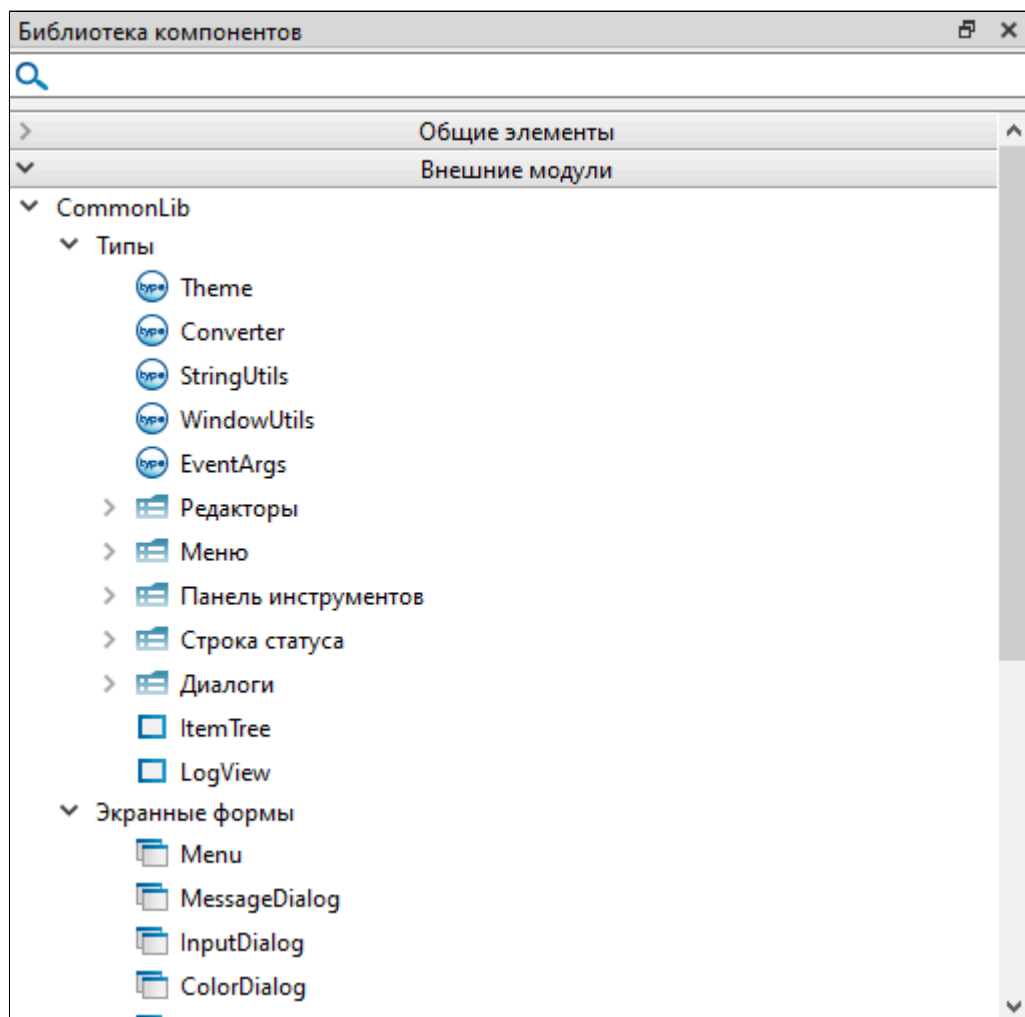
/opt/AstraRegul/Astra.HMI.Extensions

3. Откройте свой проект в дизайнера Astra.HMI.
4. Перейдите в меню Проект и выберите команду Обновить ссылки на внешние юниты.





В результате в библиотеке компонентов проекта появится внешний модуль CommonLib, содержащий компоненты расширения.



Чтобы приступить к работе, перетяните компонент на форму или вызовите из обработчика какого-либо события. В проект можно добавить столько экземпляров одного типа, сколько необходимо. Разные экземпляры одного типа можно использовать под разные цели.

# Команды

Команда	Описание
<a href="#">OnSaveTable</a>	Сохранить таблицу сигналов
<a href="#">OnDoPrintSettings</a>	Применить настройки печати
<a href="#">OnPrint</a>	Начать печать
<a href="#">OnExport</a>	Начать экспорт
<a href="#">OnSaveSignalList</a>	Сохранить лист сигналов

# OnSaveTable

Команда, которая выполняет сохранение таблицы сигналов.



OnSaveTable.Invoke()

## Примеры



```
//Вызов команды Сохранить таблицу сигналов"  
Trends.OnSaveTable.Invoke();
```

# OnDoPrintSettings

Команда, которая применяет настройки печати.



OnDoPrintSettings.Invoke()

## Примеры



```
//Вызов команды "Применить настройки печати"  
Trends.OnDoPrintSettings.Invoke();
```

# OnPrint

Команда, которая запускает печать.



OnPrint.Invoke()

## Примеры



```
//Вызов команды "Запустить печать"  
Trends.OnPrint.Invoke();
```

# OnExport

Команда, которая запускает экспорт файлов.



OnExport.Invoke()

## Примеры



```
//Вызов команды "Начать экспорт"  
Trends.OnExport.Invoke();
```

# OnSaveSignalList

Команда, которая сохраняет лист сигналов



OnSaveSignalListt.Invoke()

## Примеры



```
//Вызов команды "Сохранить лист сигналов"  
Trends.OnSaveSignalListt.Invoke();
```



## 1.1.4.4.3.3. Справочник API

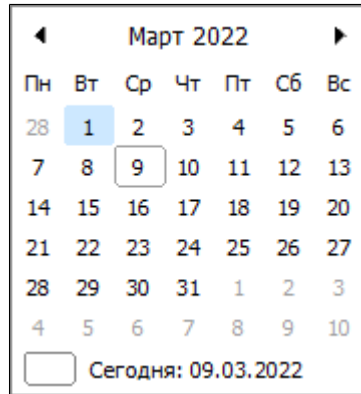
Компонент	Описание
<a href="#">Calendar</a>	Календарь для выбора даты.
<a href="#">ChangePasswordDialog</a>	Окно смены пароля текущего пользователя.
<a href="#">ColorDialog</a>	Диалоговое окно для выбора цвета из палитры.
<a href="#">ColorDialogResult</a>	Компонент, в котором сохраняется значение, введенное в окне ColorDialog.
<a href="#">ConnectionIndicator</a>	Индикатор состояния подключения к серверу.
<a href="#">Converter</a>	Компонент, использующийся для различных преобразований значений.
<a href="#">DateEditor</a>	Окно редактора даты для ввода даты в формате «день:месяц:год».
<a href="#">DateTimeResult</a>	Компонент, в котором сохраняется дата, выбранная в календаре Calendar.
<a href="#">DialogResult</a>	Компонент, в котором сохраняется путь к папке, выбранной в окне SelectFolderDialog.
<a href="#">FileDialogResult</a>	Компонент, в котором сохраняется имя и тип файла, выбранного в окне OpenFileDialog или SaveFileDialog.
<a href="#">FileList</a>	Список файлов и папок, содержащихся в указанной папке
<a href="#">InputDialog</a>	Диалоговое окно для ввода значения.
<a href="#">InputDialogResult</a>	Компонент, в котором сохраняется значение, введенное в окне InputDialog.
<a href="#">IntegerUpDown</a>	Компонент, позволяющий менять значения целочисленных переменных с указанным шагом.
<a href="#">ItemTree</a>	Дерево сигналов.
<a href="#">KeyboardLayoutButton</a>	Кнопка, показывающая текущую раскладку клавиатуры.
<a href="#">LoginButton</a>	Кнопка вызова окна авторизации пользователя LoginDialog.

<a href="#">LoginDialog</a>	Окно авторизации пользователя.
<a href="#">LoginDialogResult</a>	Компонент, через который передаются id и отображаемое имя текущего пользователя от формы LoginDialog к форме ChangePasswordDialog.
<a href="#">LogView</a>	Журнал сообщений.
<a href="#">LogViewButton</a>	Кнопка-индикатор наличия сообщений в журнале.
<a href="#">Menu</a>	Контекстное меню.
<a href="#">MenuItem</a>	Элемент контекстного меню Menu.
<a href="#">MenuItemResult</a>	Компонент, в котором сохраняется результат выбора пункта в окне Menu.
<a href="#">MenuItemSeparator</a>	Разделитель элементов.
<a href="#">MessageDialog</a>	Диалоговое окно с настраиваемым внешним видом и функциями.
<a href="#">OpenFileDialog</a>	Окно просмотра файловой системы для открытия файлов.
<a href="#">SaveFileDialog</a>	Окно просмотра файловой системы для сохранения файлов.
<a href="#">SelectFolderDialog</a>	Окно просмотра файловой системы для выбора папки.
<a href="#">StringUtils</a>	Компонент, использующийся для различных преобразований строковых значений
<a href="#">Theme</a>	Компонент, позволяющий менять тему оформления визуальных компонентов Astra.HMI.CommonLib
<a href="#">TimeEditor</a>	Окно редактора времени для ввода времени в виде «часы:минуты:секунды».
<a href="#">WindowUtils</a>	Вспомогательный тип для работы с окнами в режиме исполнения.
<a href="#">ConnectDialog</a>	Окно подключения к серверу
<a href="#">ConnectDialogResult</a>	Компонент, в котором сохраняются адрес и порт сервера, к которому выполняется подключение в окне <a href="#">ConnectDialog</a>
<a href="#">IntervalEditor</a>	Редактор интервала времени

<a href="#">EventArgs</a>	Буфер, предназначенный для передачи аргументов события, реализованного с помощью компонента Команда
---------------------------	---

## 1.1.4.4.3.3.1. Calendar

Календарь для выбора даты. Он используется в редакторе выбора даты аналогично выпадающему меню.



Используя компонент, создайте экземпляр типа [DateTimeResult](#), в котором будет сохранен результат выбора даты. Ссылка на созданный экземпляр указывается в свойстве [pResult](#).

## 1.1.4.4.3.3.1.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.1.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.1.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.1.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```



## 1.1.4.4.3.3.1.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.1.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.1.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.1.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.1.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна



## 1.1.4.4.3.3.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.1.1.20. KeyPress

Нажатие определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



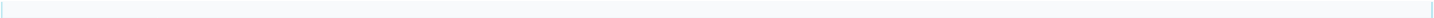
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.1.1.21. KeyRelease

Отпускание определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.1.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы



## 1.1.4.4.3.3.1.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```



## 1.1.4.4.3.3.1.2.2. SetBoundingRegion

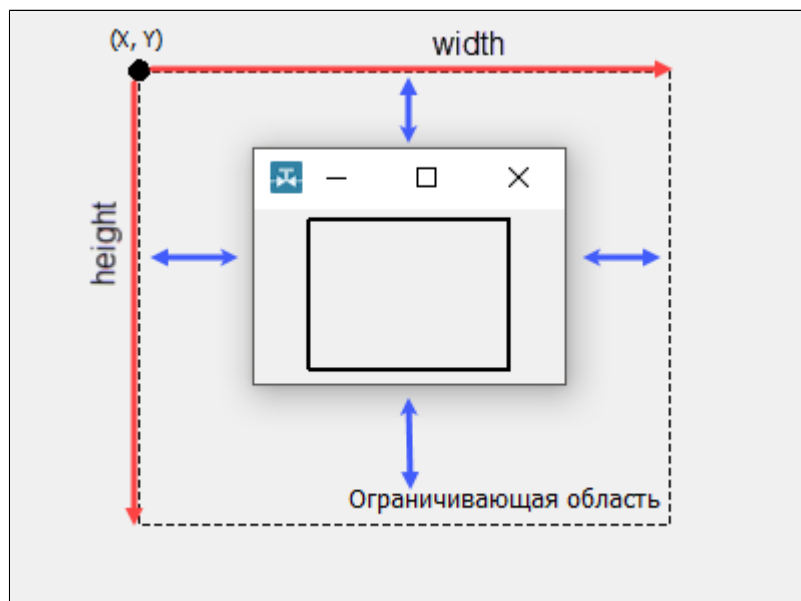
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`

## 1.1.4.4.3.3.1.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.1.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.1.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.1.2.6. SaveScreenShot

Формирует скриншот формы со всем ее видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.1.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.1.2.8. Activate

Сделать окно формы активным.



```
void Activate()
```



## 1.1.4.4.3.3.1.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.1.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.1.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.1.2.12. FormY

Получение координаты Y на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



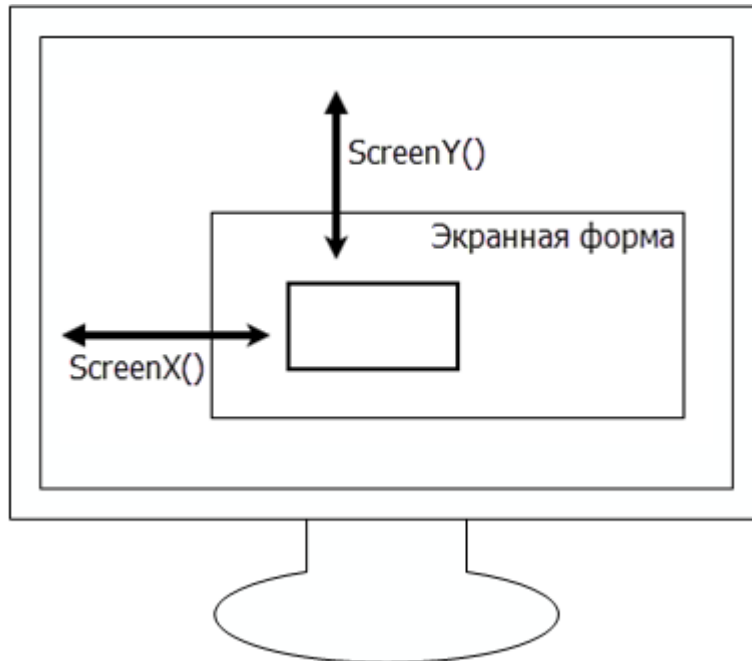
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.1.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

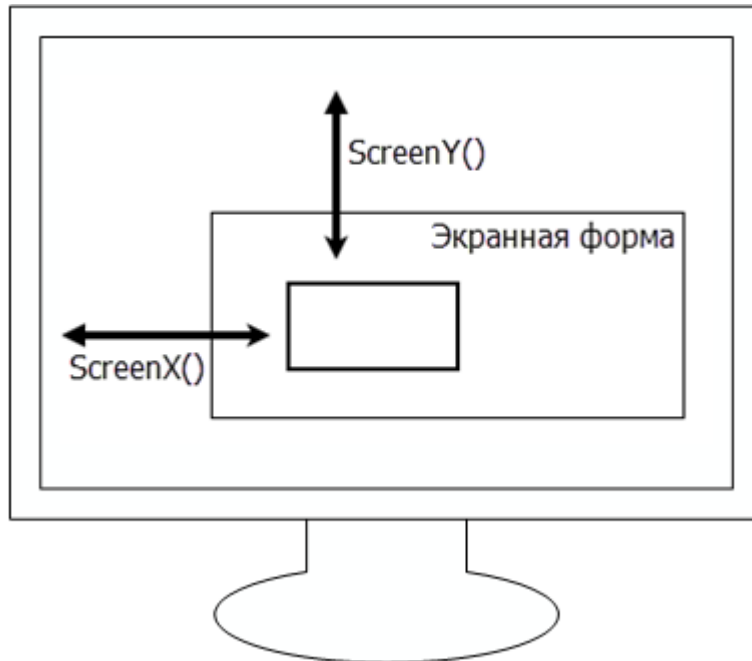


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.1.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.1.2.15. Show

Показать окно формы.



void Show()



## 1.1.4.4.3.3.1.2.16. Hide




Скрыть окно формы.



void Hide()

## 1.1.4.4.3.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна

<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Высота элемента</a>	Высота элемента, относительно которого открывается календарь
<a href="#">Дата</a>	Дата, выделенная при открытии окна
<a href="#">Ссылка на результат</a>	Ссылка на экземпляр типа <a href="#">DateTimeResult</a> , в котором будет сохранена выбранная дата
<a href="#">pSelected</a>	Ссылка на команду, выполняемую при выборе даты в окне
<a href="#">pOnClosed</a>	Ссылка на команду, выполняемую при закрытии окна
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.1.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.1.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.1.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.



## 1.1.4.4.3.3.1.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.1.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.1.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.1.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;



//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;

## 1.1.4.4.3.3.1.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.1.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.1.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.1.3.12. Непрозрачность

Непрозрачность.



float8 Opacity



## 1.1.4.4.3.3.1.3.13. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.1.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




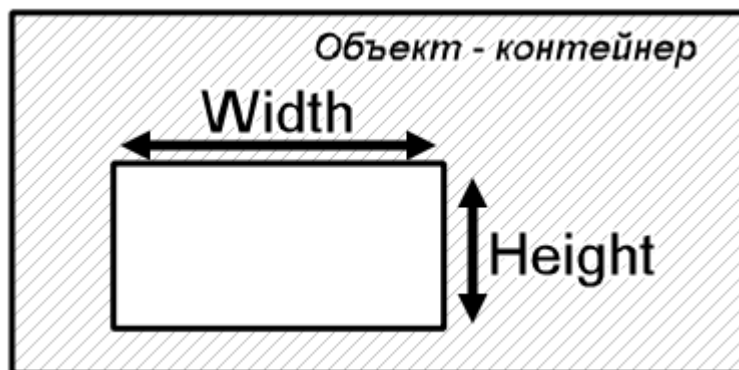
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.1.3.15. Ширина


Ширина объекта в пикселях.

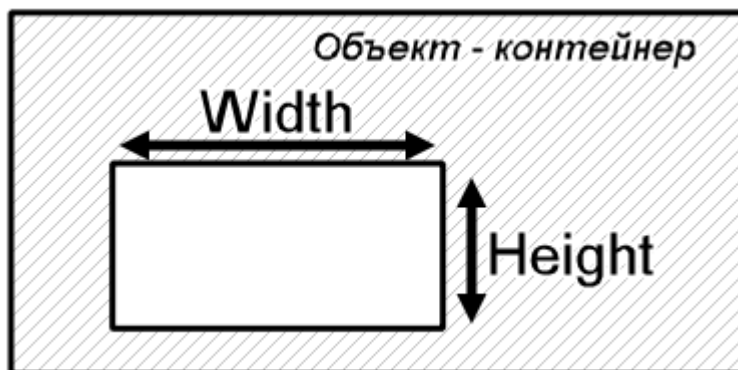
 float8 Width



# 1.1.4.4.3.3.1.3.16. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.3.1.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.1.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.1.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.1.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```



## 1.1.4.4.3.3.1.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.4.3.3.1.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.1.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.1.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.1.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.1.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.1.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```



## 1.1.4.4.3.3.1.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.1.3.29. "свернуть"

## Показывать

## кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.1.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;

## 1.1.4.4.3.3.1.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.1.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.1.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.1.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.1.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```



## 1.1.4.4.3.3.1.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.1.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.1.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.1.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.1.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

# 1.1.4.4.3.3.1.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.1.3.42. Высота элемента

Высота элемента, относительно которого открывается календарь.



int4 PointHeight

## 1.1.4.4.3.3.1.3.43. Дата

Дата, выделенная при открытии окна.



timestamp Date



## 1.1.4.4.3.3.1.3.44. Ссылка на результат

Ссылка на экземпляр типа [DateTimeResult](#), в котором будет сохранена выбранная дата.

## 1.1.4.4.3.3.1.3.45. pSelected

Ссылка на команду, выполняемую при выборе даты.

## 1.1.4.4.3.3.1.3.46. pOnClosed

Ссылка на команду, выполняемую при закрытии окна.

## 1.1.4.4.3.3.1.3.47. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.1.3.48. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



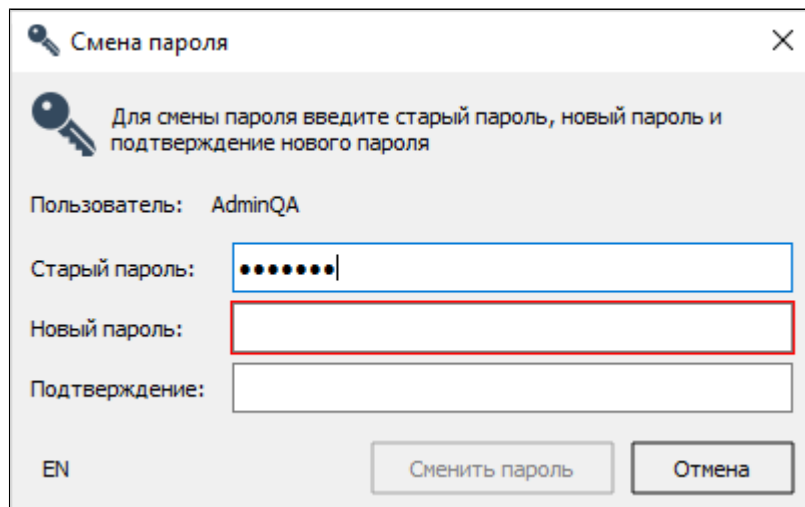
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.4.3.3.2. ChangePasswordDialog

Окно смены пароля текущего пользователя.



Смена пароля

Для смены пароля введите старый пароль, новый пароль и подтверждение нового пароля

Пользователь: AdminQA

Старый пароль: ●●●●●●●|

Новый пароль:

Подтверждение:

EN

Сменить пароль

Отмена

По умолчанию открывается при нажатии кнопки "Сменить пароль" в окне [LoginDialog](#).

## 1.1.4.4.3.2.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.2.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.2.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.2.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.2.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.2.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.2.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.2.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.2.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.2.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.2.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.2.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.2.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.2.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.2.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.2.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.2.1.16. Closed

Закрытие формы.



## 1.1.4.4.3.3.2.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.2.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.2.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.2.1.20. KeyPress

Нажатие определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



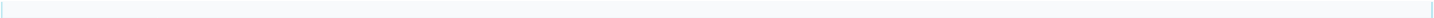
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.2.1.21. KeyRelease

Отпускание определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.2.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">FitWindowSize</a>	Подогнать по размеру окна
<a href="#">UpdateLayout</a>	Обновить макет





## 1.1.4.4.3.3.2.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.2.2. SetBoundingRegion

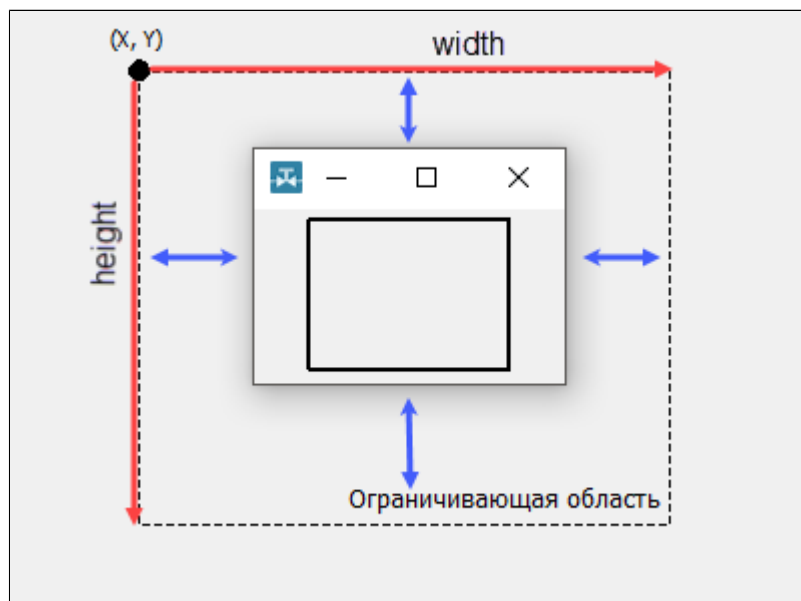
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`

## 1.1.4.4.3.3.2.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.2.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.2.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.2.2.6. SaveScreenShot

Формирует скриншот формы со всем ее видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```



## 1.1.4.4.3.3.2.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



void Print()

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.2.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.2.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.2.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.2.2.11. FormX

Получение координаты X на форме.



```
double FormX(double X, double Y)
```

### Параметры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.2.2.12. FormY

Получение координаты Y на форме.



```
double FormX(double X, double Y)
```

### Параметры



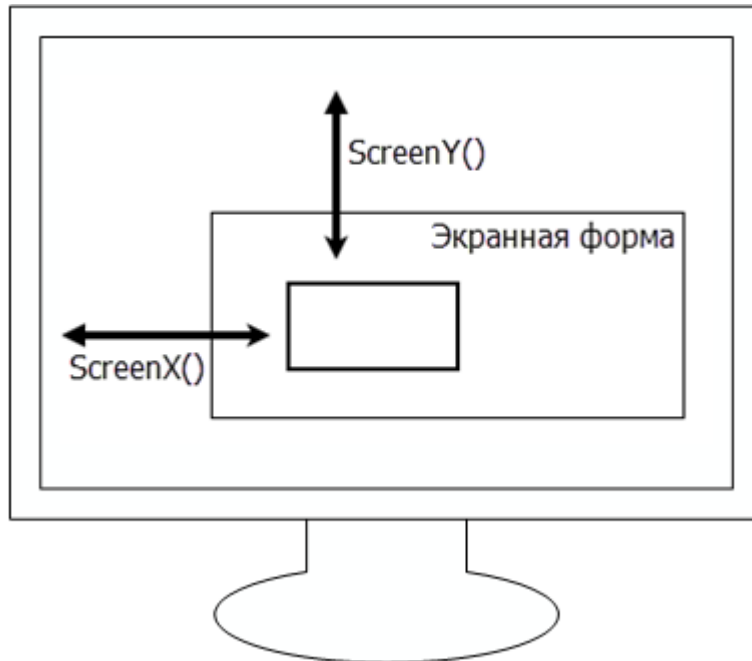
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.2.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



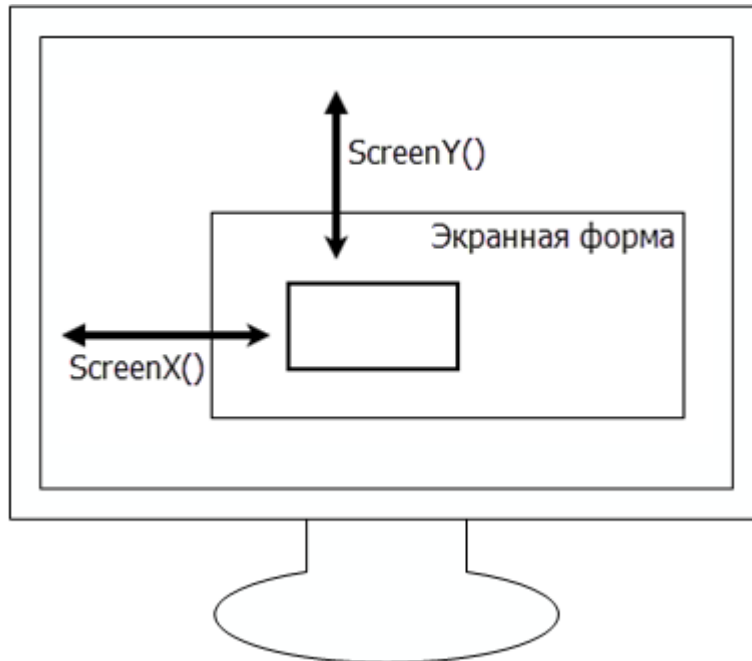
```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.2.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.2.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.2.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.2.2.17. FitWindowSize

Подогнать по размеру окна.



```
void FitWindowSize()
```

## 1.1.4.4.3.3.2.2.18. UpdateLayout




Обновить макет.



```
void UpdateLayout()
```

## 1.1.4.4.3.2.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна

<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">UserId</a>	Идентификатор текущего пользователя
<a href="#">UserDisplayName</a>	Отображаемое имя текущего пользователя
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве



## 1.1.4.4.3.3.2.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.2.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.2.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.2.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

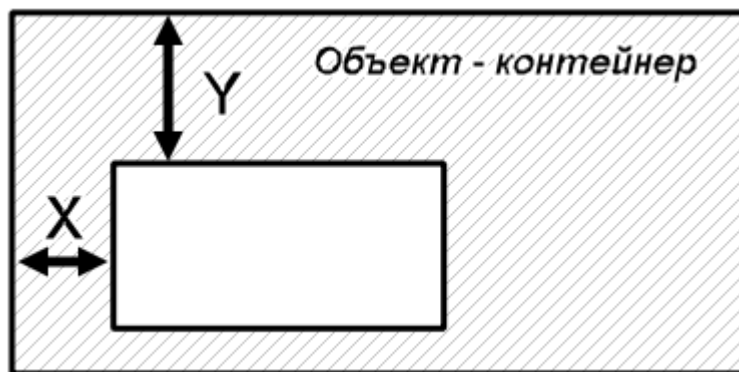
## 1.1.4.4.3.3.2.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.2.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.2.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.2.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```



## 1.1.4.4.3.3.2.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.2.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.2.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.2.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.2.3.13. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.2.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

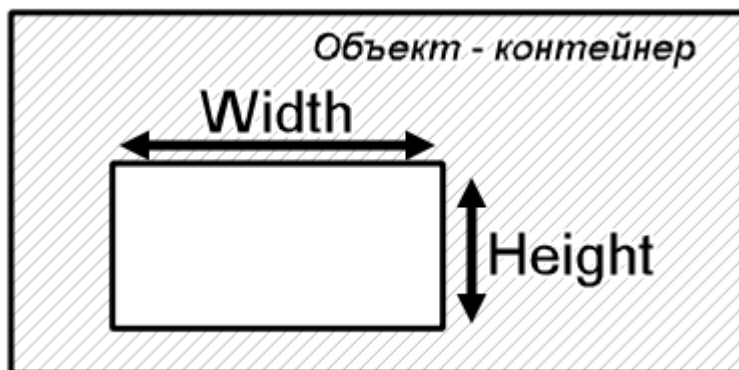
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.2.3.15. Ширина

Ширина объекта в пикселях.



float8 Width

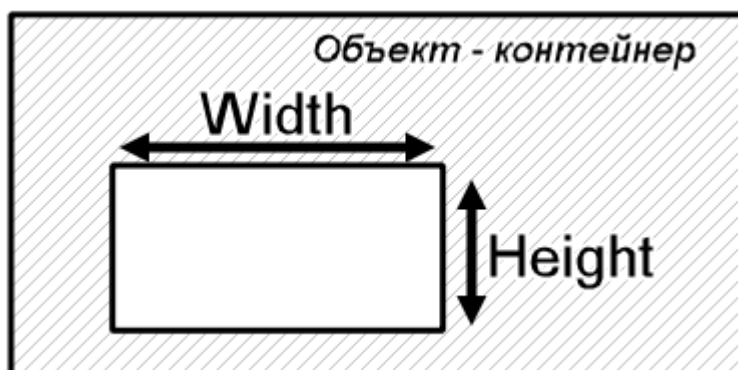


## 1.1.4.4.3.3.2.3.16. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.4.3.3.2.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.2.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.2.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.2.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.2.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;


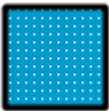

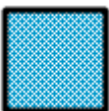

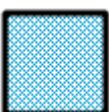
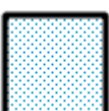
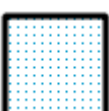
## 1.1.4.4.3.3.2.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.2.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```



## 1.1.4.4.3.3.2.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.2.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.2.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.2.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.2.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.2.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;

## 1.1.4.4.3.3.2.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

### Примеры



```
//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;
```

## 1.1.4.4.3.3.2.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```



## 1.1.4.4.3.3.2.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.2.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.2.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.2.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.2.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.2.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.2.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.2.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```



## 1.1.4.4.3.3.2.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

## 1.1.4.4.3.3.2.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.2.3.42. UserId

Хранит уникальный идентификатор (uid) текущего пользователя подсистемы безопасности Astra.Security.



string UserId

## 1.1.4.4.3.3.2.3.43. UserDisplayName

Хранит отображаемое имя текущего пользователя подсистемы безопасности Astra.Security.



string UserDisplayName

## 1.1.4.4.3.3.2.3.44. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.2.3.45. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



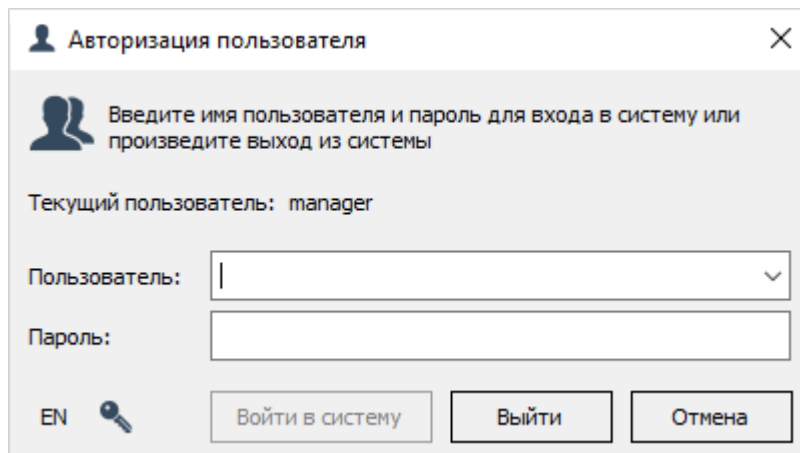
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3. LoginDialog

Окно авторизации пользователя.




Авторизация пользователя

Введите имя пользователя и пароль для входа в систему или произведите выход из системы

Текущий пользователь: manager

Пользователь:

Пароль:

EN 

Войти в систему Выйти Отмена

Чтобы не настраивать компонент самостоятельно, используйте [LoginButton](#).

## 1.1.4.4.3.3.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме



## 1.1.4.4.3.3.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.3.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.3.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.1.15. Opened

Открытие формы.



## 1.1.4.4.3.3.3.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.3.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.1.20. KeyPress

Нажатие определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



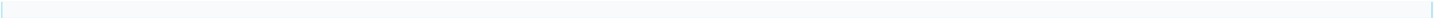
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.1.21. KeyRelease

Отпускание определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```



## 1.1.4.4.3.3.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">FitWindowSize</a>	Подогнать по размеру окна
<a href="#">UpdateLayout</a>	Обновить макет



## 1.1.4.4.3.3.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.2.2. SetBoundingRegion

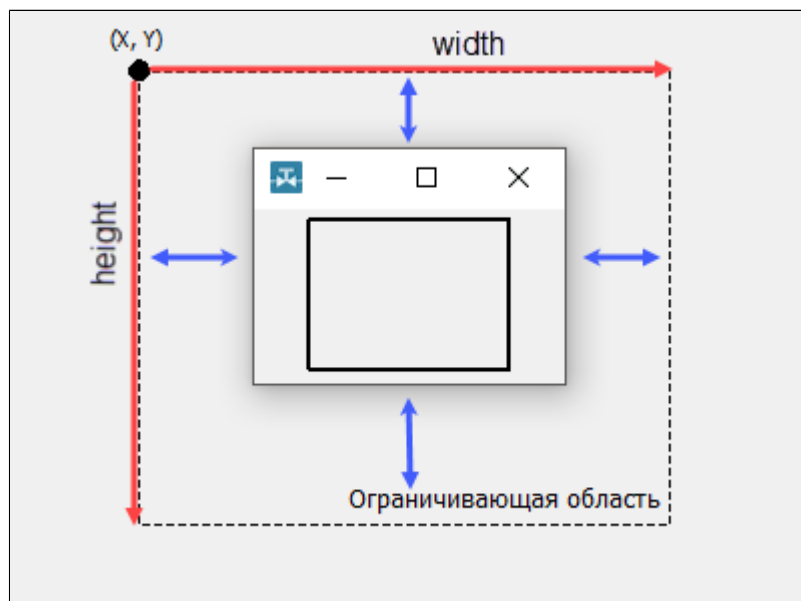
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
here.SetBoundingRegion(500, 200, 400, 400);

## 1.1.4.4.3.3.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```



## 1.1.4.4.3.3.2.6. SaveScreenShot

Формирует скриншот формы со всем ее видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



void Print()

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.3.2.8. Activate

Сделать окно формы активным.



```
void Activate()
```

## 1.1.4.4.3.3.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.2.11. FormX

Получение координаты X на форме.



```
double FormX(double X, double Y)
```

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.2.12. FormY

Получение координаты Y на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



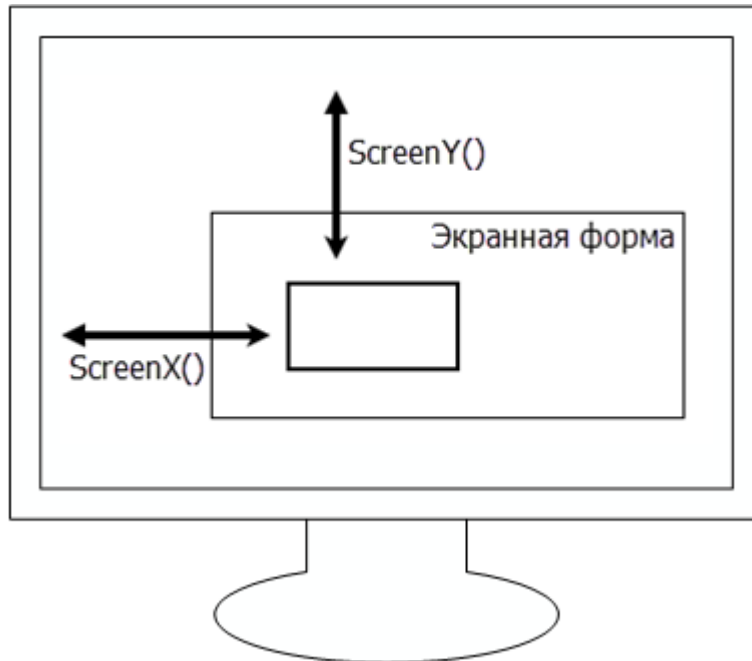
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



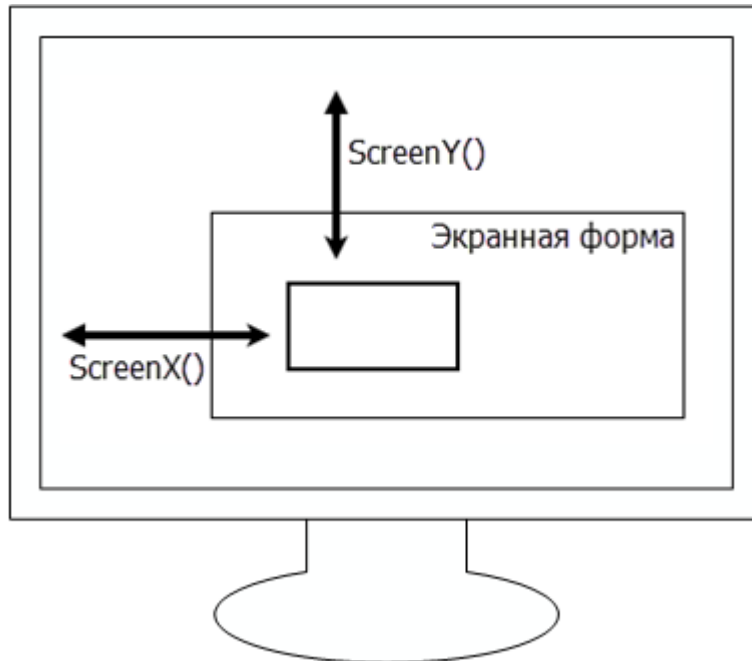
```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```



```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.3.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.3.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.3.2.17. FitWindowSize

Подогнать по размеру окна.



```
void FitWindowSize()
```

## 1.1.4.4.3.3.3.2.18. UpdateLayout




Обновить макет.



```
void UpdateLayout()
```

## 1.1.4.4.3.3.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна



<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">ChangePassword</a>	Ссылка на команду, открывающую окно <a href="#">ChangePasswordDialog</a>
<a href="#">Ссылка на результат</a>	Ссылка на <a href="#">LoginDialogResult</a> , в котором будут сохранены ID и отображаемое имя текущего пользователя
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

## 1.1.4.4.3.3.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

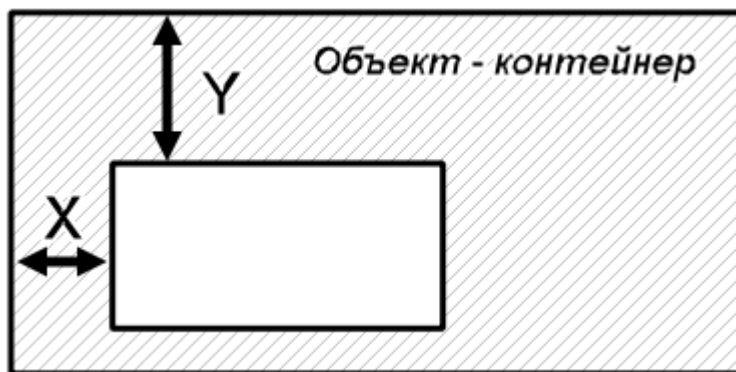
## 1.1.4.4.3.3.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.3.7. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.4.3.3.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;



//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;

## 1.1.4.4.3.3.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.3.13. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



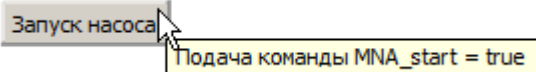
```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip



### Примеры



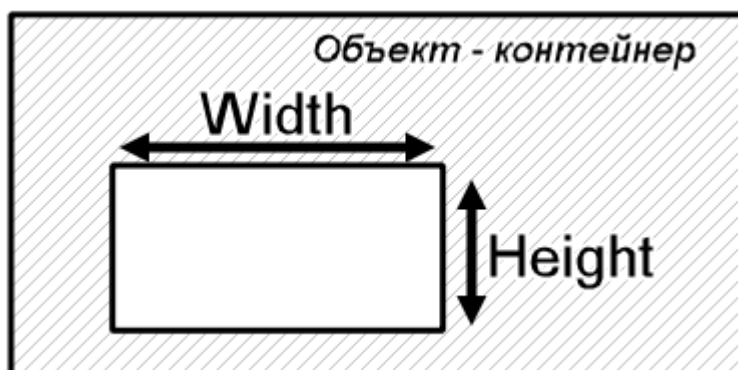
```
//Задать всплывающую подсказку для объекта:  
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.3.15. Ширина

Ширина объекта в пикселях.



float8 Width



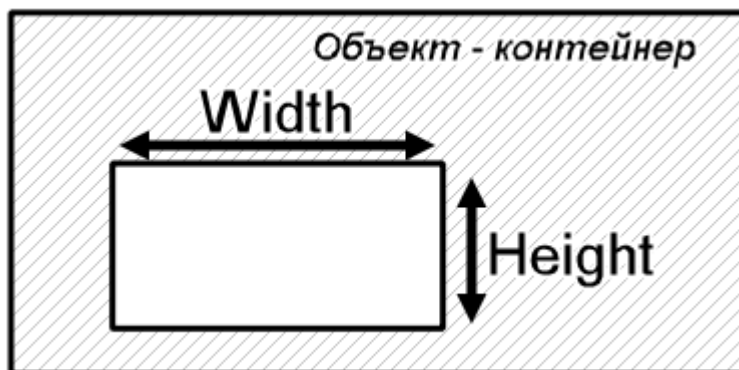


## 1.1.4.4.3.3.3.16. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;



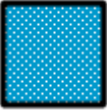
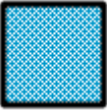

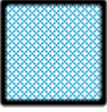
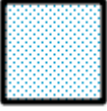
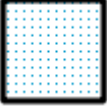
## 1.1.4.4.3.3.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```



## 1.1.4.4.3.3.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;

# 1.1.4.4.3.3.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;



## 1.1.4.4.3.3.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```



## 1.1.4.4.3.3.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

# 1.1.4.4.3.3.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.3.42. ChangePassword

Ссылка на команду, открывающую окно [ChangePasswordDialog](#).

## 1.1.4.4.3.3.3.43. Ссылка на результат

Ссылка на [LoginDialogResult](#), в котором будут сохранены ID и отображаемое имя текущего пользователя.

## 1.1.4.4.3.3.3.44. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.3.45. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



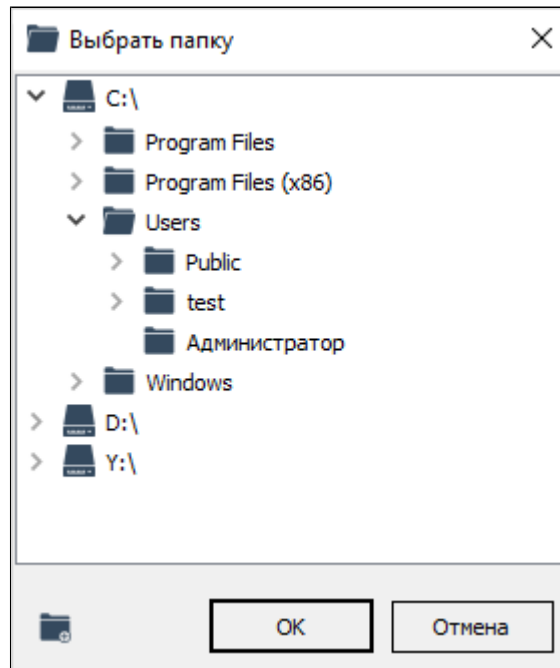
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.4. SelectFolderDialog

Окно просмотра файловой системы для выбора папки.



Используя компонент, создайте экземпляр типа [DialogResult](#), в котором будет сохранен результат выбора папки. Ссылка на созданный экземпляр указывается в свойстве [Ссылка на результат](#).



## 1.1.4.4.3.3.4.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.4.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.4.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.4.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.4.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.4.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.4.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.4.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.4.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.



## 1.1.4.4.3.3.4.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.4.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.4.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.4.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.4.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.4.1.20. KeyPress

Нажатие определенной клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



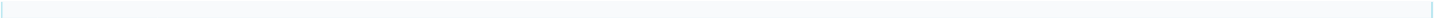
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.4.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```





//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.4.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">UpdateLayout</a>	Обновить макет



## 1.1.4.4.3.3.4.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.4.2.2. SetBoundingRegion

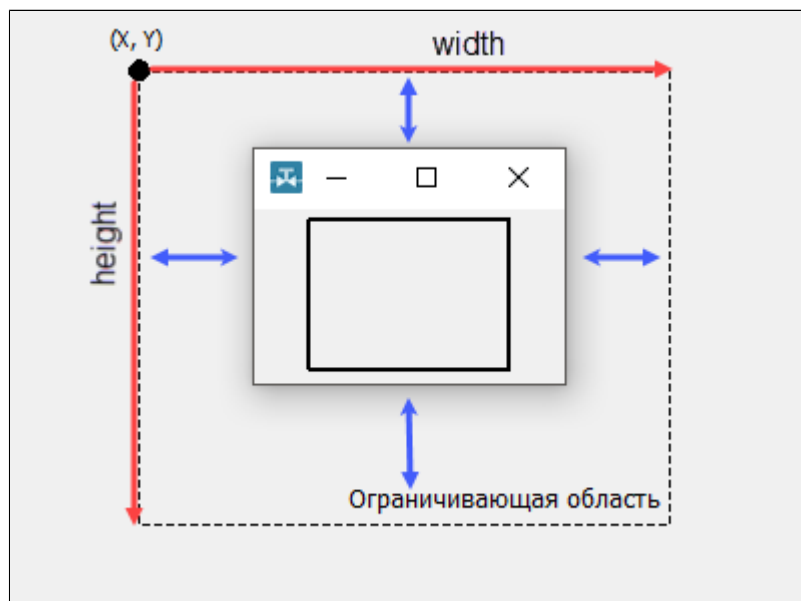
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`

## 1.1.4.4.3.3.4.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.4.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла



## 1.1.4.4.3.3.4.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.4.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.4.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



void Print()

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.4.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.4.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.4.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.4.2.11. FormX

Получение координаты X на форме.



```
double FormX(double X, double Y)
```

### Параметры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.4.2.12. FormY

Получение координаты Y на форме.



```
double FormX(double X, double Y)
```

### Параметры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

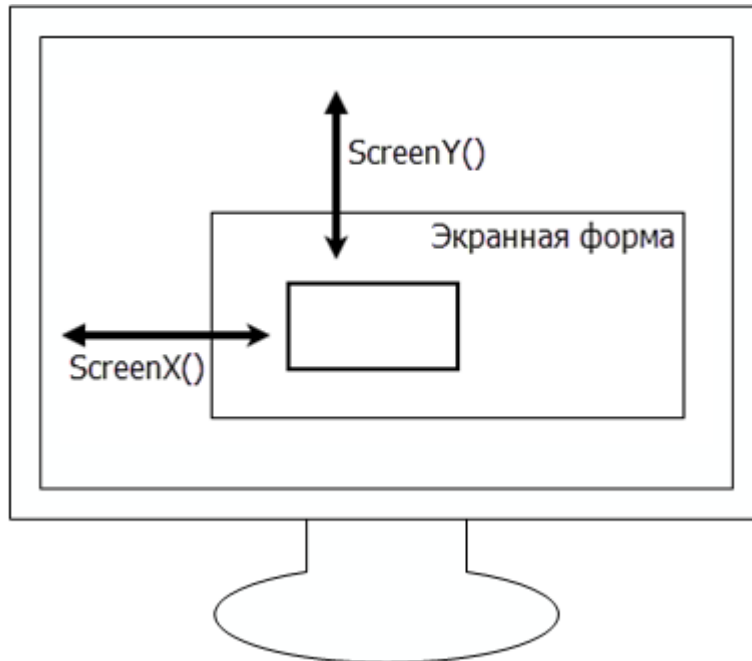
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.4.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

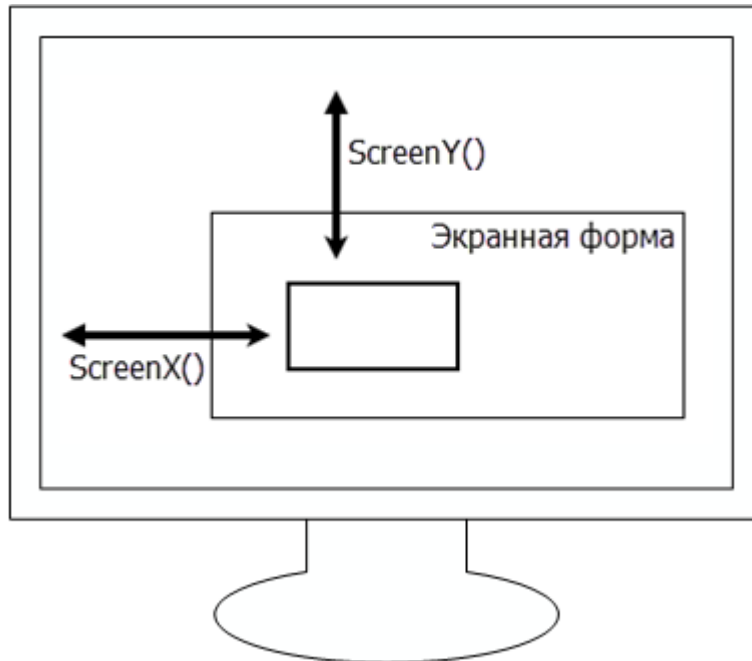


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.4.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.4.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.4.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.4.2.17. UpdateLayout




Обновить макет.



void UpdateLayout()

## 1.1.4.4.3.3.4.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна



<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Корневой путь</a>	Ограничивает область просмотра файловой системы
<a href="#">Путь по умолчанию</a>	Путь по умолчанию
<a href="#">Разрешено создание папок</a>	Указывает на возможность создания папок
<a href="#">Ссылка на результат</a>	Ссылка на экземпляр типа <a href="#">DialogResult</a> , в котором будет сохранено имя и тип выбранного файла
<a href="#">OnComplete</a>	Ссылка на команду, выполняемую в результате нажатия кнопки Открыть и закрытии окна <a href="#">SelectFolderDialog</a>
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.4.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.4.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

## 1.1.4.4.3.3.4.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.4.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.4.3.7. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.4.3.3.4.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.4.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.4.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.4.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.4.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.4.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.4.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

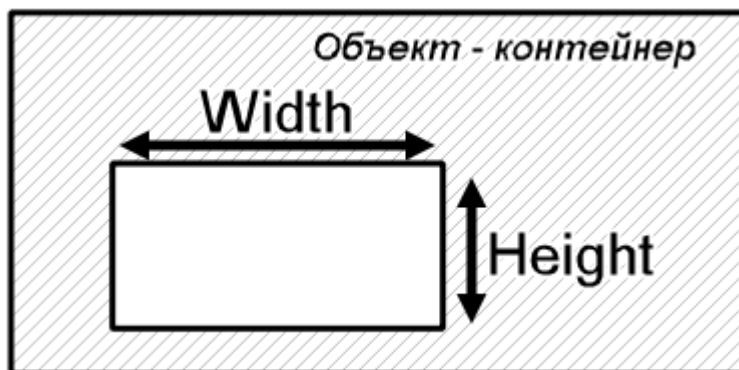
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.4.3.15. Ширина

Ширина объекта в пикселях.



float8 Width



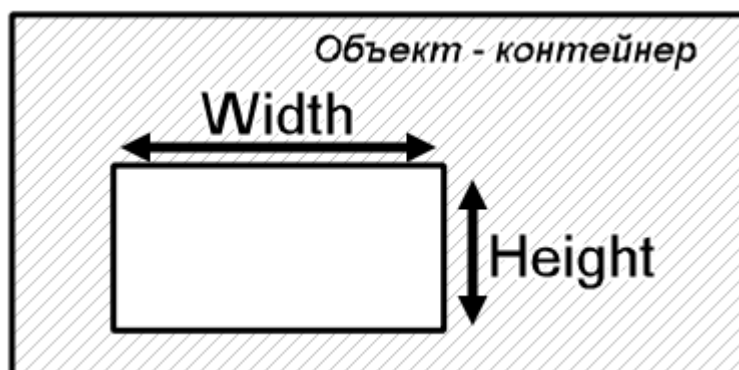


## 1.1.4.4.3.3.4.3.16. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.4.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.4.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```


## 1.1.4.4.3.3.4.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.4.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.4.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;



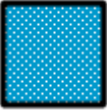
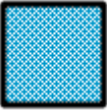

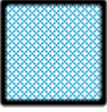
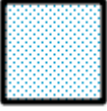
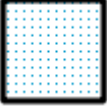
## 1.1.4.4.3.3.4.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;



## 1.1.4.4.3.3.4.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.4.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.4.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.4.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.4.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.4.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.4.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.4.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



```
//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;
```



## 1.1.4.4.3.3.4.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.4.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.4.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.4.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.4.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.4.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.4.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.4.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```



## 1.1.4.4.3.3.4.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.4.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

## 1.1.4.4.3.3.4.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.4.3.42. Корневой путь

Ограничивает область просмотра файловой системы.



string RootPath

## 1.1.4.4.3.3.4.3.43. Путь по умолчанию

Путь по умолчанию. Если указать значение, то при открытии окна `SelectFolderDialog` всегда будет открыт указанный каталог.



string `DefaultPath`

## 1.1.4.4.3.3.4.3.44. Разрешено создание папок

Указывает на возможность создания папок в окне [SelectFolderDialog](#).



bool CreateFolderAllowed

### Значение

Значение	Описание
true	Разрешено создавать папки
false	Запрещено создавать папки

## 1.1.4.4.3.3.4.3.45. Ссылка на результат

Ссылка на экземпляр типа [DialogResult](#), в котором будет сохранено имя и тип выбранного файла.

## 1.1.4.4.3.3.4.3.46. OnComplete

Ссылка на команду, выполняемую в результате нажатия кнопки Открыть и закрытии окна [SelectFolderDialog](#).



## 1.1.4.4.3.3.4.3.47. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.4.3.48. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



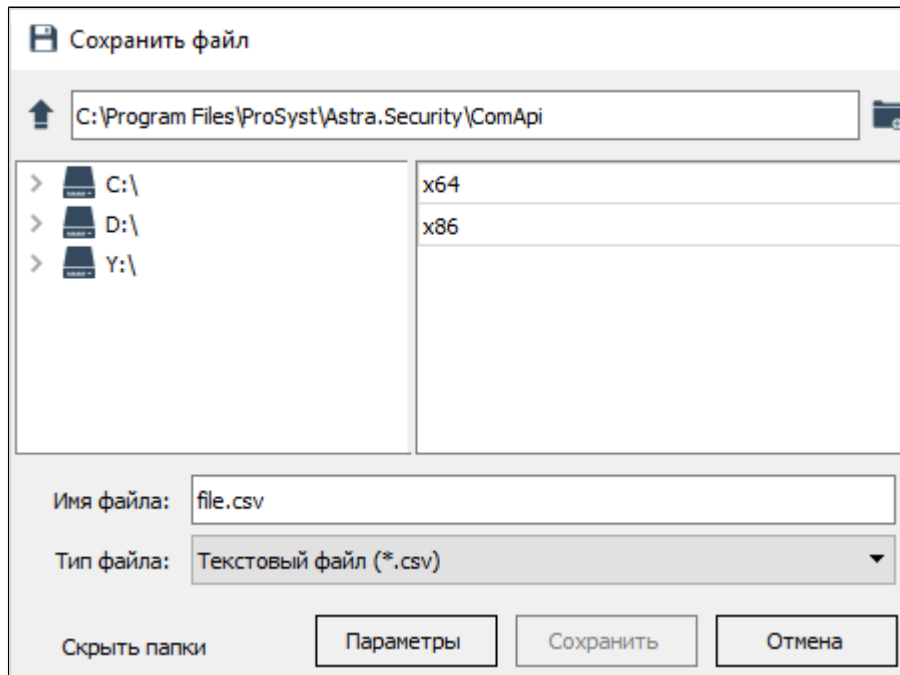
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.5. SaveFileDialog

Окно просмотра файловой системы для сохранения файлов.



Используя компонент, создайте экземпляр типа [FileDialogResult](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве [Ссылка на результат](#).

## 1.1.4.4.3.3.5.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.5.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.5.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.5.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.5.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.5.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```



## 1.1.4.4.3.3.5.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.5.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.5.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.5.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.5.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.5.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.5.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.5.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна



## 1.1.4.4.3.3.5.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



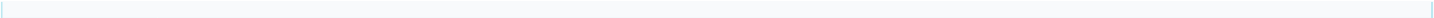
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.5.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.5.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">UpdateLayout</a>	Обновить макет
<a href="#">CoerseFileName</a>	
<a href="#">GetFolderPath</a>	Получить путь к папке



## 1.1.4.4.3.3.5.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.5.2.2. SetBoundingRegion

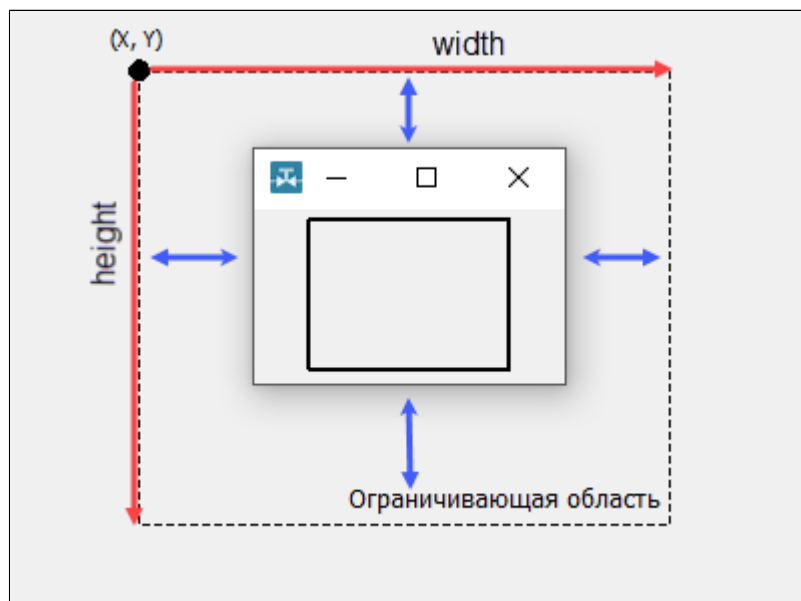
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры





//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`

## 1.1.4.4.3.3.5.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.5.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.5.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.5.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.5.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.5.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.5.2.9. Raise

Поднять окно формы на передний план.



void Raise()



## 1.1.4.4.3.3.5.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.5.2.11. FormX

Получение координаты X на форме.



```
double FormX(double X, double Y)
```

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.5.2.12. FormY

Получение координаты Y на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



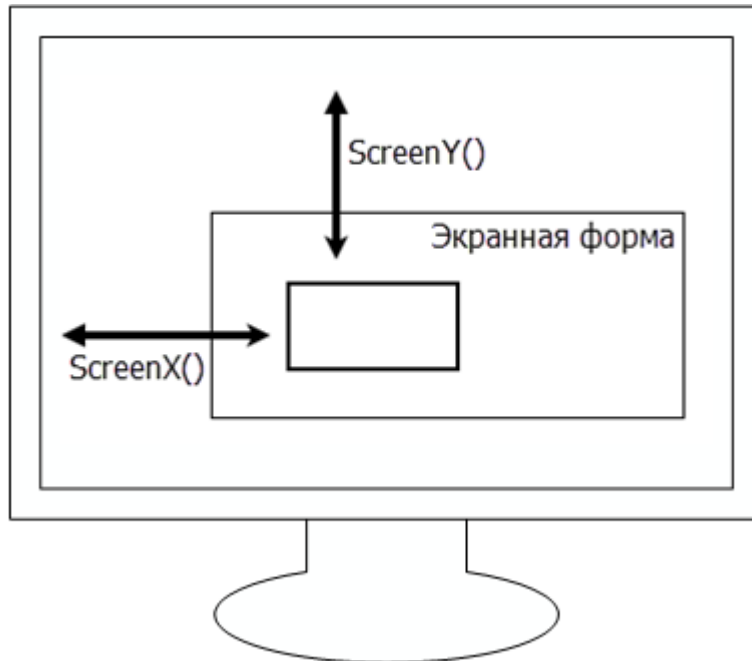
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.5.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

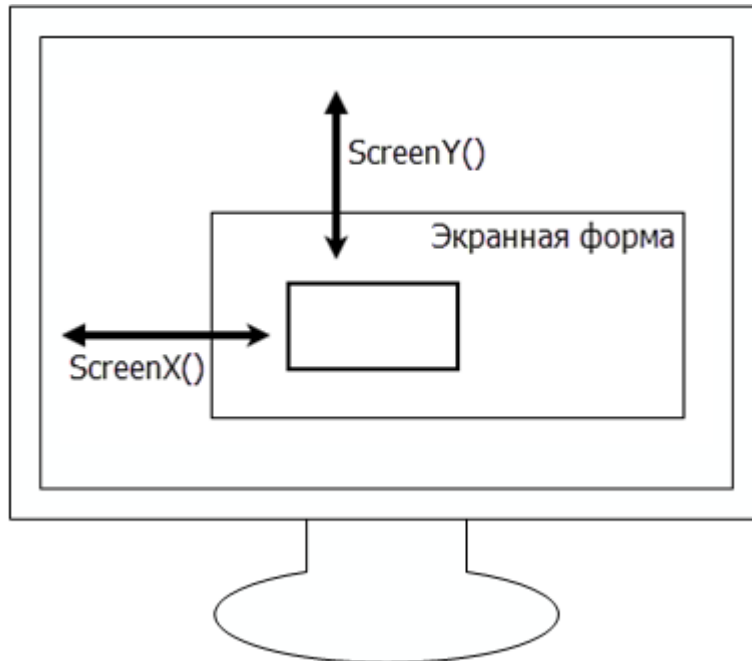


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.5.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.5.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.5.2.16. Hide

Скрыть окно формы.



void Hide()



# 1.1.4.4.3.3.5.2.17. UpdateLayout

Обновить макет.



void UpdateLayout()

## 1.1.4.4.3.3.5.2.18. CoerseFileName



string CoerseFileName(string FileName, string OldExt, string NewExt)

### Параметры

Параметр	Тип	Описание
FileName	string	
OldExt	string	
NewExt	string	

## 1.1.4.4.3.3.5.2.19. GetFolderPath

Получить путь к папке.



```
string GetFolderPath(string FilePath)
```

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь к файлу

## 1.1.4.4.3.3.5.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта

<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы

<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна
<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">OnPathChosen</a>	Ссылка на команду, выполняемую в момент выбора строки файла в списке файлов окна OpenFileDialog
<a href="#">Корневой путь</a>	Ограничивает область просмотра файловой системы
<a href="#">Путь по умолчанию</a>	Путь по умолчанию
<a href="#">Имя файла по умолчанию</a>	Имя файла по умолчанию
<a href="#">Фильтр типа файлов</a>	Фильтр отображаемого типа файлов
<a href="#">Индекс типа в фильтре</a>	Устанавливает тип файлов по умолчанию в окне OpenFileDialog
<a href="#">Показывать свойства файлов</a>	Позволяет отображать больше информации о файлах в окне OpenFileDialog
<a href="#">Разрешена смена папки</a>	Указывает на возможность смены папки, открываемой в окне OpenFileDialog по умолчанию
<a href="#">Разрешено создание папок</a>	Указывает на возможность создания папок в окне OpenFileDialog
<a href="#">Отображать панель просмотра</a>	Ограничивает возможность просмотра иерархии папок и уменьшает размеры диалога
<a href="#">Ссылка на результат</a>	Ссылка на экземпляр типа DialogResult, в котором будет сохранено имя и тип выбранного файла
<a href="#">BottomPanelHeight</a>	Высота нижней панели
<a href="#">Отображать кнопку параметры</a>	Меняет видимость кнопки Параметры в окне SaveFileDialog
<a href="#">OnEditSettings</a>	Ссылка на команду, выполняемую при нажатии на кнопку Параметры в окне SaveFileDialog

<a href="#">OnComplete</a>	Ссылка на команду, выполняемую в результате нажатия кнопки Открыть и закрытии окна OpenFileDialog
<a href="#">Lenght</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.5.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.4.3.3.5.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.5.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.5.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

## 1.1.4.4.3.3.5.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.5.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.5.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.5.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.5.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```



## 1.1.4.4.3.3.5.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.5.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.5.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.5.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



//Активировать объект в зависимости от выполнения условия:

```
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.5.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

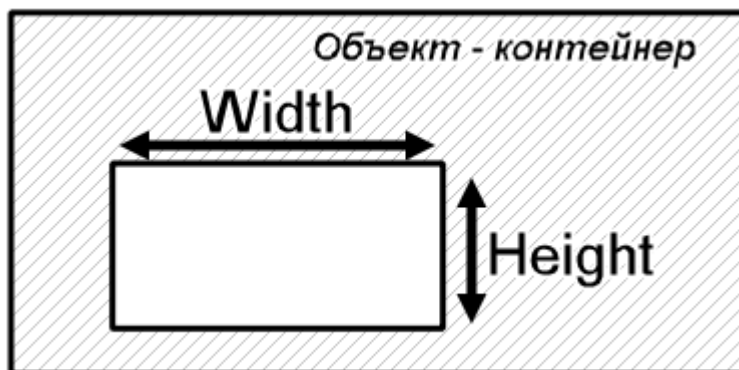
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.5.3.15. Ширина

Ширина объекта в пикселях.



float8 Width

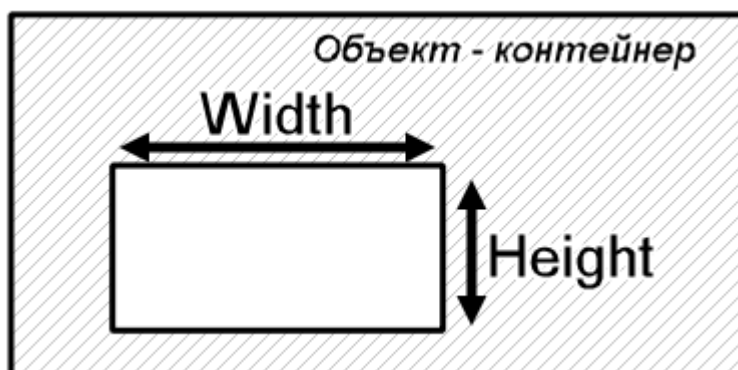


## 1.1.4.4.3.3.5.3.16. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.5.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```



## 1.1.4.4.3.3.5.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.5.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.5.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.5.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





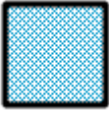
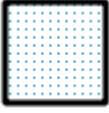
## 1.1.4.4.3.3.5.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.5.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.5.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```



## 1.1.4.4.3.3.5.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.5.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.5.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.5.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.5.3.29. "свернуть"

## Показывать

## кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

## 1.1.4.4.3.3.5.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;

## 1.1.4.4.3.3.5.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.5.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```



## 1.1.4.4.3.3.5.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.5.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.5.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.5.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.5.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.5.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.5.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.5.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath



## 1.1.4.4.3.3.5.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.5.3.42. OnPathChosen

Ссылка на команду, выполняемую в момент выбора строки файла в списке файлов окна [SaveFileDialog](#).

## 1.1.4.4.3.3.5.3.43. Корневой путь

Ограничивает область просмотра файловой системы.



string RootPath

## 1.1.4.4.3.3.5.3.44. Путь по умолчанию

Путь по умолчанию. Если указать значение, то при открытии окна [SaveFileDialog](#) всегда будет открыт указанный каталог.



string DefaultPath

## 1.1.4.4.3.3.5.3.45. Имя файла по умолчанию

Имя файла по умолчанию. Значение свойства подставляется в поле Имя файла при открытии окна [SaveFileDialog](#).



string DefaultFileName

- › Если в значении указать расширение файла (например example.xml), значение будет подставлено в поле в исходном виде.
- › Если в значении не указывать расширение (например example), к значению будет добавлено расширение, выбранное из [Фильтр типа файлов](#).

## 1.1.4.4.3.3.5.3.46. Фильтр типа файлов

Фильтр отображаемого типа файлов.



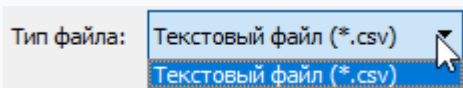
string FileTypeFilter

Формат указания значения: Описание|расширение. Чтобы указать несколько типов, перечислите их через вертикальную черту (|).

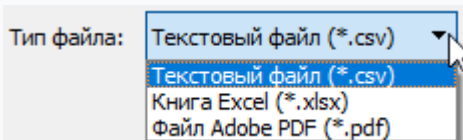
- Если указать значение, то в нижней части окна SaveFileDialog будет доступен выпадающий список – фильтр отображения файлов.



"Текстовый файл (\*.csv) | \*.csv"



"Текстовый файл (\*.csv) | \*.csv | Книга Excel (\*.xlsx) | \*.xlsx | Файл Adobe PDF (\*.pdf) | \*.pdf"



- Если оставить значение свойства пустым, то при открытии окна будут отображаться все файлы и папки. Выпадающий список с фильтром будет недоступен.

## 1.1.4.4.3.3.5.3.47. Индекс типа в фильтре

Устанавливает тип файлов по умолчанию в окне [SaveFileDialog](#).



int4 DefaultFileType

Значение – номер записи типа в свойстве [Фильтр типа файлов](#) (нумерация ведется с 0).

### Примеры

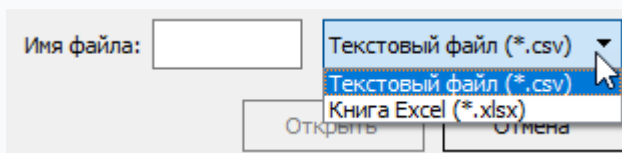


Допустим, в качестве значения свойства [Фильтр типа файлов](#) указана строка:

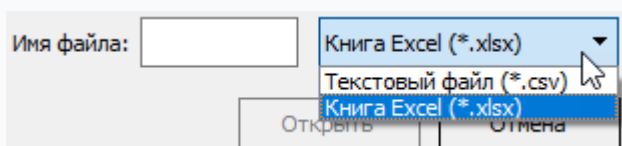
Текстовый файл (\*.csv) | \*.csv | Книга Excel (\*.xlsx) | \*.xlsx

Тогда для DefaultFileType можно указать:

- › 0 – при открытии окна [SaveFileDialog](#) по умолчанию будет предложено открыть файл типа Текстовый файл (\*.csv)



- › 1 – при открытии окна [SaveFileDialog](#) по умолчанию будет предложено открыть файл типа Книга Excel (\*.xlsx) | \*.xlsx



## 1.1.4.4.3.3.5.3.48. Показывать свойства файлов

Позволяет отображать больше информации о файлах в окне [SaveFileDialog](#).



bool ShowFileProperties

### Значение

Значение	Описание												
true	Отображаются имена, даты изменения и размеры файлов (в байтах) <table border="1"><thead><tr><th>Имя</th><th>Дата изменения</th><th>Размер</th></tr></thead><tbody><tr><td>old_backups</td><td>17.11.2022 14:12:29</td><td></td></tr><tr><td>17112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr><tr><td>18112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr></tbody></table>	Имя	Дата изменения	Размер	old_backups	17.11.2022 14:12:29		17112022_bckp.csv	17.11.2022 14:12:53	6377	18112022_bckp.csv	17.11.2022 14:12:53	6377
Имя	Дата изменения	Размер											
old_backups	17.11.2022 14:12:29												
17112022_bckp.csv	17.11.2022 14:12:53	6377											
18112022_bckp.csv	17.11.2022 14:12:53	6377											
false	Отображаются только имена файлов <table border="1"><tbody><tr><td>old_backups</td></tr><tr><td>17112022_bckp.csv</td></tr><tr><td>18112022_bckp.csv</td></tr></tbody></table>	old_backups	17112022_bckp.csv	18112022_bckp.csv									
old_backups													
17112022_bckp.csv													
18112022_bckp.csv													



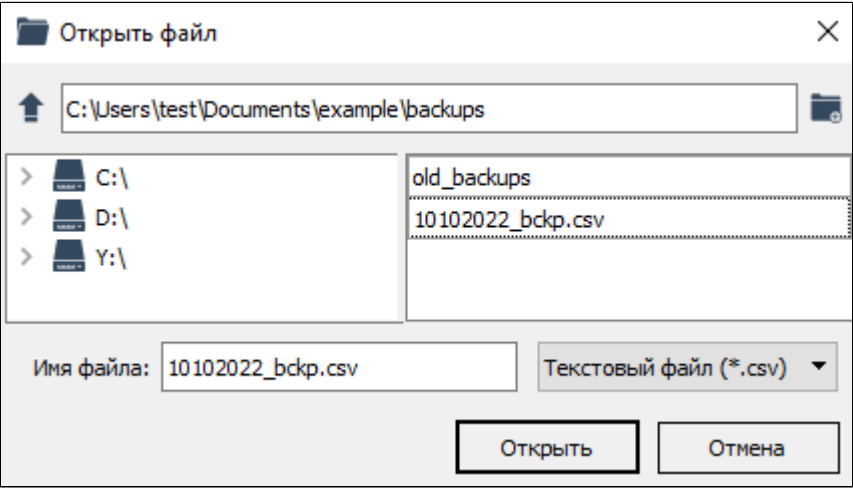
## 1.1.4.4.3.3.5.3.49. Разрешена смена папки

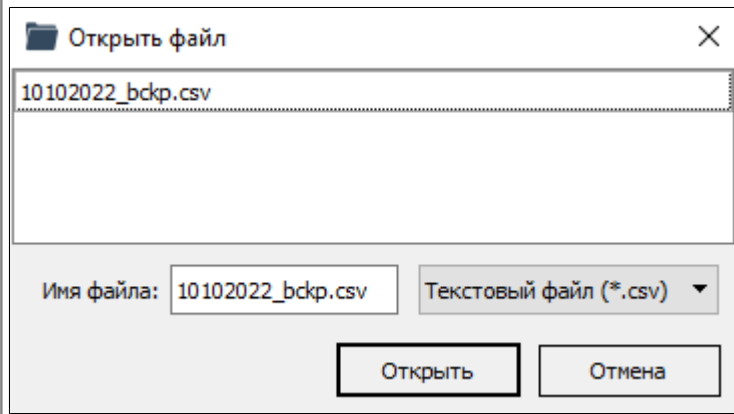
Указывает на возможность смены папки, открываемой в окне [SaveFileDialog](#) по умолчанию.



bool ChangedFolderAllowed

### Значение

Значение	Описание
true	<p>Разрешено менять папку</p> <p>В этом случае в окне SaveFileDialog отображается дерево каталогов и их содержимое (файлы и другие папки). Есть возможность перемещаться между каталогами и создавать новые папки.</p> 
false	<p>Запрещено менять папку</p> <p>В этом случае в окне SaveFileDialog отображаются только файлы папки, открываемой по умолчанию. Дерево каталогов скрыто, нет возможности создавать новые папки.</p>



## 1.1.4.4.3.3.5.3.50. Разрешено создание папок

Указывает на возможность создания папок в окне [SaveFileDialog](#).



bool CreateFolderAllowed

### Значение

Значение	Описание
true	Разрешено создавать папки
false	Запрещено создавать папки

## 1.1.4.4.3.3.5.3.51. просмотра

## Отображать

## панель

Ограничивает возможность просмотра иерархии папок и уменьшает размеры диалога.



bool ShowBrowsePanel

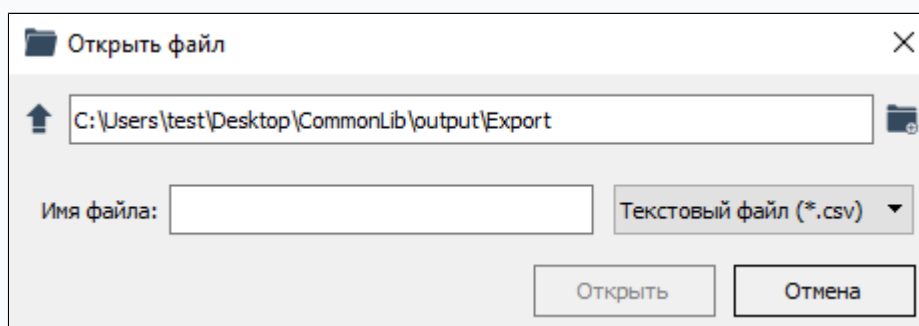
## Значение

Значение	Описание
true	Просмотр иерархии папок разрешен
false	Просмотр иерархии папок запрещен

## Примеры



Внешний вид окна, если значение свойства – false:



## 1.1.4.4.3.3.5.3.52. Ссылка на результат

Ссылка на экземпляр типа [FileDialogResult](#), в котором будет сохранено имя и тип выбранного файла.

## 1.1.4.4.3.3.5.3.53. BottomPanelHeight

Высота нижней панели.



float8 BottomPanelHeight

# 1.1.4.4.3.3.5.3.54. Параметры

## Отображать

## кнопку

Меняет видимость кнопки Параметры в окне [SaveFileDialog](#).



bool EditSettingsAllowed

При нажатии на кнопку активируется команда [OnEditSettings](#).

## Значение

Значение	Описание
true	Кнопка есть
false	Кнопки нет

## 1.1.4.4.3.3.5.3.55. OnEditSettings

Ссылка на команду, выполняемую при нажатии на кнопку Параметры в окне [SaveFileDialog](#).

Видимость кнопки зависит от значения свойства [Отображать кнопку Параметры](#).



## 1.1.4.4.3.3.5.3.56. OnComplete

Ссылка на команду, выполняемую в результате нажатия кнопки Сохранить и закрытии окна [SaveFileDialog](#).

## 1.1.4.4.3.3.5.3.57. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.5.3.58. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

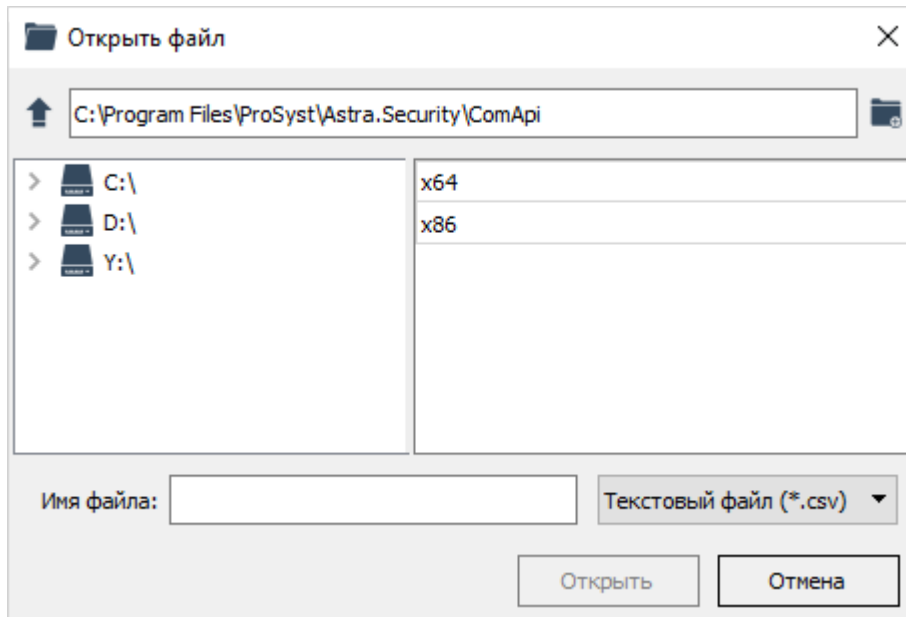


//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.6. OpenFileDialog

Окно просмотра файловой системы для открытия файлов.



Используя компонент, создайте экземпляр типа [FileDialogResult](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве [Ссылка на результат](#).

## 1.1.4.4.3.3.6.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.6.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.6.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.6.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.6.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```



## 1.1.4.4.3.3.6.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.6.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.6.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.6.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.6.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.6.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.6.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.6.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна



## 1.1.4.4.3.3.6.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.6.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



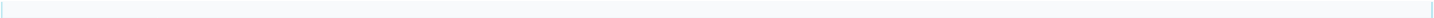
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.6.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.6.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">UpdateLayout</a>	Обновить макет
<a href="#">CoerseFileName</a>	
<a href="#">GetFolderPath</a>	Получить путь к папке



## 1.1.4.4.3.3.6.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```



## 1.1.4.4.3.3.6.2.2. SetBoundingRegion

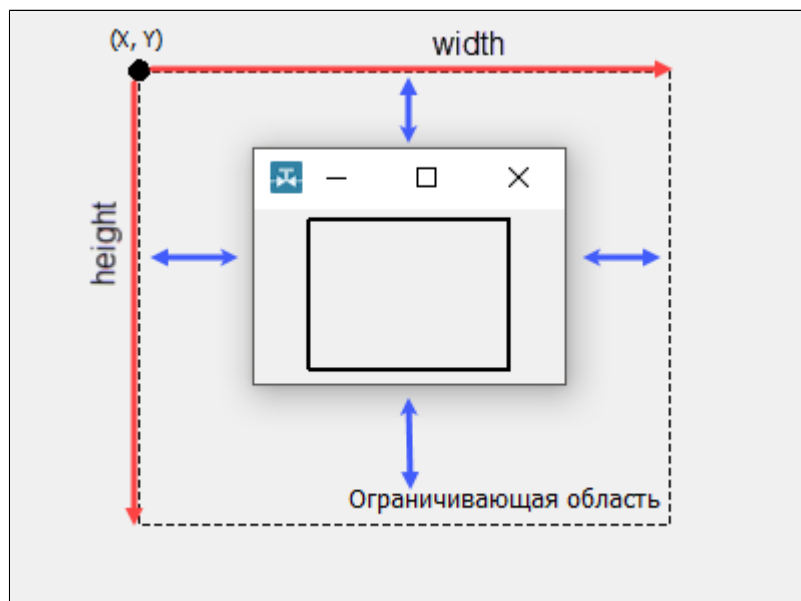
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`

## 1.1.4.4.3.3.6.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.6.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.6.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.6.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.6.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.6.2.8. Activate

Сделать окно формы активным.



void Activate()



## 1.1.4.4.3.3.6.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.6.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.6.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.6.2.12. FormY

Получение координаты Y на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



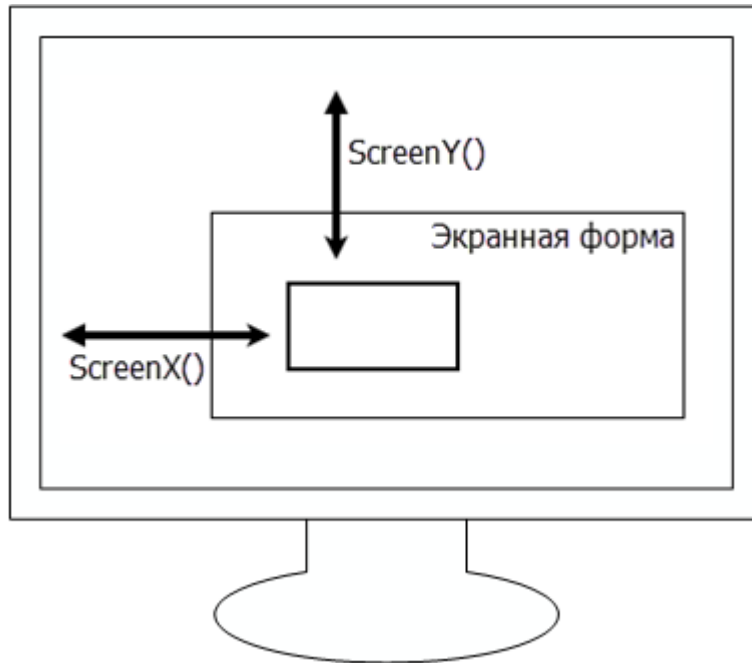
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.6.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

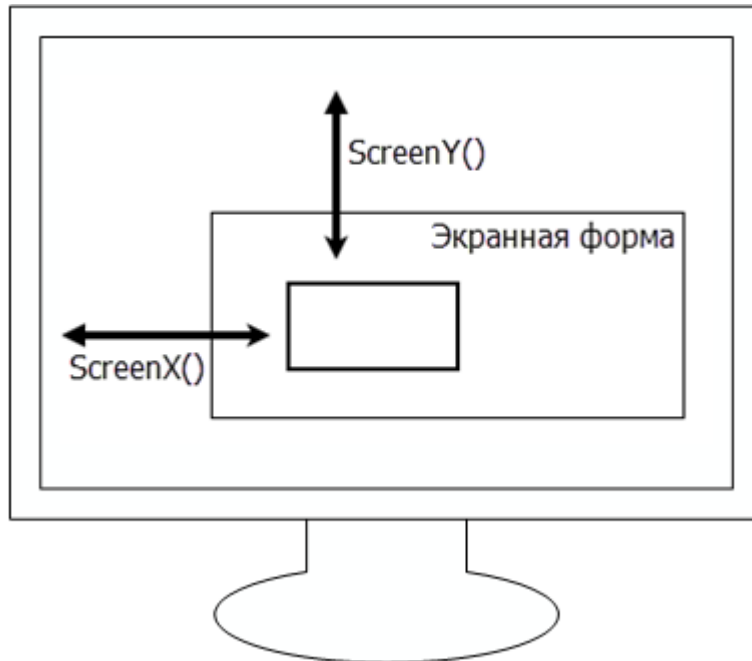


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.6.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.6.2.15. Show

Показать окно формы.



void Show()



## 1.1.4.4.3.3.6.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.6.2.17. UpdateLayout

Обновить макет.



void UpdateLayout()

## 1.1.4.4.3.3.6.2.18. CoerseFileName



string CoerseFileName(string FileName, string OldExt, string NewExt)

### Параметры

Параметр	Тип	Описание
FileName	string	
OldExt	string	
NewExt	string	

## 1.1.4.4.3.3.6.2.19. GetFolderPath

Получить путь к папке.



string GetFolderPath(string FilePath)

### Параметры

Параметр	Тип	Описание
FilePath	string	Путь к файлу

## 1.1.4.4.3.3.6.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта

<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы

<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна
<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">OnPathChosen</a>	Ссылка на команду, выполняемую в момент выбора строки файла в списке файлов окна OpenFileDialog
<a href="#">Корневой путь</a>	Ограничивает область просмотра файловой системы
<a href="#">Путь по умолчанию</a>	Путь по умолчанию
<a href="#">Имя файла по умолчанию</a>	Имя файла по умолчанию
<a href="#">Фильтр типа файлов</a>	Фильтр отображаемого типа файлов
<a href="#">Индекс типа в фильтре</a>	Устанавливает тип файлов по умолчанию в окне OpenFileDialog
<a href="#">Показывать свойства файлов</a>	Позволяет отображать больше информации о файлах в окне OpenFileDialog
<a href="#">Разрешена смена папки</a>	Указывает на возможность смены папки, открываемой в окне OpenFileDialog по умолчанию
<a href="#">Разрешено создание папок</a>	Указывает на возможность создания папок в окне OpenFileDialog
<a href="#">Отображать панель просмотра</a>	Ограничивает возможность просмотра иерархии папок и уменьшает размеры диалога
<a href="#">Ссылка на результат</a>	Ссылка на экземпляр типа DialogResult, в котором будет сохранено имя и тип выбранного файла
<a href="#">BottomPanelHeight</a>	Высота нижней панели
<a href="#">Отображать кнопку параметры</a>	Меняет видимость кнопки Параметры в окне SaveFileDialog
<a href="#">OnEditSettings</a>	Ссылка на команду, выполняемую при нажатии на кнопку Параметры в окне SaveFileDialog

<a href="#">OnComplete</a>	Ссылка на команду, выполняемую в результате нажатия кнопки Открыть и закрытии окна OpenFileDialog
<a href="#">Lenght</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве



## 1.1.4.4.3.3.6.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.6.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.6.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.6.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

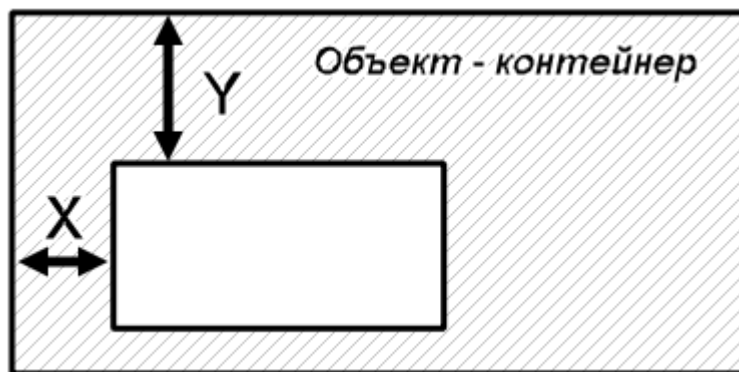
## 1.1.4.4.3.3.6.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.6.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.6.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.6.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;



//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;



## 1.1.4.4.3.3.6.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.6.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.6.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.6.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.6.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.6.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

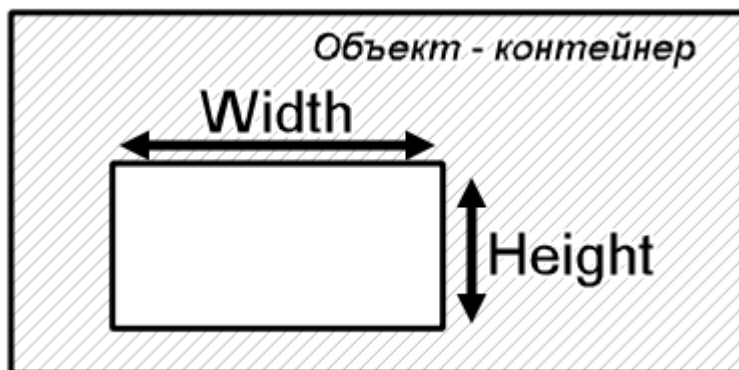
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.6.3.15. Ширина

Ширина объекта в пикселях.



float8 Width

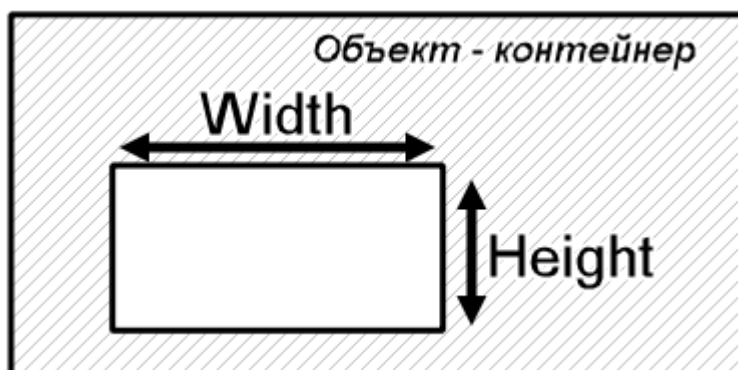


## 1.1.4.4.3.3.6.3.16. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.4.3.3.6.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.6.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.6.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.6.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.6.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.4.3.3.6.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.6.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```



## 1.1.4.4.3.3.6.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.6.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.6.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.6.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.6.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.6.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.6.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;

## 1.1.4.4.3.3.6.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```



## 1.1.4.4.3.3.6.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.6.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.6.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.6.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.6.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.6.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.6.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.6.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```



## 1.1.4.4.3.3.6.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

## 1.1.4.4.3.3.6.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.6.3.42. OnPathChosen

Ссылка на команду, выполняемую в момент выбора строки файла в списке файлов окна SaveFileDialog.

## 1.1.4.4.3.3.6.3.43. Корневой путь

Ограничивает область просмотра файловой системы.



string RootPath

## 1.1.4.4.3.3.6.3.44. Путь по умолчанию

Путь по умолчанию. Если указать значение, то при открытии окна SaveFileDialog всегда будет открыт указанный каталог.



string DefaultPath

## 1.1.4.4.3.3.6.3.45. Имя файла по умолчанию

Имя файла по умолчанию. Значение свойства подставляется в поле Имя файла при открытии окна `SaveFileDialog`.



`string DefaultFileName`

- › Если в значении указать расширение файла (например `example.xml`), значение будет подставлено в поле в исходном виде.
- › Если в значении не указывать расширение (например `example`), к значению будет добавлено расширение, выбранное из `FileTypeFilter` (Фильтр типа файлов).

## 1.1.4.4.3.3.6.3.46. Фильтр типа файлов

Фильтр отображаемого типа файлов.



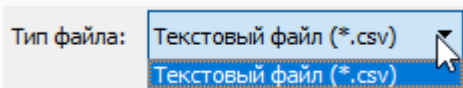
string FileTypeFilter

Формат указания значения: Описание|расширение. Чтобы указать несколько типов, перечислите их через вертикальную черту (|).

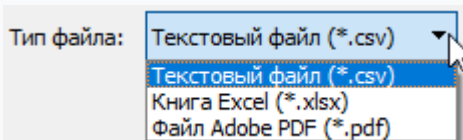
- Если указать значение, то в нижней части окна SaveFileDialog будет доступен выпадающий список – фильтр отображения файлов.



"Текстовый файл (\*.csv) | \*.csv"



"Текстовый файл (\*.csv) | \*.csv | Книга Excel (\*.xlsx) | \*.xlsx | Файл Adobe PDF (\*.pdf) | \*.pdf"



- Если оставить значение свойства пустым, то при открытии окна будут отображаться все файлы и папки. Выпадающий список с фильтром будет недоступен.

## 1.1.4.4.3.3.6.3.47. Индекс типа в фильтре

Устанавливает тип файлов по умолчанию в окне SaveFileDialog.



int4 DefaultFileType

Значение – номер записи типа в свойстве FileTypeFilter (Фильтр типа файлов) (нумерация ведется с 0).

### Примеры

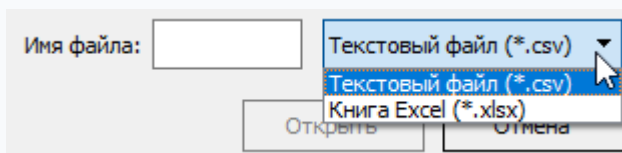


Допустим, в качестве значения свойства FileTypeFilter (Фильтр типа файлов) указана строка:

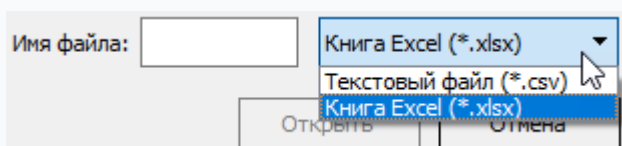
Текстовый файл (\*.csv) | \*.csv | Книга Excel (\*.xlsx) | \*.xlsx

Тогда для DefaultFileType можно указать:

- › 0 – при открытии окна SaveFileDialog по умолчанию будет предложено открыть файл типа Текстовый файл (\*.csv)



- › 1 – при открытии окна SaveFileDialog по умолчанию будет предложено открыть файл типа Книга Excel (\*.xlsx) | \*.xlsx





## 1.1.4.4.3.3.6.3.48. Показывать свойства файлов

Позволяет отображать больше информации о файлах в окне SaveFileDialog.



bool ShowFileProperties

### Значение

Значение	Описание												
true	Отображаются имена, даты изменения и размеры файлов (в байтах) <table border="1"><thead><tr><th>Имя</th><th>Дата изменения</th><th>Размер</th></tr></thead><tbody><tr><td>old_backups</td><td>17.11.2022 14:12:29</td><td></td></tr><tr><td>17112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr><tr><td>18112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr></tbody></table>	Имя	Дата изменения	Размер	old_backups	17.11.2022 14:12:29		17112022_bckp.csv	17.11.2022 14:12:53	6377	18112022_bckp.csv	17.11.2022 14:12:53	6377
Имя	Дата изменения	Размер											
old_backups	17.11.2022 14:12:29												
17112022_bckp.csv	17.11.2022 14:12:53	6377											
18112022_bckp.csv	17.11.2022 14:12:53	6377											
false	Отображаются только имена файлов <table border="1"><tbody><tr><td>old_backups</td></tr><tr><td>17112022_bckp.csv</td></tr><tr><td>18112022_bckp.csv</td></tr></tbody></table>	old_backups	17112022_bckp.csv	18112022_bckp.csv									
old_backups													
17112022_bckp.csv													
18112022_bckp.csv													

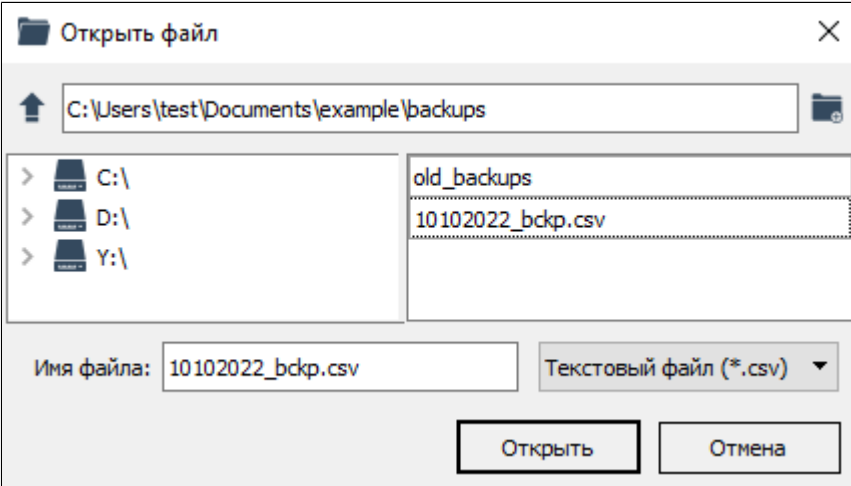
## 1.1.4.4.3.3.6.3.49. Разрешена смена папки

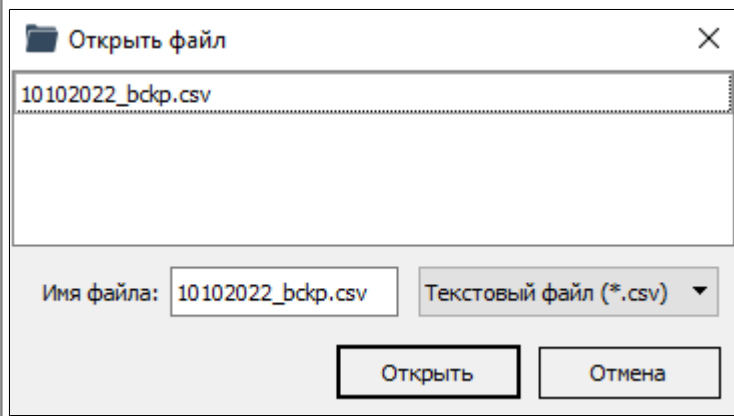
Указывает на возможность смены папки, открываемой в окне SaveFileDialog по умолчанию.



bool ChangedFolderAllowed

### Значение

Значение	Описание
true	<p>Разрешено менять папку</p> <p>В этом случае в окне SaveFileDialog отображается дерево каталогов и их содержимое (файлы и другие папки). Есть возможность перемещаться между каталогами и создавать новые папки.</p> 
false	<p>Запрещено менять папку</p> <p>В этом случае в окне SaveFileDialog отображаются только файлы папки, открываемой по умолчанию. Дерево каталогов скрыто, нет возможности создавать новые папки.</p>



## 1.1.4.4.3.3.6.3.50. Разрешено создание папок

Указывает на возможность создания папок в окне SaveFileDialog.



bool CreateFolderAllowed

### Значение

Значение	Описание
true	Разрешено создавать папки
false	Запрещено создавать папки

# 1.1.4.4.3.3.6.3.51. просмотра

## Отображать

## панель

Ограничивает возможность просмотра иерархии папок и уменьшает размеры диалога.



bool ShowBrowsePanel

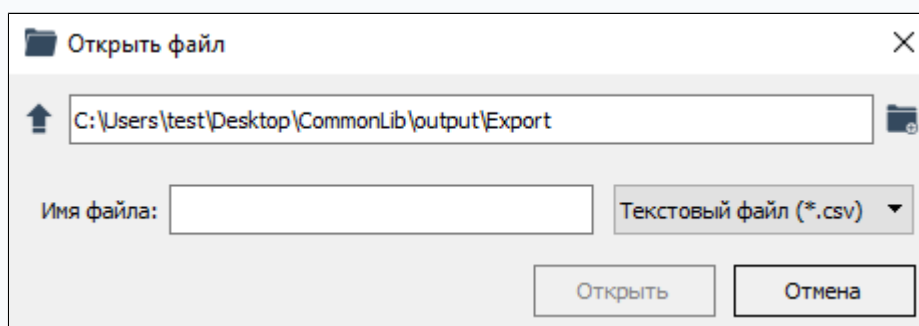
### Значение

Значение	Описание
true	Просмотр иерархии папок разрешен
false	Просмотр иерархии папок запрещен

### Примеры



Внешний вид окна, если значение свойства – false:



## 1.1.4.4.3.3.6.3.52. Ссылка на результат

Ссылка на экземпляр типа `FileDialogResult`, в котором будет сохранено имя и тип выбранного файла.

## 1.1.4.4.3.3.6.3.53. BottomPanelHeight

Высота нижней панели.



float8 BottomPanelHeight

# 1.1.4.4.3.3.6.3.54. Параметры

## Отображать

## кнопку

Меняет видимость кнопки Параметры в окне SaveFileDialog.



bool EditSettingsAllowed

При нажатии на кнопку активируется команда OnEditSettings.

## Значение

Значение	Описание
true	Кнопка есть
false	Кнопки нет



## 1.1.4.4.3.3.6.3.55. OnEditSettings

Ссылка на команду, выполняемую при нажатии на кнопку Параметры в окне SaveFileDialog.

Видимость кнопки зависит от значения свойства EditSettingsAllowed.

## 1.1.4.4.3.3.6.3.56. OnComplete

Ссылка на команду, выполняемую в результате нажатия кнопки Сохранить и закрытии окна SaveFileDialog.

## 1.1.4.4.3.3.6.3.57. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.6.3.58. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



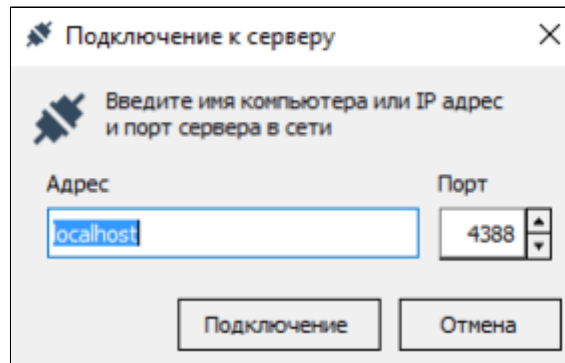
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.7. ConnectDialog

Окно подключения к серверу.



Чтобы указать учетные данные при подключении к защищенному паролем серверу, нажмите кнопку Указать учетные данные. Откроется окно ввода учетных данных CredentialsDialog.



Используя компонент, создайте экземпляр типа `ConnectDialogResult`, в котором будут сохранены данные подключения. Ссылка на созданный экземпляр указывается в свойстве `pResult`.

## 1.1.4.4.3.3.7.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.7.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.7.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.7.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.4.3.3.7.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.7.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.7.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.7.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.7.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.7.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.7.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.7.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)



## 1.1.4.4.3.3.7.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.7.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.7.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



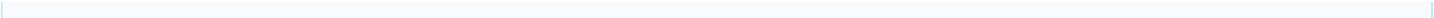
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.7.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.7.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">UpdateLayout</a>	Обновить макет





## 1.1.4.4.3.3.7.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.7.2.2. SetBoundingRegion

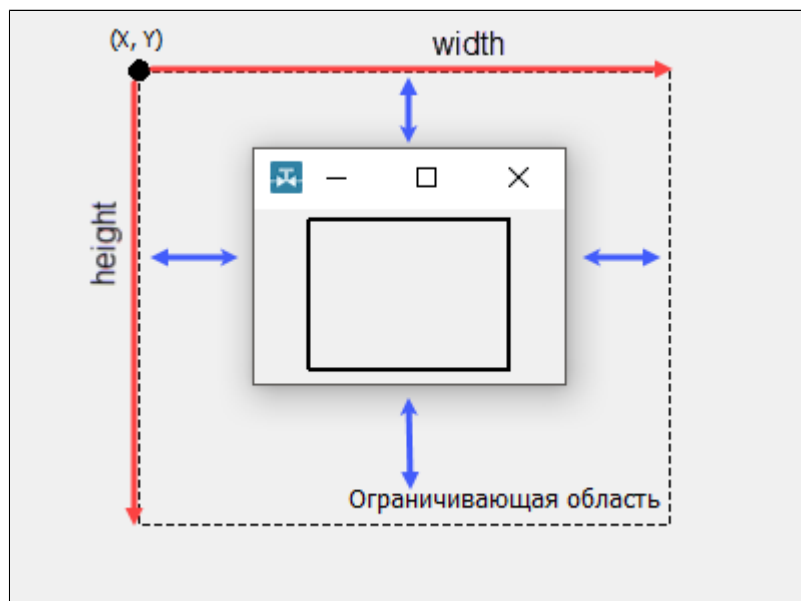
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
here.SetBoundingRegion(500, 200, 400, 400);

## 1.1.4.4.3.3.7.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.7.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.7.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.7.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.7.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```



## 1.1.4.4.3.3.7.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.7.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.7.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.7.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.7.2.12. FormY

Получение координаты Y на форме.



```
double FormX(double X, double Y)
```

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



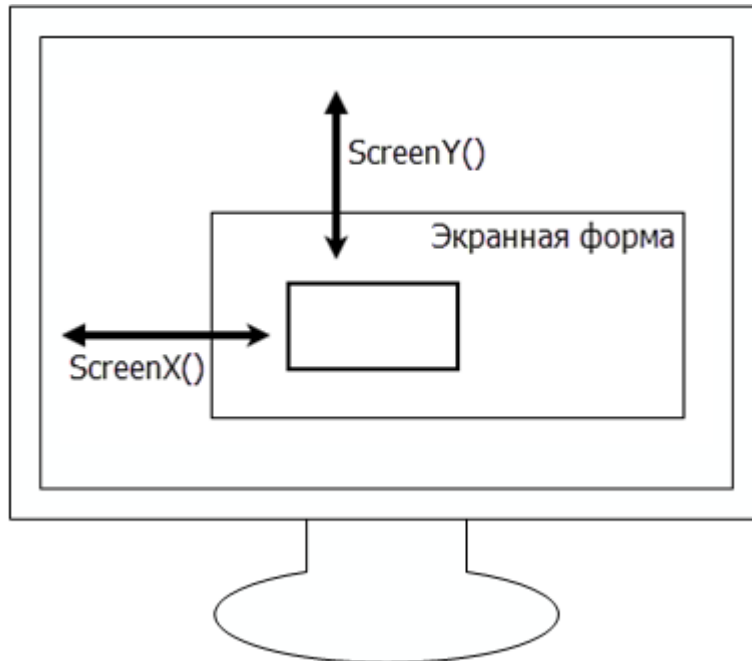
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.7.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

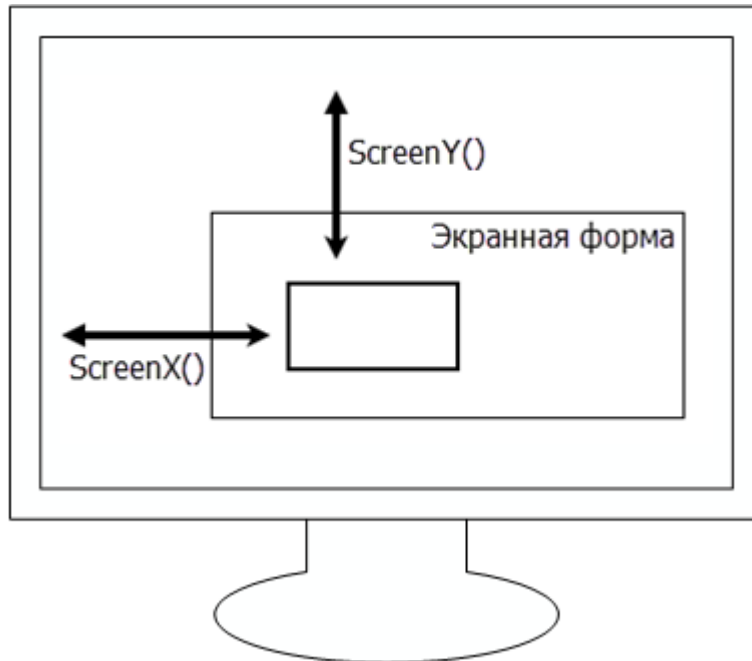


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.7.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.7.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.7.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.7.2.17. UpdateLayout




Обновить макет.



```
void UpdateLayout()
```

## 1.1.4.4.3.3.7.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна

<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Указать название</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Указать порт истории</a>	Позволяет указывать в окне подключения к серверу номер порта истории сервера
<a href="#">pResult</a>	Ссылка на ConnectDialogResult, в котором будут сохранены данные подключения
<a href="#">OkAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку Подключение в окне подключения к серверу
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

### **1.1.4.4.3.3.7.3.1. Отображаемое имя**

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.7.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



# 1.1.4.4.3.3.7.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.7.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

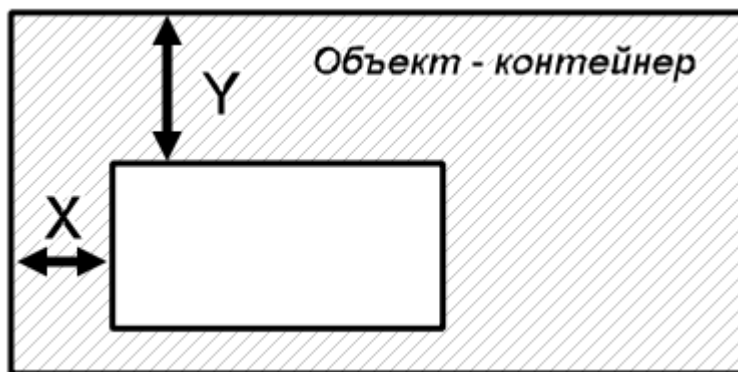
## 1.1.4.4.3.3.7.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.7.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.7.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.7.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.7.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.7.3.10. Отражение

Отражение объекта.



uint1 Flip



## 1.1.4.4.3.3.7.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.7.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.7.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



//Активировать объект в зависимости от выполнения условия:

```
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.7.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

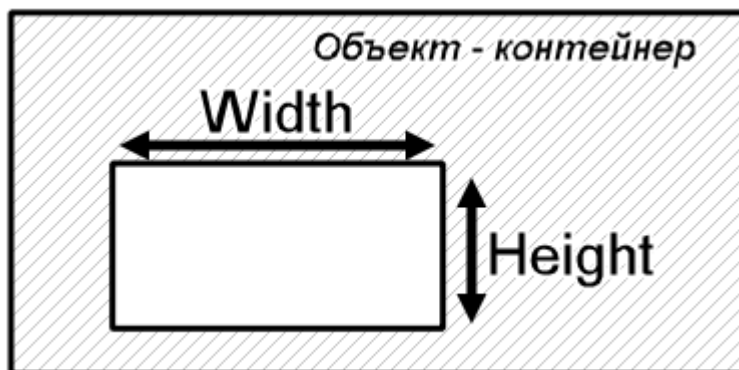
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.7.3.15. Ширина

Ширина объекта в пикселях.



float8 Width

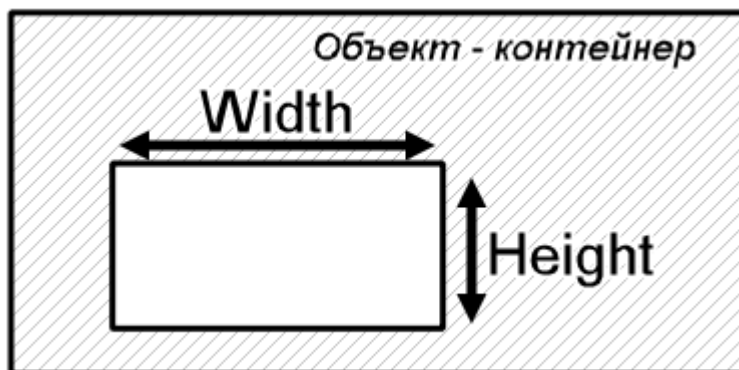


## 1.1.4.4.3.3.7.3.16. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.7.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.7.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```



## 1.1.4.4.3.3.7.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.7.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.7.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;






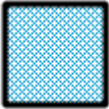

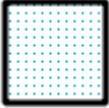
## 1.1.4.4.3.3.7.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.7.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.7.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.7.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```



## 1.1.4.4.3.3.7.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.7.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.7.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.7.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.7.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;

## 1.1.4.4.3.3.7.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.7.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.7.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```



## 1.1.4.4.3.3.7.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.7.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.7.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.7.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.7.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.7.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.7.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

## 1.1.4.4.3.3.7.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme



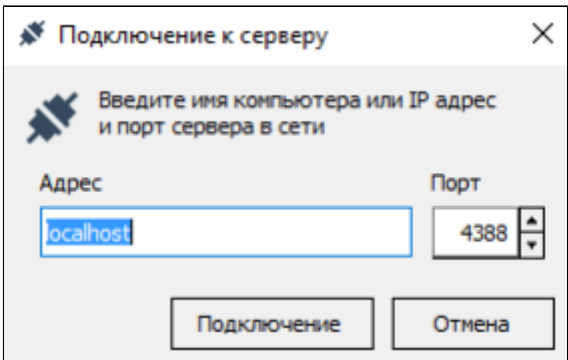
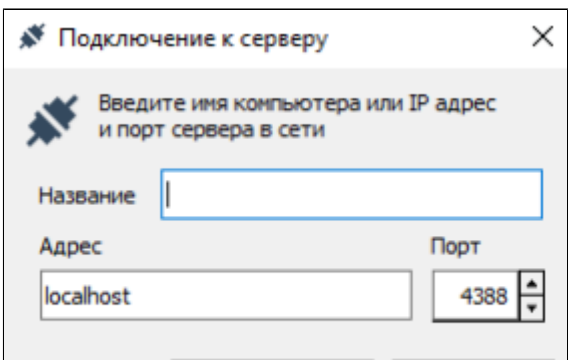
## 1.1.4.4.3.3.7.3.42. Указать название

Определяет позицию для открытия окна экранной формы.



bool SpecifyTitle

### Значение

Значение	Описание
false	<p>Обычный вид окна подключения к серверу</p> 
true	<p>В окне появляется поле ввода названия подключения</p> 

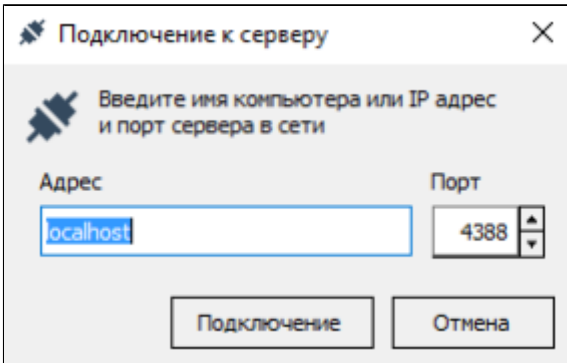
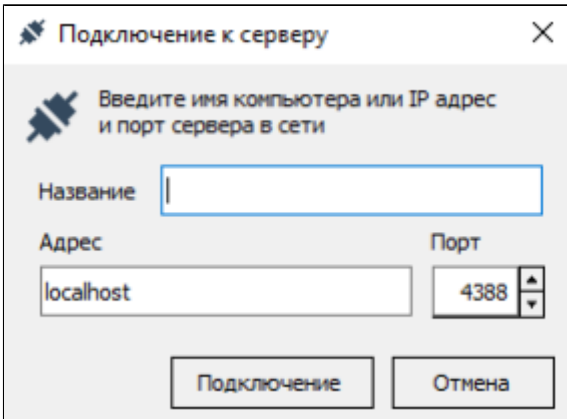
## 1.1.4.4.3.3.7.3.43. Указать порт истории

Позволяет указывать в окне подключения к серверу номер порта истории сервера.



bool SpecifyHistoryPort

### Значение

Значение	Описание
false	<p>Обычный вид окна подключения к серверу</p> 
true	<p>В окне появляется поле ввода названия подключения</p> 

## 1.1.4.4.3.3.7.3.44. pResult

Ссылка на `ConnectDialogResult`, в котором будут сохранены данные подключения.

## 1.1.4.4.3.3.7.3.45. OkAction

Ссылка на команду, выполняемую при нажатии на кнопку Подключение в окне подключения к серверу.

## 1.1.4.4.3.3.7.3.46. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.7.3.47. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

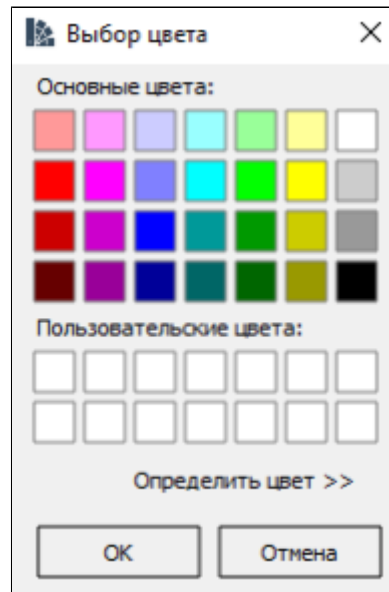


//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

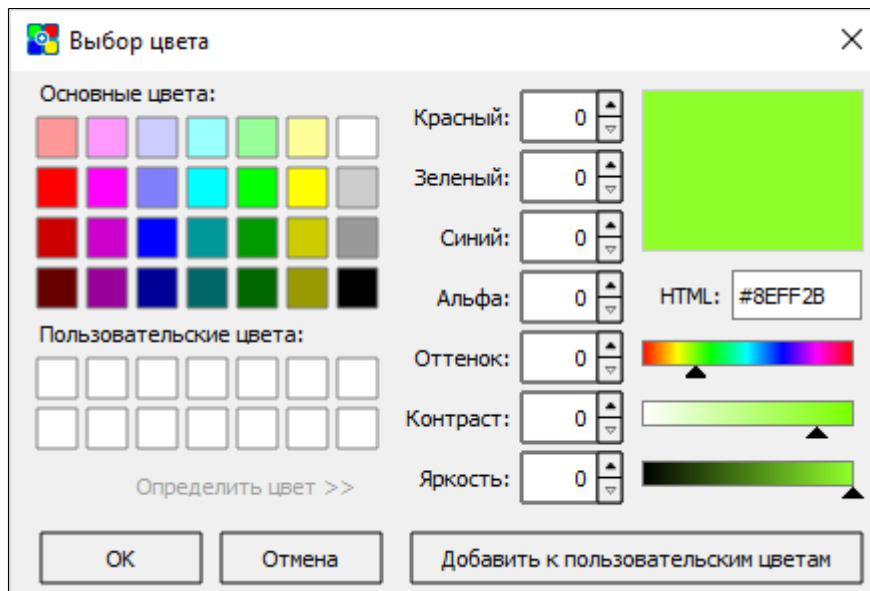
```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.8. ColorDialog

Диалоговое окно для выбора цвета из палитры.



Нажмите Определить цвет, чтобы расширить палитру.



Используя компонент, создайте экземпляр типа [ColorDialogResult](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве [Ссылка на результат](#).





## 1.1.4.4.3.3.8.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.8.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.8.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.8.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.8.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.8.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.8.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.8.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.8.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.



## 1.1.4.4.3.3.8.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.8.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.8.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.8.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.8.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.8.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



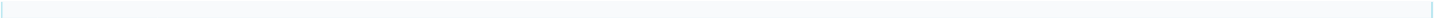
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```



## 1.1.4.4.3.3.8.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```





//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.8.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы



## 1.1.4.4.3.3.8.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.8.2.2. SetBoundingRegion

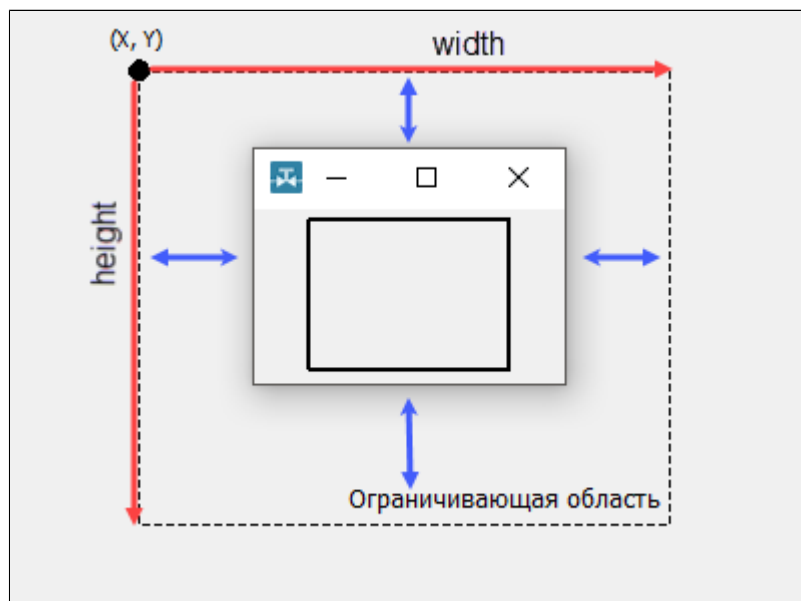
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
here.SetBoundingRegion(500, 200, 400, 400);

## 1.1.4.4.3.3.8.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.8.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла



## 1.1.4.4.3.3.8.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.8.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.8.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.8.2.8. Activate

Сделать окно формы активным.



```
void Activate()
```

## 1.1.4.4.3.3.8.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.8.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.8.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.8.2.12. FormY

Получение координаты Y на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

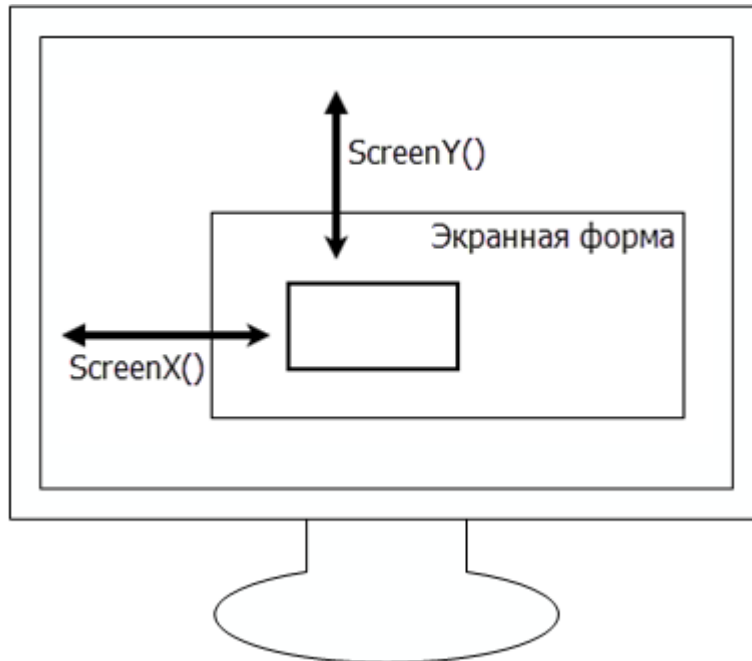
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.8.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

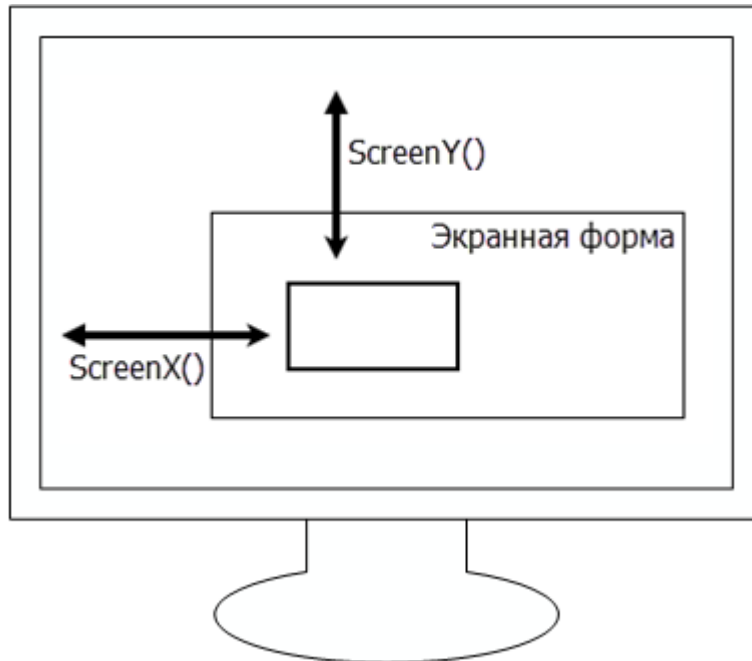


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.8.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.8.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.8.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.8.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закрыть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна

<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Палитра</a>	Вид палитры
<a href="#">IsExpanded</a>	Позволяет при вызове окна ColorDialog открывать сразу расширенный режим определения цвета
<a href="#">SelectedColor</a>	Позволяет выбрать цвет по умолчанию
<a href="#">OkAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку ОК
<a href="#">Ссылка на результат</a>	Ссылка на <a href="#">ColorDialogResult</a> , в котором будет сохранен результат выбора цвета
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Lenght</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве



## 1.1.4.4.3.3.8.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.8.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.8.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.8.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

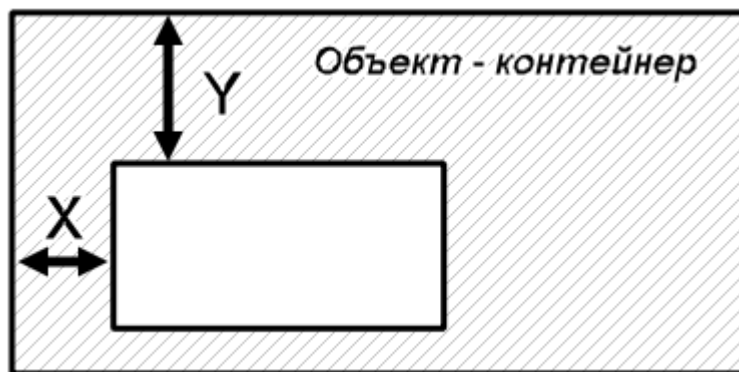
## 1.1.4.4.3.3.8.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.8.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.8.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.8.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;



//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;



## 1.1.4.4.3.3.8.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.8.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.8.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.8.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.8.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.8.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

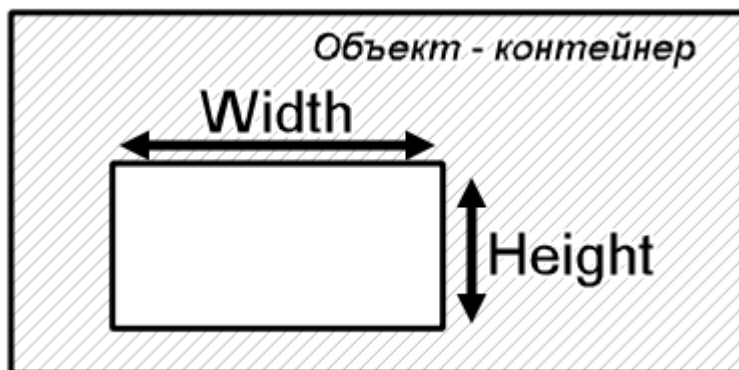
```
Object.Tooltip = "Подача команды MNA_start = true";
```

## 1.1.4.4.3.3.8.3.15. Ширина

Ширина объекта в пикселях.




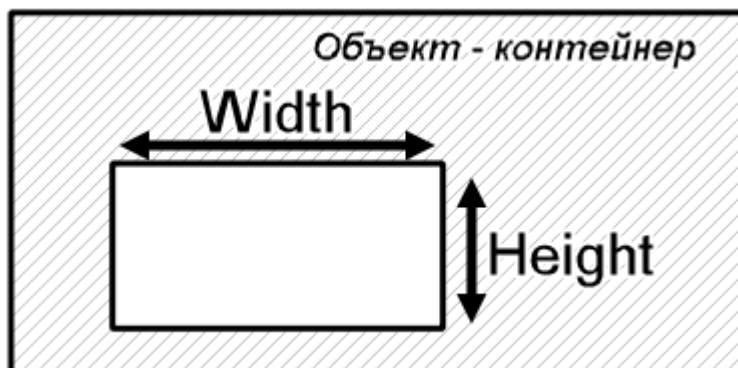
float8 Width



# 1.1.4.4.3.3.8.3.16. Высота

Высота объекта в пикселях.

 float8 Height





## 1.1.4.4.3.3.8.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.8.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.8.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.8.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.8.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.4.3.3.8.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.8.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```



## 1.1.4.4.3.3.8.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.8.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.8.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.8.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.8.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.8.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.8.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;

## 1.1.4.4.3.3.8.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```



## 1.1.4.4.3.3.8.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.8.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.8.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.8.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.8.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.8.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.8.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.8.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```



## 1.1.4.4.3.3.8.3.40. Путь к файлу иконки


Путь к файлу иконки.



string WindowIconPath

# 1.1.4.4.3.3.8.3.41. Палитра

Вид палитры.

 uint1 PaletteType

Может принимать только одно из допустимых значений, приведенных в таблице.

Допустимое значение	Имя значения	Результат выбора значения
0	Градиент	
1	Веб-цвета	

## 1.1.4.4.3.3.8.3.42. IsExpanded

Позволяет при вызове окна ColorDialog открывать сразу расширенный режим определения цвета.



bool IsExpanded

### Значение

Значение	Описание
true	Запускается расширенный режим определения цвета
false	Запускается обычный режим определения цвета

## 1.1.4.4.3.3.8.3.43. SelectedColor

Позволяет выбрать цвет по умолчанию. При запуске окна ColorDialog в палитре будет выбран указанный цвет.



uint4 SelectedColor

## 1.1.4.4.3.3.8.3.44. OkAction

Ссылка на команду, выполняемую при нажатии на кнопку ОК.

## 1.1.4.4.3.3.8.3.45. Ссылка на результат

Ссылка на [ColorDialogResult](#), в котором будет сохранен результат выбора цвета.

## 1.1.4.4.3.3.8.3.46. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.8.3.47. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.4.3.3.8.3.48. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



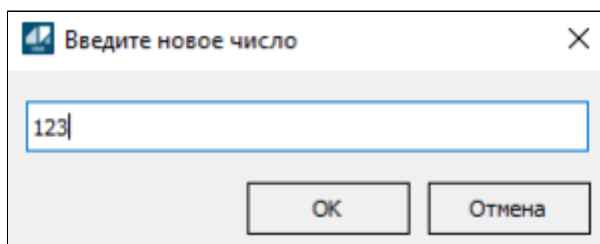
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.9. InputDialog

Диалоговое окно для ввода значения.



Используя компонент, создайте экземпляр типа [InputDialogResult](#), в котором будет сохранено имя и тип выбранного файла. Ссылка на созданный экземпляр указывается в свойстве [Ссылка на результат](#).

## 1.1.4.4.3.3.9.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.9.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.9.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.9.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.9.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.9.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.9.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте



## 1.1.4.4.3.3.9.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.9.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.9.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.9.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.9.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.9.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.9.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.9.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```

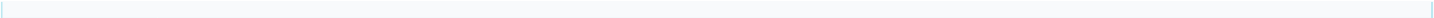


```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```





## 1.1.4.4.3.3.9.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.9.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы
<a href="#">UpdateLayout</a>	Обновить макет



## 1.1.4.4.3.3.9.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.9.2.2. SetBoundingRegion

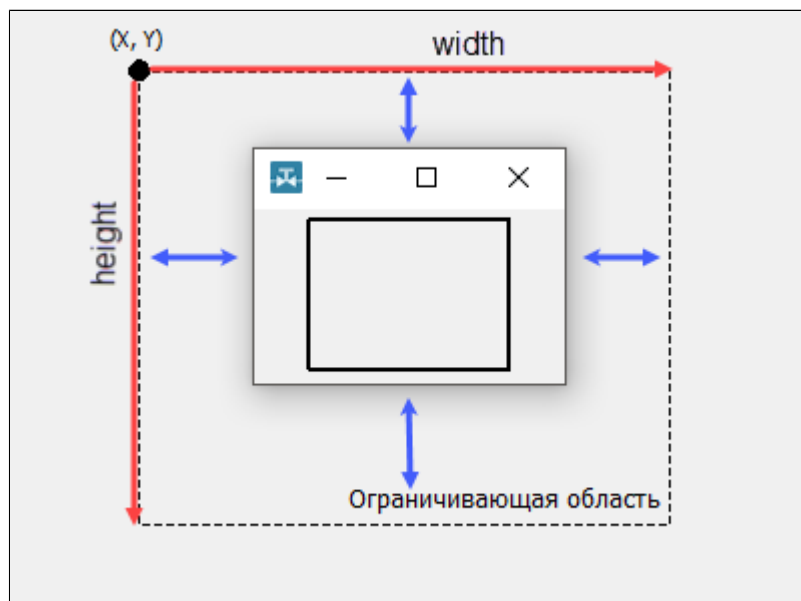
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`



## 1.1.4.4.3.3.9.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.9.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.9.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.9.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.9.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



void Print()

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.9.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.9.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.9.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```



## 1.1.4.4.3.3.9.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.9.2.12. FormY

Получение координаты Y на форме.



```
double FormX(double X, double Y)
```

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



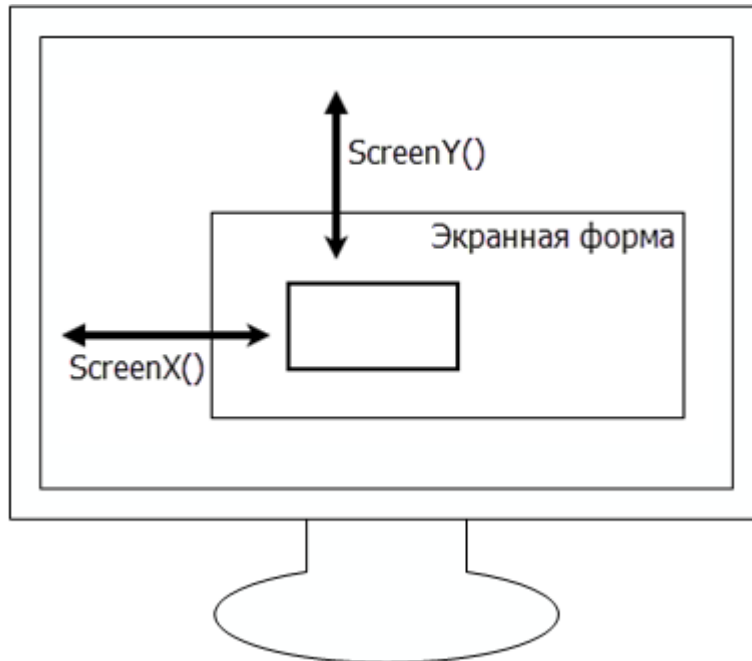
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.9.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

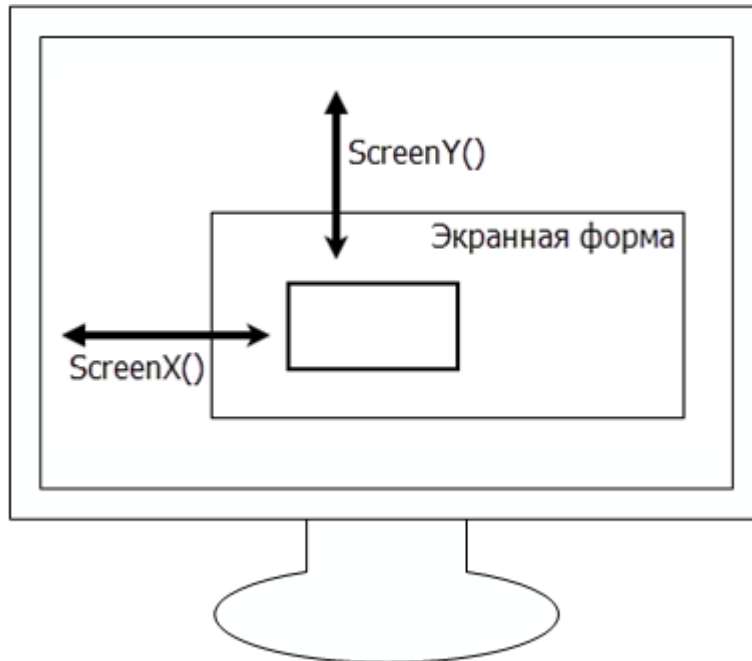


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.9.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.9.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.9.2.16. Hide

Скрыть окно формы.



void Hide()

## 1.1.4.4.3.3.9.2.17. UpdateLayout

Обновить макет.






void UpdateLayout()



## 1.1.4.4.3.3.9.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта

<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна

<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Ссылка на результат</a>	Ссылка на <a href="#">InputDialogResult</a> , в котором будет сохранен ввод пользователя
<a href="#">OkAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку ОК
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.9.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.9.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.9.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.9.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

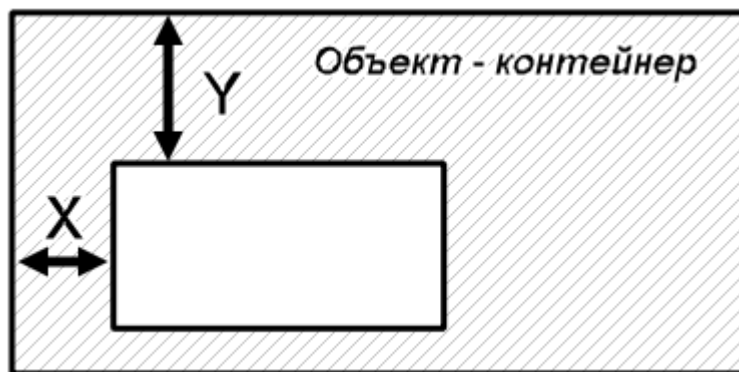
## 1.1.4.4.3.3.9.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```



## 1.1.4.4.3.3.9.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.9.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.9.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;



//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;

## 1.1.4.4.3.3.9.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.9.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.9.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.9.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.9.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.4.3.3.9.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




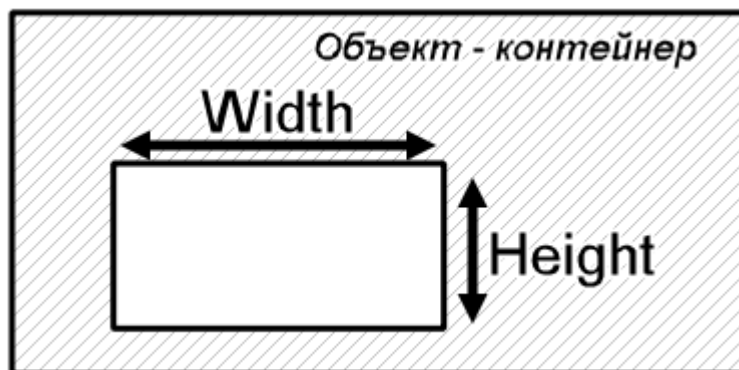
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.9.3.15. Ширина


Ширина объекта в пикселях.

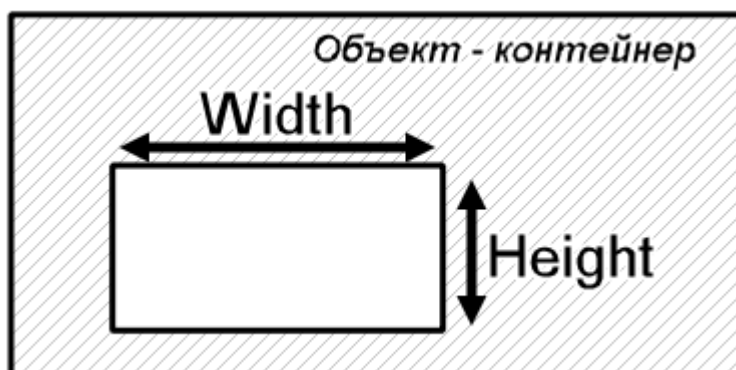
 float8 Width



# 1.1.4.4.3.3.9.3.16. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.3.9.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.9.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.9.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.9.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.9.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;






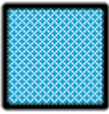

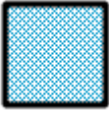

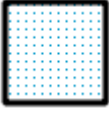
## 1.1.4.4.3.3.9.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.9.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.9.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.9.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.9.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.9.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.9.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```



# 1.1.4.4.3.3.9.3.29. "свернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.9.3.30. "развернуть"

Показывать

кнопку

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;

## 1.1.4.4.3.3.9.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

### Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.9.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.9.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.9.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.9.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.9.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```



## 1.1.4.4.3.3.9.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.9.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.9.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.9.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

# 1.1.4.4.3.3.9.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.9.3.42. Ссылка на результат

Ссылка на [InputDialogResult](#), в котором будет сохранен ввод пользователя.

## 1.1.4.4.3.3.9.3.43. OkAction

Ссылка на команду, выполняемую при нажатии на кнопку ОК.

## 1.1.4.4.3.3.9.3.44. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.4.3.3.9.3.45. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



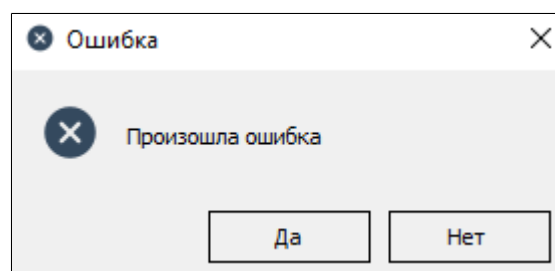
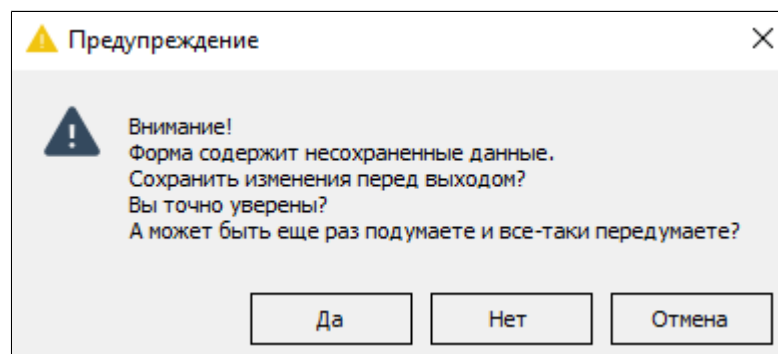
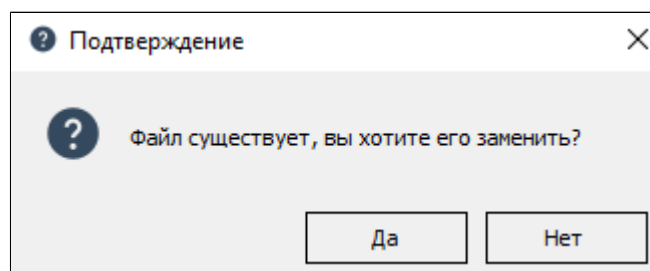
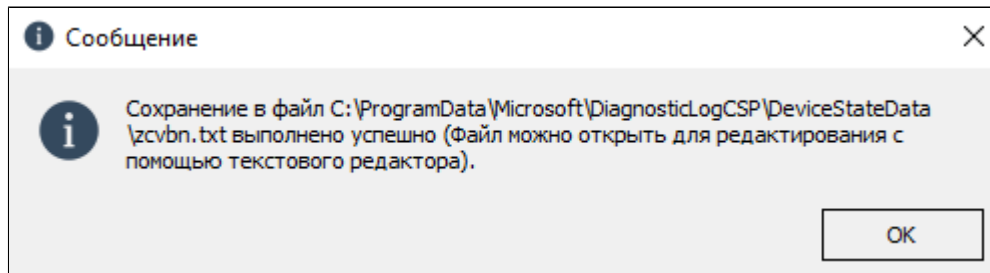
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.10. MessageDialog

Диалоговое окно с настраиваемым внешним видом и функциями.



Для изменения внешнего вида и функций диалогового окна, используйте свойства.

## 1.1.4.4.3.3.10.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.10.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.10.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.10.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.10.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.10.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.10.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте



## 1.1.4.4.3.3.10.1.13. CloseRequested

Запрос на закрытие формы.

## 1.1.4.4.3.3.10.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.10.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.10.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.10.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.10.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.10.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.10.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```

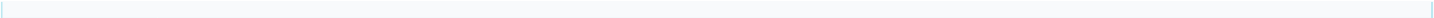


```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```





## 1.1.4.4.3.3.10.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.10.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы



## 1.1.4.4.3.3.10.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.10.2.2. SetBoundingRegion

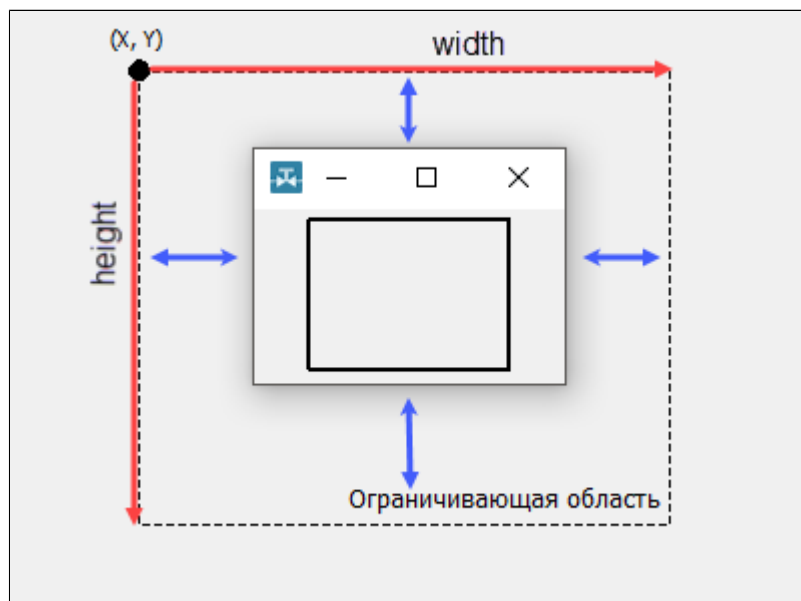
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`



## 1.1.4.4.3.3.10.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```

## 1.1.4.4.3.3.10.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.10.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.10.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.10.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



```
void Print()
```

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.10.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.10.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.10.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```



## 1.1.4.4.3.3.10.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.10.2.12. FormY

Получение координаты Y на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



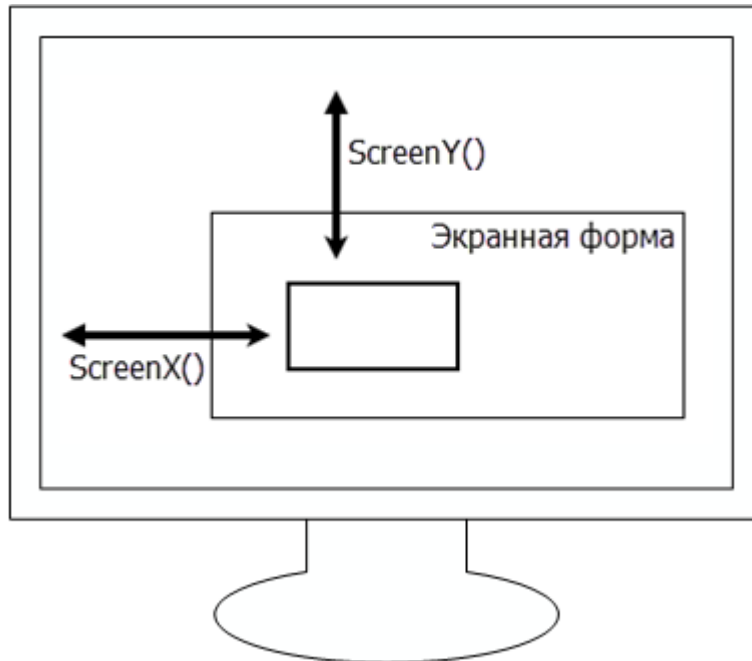
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.10.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

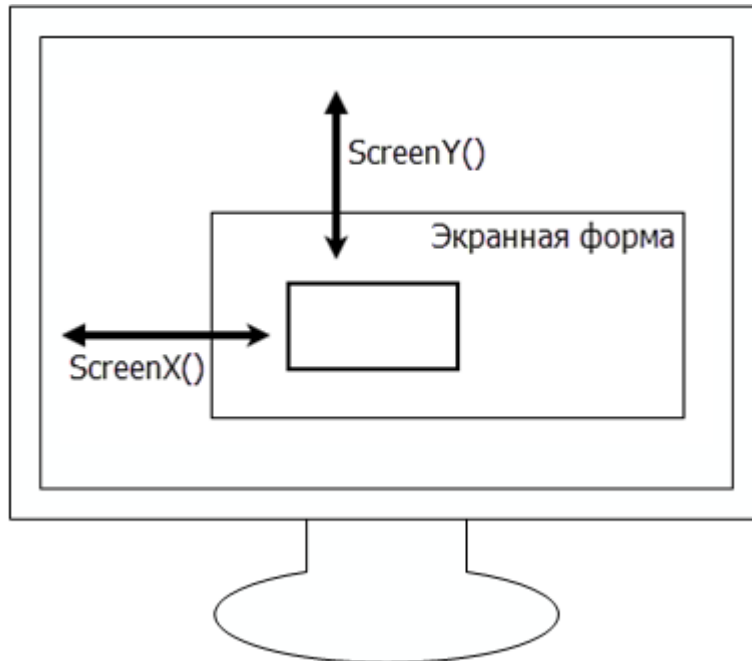


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.10.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.10.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.10.2.16. Hide

Скрыть окно формы.



void Hide()

### 1.1.4.4.3.3.10.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта



<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закрыть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна

<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Высота строки текста</a>	Высота строки текста сообщения
<a href="#">Символов в строке</a>	Количество символов в строке
<a href="#">Сообщение</a>	Текст сообщения
<a href="#">Иконка</a>	Иконка рядом с текстом сообщения
<a href="#">Кнопки</a>	Набор кнопок в диалоговом окне
<a href="#">OkAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку ОК
<a href="#">YesAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку Да
<a href="#">NoAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку Нет
<a href="#">CancelAction</a>	Ссылка на команду, выполняемую при нажатии на кнопку Отмена
<a href="#">Результат</a>	Хранит значение кнопки, выбранной в диалоговом окне
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.10.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.10.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.10.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.10.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

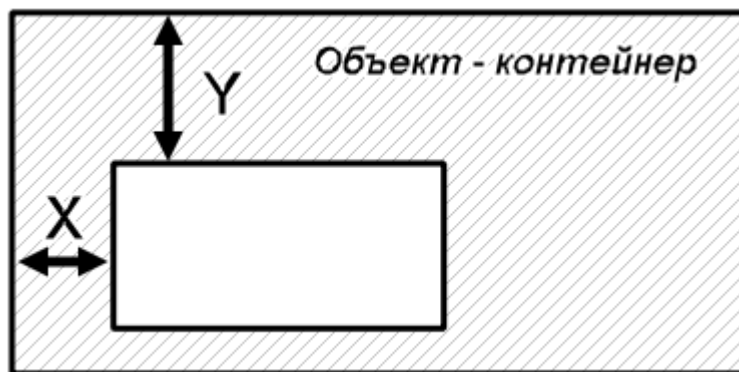
## 1.1.4.4.3.3.10.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.10.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.10.3.7. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.10.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.10.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.10.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.10.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.10.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.10.3.13. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.10.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




//Задать всплывающую подсказку для объекта:

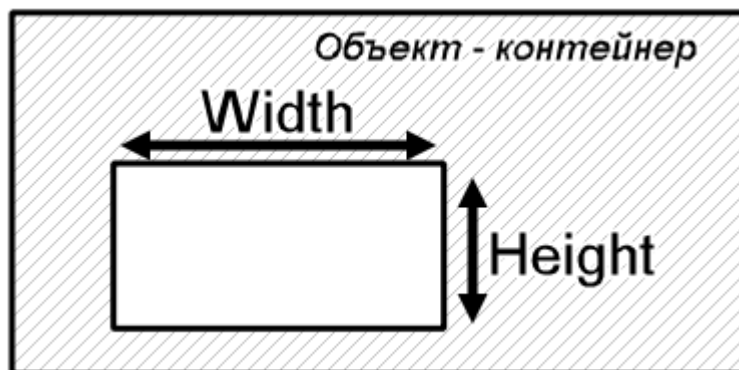
```
Object.Tooltip = "Подача команды MNA_start = true";
```



# 1.1.4.4.3.3.10.3.15. Ширина

Ширина объекта в пикселях.

 float8 Width

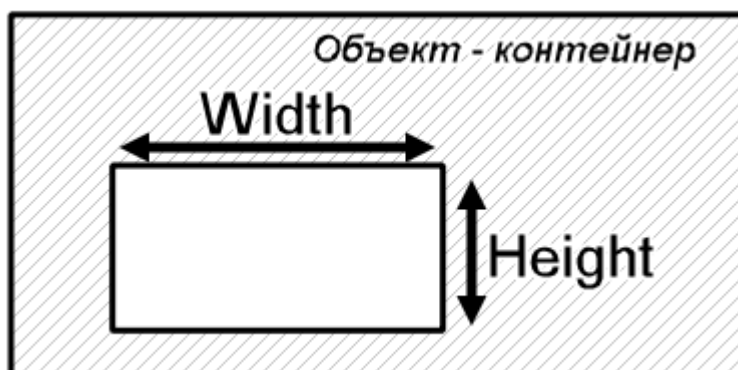


## 1.1.4.4.3.3.10.3.16. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.10.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.10.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.4.3.3.10.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.10.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.10.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;




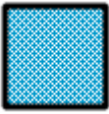

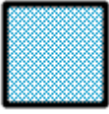

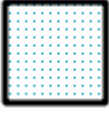
## 1.1.4.4.3.3.10.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	



Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.10.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.10.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.10.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.10.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.10.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.10.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.10.3.29. Показывать кнопку "свернуть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```



# 1.1.4.4.3.3.10.3.30. Показывать кнопку "развернуть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



```
//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;
```

# 1.1.4.4.3.3.10.3.31. Показывать кнопку "закреть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

## Примеры



```
//Убрать кнопку "закреть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.10.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.10.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

## 1.1.4.4.3.3.10.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

### Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

### Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.10.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

## 1.1.4.4.3.3.10.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

### Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

### Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.10.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```



## 1.1.4.4.3.3.10.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```

## 1.1.4.4.3.3.10.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.10.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

# 1.1.4.4.3.3.10.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.10.3.42. Высота строки текста

Высота строки текста сообщения.



float8 LineHeight

Используется для изменения размера диалогового окна.

## 1.1.4.4.3.3.10.3.43. Символов в строке

Количество символов в строке.



int4 CharactersPerLine

Используется для изменения размера диалогового окна.

# 1.1.4.4.3.3.10.3.44. Сообщение

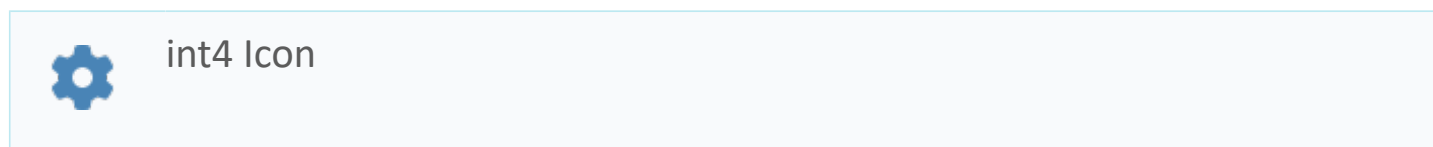
Текст сообщения.







string Message

## 1.1.4.4.3.3.10.3.45. Иконка

Иконка рядом с текстом сообщения.



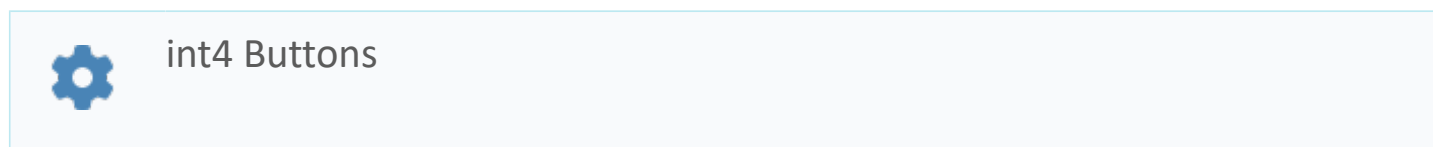
Может принимать только одно из допустимых значений, приведенных в таблице.

Допустимое значение	Имя значения	Результат выбора значения
0	Info	
1	Question	
2	Warning	
3	Error	



## 1.1.4.4.3.3.10.3.46. Кнопки

Набор кнопок в диалоговом окне.



Может принимать только одно из допустимых значений, приведенных в таблице.

Допустимое значение	Набор кнопок
0	ОК
1	ОК и Отмена
2	Да и Нет
3	Да, Нет и Отмена

## 1.1.4.4.3.3.10.3.47. OkAction

Ссылка на команду, выполняемую при нажатии на кнопку ОК.

## 1.1.4.4.3.3.10.3.48. YesAction

Ссылка на команду, выполняемую при нажатии на кнопку Да.

## 1.1.4.4.3.3.10.3.49. NoAction

Ссылка на команду, выполняемую при нажатии на кнопку Нет.

## 1.1.4.4.3.3.10.3.50. CancelAction

Ссылка на команду, выполняемую при нажатии на кнопку Отмена.

## 1.1.4.4.3.3.10.3.51. Результат

Хранит значение кнопки, выбранной в диалоговом окне.



int4 Result

Значения кнопок приведены в таблице.

<b>Выбранная кнопка</b>	<b>Значение свойств</b>
ОК	1
Да	2
Нет	3
Отмена	4

## 1.1.4.4.3.3.10.3.52. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.10.3.53. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

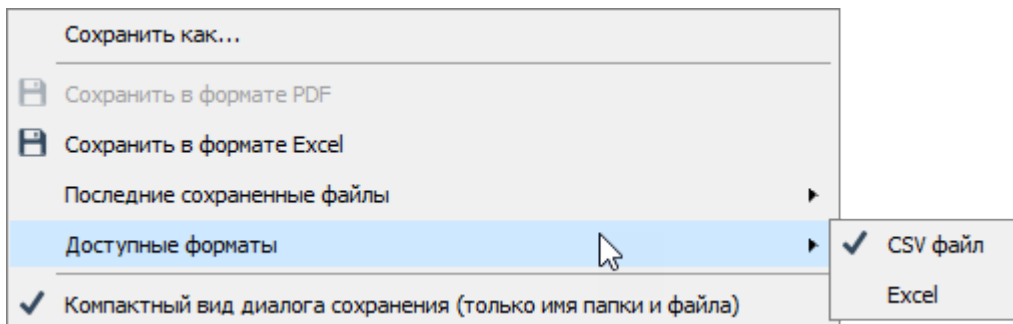


```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.4.3.3.11. Menu

Контекстное меню.



Меню следует настроить. Перечень элементов меню укажите в качестве значения свойства `Items` в виде JSON массива.

Пример использования приведен в Выбор цвета на панели инструментов из палитры или из меню.

## 1.1.4.4.3.3.11.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">CloseRequested</a>	Запрос на закрытие формы
<a href="#">AboutToOpen</a>	Событие, срабатывающее перед открытием окна
<a href="#">Opened</a>	Открытие формы
<a href="#">Closed</a>	Закрытие формы
<a href="#">WindowActivationChanged</a>	Изменение статуса активности окна
<a href="#">WindowSizeChanged</a>	Изменение размеров окна
<a href="#">WindowPosChanged</a>	Изменение положения окна
<a href="#">KeyPress</a>	Нажатие определенной клавиши или сочетания клавиш на форме
<a href="#">KeyRelease</a>	Отпускание определенной клавиши или сочетания клавиш на форме

## 1.1.4.4.3.3.11.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.11.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.11.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.11.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.11.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.11.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.11.1.13. CloseRequested

Запрос на закрытие формы.



## 1.1.4.4.3.3.11.1.14. AboutToOpen

Событие, срабатывающее перед открытием окна.

## 1.1.4.4.3.3.11.1.15. Opened

Открытие формы.

## 1.1.4.4.3.3.11.1.16. Closed

Закрытие формы.

## 1.1.4.4.3.3.11.1.17. WindowActivationChanged

Изменение статуса активности окна. Окно активно, когда на нем фокус.

### Параметры

Параметр	Тип	Описание
active	bool	Статус активности окна: активное (true) или неактивное (false)

## 1.1.4.4.3.3.11.1.18. WindowSizeChanged

Изменение размеров окна.

### Параметры

Параметр	Тип	Описание
width	uint4	Новая ширина окна
height	uint4	Новая высота окна

## 1.1.4.4.3.3.11.1.19. WindowPosChanged

Изменение положения окна.

### Параметры

Параметр	Тип	Описание
x	int4	Новое положение по горизонтали верхней левой точки окна
y	int4	Новое положение по вертикали верхней левой точки окна

## 1.1.4.4.3.3.11.1.20. KeyPress

Нажатие определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
<code>KeyCode</code>	<code>uint4</code>	Код клавиши
<code>Modifiers</code>	<code>int4</code>	Код клавиши-модификатора
<code>Repeated</code>	<code>bool</code>	Флаг удержания клавиши или сочетания любых клавиш. Не становится <code>true</code> при удержании клавиш-модификаторов: <code>Ctrl</code> , <code>Shift</code> , <code>Alt</code>
<code>Text</code>	<code>string</code>	Текст клавиши

### Примеры



```
//Вывести код нажатой клавиши в текстовое поле:  
TextEdit_1.Text = String.ToString(event.KeyCode);
```



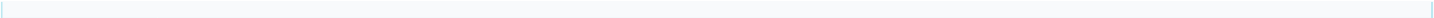
```
//Вывести код нажатой клавиши-модификатора в текстовое поле:  
TextEdit_1.Text = String.ToString(event.Modifiers);
```



```
//При удержании клавиши F5 записать текст из поля ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



```
//Вывести текст нажатой клавиши в текстовое поле:  
TextEdit_1.Text = event.Text;
```





## 1.1.4.4.3.3.11.1.21. KeyRelease

Отпускание определённой клавиши или сочетания клавиш на форме. Чтобы обратиться к параметрам события, используйте ключевое слово `event`.

### Параметры

Параметр	Тип	Описание
KeyCode	uint4	Код клавиши
Modifiers	int4	Код клавиши-модификатора
Repeated	bool	Флаг удержания клавиши или сочетания любых клавиш. Не становится true при удержании клавиш-модификаторов: Ctrl, Shift, Alt
Text	string	Текст клавиши

### Примеры



```
//Увеличить размер прямоугольника при отпускании пробела:  
if (event.KeyCode == 84)  
{Rectangle_1.Width = 500;Rectangle_1.Height = 500;}
```



```
//Уведомить о сохранении данных:  
if (event.Modifiers == 67108864 && event.KeyCode == 28)  
{Text_1.Text = "Сохранено";}
```



```
//При удержании и отпускании клавиши F5 записать текст из поля  
ввода в текстовое поле:  
if (event.KeyCode == 55 && event.Repeated == true)  
{Text_1.Text = TextEdit_1.Text;}
```



//Вывести текст нажатой клавиши в текстовое поле:

```
Text_1.Text = event.Text;
```

## 1.1.4.4.3.3.11.2. Функции

Компонент	Описание
<a href="#">ResetBoundingRegion</a>	Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы
<a href="#">SetBoundingRegion</a>	Определяет ограничивающую область для перемещения экранной формы
<a href="#">DoAutoSize</a>	Автоматически подгоняет размер окна под размер экранной формы
<a href="#">SavePdf</a>	Сохранить форму в PDF файл
<a href="#">PrintDialog</a>	Открывает диалоговое окно Просмотр печати
<a href="#">SaveScreenShot</a>	Формирует скриншот формы и сохраняет его в файл указанного формата по указанному пути
<a href="#">Print</a>	Печать экранной формы на принтере, настроенном по умолчанию
<a href="#">Activate</a>	Сделать окно формы активным
<a href="#">Raise</a>	Поднять окно формы на передний план
<a href="#">Close</a>	Закрывает экранную форму
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Show</a>	Показать окно формы
<a href="#">Hide</a>	Скрыть окно формы



## 1.1.4.4.3.3.11.2.1. ResetBoudingRegion

Отменяет все ограничения перемещения для диалогового окна внутри мнемосхемы.



```
void ResetBoudingRegion()
```

### Примеры



```
//Отменить ограничения перемещения окна:  
ResetBoudingRegion();
```

## 1.1.4.4.3.3.11.2.2. SetBoundingRegion

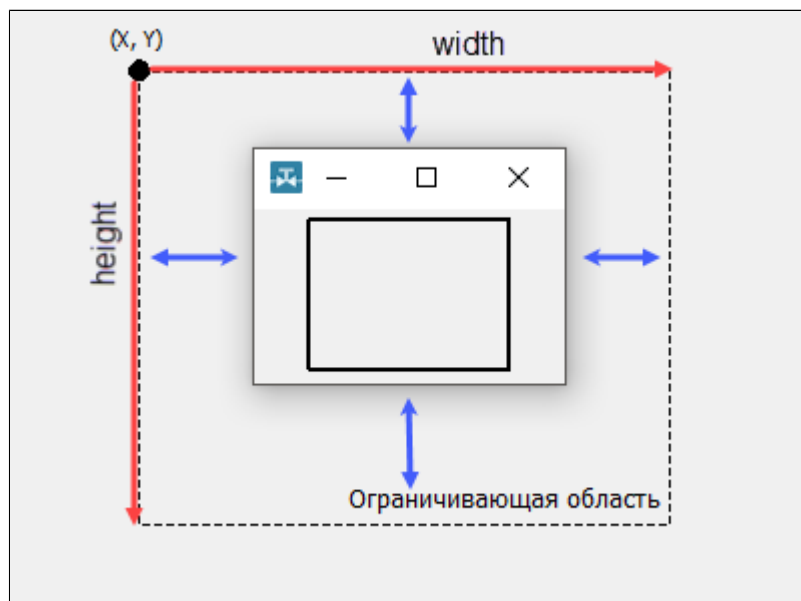
Определяет ограничивающую область для перемещения экранной формы. Экранную форму невозможно переместить за пределы ограничивающей области.



```
void SetBoundingRegion(int x, int y, uint width, uint height)
```

### Параметры

Параметр	Тип	Описание
x	int	Координата X начала области ограничения
y	int	Координата Y начала области ограничения
width	uint	Ширина области ограничения, отсчитывается от точки начала области
height	uint	Высота области ограничения, отсчитывается от точки начала области



### Примеры



//Установить ограничения перемещения окна:  
`here.SetBoundingRegion(500, 200, 400, 400);`

## 1.1.4.4.3.3.11.2.3. DoAutoSize

Автоматически подгоняет размер окна под размер экранной формы.



```
void DoAutoSize()
```

### Примеры



```
//Установить размер окна по размеру экранной формы:  
here.DoAutoSize();
```



## 1.1.4.4.3.3.11.2.4. SavePdf

Сохранить форму в PDF файл.



void SavePdf(string Path)

### Параметры

Параметр	Тип	Описание
Path	string	Путь файла

## 1.1.4.4.3.3.11.2.5. PrintDialog

Открывает диалоговое окно Просмотр печати.



```
void PrintDialog()
```

### Примеры



```
//Открыть диалоговое окно печати:  
here.PrintDialog();
```

## 1.1.4.4.3.3.11.2.6. SaveScreenShot

Формирует скриншот формы со всем её видимым содержимым и сохраняет его в файл указанного формата по указанному пути. Формат файла должен быть графическим растровым (png, jpg, bmp и т.д.).



```
void SaveScreenShot(string filePath)
```

### Параметры

Параметр	Тип	Описание
filePath	string	Путь к файлу

### Примеры



```
//Сделать скриншот формы:  
here.SaveScreenShot("C:/WORK/TU_UK/KP_635.jpeg");
```

## 1.1.4.4.3.3.11.2.7. Print

Печать экранной формы на принтере, настроенном по умолчанию.



void Print()

### Примеры



```
//Распечатать форму на принтере по умолчанию:  
here.Print();
```

## 1.1.4.4.3.3.11.2.8. Activate

Сделать окно формы активным.



void Activate()

## 1.1.4.4.3.3.11.2.9. Raise

Поднять окно формы на передний план.



void Raise()

## 1.1.4.4.3.3.11.2.10. Close

Закрывает экранную форму. Функция связана с свойством Режим обработки закрытия окна.



void Close()

### Примеры



```
//При выборе флажка закрыть форму  
if (Checkbox.State == true) me.Close();
```

## 1.1.4.4.3.3.11.2.11. FormX

Получение координаты X на форме.



double FormX(double X, double Y)

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.11.2.12. FormY

Получение координаты Y на форме.



```
double FormX(double X, double Y)
```

### Параметры

Параметр	Тип	Описание
X	double	Значение свойства X объекта
Y	double	Значение свойства Y объекта

### Примеры



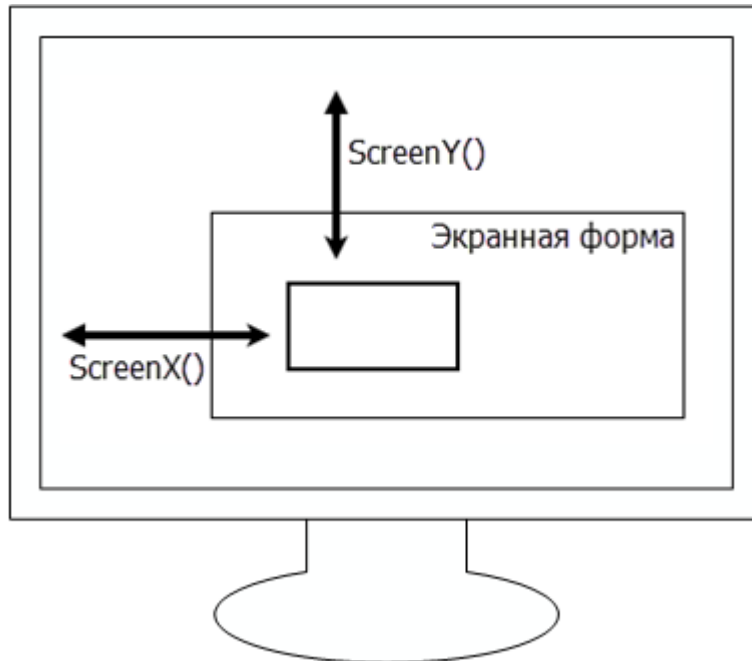
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.11.2.13. ScreenX

Вычисляют координату X объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenX(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

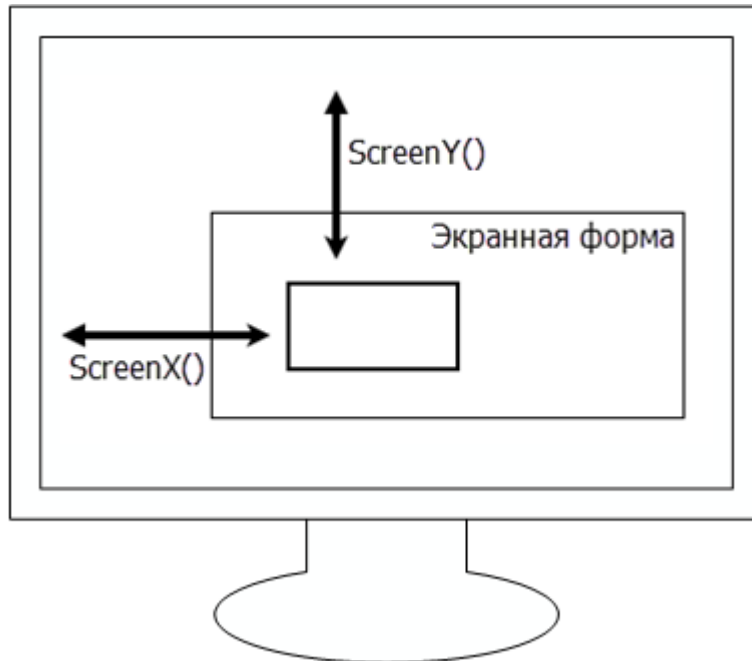


```
//Записать координаты X и Y объекта относительно экрана в поля ввода:  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.11.2.14. ScreenY

Вычисляют координату Y объекта относительно экрана монитора. Координаты измеряются в пикселях.



int4 ScreenY(Object.X, Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.11.2.15. Show

Показать окно формы.



void Show()

## 1.1.4.4.3.3.11.2.16. Hide

Скрыть окно формы.



void Hide()

### 1.1.4.4.3.11.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Горизонтальная полоса прокрутки</a>	Горизонтальная полоса прокрутки
<a href="#">Вертикальная полоса прокрутки</a>	Вертикальная полоса прокрутки
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки

<a href="#">X окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Y окна</a>	Позиция окна экранной формы по горизонтали и вертикали
<a href="#">Ширина окна</a>	Ширина окна экранной формы в пикселях
<a href="#">Высота окна</a>	Высота окна экранной формы в пикселях
<a href="#">Заголовок окна</a>	Надпись в строке заголовка
<a href="#">Показывать заголовок</a>	Включает/отключает заголовок окна вместе со всеми кнопками
<a href="#">Показывать кнопку "свернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "развернуть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Показывать кнопку "закреть"</a>	Включает/отключает кнопку  в заголовке окна
<a href="#">Поверх всех окон</a>	Определяет режим отображения экранной формы поверх остальных окон мнемосхемы
<a href="#">Размеры окна</a>	Определяет режим вычисления размеров окна
<a href="#">Стиль рамки окна</a>	Стиль рамки окна экранной формы
<a href="#">Состояние окна</a>	Определяет режим, в котором будет открываться окно экранной формы
<a href="#">Режим масштабирования</a>	Автоматически подгоняет размеры контейнера экранной формы до размеров окна
<a href="#">Монитор</a>	Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения
<a href="#">Положение окна</a>	Определяет позицию для открытия окна экранной формы
<a href="#">Режим обработки закрытия окна</a>	Позволяет пользователю определять условия закрытия окна
<a href="#">Путь к файлу иконки</a>	Путь к файлу иконки



<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Высота элемента</a>	Высота элемента, относительно которого открывается календарь.
<a href="#">IsSubmenu</a>	Указывает, является ли данный экземпляр Menu подменю другого экземпляра Menu.
<a href="#">Items</a>	Хранит перечень элементов меню в виде JSON массива, содержащего описание каждого элемента
<a href="#">pResult</a>	Ссылка на экземпляр типа MenuResult, в котором будет сохранен результат выбора пункта меню
<a href="#">pAction</a>	Ссылка на команду, выполняемую при выборе элемента меню
<a href="#">pPrevMenu</a>	Ссылка на команду, выполняемую при переходе к предыдущему пункту меню (при управлении меню клавишами)
<a href="#">pNextMenu</a>	Ссылка на команду, выполняемую при переходе к следующему пункту меню (при управлении меню клавишами)
<a href="#">pOnDoClose</a>	Ссылка на команду
<a href="#">pOnClosed</a>	Ссылка на команду, выполняемую при закрытии меню
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.11.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.11.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

# 1.1.4.4.3.3.11.3.3. Горизонтальная полоса прокрутки

Горизонтальная полоса прокрутки.

# 1.1.4.4.3.3.11.3.4. Вертикальная полоса прокрутки

Вертикальная полоса прокрутки.

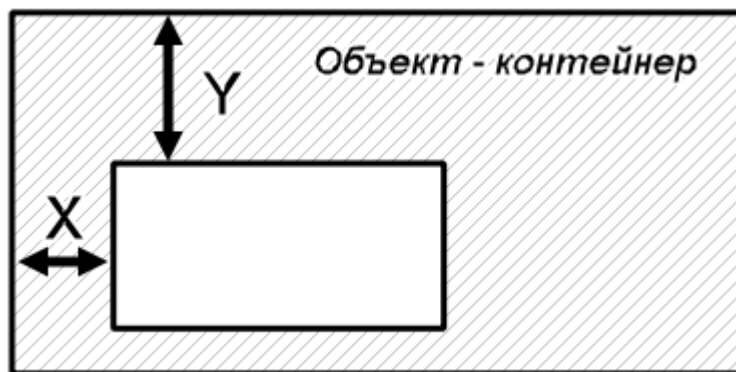
## 1.1.4.4.3.3.11.3.5. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

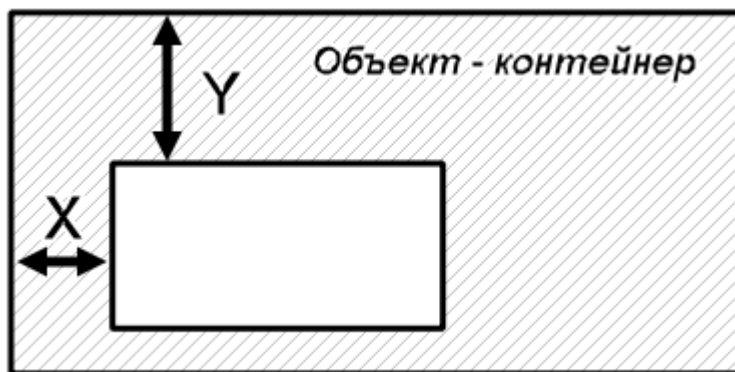
## 1.1.4.4.3.3.11.3.6. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.11.3.7. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.4.3.3.11.3.8. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;



//Увеличить угол поворот объекта на 1 градус относительно текущего значения:  
Object.Rotation += 1;

## 1.1.4.4.3.3.11.3.9. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.11.3.10. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.11.3.11. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.11.3.12. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.11.3.13. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.11.3.14. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




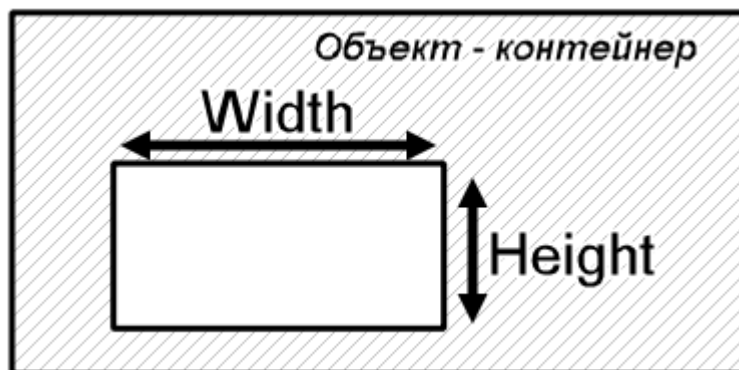
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.11.3.15. Ширина

Ширина объекта в пикселях.

 float8 Width



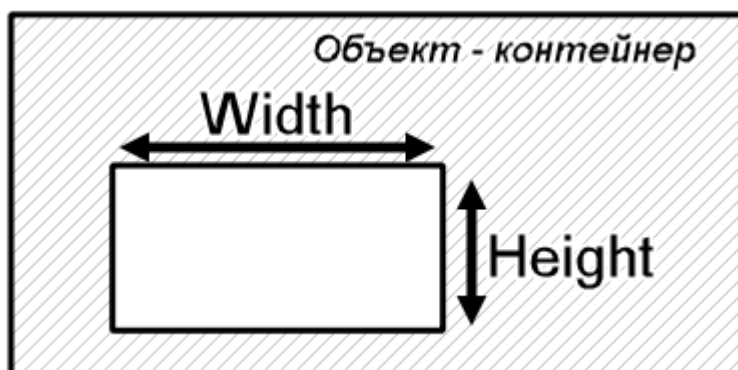


## 1.1.4.4.3.3.11.3.16. Высота

Высота объекта в пикселях.



float8 Height



# 1.1.4.4.3.3.11.3.17. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

## Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

## Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.11.3.18. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.11.3.19. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.11.3.20. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.11.3.21. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;



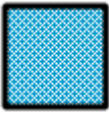

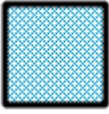

## 1.1.4.4.3.3.11.3.22. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```



## 1.1.4.4.3.3.11.3.23. X окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству X для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowX

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowX = 10;
```

## 1.1.4.4.3.3.11.3.24. Y окна

Позиция окна экранной формы по горизонтали и вертикали. Измеряется в пикселях. Действует аналогично свойству Y для объектов. За начало отчета (точка 0;0) берется левый верхний угол монитора.



int4 WindowY

### Примеры



```
//Установить позицию окна экранной формы  
here.WindowY = 50;
```

## 1.1.4.4.3.3.11.3.25. Ширина окна

Ширина окна экранной формы в пикселях. Может отличаться от свойства Ширина, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowWidth

### Примеры



```
//Установить ширину окна  
here.WindowWidth = 800;
```

## 1.1.4.4.3.3.11.3.26. Высота окна

Высота окна экранной формы в пикселях. Может отличаться от свойства Высота, которое относится к контейнеру экранной формы. Если размеры окна больше размеров контейнера экранной формы, то в режиме исполнения будут видны свободные области. Если размеры окна меньше размеров контейнера экранной формы, то снизу и справа появятся полосы прокрутки.



int4 WindowHeight

### Примеры



```
//Установить высоту окна  
here.WindowHeight = 600;
```

## 1.1.4.4.3.3.11.3.27. Заголовок окна

Надпись в строке заголовка.



string WindowCaption

### Примеры



```
//Установить заголовок окна:  
here.WindowCaption = "Заголовок окна";
```

## 1.1.4.4.3.3.11.3.28. Показывать заголовок

Включает/отключает заголовок окна вместе со всеми кнопками.



bool ShowWindowCaption

Чтобы скрыть заголовок окна, и тем самым скрыть все кнопки управления окном и запретить пользователю захват окна для перемещения, укажите для свойства Показывать заголовок значение false.

### Примеры



```
//Убрать заголовок окна  
here.ShowWindowCaption = false;
```

# 1.1.4.4.3.3.11.3.29. Показывать кнопку "свернуть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMinimizen

## Примеры



```
//Убрать кнопку "свернуть":  
here.ShowWindowMinimizen = false;
```

# 1.1.4.4.3.3.11.3.30. Показывать кнопку "развернуть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowMaximize

## Примеры



```
//Убрать кнопку "развернуть":  
here.ShowWindowMaximize = false;
```



# 1.1.4.4.3.3.11.3.31. Показывать кнопку "заккрыть"

Включает/отключает кнопку  в заголовке окна.



bool ShowWindowClose

## Примеры



```
//Убрать кнопку "заккрыть":  
here.ShowWindowClose = false;
```

## 1.1.4.4.3.3.11.3.32. Поверх всех окон

Определяет режим отображения экранной формы поверх остальных окон мнемосхемы.



bool AlwaysOnTop

### Примеры



```
//Установить отображение экранной формы поверх остальных окон:  
here.AlwaysOnTop = true;
```

## 1.1.4.4.3.3.11.3.33. Размеры окна

Определяет режим вычисления размеров окна.



uint4 WindowSizeMode

### Значение

Значение	Описание
0	Вручную
1	Автоподбор при открытии
2	Автоподбор

### Примеры



```
//Отобразить режим вычисления размеров окна в текстовом поле:  
TextEdit.Text = here.WindowSizeMode;
```

# 1.1.4.4.3.3.11.3.34. Стиль рамки окна

Стиль рамки окна экранной формы.



uint4 WindowBorderStyle

## Значение

Значение	Описание
0	Без рамки
1	Изменяемый размер
2	Фиксированный размер

## Примеры



```
//Отобразить код текущего стиля рамки окна в текстовом поле:  
TextEdit.Text = here.WindowBorderStyle;
```

## 1.1.4.4.3.3.11.3.35. Состояние окна

Определяет режим, в котором будет открываться окно экранной формы.



uint4 WindowState

### Значение

Значение	Описание
0	По умолчанию
1	Развернуто
2	На весь экран
3	Свернуто

### Примеры



```
//Показать код текущего режима в текстовом поле:  
TextEdit.Text = here.WindowState;
```

# 1.1.4.4.3.3.11.3.36. Режим масштабирования

Автоматически подгоняет размеры контейнера экранной формы до размеров окна.



uint4 WindowScalingMode

## Значение

Значение	Описание
0	Не масштабировать
1	Всегда масштабировать: автоматическая подгонка размеров будет работать в любых ситуациях
2	Только уменьшать: контейнер экранной формы, размеры которого выходят за границы окна, будет автоматически уменьшен до размеров окна

## Примеры



```
//Включить режим "Всегда масштабировать":  
here.WindowScalingMode = 1;
```

## 1.1.4.4.3.3.11.3.37. Монитор

Определяет номер монитора для открытия окна экранной формы, если используется многомониторная система отображения.



int4 MonitorNumber



Если вместо многомониторной системы отображения используется единственный монитор, то во всех случаях значение свойства Монитор нужно оставлять равным 0.

### Примеры



```
//Отобразить номер монитора в текстовом поле:  
TextEdit.Text = here.MonitorNumber;
```

## 1.1.4.4.3.3.11.3.38. Положение окна

Определяет позицию для открытия окна экранной формы.



uint4 WindowPosition

### Значение

Значение	Описание
0	Вручную
1	По центру монитора
2	По центру родительского окна
3	По центру родительской формы
4	Вручную относительно монитора

### Примеры



```
//Отобразить код позиции в текстовом поле:  
TextEdit.Text = here.WindowPosition;
```



## 1.1.4.4.3.3.11.3.39. Режим обработки закрытия окна

Позволяет пользователю определять условия закрытия окна. Для определения условия закрытия окна следует выбрать в редакторе свойств значение Отправить запрос. Свойство связано с функцией Close().



uint2 WindowCloseMode



Свойство игнорируется для окон, открытых с помощью элемента Фрейм.

### Примеры



```
//Режим обработки закрытия окна - Отправить запрос:  
here.WindowCloseMode = 2;
```

## 1.1.4.4.3.3.11.3.40. Путь к файлу иконки

Путь к файлу иконки.



string WindowIconPath

# 1.1.4.4.3.3.11.3.41. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.11.3.42. Высота элемента

Высота элемента, относительно которого открывается календарь.



int4 PointHeight

## 1.1.4.4.3.3.11.3.43. IsSubmenu

Указывает, является ли данный экземпляр Menu подменю другого экземпляра Menu.



bool IsSubmenu


## 1.1.4.4.3.3.11.3.44. Элементы меню

Хранит перечень элементов меню в виде JSON массива, содержащего описание каждого элемента.



string Items

Структура:

- › идентификатор команды `id` и заголовок `title` (обязательные): «`{ 'id': "cmd", 'title': "Команда" }`»;
- › параметр `checked` со значением «`true`» – создаст элемент выбора с иконкой : «`{ 'id': "slim", 'title': "Компактный вид", 'checked': true }`»;
- › параметр `icon` добавит элементу меню иконку, находящуюся по указанному пути: «`{ 'id': "save", 'title': "Сохранить", 'icon': "Files_Icons/SaveToFile16.png" }`»;
- › параметр `staysOpen` со значением «`true`» – оставляет окно меню открытым при выборе элемента: «`{ 'id': i, 'title': "Сохранить как...", 'staysOpen': true }`»;
- › разделитель элементов меню задается в виде значения параметра `title`: «`{ 'title': "-" }`».

## Примеры

Создана функция `GetItems()`. JSON массив пунктов меню описан в функции на языке JavaScript.



```
var items = [];  
items.push({  
  'id': "addSignal",  
  'title': "Добавить новый сигнал...",  
  'active': true  
});
```

```

items.push({
  'id': "saveSignal",
  'title': "Сохранить сигнал...",
  'active': true
});
items.push({
  'id': "delSignal",
  'title': "Удалить сигнал...",
  'active': true
});
return JSON.stringify(items);

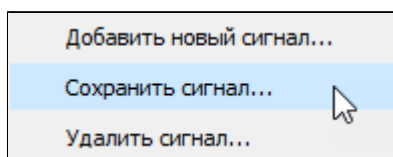
```

Тогда в качестве значения свойства указана ссылка на функцию.

The screenshot displays two windows from the Visual Studio IDE:

- Структура объекта (Object Structure):** A tree view showing the application's object model. The 'MainForm' object is expanded to show 'Графические объекты' (Visual Objects), which includes 'ShowMenu\_Button' (a Button). Under 'ShowMenu\_Button', there is a 'Данные' (Data) section containing 'showMenu' (a Command), 'onComplete' (a Command), and 'MenuResult' (a MenuResult). Below this is a 'Функции' (Functions) section where 'GetItems' (a Function) is highlighted with a blue box.
- События (Events):** A window showing the event handlers for the application. The 'Invoker' is 'Handler\_2' (Open <Menu> in dialog window). The 'Items' property is selected, and its handler is 'GetItems()' (Function), which is also highlighted with a blue box.

Приведенный в примере JSON массив в проекте в режиме исполнения выглядит так:





## 1.1.4.4.3.3.11.3.45. Ссылка на результат

Ссылка на экземпляр типа [MenuResult](#), в котором будет сохранен результат выбора пункта меню.

## 1.1.4.4.3.3.11.3.46. pAction

Ссылка на команду, выполняемую при выборе элемента меню.

## 1.1.4.4.3.3.11.3.47. pPrevMenu

Ссылка на команду, выполняемую при переходе к предыдущему пункту меню (при управлении меню клавишами).

## 1.1.4.4.3.3.11.3.48. pNextMenu

Ссылка на команду, выполняемую при переходе к следующему пункту меню (при управлении меню клавишами).

## 1.1.4.4.3.3.11.3.49. pOnDoClose

Ссылка на команду.

## 1.1.4.4.3.3.11.3.50. pOnClosed

Ссылка на команду, выполняемую при закрытии меню.

## 1.1.4.4.3.3.11.3.51. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.11.3.52. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```

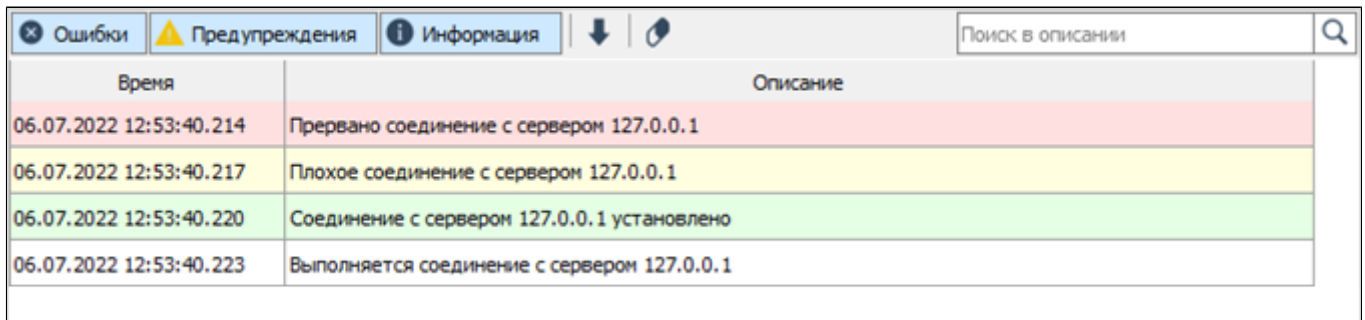


```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.4.3.3.12. LogView

Журнал сообщений.



Время	Описание
06.07.2022 12:53:40.214	Прервано соединение с сервером 127.0.0.1
06.07.2022 12:53:40.217	Плохое соединение с сервером 127.0.0.1
06.07.2022 12:53:40.220	Соединение с сервером 127.0.0.1 установлено
06.07.2022 12:53:40.223	Выполняется соединение с сервером 127.0.0.1

Отображает сообщения в виде таблицы. По умолчанию в таблице отображаются только Время и Описание сообщений.

Чтобы выводить дополнительную информацию, измените значение свойства [AdditionalColumns](#).

Сообщения в журнале могут иметь разную важность, обозначаемую цветом фона сообщения:

- ошибки – красные;
- предупреждения – желтые;
- информационные сообщения: обычные – зеленые, и отладочные – белые.

Подсказки по работе с журналом, добавленным в проект, в режиме исполнения:

- чтобы в журнале не отображались сообщения определенного типа, следует отжать кнопку с названием типа на панели инструментов журнала;
- чтобы следить за новыми сообщениями, нужно нажать кнопку Автоматическое прокручивание списка;
- чтобы очистить журнал сообщений, нажмите кнопку Очистить список;
- чтобы найти в журнале сообщения с определенной фразой, введите ее в окне Поиск в списке.

## 1.1.4.4.3.12.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.3.12.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.12.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.12.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.12.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.12.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.12.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.12.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">AddMessage</a>	Добавляет в журнал сообщение с заданной важностью
<a href="#">Add</a>	Добавляет в журнал сообщение с заданной важностью и дополнительными данными
<a href="#">Clear</a>	Очищает журнал сообщений
<a href="#">GetViewSettings</a>	Возвращает настройки внешнего вида журнала в виде json-строки
<a href="#">UpdateView</a>	Настраивает внешний вид журнала в соответствии с настройками, которые могут быть получены в результате вызова функции GetViewSettings()



## 1.1.4.4.3.3.12.2.1. AddMessage

Добавляет в журнал сообщение с заданной важностью.



```
void AddMessage(uint4 Severity, string Message)
```



Если в свойстве `AdditionalColumns` указаны дополнительные столбцы, используйте функцию `Add()`.

### Параметры

Параметр	Тип	Описание
Severity	uint4	Важность: <ul style="list-style-type: none"><li>&gt; 0 – отладочное информационное сообщение, цвет фона – белый</li><li>&gt; 1 – информационное сообщение, цвет фона – зеленый</li><li>&gt; 2 – важное сообщение, цвет фона – желтый</li><li>&gt; 3 – сообщение об ошибке, цвет фона – красный</li></ul>
Message	string	Текст сообщения

### Примеры



```
LogView.AddMessage(3, "Прервано соединение с сервером  
127.0.0.1");
```

## 1.1.4.4.3.3.12.2.2. Add

Добавляет в журнал сообщение с заданной важностью и дополнительными данными.



```
void AddMessage(uint4 Severity, string Message, string  
AdditionalColumns)
```



Эта функция используется, если в свойстве `AdditionalColumns` указаны дополнительные столбцы.

### Параметры

Параметр	Тип	Описание
Severity	uint4	Важность: <ul style="list-style-type: none"><li>&gt; 0 – отладочное информационное сообщение, цвет фона – белый</li><li>&gt; 1 – информационное сообщение, цвет фона – зеленый</li><li>&gt; 2 – важное сообщение, цвет фона – желтый</li><li>&gt; 3 – сообщение об ошибке, цвет фона – красный</li></ul>
Message	string	Текст сообщения
AdditionalColumns	string	Дополнительные данные в виде json-строки: { "id1":value1, "id2":value2, ... }, помещаемые в дополнительные столбцы таблицы сообщений

### Примеры



//Вызов функции в JavaScript

```
Log.Add(3, "Прервано соединение с сервером 127.0.0.1",  
JSON.stringify({"address":"127.0.0.1", "port":4388}));
```



//Вызов функции в Astra.Om

```
Log.Add(3, "Прервано соединение с сервером 127.0.0.1", "{ \"address  
\": \"127.0.0.1\", \"port\": 4388 }");
```

## 1.1.4.4.3.3.12.2.3. Clear

Очищает журнал сообщений.



```
void Clear()
```

### Примеры



```
LogView.Clear();
```

## 1.1.4.4.3.3.12.2.4. GetViewSettings

Возвращает настройки внешнего вида журнала в виде json-строки.



string GetViewSettings()

Настройки внешнего вида содержат информацию о ширине столбцов, порядке сортировки, включенных кнопках фильтрации, текста поиска, режим слежения за последним сообщением. Эти настройки могут использоваться для восстановления внешнего вида журнала с помощью функции UpdateView() при перезапуске проекта.

### Пример json-строки



```
{"Columns":{"importance":{"Index":0,"Width":20,"Hidden":true},
"time":{"Index":1,"Width":150,"Hidden":false},
"text":{"Index":2,"Width":424,"Hidden":false,"SortOrder":0,"SortPriority":0},
"address":{"Index":3,"Width":100,"Hidden":false},
"port":{"Index":4,"Width":100,"Hidden":false}},
"Filter":{"Errors":true,"Warnings":true,"Informations":true,"Text":""},
"FollowLatest":true}
```

### Примеры



```
settings: string = LogView.GetViewSettings();
```

## 1.1.4.4.3.3.12.2.5. UpdateView

Настраивает внешний вид журнала в соответствии с настройками, которые могут быть получены в результате вызова функции `GetViewSettings()`.



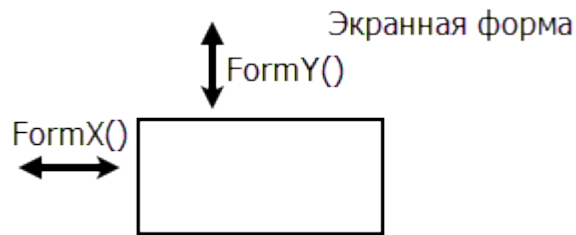
```
void UpdateView(string Settings)
```

### Параметры

Параметр	Тип	Описание
Settings	string	Настройки внешнего вида журнала в виде json-строки

## 1.1.4.4.3.3.12.2.6. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



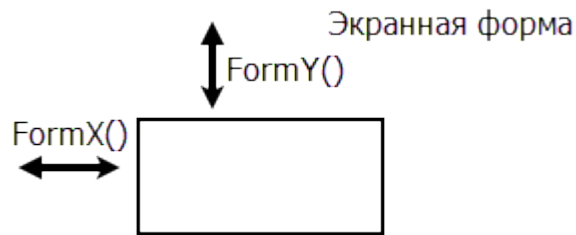
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.12.2.7. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

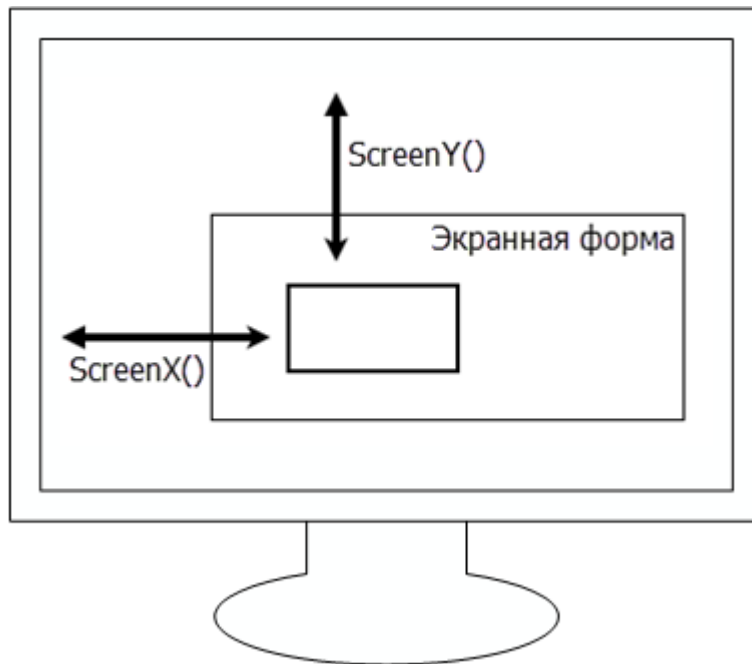
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.12.2.8. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

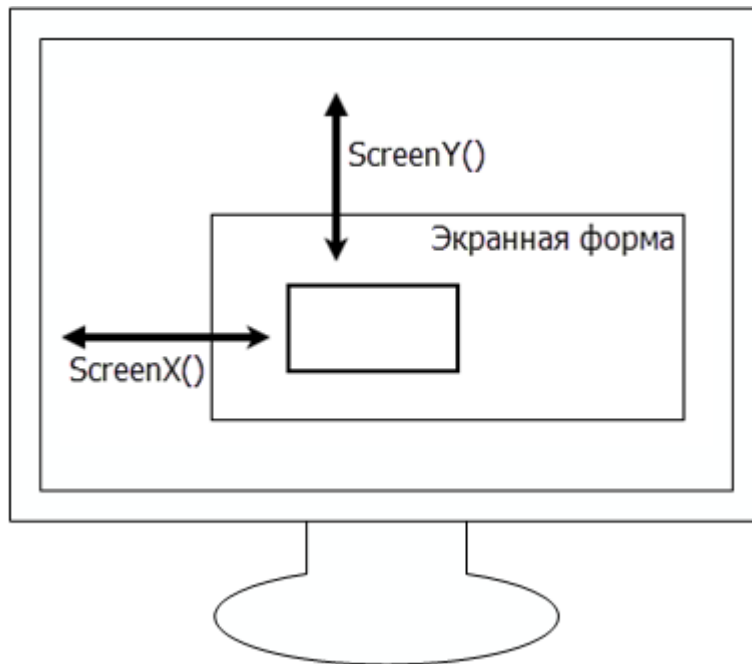
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.12.2.9. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.12.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">IsVisible</a>	Меняет видимость журнала

<a href="#">Количество ошибок</a>	Хранит текущее количество сообщений об ошибках в журнале
<a href="#">Количество предупреждений</a>	Хранит текущее количество предупреждений в журнале
<a href="#">Количество сообщений</a>	Хранит текущее количество информационных сообщений в журнале
<a href="#">Дополнительные столбцы</a>	Перечень дополнительных столбцов таблицы сообщений в виде json-строки
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.12.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.12.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.12.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

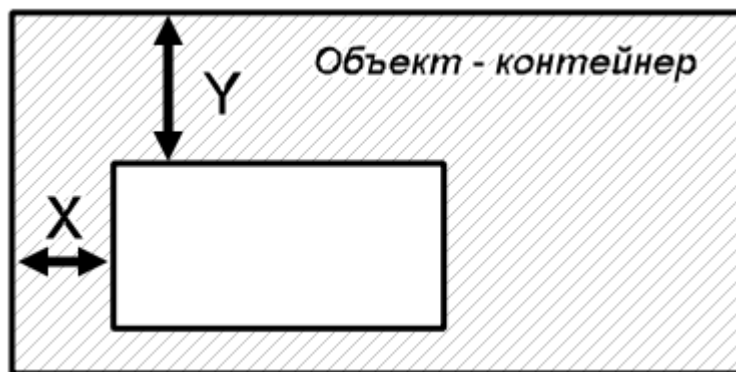
## 1.1.4.4.3.3.12.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.12.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.12.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.12.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.12.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.12.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.12.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.12.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.12.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

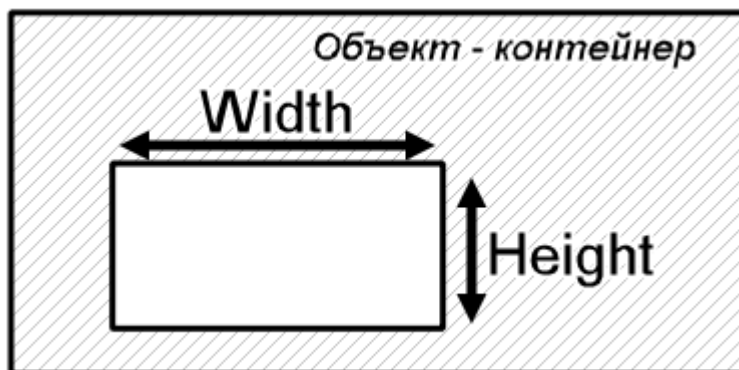


## 1.1.4.4.3.3.12.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

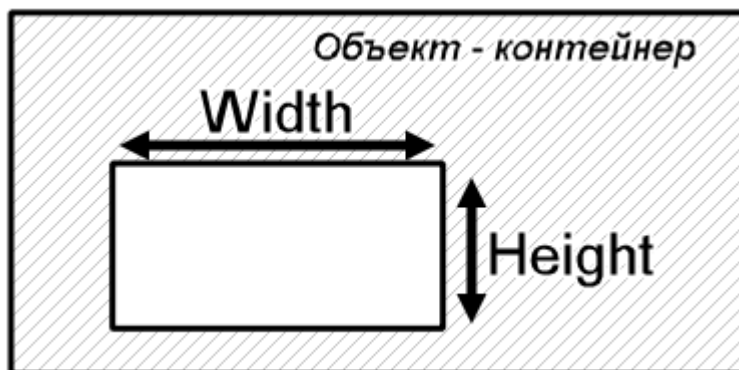


## 1.1.4.4.3.3.12.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.12.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

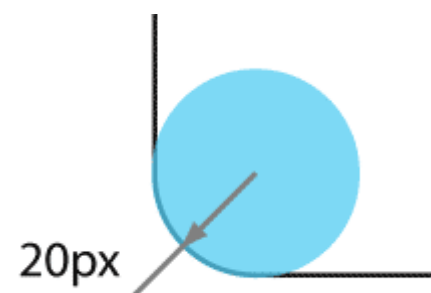
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.12.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.12.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.12.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.12.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.12.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;



## 1.1.4.4.3.3.12.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.12.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.12.3.23. IsVisible

Меняет видимость журнала.



bool IsVisible

### Значение

Значение	Описание
true	Показать журнал
false	Скрыть журнал

## 1.1.4.4.3.3.12.3.24. Количество ошибок

Хранит текущее количество сообщений об ошибках в журнале.



int4 ErrorCount

# 1.1.4.4.3.3.12.3.25. предупреждений

Количество

Хранит текущее количество предупреждений в журнале.



int4 WarningCount

## 1.1.4.4.3.3.12.3.26. Количество сообщений

Хранит текущее количество информационных сообщений в журнале.



int4 InformationCount

## 1.1.4.4.3.3.12.3.27. Дополнительные столбцы

Перечень дополнительных столбцов таблицы сообщений в виде json-строки.



string AdditionalColumns



## 1.1.4.4.3.3.12.3.28. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.12.3.29. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



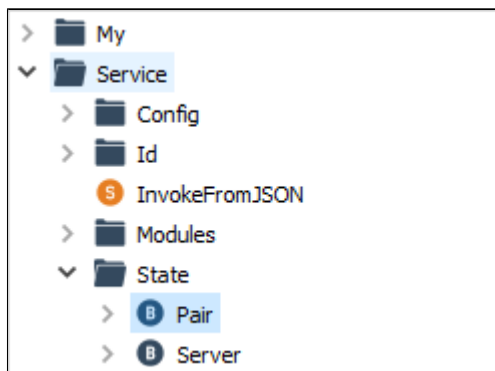
```
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.13. ItemTree

Дерево сигналов. Предназначено для отображения адресного пространства сервера в виде дерева.



## 1.1.4.4.3.3.13.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.3.13.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.13.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.13.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.13.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.13.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```



## 1.1.4.4.3.3.13.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

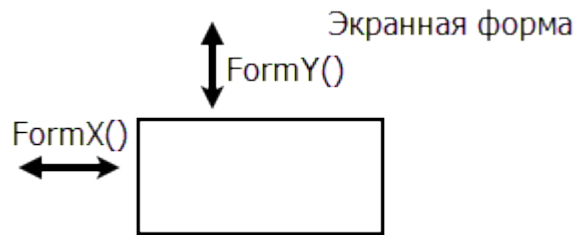
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.13.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">GetCurrentItemInfo</a>	Возвращает описание выделенного в дереве сигнала в виде json-строки

## 1.1.4.4.3.3.13.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



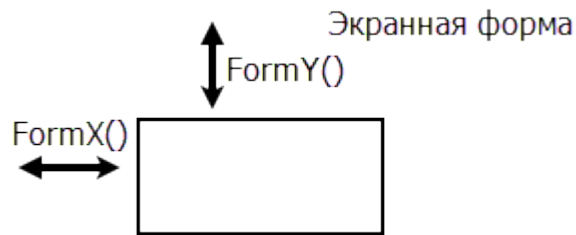
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.13.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



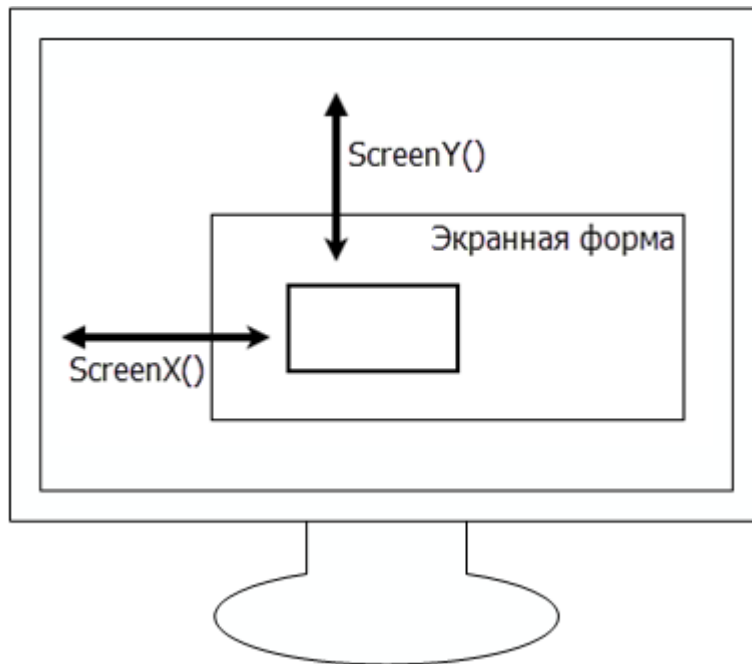
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.13.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

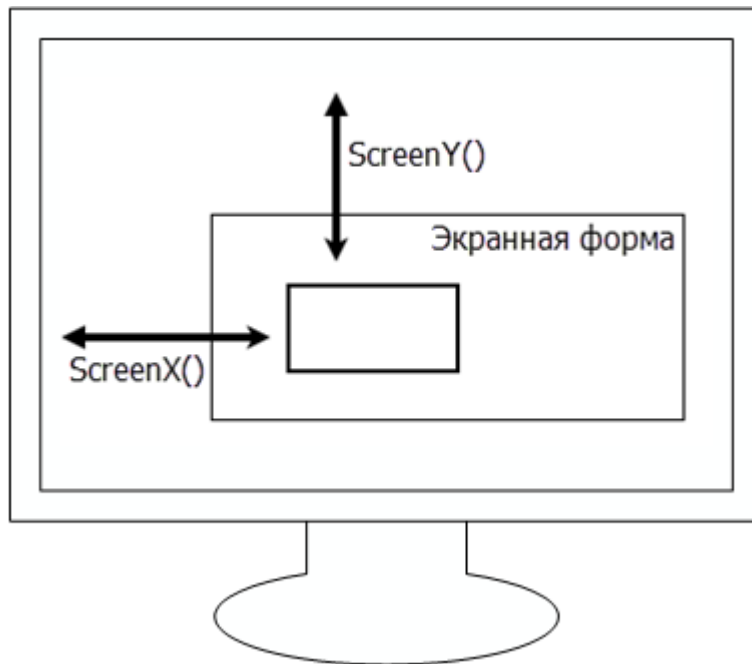
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.13.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.13.2.5. GetCurrentItemInfo

Возвращает описание выделенного в дереве сигнала в виде json-строки.  
Функция не требует входных параметров.



```
string GetCurrentItemInfo()
```

### Формат json-строки



Пример возвращаемого результата:

```
{  
  "tag": "Messages.AdminMessageSignal",  
  "dataType": "string"  
}
```



Пример использования в обработчике команды, на которую ссылается [pDragStarted](#):

```
// получение значения, возвращаемого функцией  
var json = GetCurrentItemInfo();  
if (json != "")  
{  
  // разбираем объект info  
  var info = JSON.parse(json); //  
  записываем в журнал сообщения о начале перетаскивания  
  сигнала с именем  
  <tag>  
  LogView.AddMessage(0, "Начато перетаскивание " + info.tag);  
}
```

### Пример



```
info: string = GetCurrentItemInfo();
```



## 1.1.4.4.3.3.13.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуального компонента
<a href="#">Вывод описания предпочтительнее имени</a>	Позволяет выбрать, что отображать в дереве

<a href="#">Ссылка на источник</a>	Сервер, дерево сигналов которого должно отображаться в ItemTree
<a href="#">pDragStarted</a>	Ссылка на команду, выполняемую при перетаскивании узла дерева левой кнопкой мыши
<a href="#">pDoubleClicked</a>	Ссылка на команду, выполняемую при двойном клике левой кнопкой мыши по узлу дерева
<a href="#">pMouseRightClicked</a>	Ссылка на команду, выполняемую при клике правой кнопкой мыши по узлу дерева
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.13.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.13.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.13.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.13.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.13.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.13.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```



## 1.1.4.4.3.3.13.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.13.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.13.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.13.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.13.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.13.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true


### Примеры

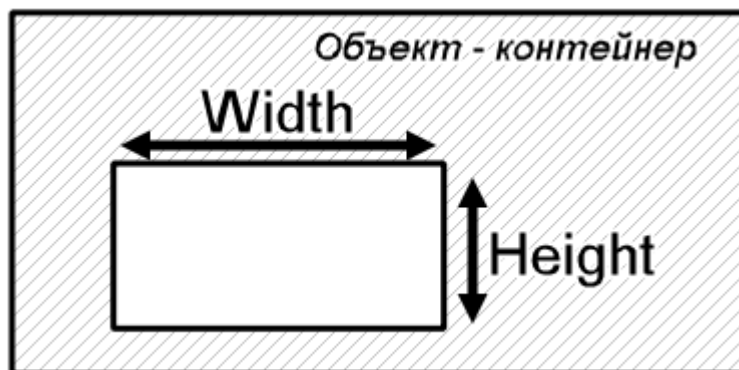


```
//Задать всплывающую подсказку для объекта:  
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.13.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

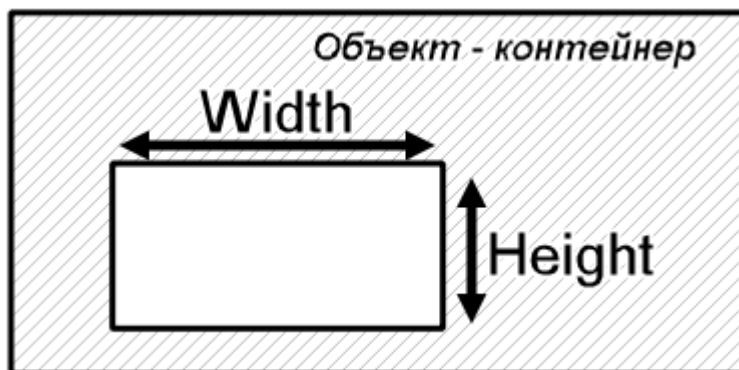


## 1.1.4.4.3.3.13.3.14. Высота

Высота объекта в пикселях.



float8 Height





## 1.1.4.4.3.3.13.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

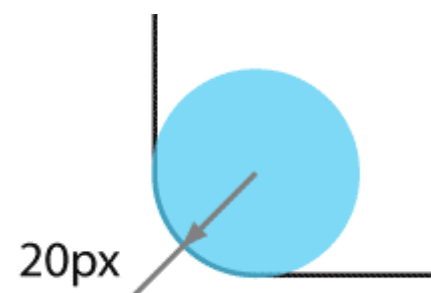
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.13.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.13.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.13.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.13.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.13.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;



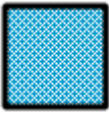

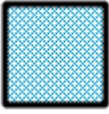

## 1.1.4.4.3.3.13.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;



## 1.1.4.4.3.3.13.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

# 1.1.4.4.3.3.13.3.23. Вывод описания предпочтительнее имени

Позволяет выбрать, что отображать в дереве: имя сигнала или его описание:

- › «true» – отображать описание сигнала (если не указано, то имя);
- › «false» – отображать имя сигнала.



bool DescriptionPreferred

## 1.1.4.4.3.3.13.3.24. Ссылка на источник

Здесь указывается сервер, дерево сигналов которого должно отображаться в ItemTree. Тип значения – ссылка на источник данных типа ApSource.



ApSource init\_ApSource

## 1.1.4.4.3.3.13.3.25. pDragStarted

Ссылка на команду, выполняемую при перетаскивании узла дерева левой кнопкой мыши.

## 1.1.4.4.3.3.13.3.26. pDoubleClicked

Ссылка на команду, выполняемую при двойном клике левой кнопкой мыши по узлу дерева.

## 1.1.4.4.3.3.13.3.27. pMouseRightClicked

Ссылка на команду, выполняемую при клике правой кнопкой мыши по узлу дерева. Может использоваться, например, для вызова контекстного меню.

## 1.1.4.4.3.3.13.3.28. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.13.3.29. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.4.3.3.14. FileList

Список файлов и папок, содержащихся в указанной папке.

Имя	Дата изменения	Размер	
configuration_files	17.11.2022 14:12:29		▲
new_backups	17.11.2022 14:12:29		
old_backups	17.11.2022 14:12:29		
17112022_bckp.csv	17.11.2022 14:12:53	6377	
17112022_bckp.pdf	17.11.2022 14:12:53	6377	
18112022_bckp.csv	17.11.2022 14:12:53	6377	
18112022_bckp.pdf	17.11.2022 14:12:53	6377	▼

Указать папку, содержимое которой будет отображаться в компоненте, настроить фильтры отображения файлов и настроить ограничения на просмотр файловой системы можно с помощью свойств компонента.

## 1.1.4.4.3.3.14.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.3.14.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.14.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.14.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.14.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.14.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.14.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

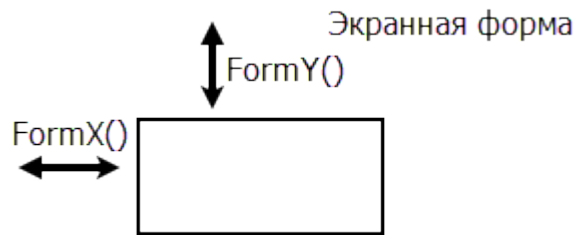
## 1.1.4.4.3.3.14.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y



## 1.1.4.4.3.3.14.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



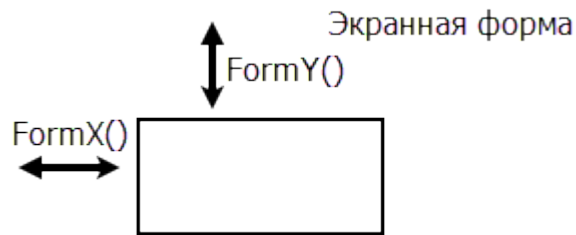
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.14.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



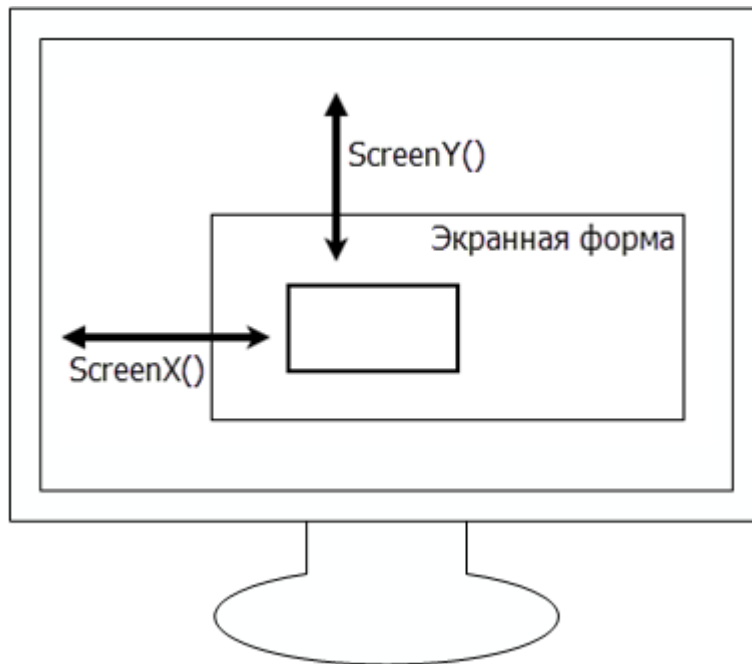
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.14.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

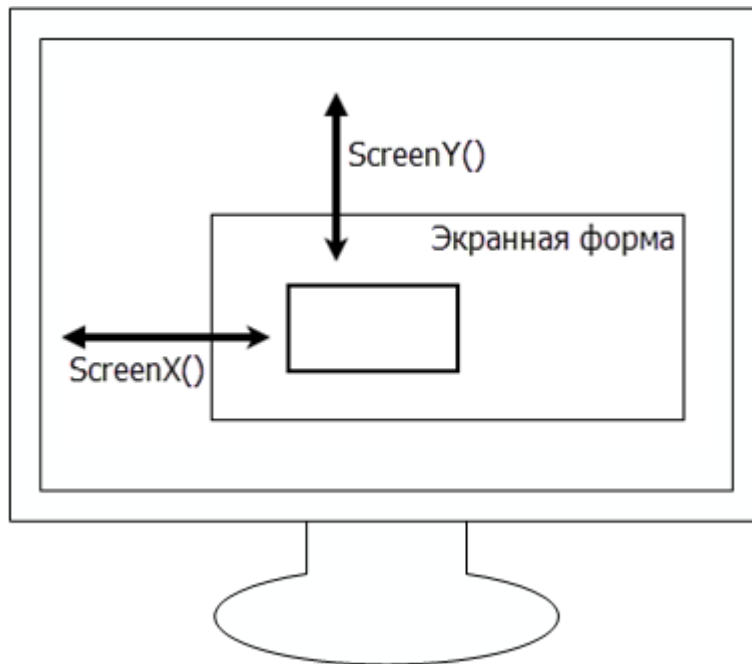
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.14.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.14.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов

<a href="#">Корневой путь</a>	Ограничивает область просмотра файловой системы
<a href="#">Отображаемая папка</a>	Путь к папке, содержимое которой должно отображаться в окне FileList
<a href="#">Фильтр по типу файла</a>	Фильтр типа файлов, отображаемых в окне FileList
<a href="#">Показывать свойства файлов</a>	Позволяет отображать больше информации о файлах и папках в окне FileList
<a href="#">Разрешена смена папки</a>	Указывает на возможность смены папки, открываемой в окне FileList
<a href="#">Выделенный путь</a>	Путь к файлу или папке, выбранной в окне FileList
<a href="#">Выделенный файл</a>	Название файла, выбранного в окне FileList
<a href="#">OnPathChosen</a>	Ссылка на команду, выполняемую в результате двойного щелчка ЛКМ по строке файла в списке
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.14.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.14.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



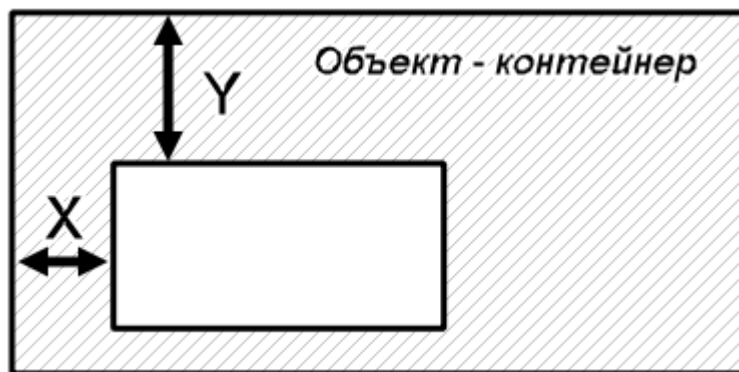
## 1.1.4.4.3.3.14.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

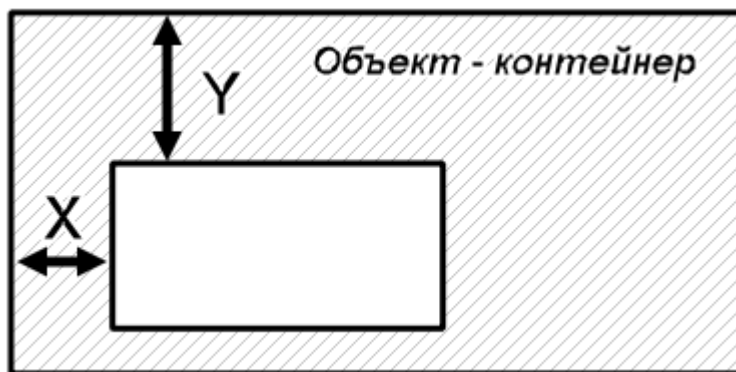
## 1.1.4.4.3.3.14.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.14.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.14.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.14.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.14.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.14.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.14.3.10. Непрозрачность

Непрозрачность.



float8 Opacity



## 1.1.4.4.3.3.14.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.14.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

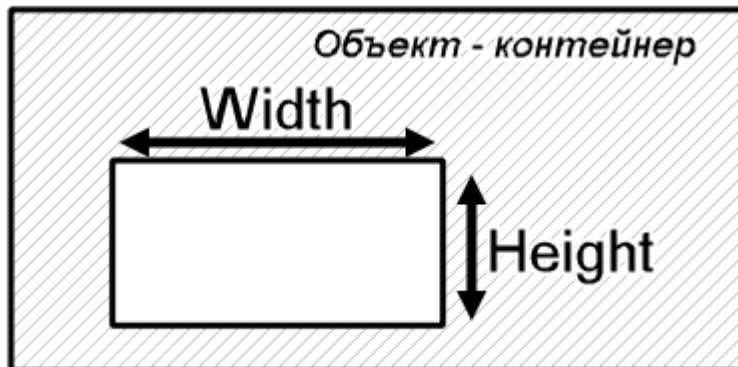
```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.14.3.13. Ширина

Ширина объекта в пикселях.



float8 Width

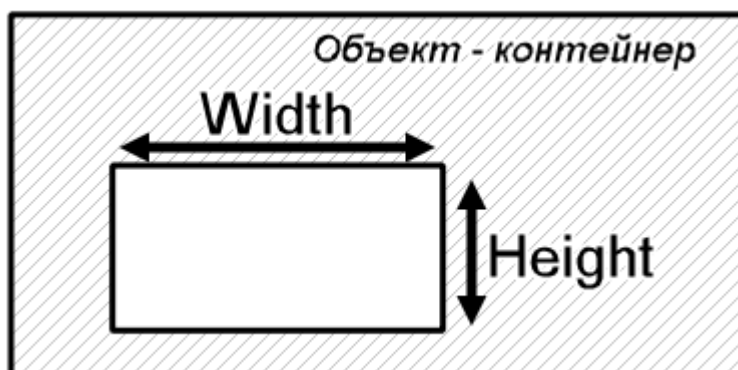


## 1.1.4.4.3.3.14.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.14.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

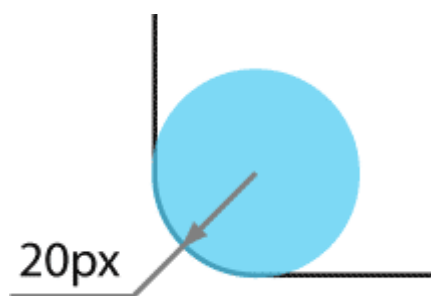
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.14.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.14.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.14.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;



## 1.1.4.4.3.3.14.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.14.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;



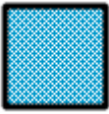

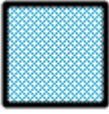
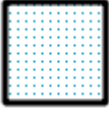
## 1.1.4.4.3.3.14.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.14.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.14.3.23. Корневой путь

Ограничивает область просмотра файловой системы.



string RootPath

## 1.1.4.4.3.3.14.3.24. Отображаемая папка

Путь к папке, содержимое которой должно отображаться в окне FileList.



string BrowsedPath

Полнота отображения – только папки, папки и все файлы или папки и файлы определенного типа – указывается в свойстве ExtensionFilter (Фильтр по типу файла).

## 1.1.4.4.3.3.14.3.25. Фильтр по типу файла

Фильтр типа файлов, отображаемых в окне FileList.

- › Если не указывать значение, то в списке будут отображаться все файлы и папки.
- › Если указать конкретное расширение, например \*.csv, то будут отображаться папки и файлы с указанным расширением.



string ExtensionFilter



# 1.1.4.4.3.3.14.3.26. Показывать свойства файлов

Позволяет отображать больше информации о файлах и папках в окне FileList.



bool ShowFileProperties

## Значение

Значение	Описание																		
true	Для файлов и папок отображаются имена и даты изменений, а для файлов – еще размеры (в байтах) <table border="1"><thead><tr><th>Имя</th><th>Дата изменения</th><th>Размер</th></tr></thead><tbody><tr><td>new_backups</td><td>17.11.2022 14:12:29</td><td></td></tr><tr><td>old_backups</td><td>17.11.2022 14:12:29</td><td></td></tr><tr><td>17112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr><tr><td>18112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr><tr><td>19112022_bckp.csv</td><td>17.11.2022 14:12:53</td><td>6377</td></tr></tbody></table>	Имя	Дата изменения	Размер	new_backups	17.11.2022 14:12:29		old_backups	17.11.2022 14:12:29		17112022_bckp.csv	17.11.2022 14:12:53	6377	18112022_bckp.csv	17.11.2022 14:12:53	6377	19112022_bckp.csv	17.11.2022 14:12:53	6377
Имя	Дата изменения	Размер																	
new_backups	17.11.2022 14:12:29																		
old_backups	17.11.2022 14:12:29																		
17112022_bckp.csv	17.11.2022 14:12:53	6377																	
18112022_bckp.csv	17.11.2022 14:12:53	6377																	
19112022_bckp.csv	17.11.2022 14:12:53	6377																	
false	Отображаются только имена файлов и папок <table border="1"><tbody><tr><td>new_backups</td></tr><tr><td>old_backups</td></tr><tr><td>17112022_bckp.csv</td></tr><tr><td>18112022_bckp.csv</td></tr><tr><td>19112022_bckp.csv</td></tr></tbody></table>	new_backups	old_backups	17112022_bckp.csv	18112022_bckp.csv	19112022_bckp.csv													
new_backups																			
old_backups																			
17112022_bckp.csv																			
18112022_bckp.csv																			
19112022_bckp.csv																			

## 1.1.4.4.3.3.14.3.27. Разрешена смена папки

Указывает на возможность смены папки, открываемой в окне FileList.



bool ChangeFolderAllowed

### Значение

Значение	Описание
true	Можно сменить открывшуюся папку
false	Нельзя сменить открывшуюся папку

## 1.1.4.4.3.3.14.3.28. Выделенный путь

Путь к файлу или папке, выбранной в окне FileList.



string SelectedPath

### Примеры



Допустим, в свойстве BrowsedPath (Отображаемая папка) указано значение D:\CommonLib\backups, тогда в окне FileList отображается содержимое этой папки.

Имя	Дата изменения	Размер
configuration_files	17.11.2022 14:12:29	
new_backups	17.11.2022 14:12:29	
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377

В таком случае свойство SelectedPath (Выделенный путь):

- › при выборе папки new\_backups примет значение D:\CommonLib\backups\new\_backups;
- › при выборе файла 17112022\_bckp.csv примет значение D:\CommonLib\backups\17112022\_bckp.csv.

## 1.1.4.4.3.3.14.3.29. Выделенный файл

Название файла, выбранного в окне FileList.



string SelectedFileName

### Примеры



Допустим, в свойстве BrowsedPath (Отображаемая папка) указано значение D:\CommonLib\backups, тогда в окне FileList отображается содержимое этой папки.

В таком случае:

Имя	Дата изменения	Размер
configuration_files	17.11.2022 14:12:29	
new_backups	17.11.2022 14:12:29	
old_backups	17.11.2022 14:12:29	
17112022_bckp.csv	17.11.2022 14:12:53	6377
18112022_bckp.csv	17.11.2022 14:12:53	6377

› при выборе файла 17112022\_bckp.csv свойство SelectedFileName (Выделенный файл) примет значение 17112022\_bckp.csv;

› при выборе папки:

› если ранее не был выбран ни один файл, то значение свойства SelectedFileName (Выделенный файл) останется пустым;

› если ранее был выбран какой-либо файл, то имя этого файла будет храниться в качестве значения свойства SelectedFileName (Выделенный файл), пока не будет выбран другой файл и значение не перезапишется.

## 1.1.4.4.3.3.14.3.30. OnPathChosen

Ссылка на команду, выполняемую в результате двойного щелчка ЛКМ по строке файла в списке.

## 1.1.4.4.3.3.14.3.31. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.14.3.32. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.15. LoginDialogResult

Компонент, через который передаются id и отображаемое имя текущего пользователя от формы [LoginDialog](#) к форме [ChangePasswordDialog](#).

Чтобы не настраивать компонент самостоятельно, используйте [LoginButton](#). Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.



## 1.1.4.4.3.3.15.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">UserId</a>	Хранит уникальный идентификатор (uid) текущего пользователя подсистемы безопасности Astra.Security
<a href="#">UserDisplayName</a>	Хранит отображаемое имя текущего пользователя подсистемы безопасности Astra.Security
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.15.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.15.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.15.1.3. UserId

Хранит уникальный идентификатор (uid) текущего пользователя подсистемы безопасности Astra.Security.



string UserId

### Примеры



Вызов: LoginDialogResult\_1.UserId

Вид значения: alpha:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Пример значения: alpha:c3a997b1-326e-42ee-8907-e359fc17feb7.

## 1.1.4.4.3.3.15.1.4. UserDisplayName

Хранит отображаемое имя текущего пользователя подсистемы безопасности Astra.Security.



string UserDisplayName

### Примеры



Вызов: LoginDialogResult\_1.UserDisplayName

Пример значения: Иванов.

## 1.1.4.4.3.3.15.1.5. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.15.1.6. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.4.3.3.16. DialogResult

Компонент, в котором сохраняется имя и тип файла, выбранного в окне OpenFileDialog или SaveFileDialog.

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.



## 1.1.4.4.3.3.16.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Путь к папке</a>	Хранит полный путь к папке, в которой находится файл
<a href="#">Имя файла</a>	Хранит имя файла
<a href="#">Путь к файлу</a>	Хранит полный путь к файлу
<a href="#">Тип файла</a>	Хранит индекс типа файла
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.16.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.16.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.16.1.3. Путь к папке

Хранит полный путь к папке, в которой находится файл, выбранный в окне [OpenFileDialog](#) или [SaveFileDialog](#).



string FolderPath

## 1.1.4.4.3.3.16.1.4. Имя файла

Хранит имя файла, выбранного в окне [OpenFileDialog](#) или [SaveFileDialog](#).



string FileName

## 1.1.4.4.3.3.16.1.5. Путь к файлу

Хранит полный путь к файлу, выбранному в окне [OpenFileDialog](#) или [SaveFileDialog](#).



string FilePath

## 1.1.4.4.3.3.16.1.6. Тип файла

Хранит индекс типа файла, выбранного в окне [OpenFileDialog](#) или [SaveFileDialog](#).



int4 FileType

## 1.1.4.4.3.3.16.1.7. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.4.3.3.16.1.8. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.17. ConnectDialogResult

Компонент, в котором сохраняются адрес и порт сервера, к которому выполняется подключение в окне [ConnectDialog](#).

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.3.17.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Название</a>	Название подключения
<a href="#">Хост</a>	IP-адрес или имя сервера (компьютера), к которому выполняется подключение
<a href="#">Порт</a>	Номер порта, через который выполняется подключение к указанному серверу (компьютеру)
<a href="#">Порт истории</a>	Номер порта, через который выполняется подключение к указанному серверу (компьютеру)
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.17.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.17.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.17.1.3. Название

Название подключения.



string Title

## 1.1.4.4.3.3.17.1.4. Хост

IP-адрес или имя сервера (компьютера), к которому выполняется подключение.



string Location

## 1.1.4.4.3.3.17.1.5. Порт

Номер порта, через который выполняется подключение к указанному серверу (компьютеру).



uint2 Port



## 1.1.4.4.3.3.17.1.6. Порт истории

Номер порта, через который выполняется подключение к указанному серверу (компьютеру).



uint2 HistoryPort

## 1.1.4.4.3.3.17.1.7. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.17.1.8. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.18. ColorDialogResult

Компонент, в котором сохраняется значение, введенное в окне ColorDialog. Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.3.18.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Цвет</a>	Хранит цвет, выбранный в окне <a href="#">ColorDialog</a>
<a href="#">Пользовательская палитра</a>	Хранит перечень пользовательских цветов
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.18.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.18.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.18.1.3. Цвет

Хранит цвет, выбранный в окне ColorDialog.



uint4 Color



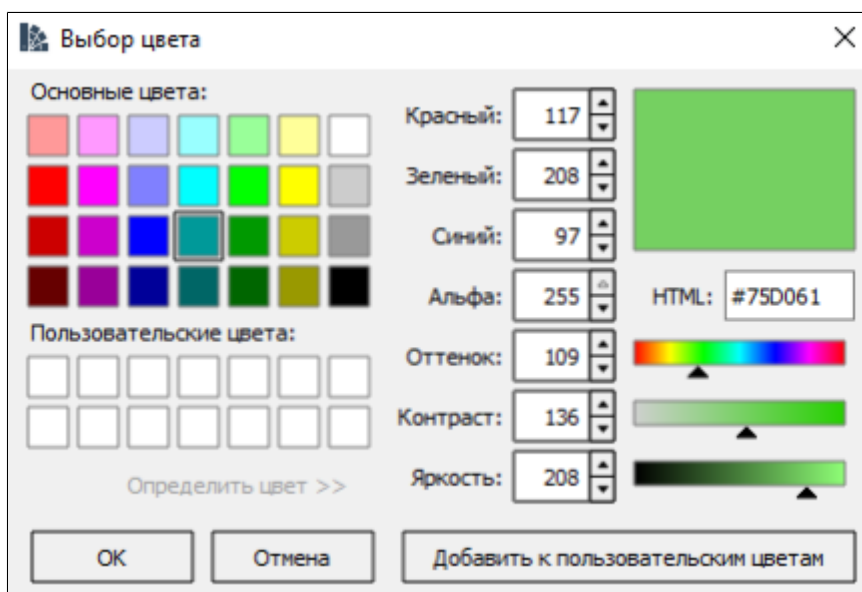
## 1.1.4.4.3.3.18.1.4. Пользовательская палитра

Хранит перечень пользовательских цветов.



string CustomColor

Добавить цвет в список пользовательских можно, расширив диалоговое окно и нажав кнопку **Добавить к пользовательским цветам**.



Набор пользовательских цветов сохраняется в виде JSON-массива в файл `colors.json` в папке проекта.

## 1.1.4.4.3.3.18.1.5. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.18.1.6. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.4.3.3.19. InputDialogResult

Компонент, в котором сохраняется значение, введенное в окне InputDialog. Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.3.19.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Значение</a>	Итоговое значение, введенное пользователем
<a href="#">Вводимый текст</a>	Вводимый текст
<a href="#">Описание ошибок</a>	Текст ошибки ввода даты
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.19.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.19.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.19.1.3. Значение

Итоговое значение, введенное пользователем.



string Value



## 1.1.4.4.3.3.19.1.4. Вводимый текст

Вводимый текст.

Может использоваться:

- › для проверки введенного текста;
- › в качестве начального значения при открытии окна `InputDialog`.

## 1.1.4.4.3.3.19.1.5. Описание ошибки

Текст ошибки ввода даты.



string Error

При возникновении ошибки ввода:

- › вокруг поля ввода появляется красная рамка;
- › во всплывающей подсказке появляется текст ошибки;
- › кнопка ОК блокируется.

### Примеры

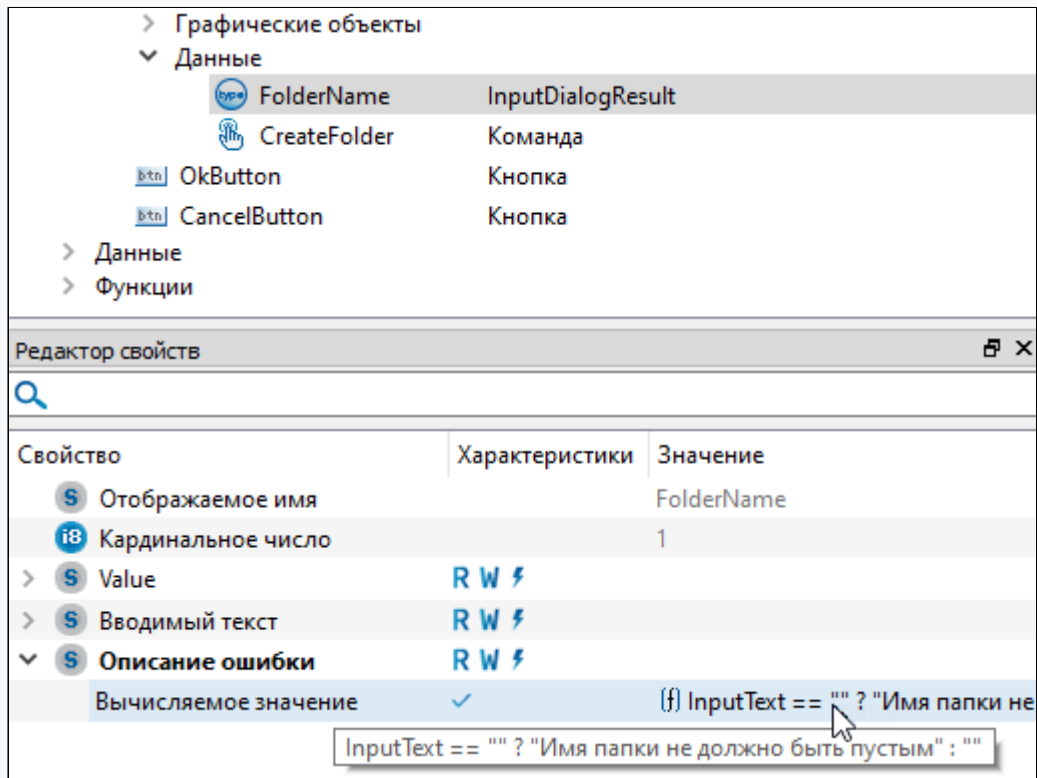


Допустим, значение в поле ввода не должно быть пустым.

Проверка условия может быть описана на языке Astra.От компактной тернарной операцией вида:

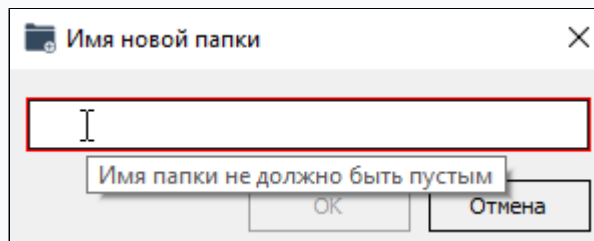


```
<condition>?<statement>:<statement2>
```



В данном случае проверяется вводимый текст (InputText). При пустом значении ввода:

- вокруг поля ввода появится красная рамка;
- во всплывающей подсказке появится текст ошибки;
- кнопка ОК заблокируется.



## 1.1.4.4.3.3.19.1.6. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.19.1.7. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.20. DateTimeResult

Компонент, в котором сохраняется дата, выбранная в календаре Calendar.

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.3.20.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Значение</a>	Хранит дату, выбранную в календаре (Calendar)
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.20.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.4.3.3.20.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.20.1.3. Значение

Хранит дату, выбранную в календаре (Calendar).



timestamp Value

## 1.1.4.4.3.3.20.1.4. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.20.1.5. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.21. DialogResult

Компонент, в котором сохраняется путь к папке, выбранной в окне [SelectFolderDialog](#).

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.3.21.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Значение</a>	Хранит введенные в окне редактора дату и время, округленное до секунд
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.21.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.21.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.4.3.3.21.1.3. Значение

Хранит путь к папке, выбранной в окне [SelectFolderDialog](#).



string Value

## 1.1.4.4.3.3.21.1.4. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.21.1.5. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



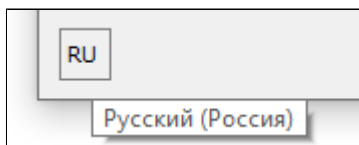
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.22. KeyboardLayoutButton

Кнопка, показывающая текущую раскладку клавиатуры.



При нажатии на кнопку раскладка меняется.



Кнопка применяется в компонентах [LoginDialog](#) и [ChangePasswordDialog](#).

## 1.1.4.4.3.3.22.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">ButtonPressed</a>	Нажатие на кнопку

## 1.1.4.4.3.3.22.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.22.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.22.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.22.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```



## 1.1.4.4.3.3.22.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.22.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.22.1.13. ButtonPressed

Нажатие на кнопку.

### Параметры

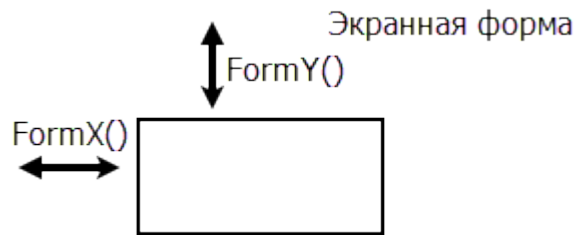
Параметр	Тип	Описание
checked	bool	Указывает была ли нажата (true) или отпущена (false) кнопка

## 1.1.4.4.3.3.22.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Click</a>	Имитировать нажатие кнопки

## 1.1.4.4.3.3.22.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



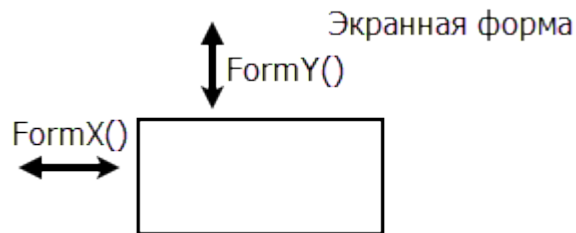
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.22.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



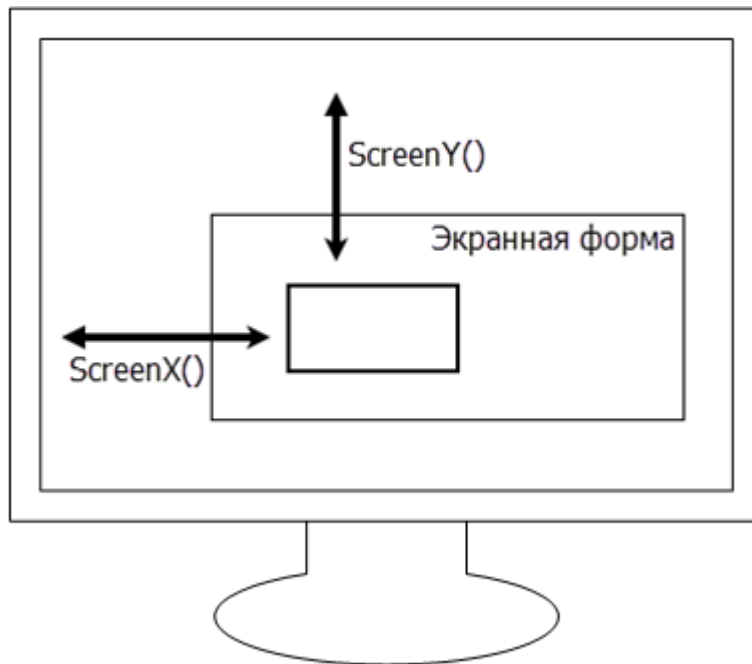
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.22.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

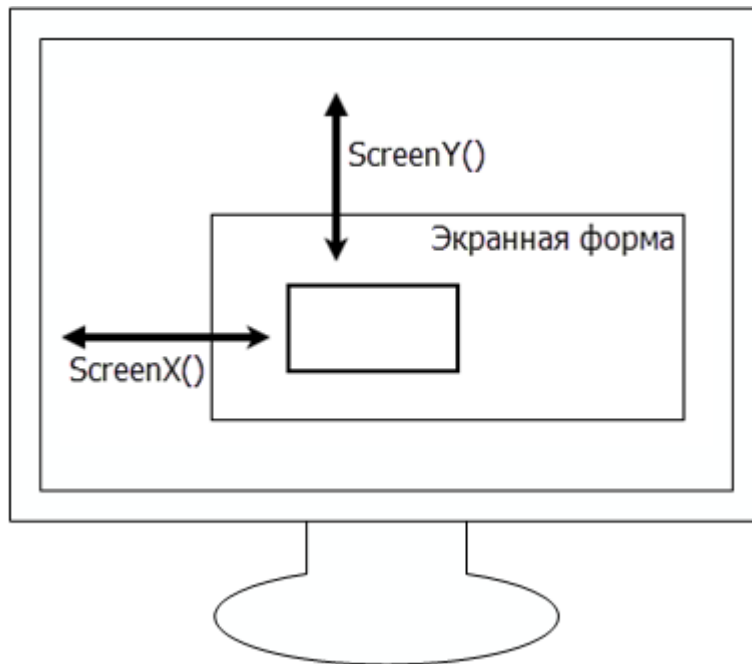
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.22.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.22.2.5. Click

Имитировать нажатие кнопки.



void Click()

## 1.1.4.4.3.22.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Удерживаемая</a>	Оставляет кнопку удерживаемой/отпускаемой после нажатия
<a href="#">Нажата</a>	Нажата ли кнопка
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Выравнивание текста</a>	Выравнивание текста
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Цвет шрифта</a>	Цвет шрифта
<a href="#">Цвет шрифта при нажатии</a>	Цвет шрифта при нажатии

<a href="#">Цвет шрифта при наведении мыши</a>	Цвет шрифта при наведении мыши
<a href="#">Цвет шрифта при выключении</a>	Цвет шрифта при выключении
<a href="#">Цвет заливки при нажатии</a>	Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту
<a href="#">Стиль заливки при нажатии</a>	Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту
<a href="#">Цвет заливки при наведении мыши</a>	Цвет внутренней заливки объекта при вводе курсора в область объекта
<a href="#">Стиль заливки при наведении мыши</a>	Заливает объект одним из стилей заливки при вводе курсора в область объекта
<a href="#">Цвет заливки при выключении</a>	Цвет заливки при выключении
<a href="#">Стиль заливки при выключении</a>	Стиль заливки при выключении
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет пера при нажатии</a>	Цвет внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Стиль пера при нажатии</a>	Стиль внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Толщина пера при нажатии</a>	Толщина внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Цвет пера при наведении мыши</a>	Цвет внешней границы объекта при вводе курсора в область объекта
<a href="#">Стиль пера при наведении мыши</a>	Стиль внешней границы объекта при вводе курсора в область объекта

<a href="#">Толщина пера при наведении мыши</a>	Толщина пера при наведении мыши
<a href="#">Цвет пера при выключении</a>	Цвет пера при выключении
<a href="#">Стиль пера при выключении</a>	Стиль пера при выключении
<a href="#">Толщина пера при выключении</a>	Толщина пера при выключении
<a href="#">Тема оформления</a>	Тема оформления визуального компонента
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.22.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.22.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

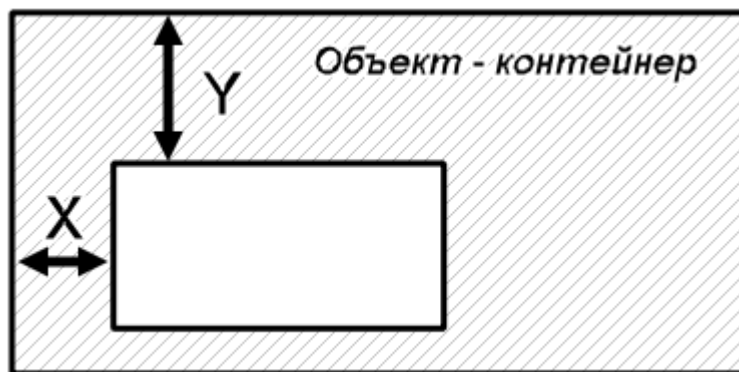
## 1.1.4.4.3.3.22.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.22.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.22.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.22.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.22.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.22.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.22.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.22.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.22.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.22.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




//Задать всплывающую подсказку для объекта:

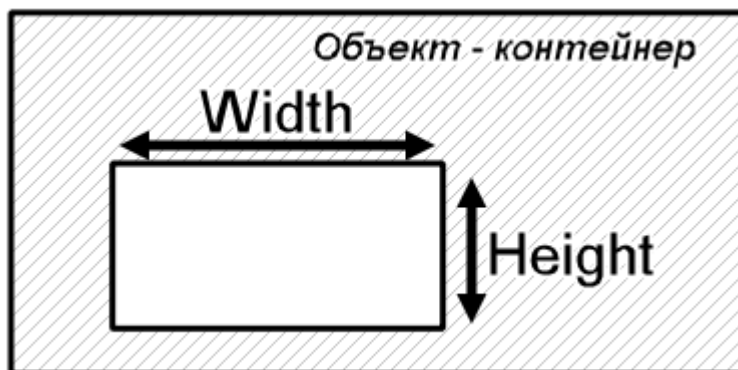
```
Object.Tooltip = "Подача команды MNA_start = true";
```



# 1.1.4.4.3.3.22.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

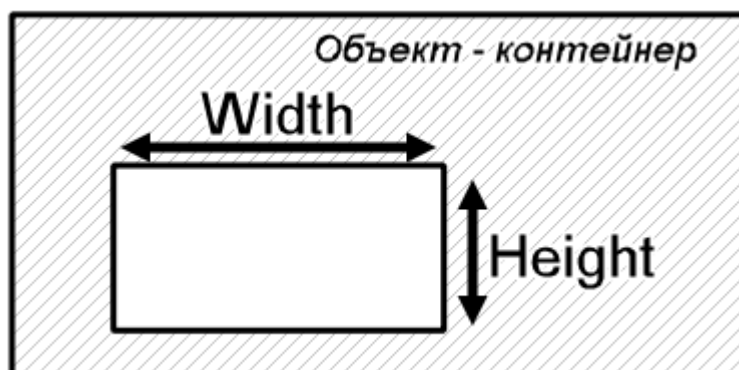


## 1.1.4.4.3.3.22.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.22.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.22.3.16. Удерживаемая

Оставляет кнопку удерживаемой/отпускаемой после нажатия.



bool Checkable

### Значение

Значение	Описание
true	Удерживать кнопку после нажатия
false	Отпускать кнопку после нажатия

## 1.1.4.4.3.3.22.3.17. Нажата

Нажата ли кнопка.



bool Checked

## 1.1.4.4.3.3.22.3.18. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.4.3.3.22.3.19. Выравнивание текста



int4 TextAlignment

### Значение

Значение	Описание
33	Слева сверху
65	Слева снизу
129	Слева по центру
34	Справа сверху
66	Справа снизу
130	Справа по центру
36	По центру сверху
68	По центру снизу
132	По центру объекта
40	По всей ширине сверху
72	По всей ширине снизу
136	По всей ширине по центру

## 1.1.4.4.3.3.22.3.20. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку.



string Font



## 1.1.4.4.3.3.22.3.21. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

## 1.1.4.4.3.3.22.3.22. Цвет шрифта при нажатии

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickFontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.OnClickFontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.OnClickFontColor = 4294901760;
```

## 1.1.4.4.3.3.22.3.23. Цвет шрифта при наведении МЫШИ

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverFontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.OnHoverFontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.OnHoverFontColor = 4294901760;
```

# 1.1.4.4.3.3.22.3.24. ВЫКЛЮЧЕНИЯ

## Цвет шрифта при

Цвет шрифта при выключении.



uint4 DisabledFontColor

## 1.1.4.4.3.3.22.3.25. Цвет заливки при нажатии

Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickBrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.OnClickBrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.OnClickBrushColor = 4294901760;
```

## 1.1.4.4.3.3.22.3.26. Стиль заливки при нажатии

Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту.



uint2 OnClickBrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.OnClickBrushStyle = 1;

## 1.1.4.4.3.3.22.3.27. Цвет заливки при наведении МЫШИ

Цвет внутренней заливки объекта при вводе курсора в область объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverBrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.OnHoverBrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.OnHoverBrushColor = 4294901760;
```






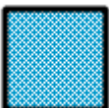

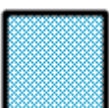
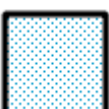
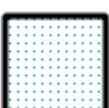
## 1.1.4.4.3.3.22.3.28. Стиль заливки при наведении мыши

Заливает объект одним из стилей заливки при вводе курсора в область объекта.



uint2 OnHoverBrushStyle

### Значение

Значение в дизайнере	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.OnHoverBrushStyle = 1;

# 1.1.4.4.3.3.22.3.29. Цвет заливки при выключении

Цвет заливки при выключении.



uint4 DisabledBrushColor

# 1.1.4.4.3.3.22.3.30.      Стиль      заливки      при выключении

Стиль заливки при выключении.



uint2 DisabledBrushStyle

## 1.1.4.4.3.3.22.3.31. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.22.3.32. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.22.3.33. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.22.3.34. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





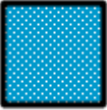
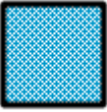

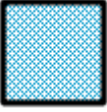
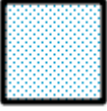
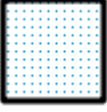
## 1.1.4.4.3.3.22.3.35. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.22.3.36. Цвет пера при нажатии

Цвет внешней границы объекта при щелчке левой кнопкой по объекту. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickPenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:

```
Object.OnClickPenColor = 0xff000000;
```



//Задать черный цвет пера в десятичной системе счисления:

```
Object.OnClickPenColor = 4278190080;
```


## 1.1.4.4.3.3.22.3.37. Стиль пера при нажатии

Стиль внешней границы объекта при щелчке левой кнопкой по объекту.



uint2 OnClickPenStyle

### Значение

Значение	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.OnClickPenStyle = 3;
```

## 1.1.4.4.3.3.22.3.38. Толщина пера при нажатии

Толщина внешней границы объекта при щелчке левой кнопкой по объекту. Измеряется в пикселях.



float8 OnClickPenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.OnClickPenWidth = 5;
```

## 1.1.4.4.3.3.22.3.39. Цвет пера при наведении МЫШИ

Цвет внешней границы объекта при вводе курсора в область объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverPenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.OnHoverPenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.OnHoverPenColor = 4278190080;
```






## 1.1.4.4.3.3.22.3.40. Стиль пера при наведении МЫШИ

Стиль внешней границы объекта при вводе курсора в область объекта.



uint2 OnHoverPenStyle

### Значение

Значение	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.OnHoverPenStyle = 3;
```

# 1.1.4.4.3.3.22.3.41. Толщина пера при наведении мыши

Толщина пера при наведении мыши.



float8 OnHoverPenWidth



## 1.1.4.4.3.3.22.3.42. Цвет пера при выключении

Цвет пера при выключении.



uint4 DisabledPenColor

## 1.1.4.4.3.3.22.3.43. Стил ь пера при выключении

Стил ь пера при выключении.



uint2 DisabledPenStyle

# 1.1.4.4.3.3.22.3.44. Толщина пера при выключении

Толщина пера при выключении.



float8 DisabledPenWidth

# 1.1.4.4.3.3.22.3.45. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.22.3.46. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.22.3.47. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



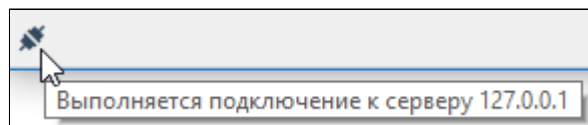
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.23. ConnectionIndicator

Индикатор состояния подключения к серверу.



Во всплывающей подсказке выводится информация о состоянии подключения.

## 1.1.4.4.3.3.23.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта



## 1.1.4.4.3.3.23.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры

v

## 1.1.4.4.3.3.23.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.23.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.23.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.23.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.23.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

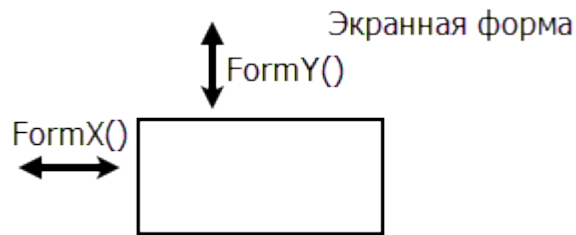
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.23.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.3.23.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



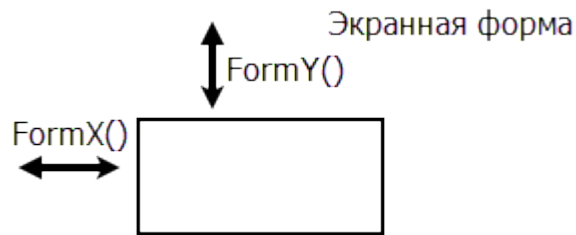
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.23.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

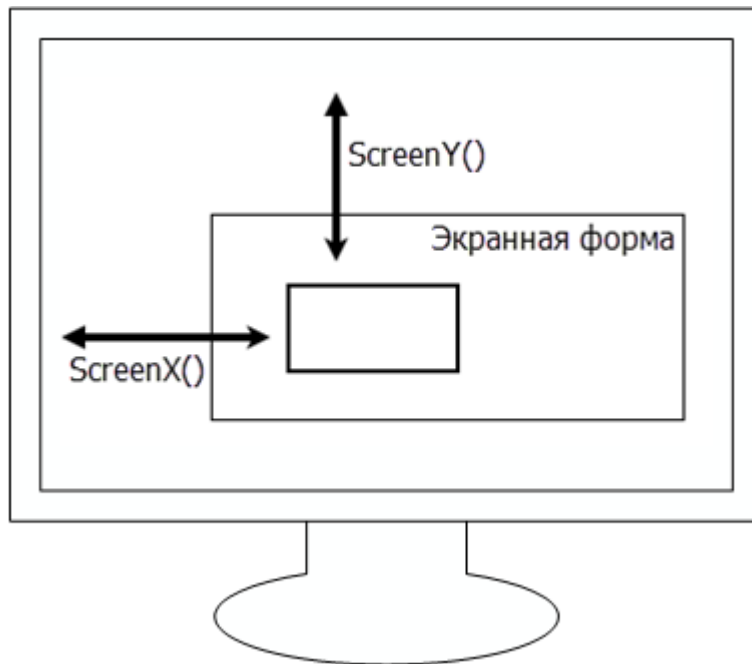
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.23.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

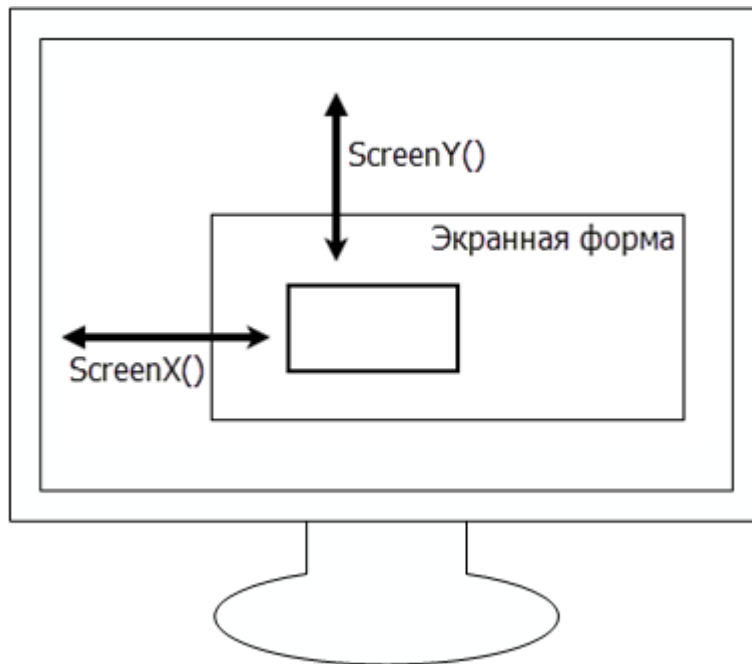
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.23.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.23.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Ссылка на источник</a>	Ссылка на сервер, о состоянии подключения с которым должен сообщать <a href="#">ConnectionIndicator</a>
<a href="#">Описание состояния</a>	Информация о подключении в виде json-строки

<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.23.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.23.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.23.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

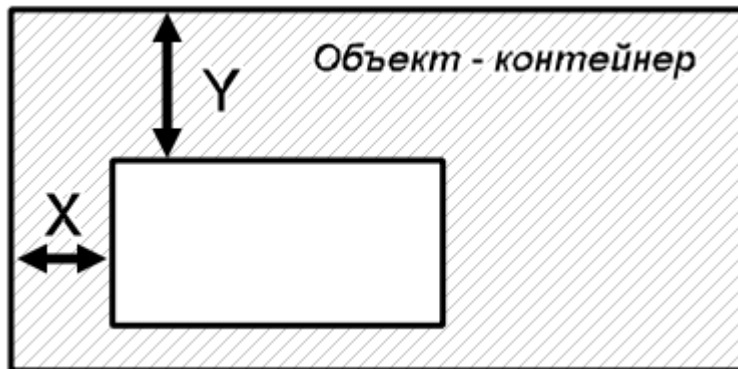
## 1.1.4.4.3.3.23.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.23.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.23.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.23.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.23.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.23.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.23.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.23.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.23.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




//Задать всплывающую подсказку для объекта:

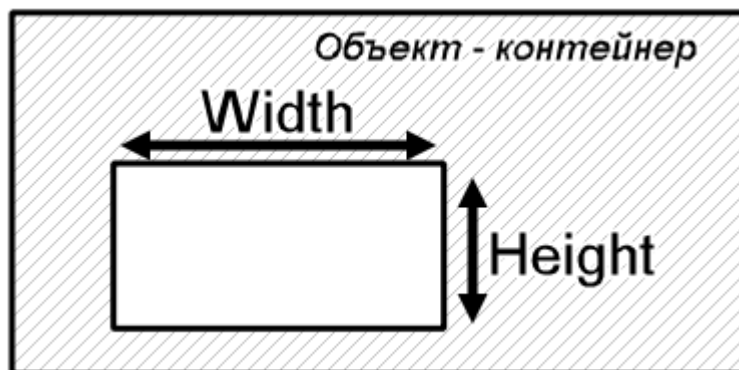
```
Object.Tooltip = "Подача команды MNA_start = true";
```



# 1.1.4.4.3.3.23.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

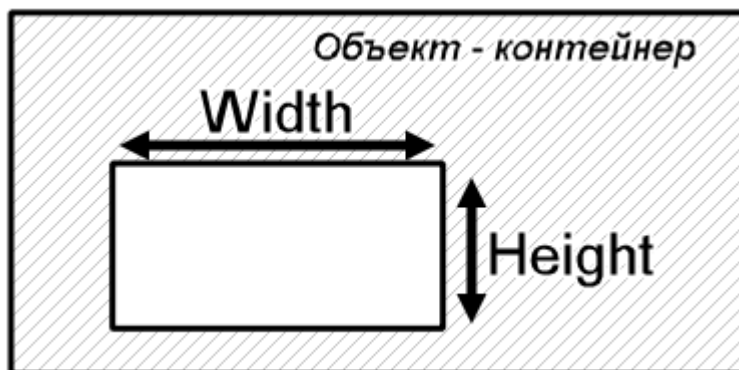


## 1.1.4.4.3.3.23.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.23.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

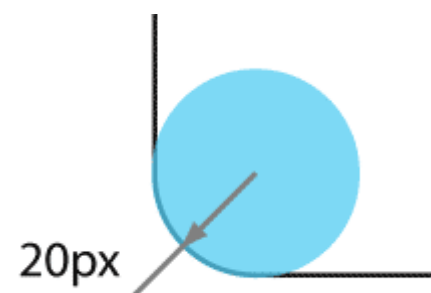
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.23.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.23.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.4.3.3.23.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.23.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.23.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;








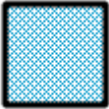

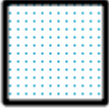
## 1.1.4.4.3.3.23.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.23.3.22. Ссылка на источник

Здесь указывается сервер, о состоянии подключения с которым должен сообщать [ConnectionIndicator](#).

## 1.1.4.4.3.3.23.3.23. Описание состояния

Содержит информацию о подключении в виде json-строки.



string StateInfo

### Примеры



```
{  
  "importance": "info",  
  "description": "Установлено соединение с сервером 127.0.0.1"  
}
```



```
{  
  "importance": "error",  
  "description": "Прервано соединение с сервером 127.0.0.1"  
}
```

## 1.1.4.4.3.3.23.3.24. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.23.3.25. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



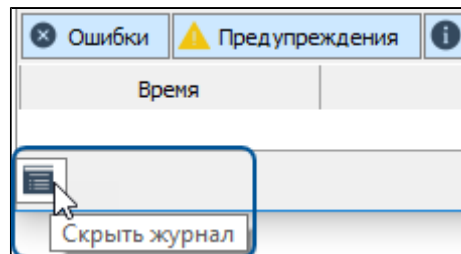
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



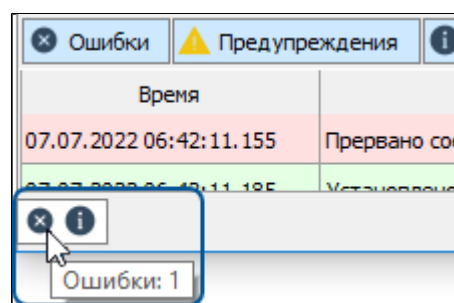
```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.24. LogViewButton

Кнопка-индикатор наличия сообщений в журнале. Позволяет менять видимость журнала. Если журнал сообщений пуст, в рантайме кнопка выглядит следующим образом:



Если в журнале есть сообщения, на кнопке появляются иконки, соответствующие типам сообщений в журнале. При наведении на иконку всплывает подсказка с количеством сообщений этого типа в журнале:



## 1.1.4.4.3.3.24.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта



## 1.1.4.4.3.3.24.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.24.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.24.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.24.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.24.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

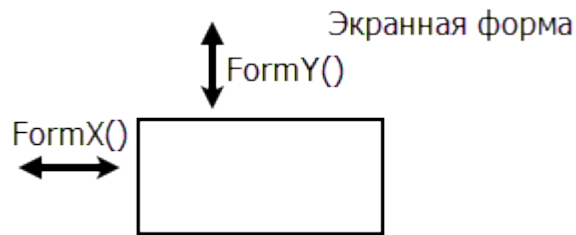
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.24.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.3.24.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



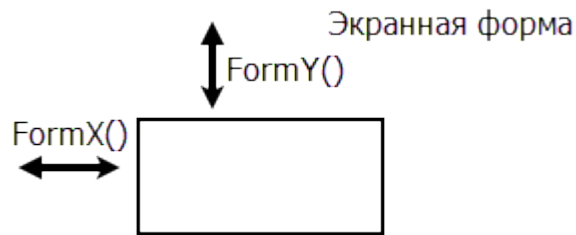
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.24.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

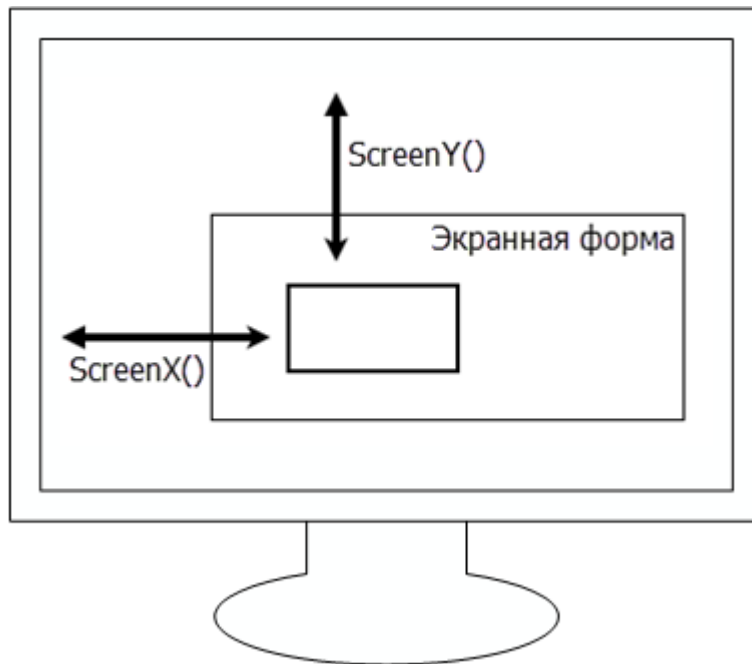
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.24.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

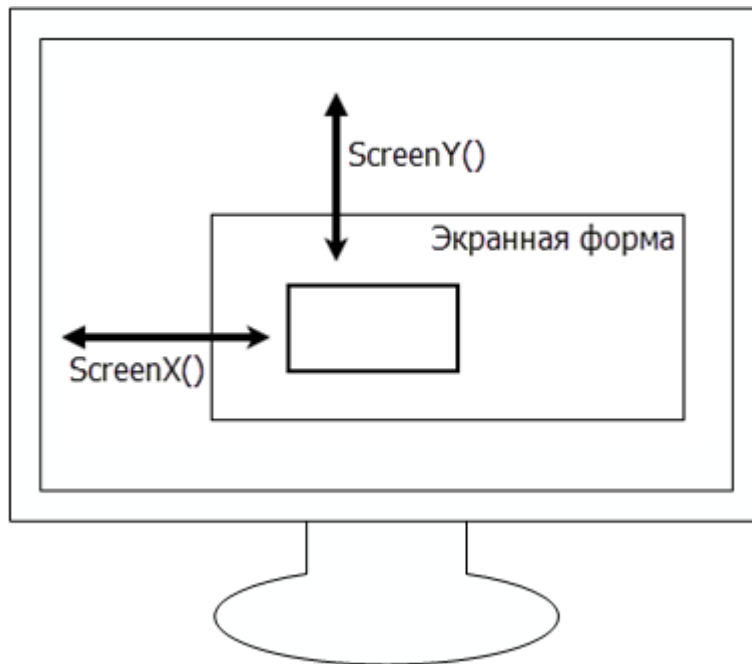
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.24.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.24.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов

<a href="#">pLogView</a>	Ссылка на определенный экземпляр журнала сообщений
<a href="#">ButtonHeight</a>	Высота кнопки
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.24.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.24.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

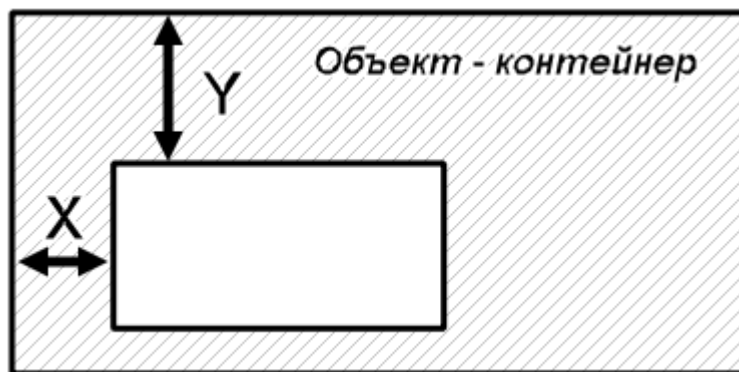
## 1.1.4.4.3.3.24.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.24.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.24.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.24.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.24.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.24.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.24.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.24.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.24.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.24.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




//Задать всплывающую подсказку для объекта:

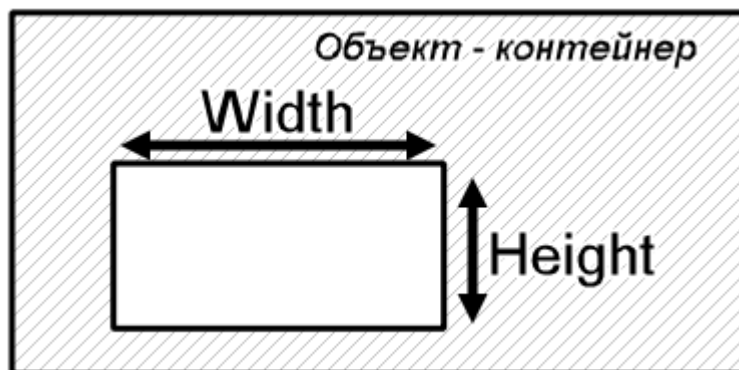
```
Object.Tooltip = "Подача команды MNA_start = true";
```



# 1.1.4.4.3.3.24.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

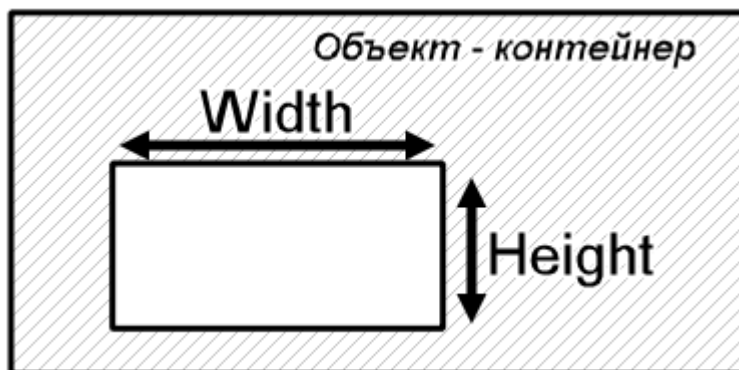


## 1.1.4.4.3.3.24.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.24.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

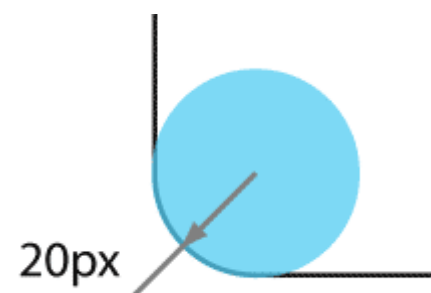
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.24.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.24.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.4.3.3.24.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.24.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.24.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





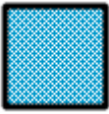

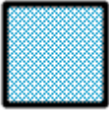

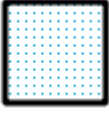
## 1.1.4.4.3.3.24.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.24.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.24.3.23. pLogView

Ссылка на определенный экземпляр журнала сообщений.

## 1.1.4.4.3.3.24.3.24. ButtonHeight

Высота кнопки.



uint2 ButtonHeight

## 1.1.4.4.3.3.24.3.25. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.24.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```





```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.25. LoginButton

Кнопка вызова окна авторизации пользователя LoginDialog.



Внешний вид кнопки зависит от текущего состояния входа:

-  – пользователь авторизован в системе;
-  – гостевой режим.

Во всплывающей подсказке показывается отображаемое имя текущего пользователя.



## 1.1.4.4.3.25.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта
<a href="#">ButtonPressed</a>	Нажатие на кнопку

## 1.1.4.4.3.3.25.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.25.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.25.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.25.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.25.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.25.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.25.1.13. ButtonPressed

Нажатие на кнопку.

### Параметры

Параметр	Тип	Описание
checked	bool	Указывает была ли нажата (true) или отпущена (false) кнопка

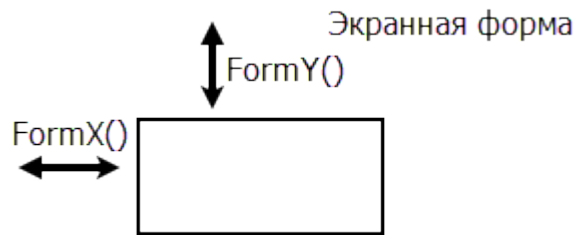
## 1.1.4.4.3.3.25.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Click</a>	Имитировать нажатие кнопки
<a href="#">UpdateLayout</a>	Обновить макет



## 1.1.4.4.3.3.25.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



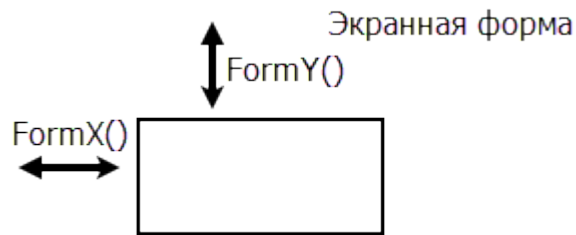
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.25.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



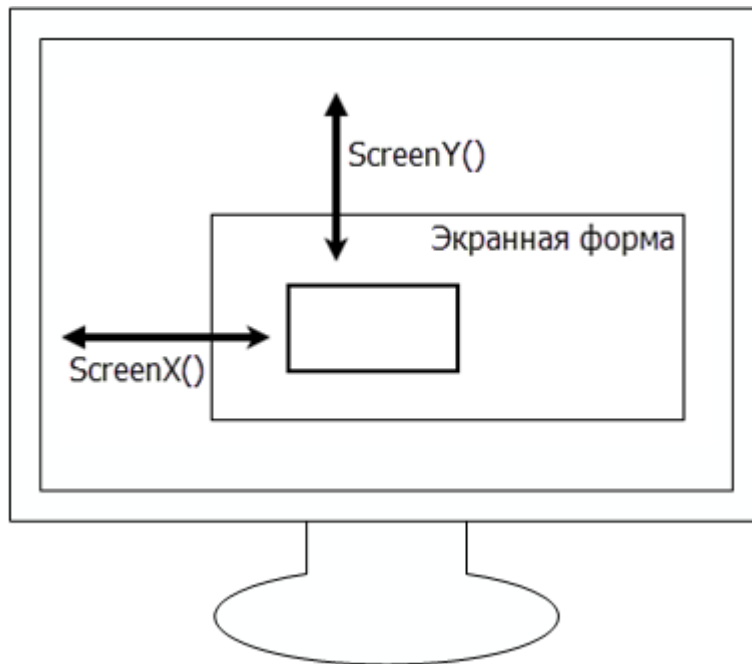
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.25.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

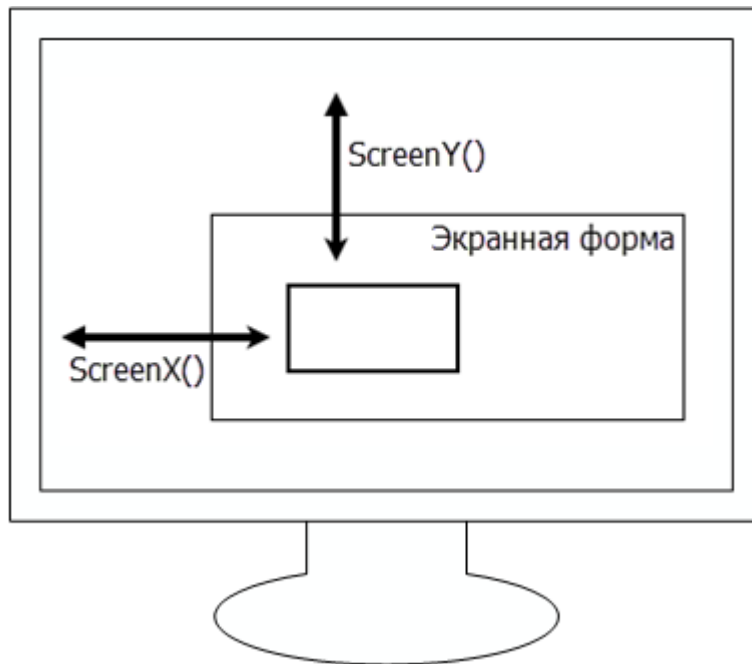
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.25.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.25.2.5. Click

Имитировать нажатие кнопки.



void Click()

## 1.1.4.4.3.3.25.2.6. UpdateLayout

Обновить макет.



void UpdateLayout()

### 1.1.4.4.3.25.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Удерживаемая</a>	Оставляет кнопку удерживаемой/отпускаемой после нажатия
<a href="#">Нажата</a>	Нажата ли кнопка
<a href="#">Текст</a>	Содержимое текстовой области объекта
<a href="#">Выравнивание текста</a>	Выравнивание текста
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Цвет шрифта</a>	Цвет шрифта
<a href="#">Цвет шрифта при нажатии</a>	Цвет шрифта при нажатии

<a href="#">Цвет шрифта при наведении мыши</a>	Цвет шрифта при наведении мыши
<a href="#">Цвет шрифта при выключении</a>	Цвет шрифта при выключении
<a href="#">Цвет заливки при нажатии</a>	Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту
<a href="#">Стиль заливки при нажатии</a>	Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту
<a href="#">Цвет заливки при наведении мыши</a>	Цвет внутренней заливки объекта при вводе курсора в область объекта
<a href="#">Стиль заливки при наведении мыши</a>	Заливает объект одним из стилей заливки при вводе курсора в область объекта
<a href="#">Цвет заливки при выключении</a>	Цвет заливки при выключении
<a href="#">Стиль заливки при выключении</a>	Стиль заливки при выключении
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Цвет пера при нажатии</a>	Цвет внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Стиль пера при нажатии</a>	Стиль внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Толщина пера при нажатии</a>	Толщина внешней границы объекта при щелчке левой кнопкой по объекту
<a href="#">Цвет пера при наведении мыши</a>	Цвет внешней границы объекта при вводе курсора в область объекта
<a href="#">Стиль пера при наведении мыши</a>	Стиль внешней границы объекта при вводе курсора в область объекта



<a href="#">Толщина пера при наведении мыши</a>	Толщина пера при наведении мыши
<a href="#">Цвет пера при выключении</a>	Цвет пера при выключении
<a href="#">Стиль пера при выключении</a>	Стиль пера при выключении
<a href="#">Толщина пера при выключении</a>	Толщина пера при выключении
<a href="#">Тема оформления</a>	Тема оформления визуального компонента
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.25.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.25.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

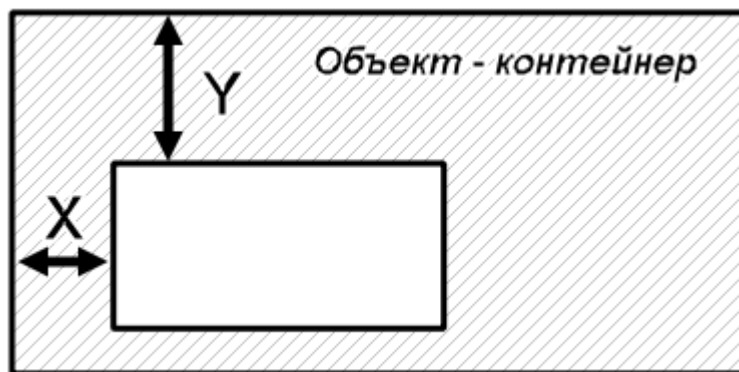
## 1.1.4.4.3.3.25.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.3.25.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.25.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.25.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.25.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```



## 1.1.4.4.3.3.25.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.25.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.25.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.25.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.25.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




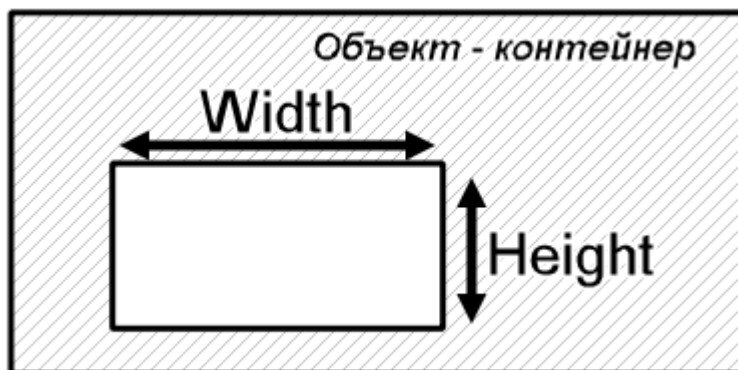
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.25.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

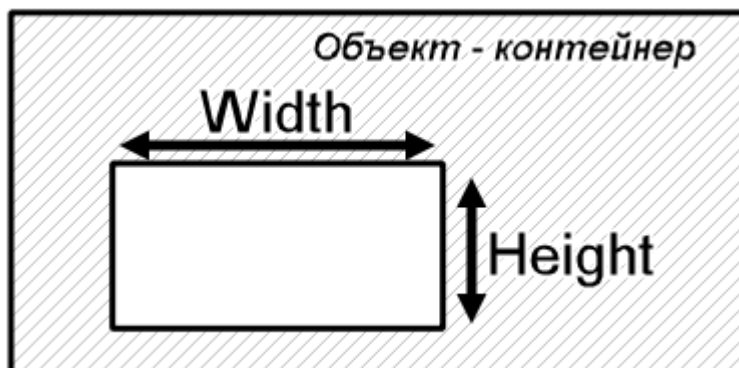


## 1.1.4.4.3.3.25.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.25.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```



## 1.1.4.4.3.3.25.3.16. Удерживаемая

Оставляет кнопку удерживаемой/отпускаемой после нажатия.



bool Checkable

### Значение

Значение	Описание
true	Удерживать кнопку после нажатия
false	Отпускать кнопку после нажатия

## 1.1.4.4.3.3.25.3.17. Нажата

Нажата ли кнопка.



bool Checked

## 1.1.4.4.3.3.25.3.18. Текст

Содержимое текстовой области объекта.



string Text

## 1.1.4.4.3.3.25.3.19. Выравнивание текста



int4 TextAlignment

### Значение

Значение	Описание
33	Слева сверху
65	Слева снизу
129	Слева по центру
34	Справа сверху
66	Справа снизу
130	Справа по центру
36	По центру сверху
68	По центру снизу
132	По центру объекта
40	По всей ширине сверху
72	По всей ширине снизу
136	По всей ширине по центру

## 1.1.4.4.3.3.25.3.20. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку.



string Font

## 1.1.4.4.3.3.25.3.21. Цвет шрифта

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 FontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.FontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.FontColor = 4294901760;
```

## 1.1.4.4.3.3.25.3.22. Цвет шрифта при нажатии

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickFontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.OnClickFontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.OnClickFontColor = 4294901760;
```

## 1.1.4.4.3.3.25.3.23. Цвет шрифта при наведении МЫШИ

Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverFontColor

### Примеры



//Задать красный цвет шрифта в шестнадцатеричной системе счисления:

```
Object.OnHoverFontColor = 0xffff0000;
```



//Задать красный цвет шрифта в десятичной системе счисления:

```
Object.OnHoverFontColor = 4294901760;
```



# 1.1.4.4.3.3.25.3.24. ВЫКЛЮЧЕНИЯ

## Цвет шрифта при

Цвет шрифта при выключении.



uint4 DisabledFontColor

## 1.1.4.4.3.3.25.3.25. Цвет заливки при нажатии

Цвет внутренней заливки объекта при щелчке левой кнопкой по объекту. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickBrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.OnClickBrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.OnClickBrushColor = 4294901760;
```

## 1.1.4.4.3.3.25.3.26. Стиль заливки при нажатии

Заливает объект одним из стилей заливки при щелчке левой кнопкой по объекту.



uint2 OnClickBrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.OnClickBrushStyle = 1;

## 1.1.4.4.3.3.25.3.27. Цвет заливки при наведении МЫШИ

Цвет внутренней заливки объекта при вводе курсора в область объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverBrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):

```
Object.OnHoverBrushColor = 0xffff0000;
```



//Залить объект красным цветом (цвет в десятичной системе счисления):

```
Object.OnHoverBrushColor = 4294901760;
```


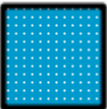
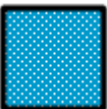
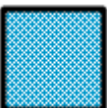

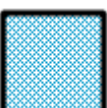
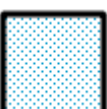
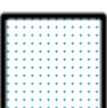
## 1.1.4.4.3.3.25.3.28. Стиль заливки при наведении мыши

Заливает объект одним из стилей заливки при вводе курсора в область объекта.



uint2 OnHoverBrushStyle

### Значение

Значение в дизайнере	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.OnHoverBrushStyle = 1;
```

# 1.1.4.4.3.3.25.3.29. Цвет заливки при выключении

Цвет заливки при выключении.



uint4 DisabledBrushColor



# 1.1.4.4.3.3.25.3.30.      Стиль      заливки      при выключении

Стиль заливки при выключении.



uint2 DisabledBrushStyle

## 1.1.4.4.3.3.25.3.31. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.25.3.32. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.25.3.33. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.25.3.34. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.4.3.3.25.3.35. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

## 1.1.4.4.3.3.25.3.36. Цвет пера при нажатии

Цвет внешней границы объекта при щелчке левой кнопкой по объекту. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnClickPenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:

```
Object.OnClickPenColor = 0xff000000;
```



//Задать черный цвет пера в десятичной системе счисления:

```
Object.OnClickPenColor = 4278190080;
```





## 1.1.4.4.3.3.25.3.37. Стиль пера при нажатии

Стиль внешней границы объекта при щелчке левой кнопкой по объекту.



uint2 OnClickPenStyle

### Значение

Значение	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.OnClickPenStyle = 3;
```

## 1.1.4.4.3.3.25.3.38. Толщина пера при нажатии

Толщина внешней границы объекта при щелчке левой кнопкой по объекту. Измеряется в пикселях.



float8 OnClickPenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.OnClickPenWidth = 5;
```

## 1.1.4.4.3.3.25.3.39. Цвет пера при наведении МЫШИ

Цвет внешней границы объекта при вводе курсора в область объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 OnHoverPenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.OnHoverPenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.OnHoverPenColor = 4278190080;
```






## 1.1.4.4.3.3.25.3.40. Стиль пера при наведении МЫШИ

Стиль внешней границы объекта при вводе курсора в область объекта.



uint2 OnHoverPenStyle

### Значение

Значение	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



```
//Применить стиль границы (короткий пунктир) для объекта:  
Object.OnHoverPenStyle = 3;
```

# 1.1.4.4.3.3.25.3.41. Толщина пера при наведении мыши

Толщина пера при наведении мыши.



float8 OnHoverPenWidth

## 1.1.4.4.3.3.25.3.42. Цвет пера при выключении

Цвет пера при выключении.



uint4 DisabledPenColor

## 1.1.4.4.3.3.25.3.43. Стил ь пера при выключении

Стил ь пера при выключении.



uint2 DisabledPenStyle

# 1.1.4.4.3.3.25.3.44. Толщина пера при выключении

Толщина пера при выключении.



float8 DisabledPenWidth



# 1.1.4.4.3.3.25.3.45. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.25.3.46. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.25.3.47. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.26. MenuResult

Компонент, в котором сохраняется результат выбора пункта в окне [Menu](#).

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.3.26.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Идентификатор команды</a>	Идентификатор выбранного (выделенного) пункта меню
<a href="#">Меню остается открытым</a>	Указывает, остается ли открытым меню при выборе (выделении) этого пункта меню
<a href="#">Обновить элементы меню</a>	Позволяет обновлять элементы меню после его открытия
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.26.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.26.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.26.1.3. Идентификатор команды

Идентификатор выбранного (выделенного) пункта меню.



string Id



## 1.1.4.4.3.3.26.1.4. Меню остается открытым

Указывает, остается ли открытым меню при выборе (выделении) этого пункта меню.



bool StaysOpen

## 1.1.4.4.3.3.26.1.5. Обновить элементы меню

Позволяет обновлять элементы меню после его открытия. Значение данного свойства представляет собой json-строку.



string UpdateItems

Подробнее структура json-строки описана в свойстве Items компонента Menu. Строка-значение данного свойства должна описывать измененное состояние пунктов меню.

Используется, например, если выбор одного пункта меню должен приводить к изменению других пунктов меню: их заголовков, иконок, доступности.

### Примеры



```
MenuResult.UpdateItems = JSON.stringify([
  {
    'id': "addSignal",
    'title': "Добавить новый сигнал...",
    'active': true
  }
  {
    'id': "saveSignal",
    'title': "Сохранить сигнал...",
    'active': true
  }
]);
```

## 1.1.4.4.3.3.26.1.6. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.26.1.7. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.27. MenuSeparator

Разделитель элементов. Используется в Menu и как отдельный компонент.



После изменения размеров компонента для обновления разметки необходимо вызвать метод [UpdateLayout](#).

## 1.1.4.4.3.27.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.3.27.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.27.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.27.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.4.3.3.27.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.27.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.27.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

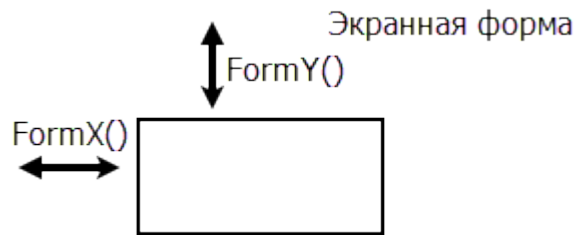
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.27.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">UpdateLayout</a>	Обновить макет

## 1.1.4.4.3.3.27.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



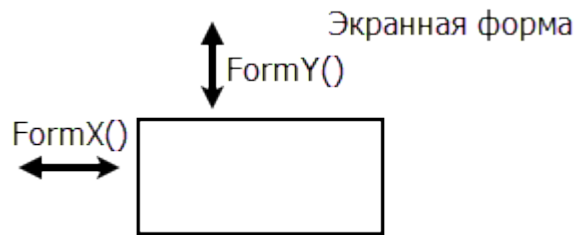
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.27.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



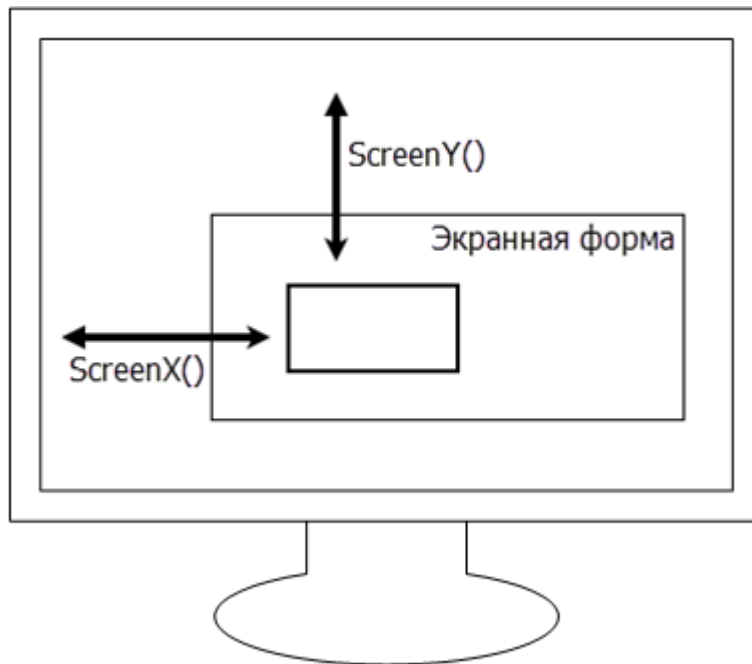
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.27.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

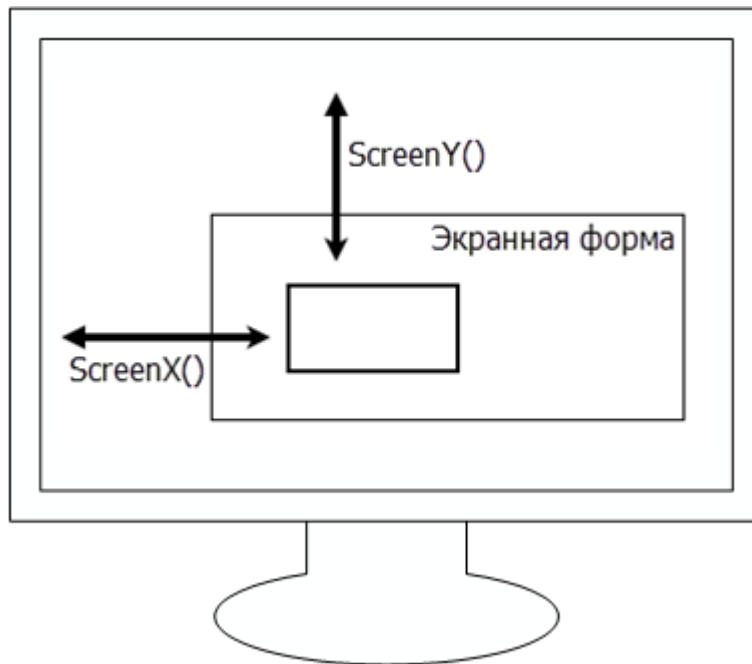
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.27.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



# 1.1.4.4.3.3.27.2.5. UpdateLayout

Обновить макет.



```
void UpdateLayout()
```

### 1.1.4.4.3.27.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">Length</a>	Размер массива (количество элементов в массиве)



## 1.1.4.4.3.3.27.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.27.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

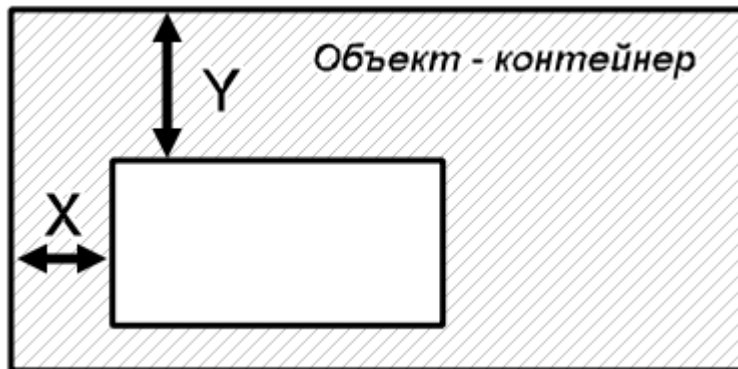
## 1.1.4.4.3.3.27.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

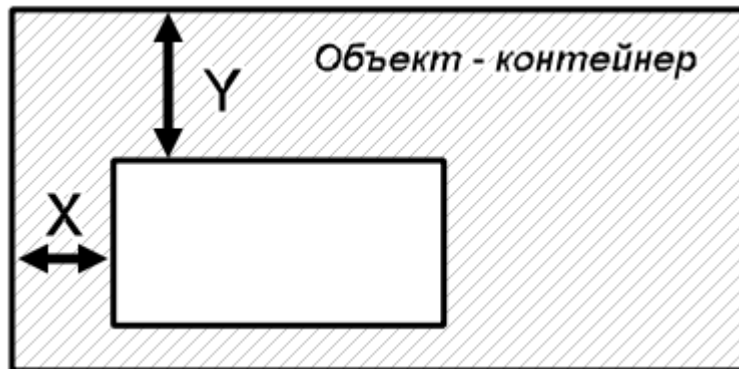
## 1.1.4.4.3.3.27.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.27.3.5. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.4.3.3.27.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

# 1.1.4.4.3.3.27.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

## Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.27.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.27.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.27.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.3.27.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.27.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




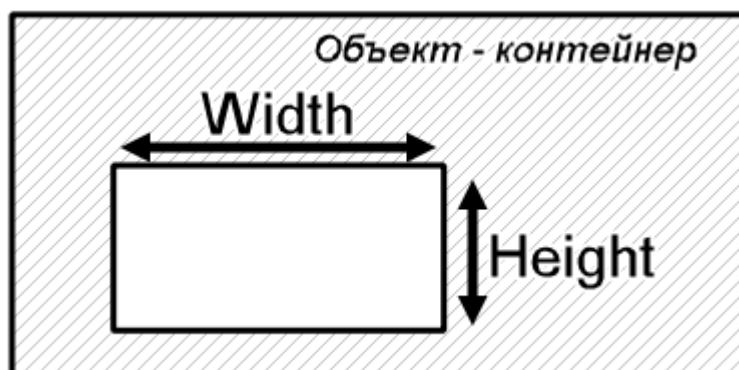
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.27.3.13. Ширина

Ширина объекта в пикселях.


 float8 Width

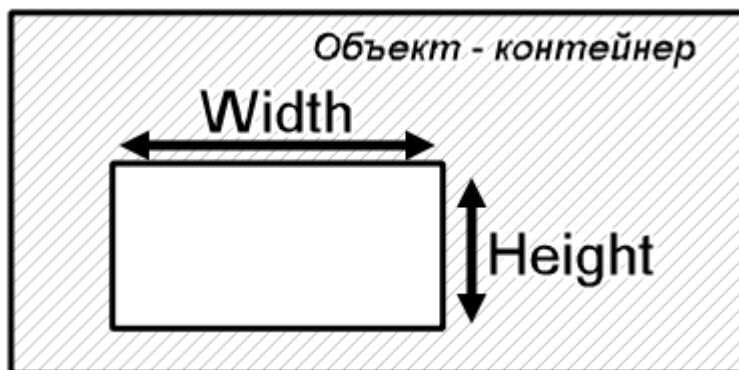




# 1.1.4.4.3.3.27.3.14. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.3.27.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

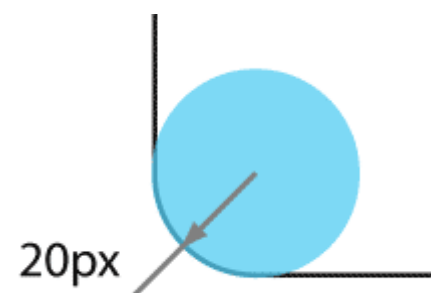
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.27.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.27.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.27.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.27.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.27.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.4.3.3.27.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	



Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

# 1.1.4.4.3.3.27.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.27.3.23. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.27.3.24. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.28. MenuItem

Элемент контекстного меню. Используется в [Menu](#) и как отдельный компонент.



После изменения размеров компонента для обновления разметки необходимо вызвать метод [UpdateLayout](#).

## 1.1.4.4.3.28.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.3.28.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.28.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.28.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.4.3.3.28.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.28.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.28.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

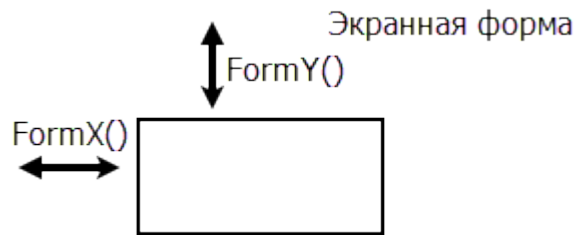
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.28.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">UpdateLayout</a>	Обновить макет

## 1.1.4.4.3.3.28.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



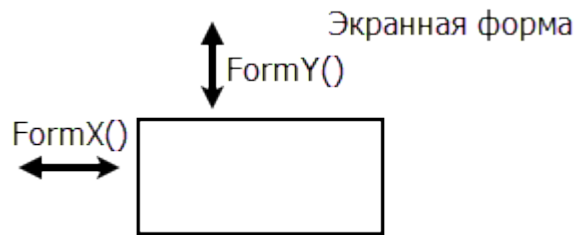
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.28.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



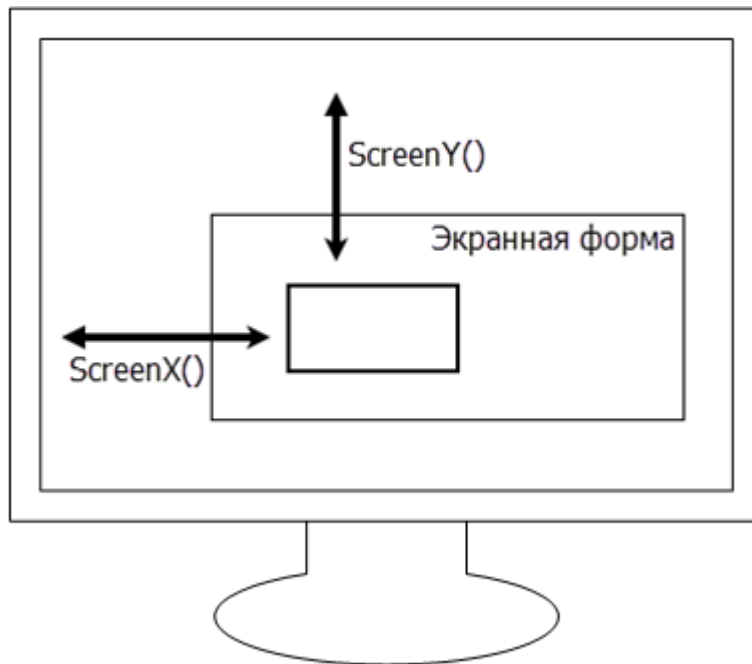
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.28.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

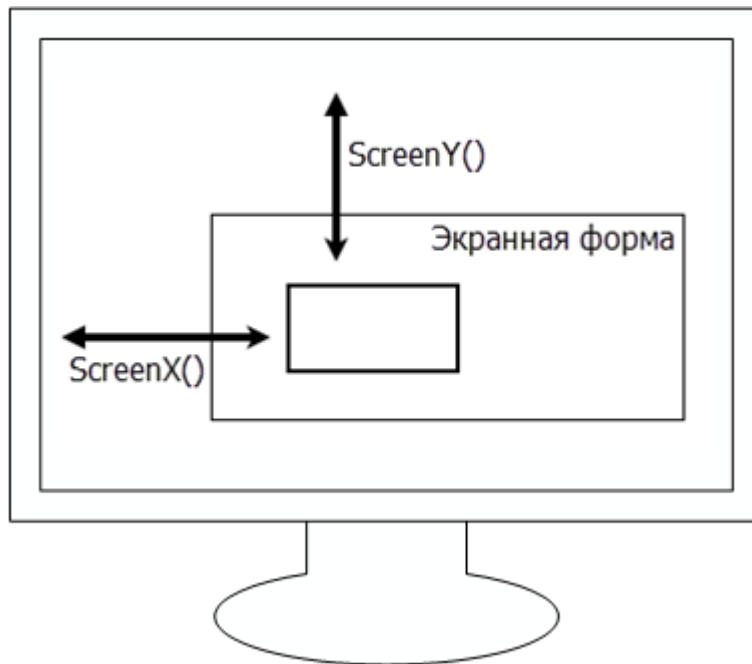
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.28.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.28.2.5. UpdateLayout

Обновить макет.



```
void UpdateLayout()
```

## 1.1.4.4.3.3.28.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки

<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">IconFile</a>	Добавляет элементу меню иконку
<a href="#">Title</a>	Описание команды меню
<a href="#">Checked</a>	Преобразует элемент меню в элемент выбора с иконкой <input checked="" type="checkbox"/>
<a href="#">Selected</a>	Указывает, выбран (выделен) ли элемент меню
<a href="#">Active</a>	Активность элемента меню
<a href="#">Items</a>	Дочерние элементы меню
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.28.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.28.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

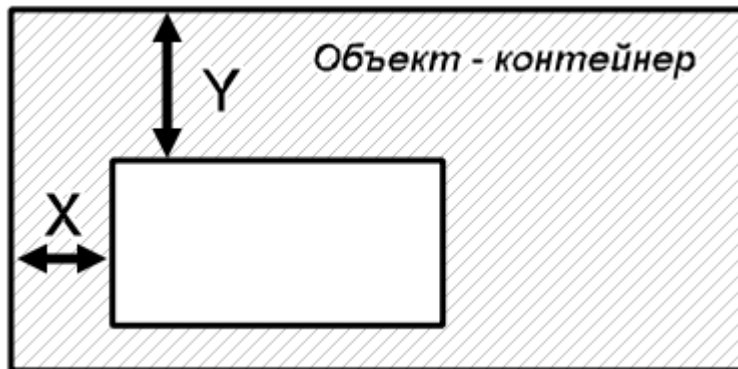
## 1.1.4.4.3.3.28.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.28.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.28.3.5. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.4.3.3.28.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.28.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.28.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.28.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.28.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.28.3.11. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.28.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




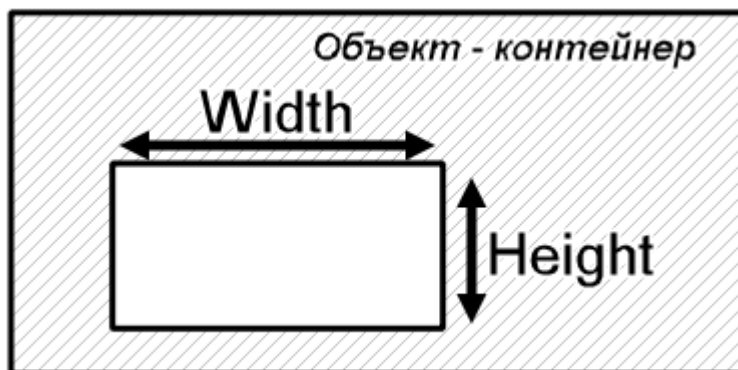
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.28.3.13. Ширина

Ширина объекта в пикселях.


 float8 Width

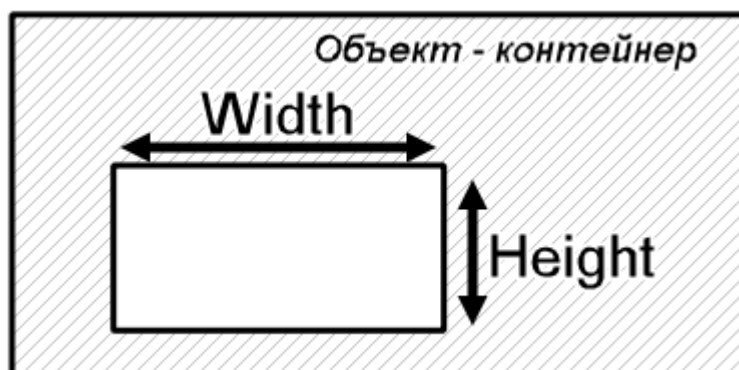




# 1.1.4.4.3.3.28.3.14. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.3.28.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

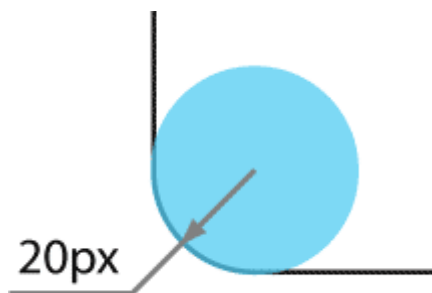
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.28.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.28.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



//Задать черный цвет пера в шестнадцатеричной системе счисления:  
Object.PenColor = 0xff000000;



//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;

## 1.1.4.4.3.3.28.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.28.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.28.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



```
//Залить объект красным цветом (цвет в шестнадцатеричной  
системе счисления):  
Object.BrushColor = 0xffff0000;
```



```
//Залить объект красным цветом (цвет в десятичной системе  
счисления):  
Object.BrushColor = 4294901760;
```

## 1.1.4.4.3.3.28.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	



Горизонтальные линии	9	
----------------------	---	---

## Примеры



```
//Залить объект сплошным цветом:  
Object.BrushStyle = 1;
```

# 1.1.4.4.3.3.28.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.28.3.23. IconFile

Добавляет элементу меню иконку. В качестве значения укажите путь к файлу иконки.



string IconFile

### Примеры



"Files\_Icons/SaveToFile16.png"


# 1.1.4.4.3.3.28.3.24. Title

Описание команды меню.



string Title

## 1.1.4.4.3.3.28.3.25. Checked

Преобразует элемент меню в элемент выбора с иконкой .



bool Checked

### Значение

Значение	Описание
true	Элемент выбора
false	Обычный элемент меню

## 1.1.4.4.3.3.28.3.26. Selected

Указывает, выбран (выделен) ли элемент меню.



bool Selected

### Значение

Значение	Описание
true	Элемент выбран
false	Элемент не выбран

## 1.1.4.4.3.3.28.3.27. Active

Активность элемента меню.



bool Active

### Значение

Значение	Описание
true	Элемент активен
false	Элемент неактивен

# 1.1.4.4.3.3.28.3.28. Items

Дочерние элементы меню.



string Items



## 1.1.4.4.3.3.28.3.29. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.28.3.30. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.29. IntervalEditor

Редактор интервала времени.



Позволяет указать интервал времени с точностью до секунд.

Значения полей редактора можно менять:

- › с помощью ввода цифр;
- › клавишами-стрелками;
- › кнопками-стрелками в окне редактора.



Также можно использовать кнопки смещения к предыдущему и следующему интервалу времени.

## 1.1.4.4.3.29.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.3.29.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.3.29.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.3.29.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.29.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.3.29.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;
```

```
Ellipse_1.Y = 341;
```

```
SW_1.OpenSW = 1;
```



## 1.1.4.4.3.3.29.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

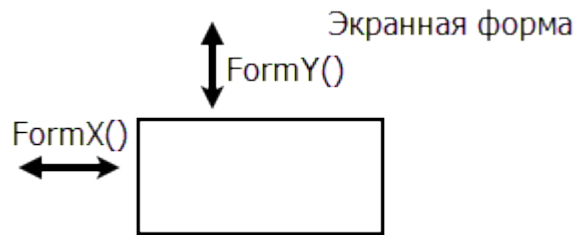
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.29.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.3.29.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



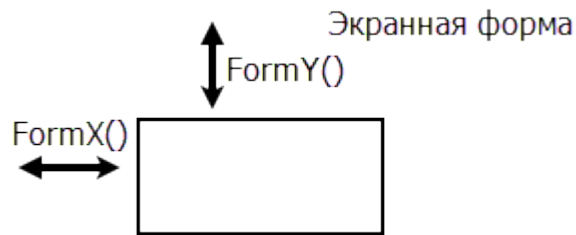
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.29.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



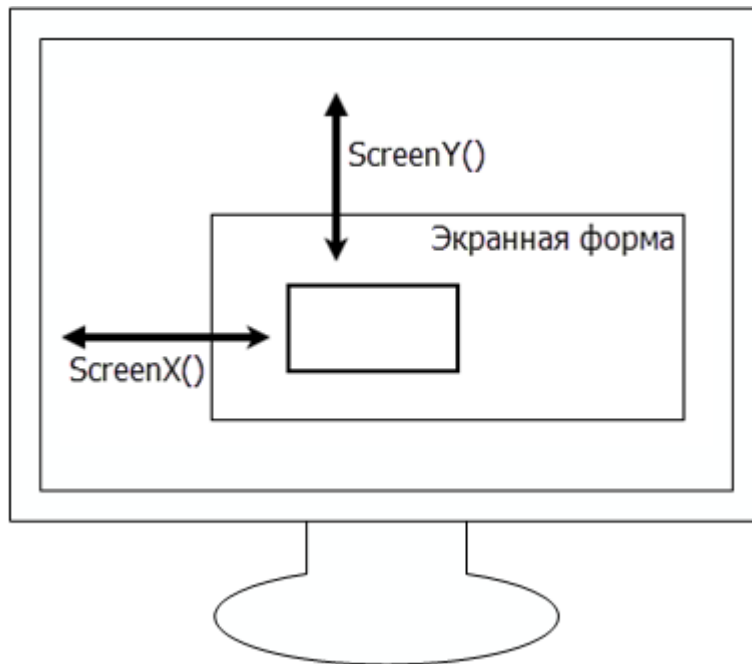
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.3.29.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

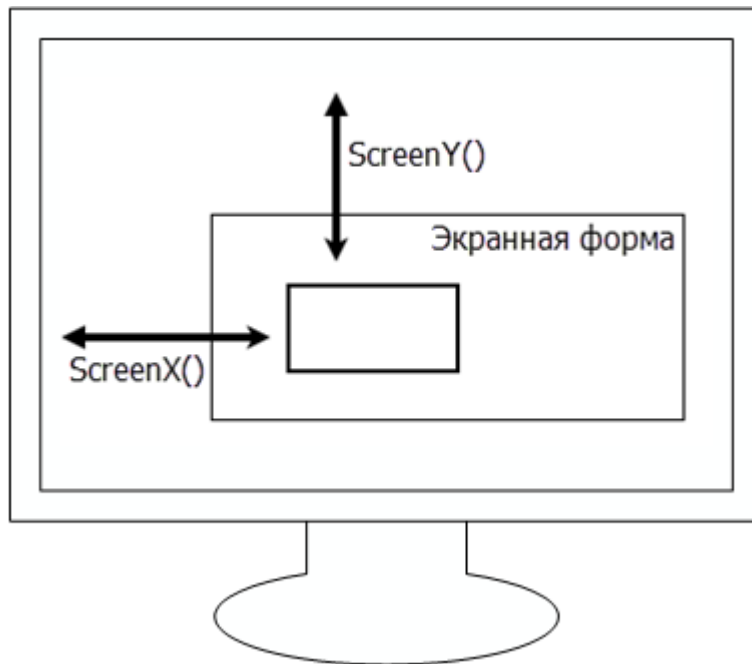
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.3.29.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.29.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки

<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">IsVisible</a>	Меняет видимость редактора, меняя его ширину
<a href="#">Begin</a>	Хранит значение, соответствующее началу интервала, указанному в редакторе
<a href="#">End</a>	Хранит значение, соответствующее концу интервала, указанному в редакторе
<a href="#">pIntervalChanged</a>	Ссылка на команду, выполняемую при изменении даты в окне редактора
<a href="#">Error</a>	Текст ошибки ввода даты
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве



## 1.1.4.4.3.3.29.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.29.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

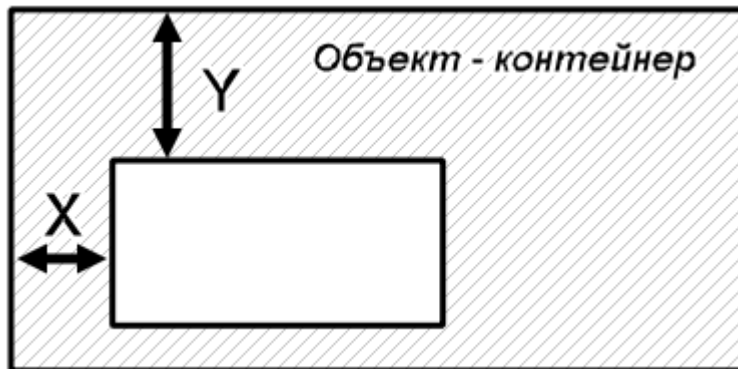
## 1.1.4.4.3.3.29.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.3.29.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.29.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.29.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.3.29.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

# 1.1.4.4.3.3.29.3.8. Отражение

Отражение объекта.



uint1 Flip



## 1.1.4.4.3.3.29.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.29.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.3.29.3.11. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.29.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




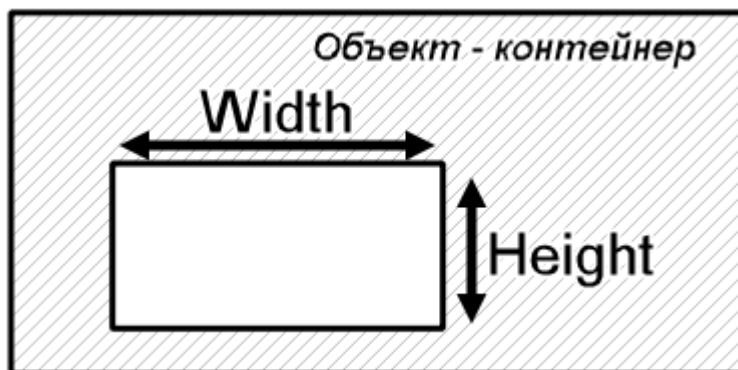
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.29.3.13. Ширина


Ширина объекта в пикселях.

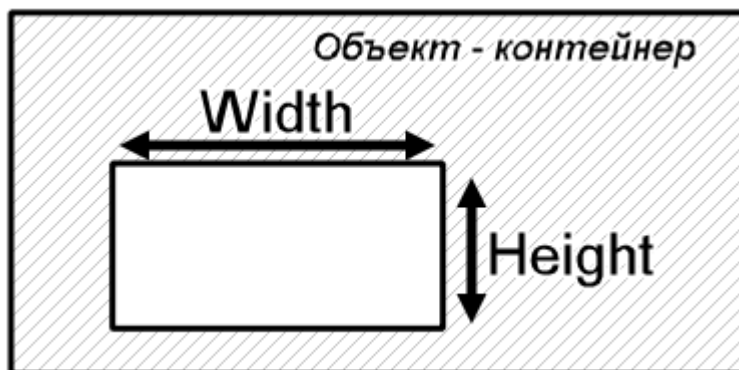
 float8 Width



# 1.1.4.4.3.3.29.3.14. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.3.29.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

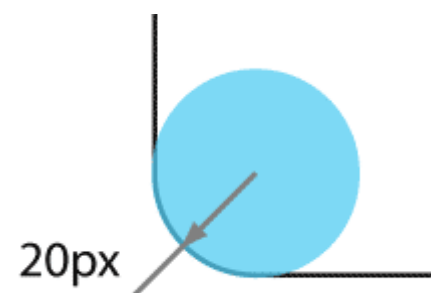
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.29.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius





## 1.1.4.4.3.3.29.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```


## 1.1.4.4.3.3.29.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.29.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.29.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





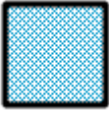
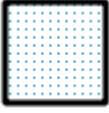
## 1.1.4.4.3.3.29.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

# 1.1.4.4.3.3.29.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.29.3.23. IsVisible

Меняет видимость редактора, меняя его ширину:

- › «true» – редактор виден, ширина редактора обычная;
- › «false» – редактор скрывается, так как его ширина становится нулевой, благодаря чему выравниваются другие элементы.



bool IsVisible



## 1.1.4.4.3.3.29.3.24. Begin

Хранит значение, соответствующее началу интервала, указанному в редакторе.



timestamp Begin

## 1.1.4.4.3.3.29.3.25. End

Хранит значение, соответствующее концу интервала, указанному в редакторе.



timestamp End

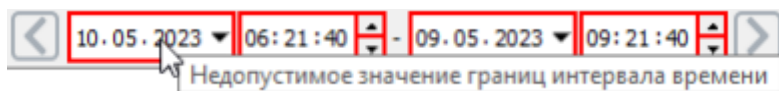
## 1.1.4.4.3.3.29.3.26. Error

Текст ошибки, возникшей при указании границ интервала.



string Error

Если указать в качестве начала интервала значение, большее или равное значению окончания интервала, возникнет ошибка. Текст ошибки записывается в компонент в качестве значения описываемого свойства.



## 1.1.4.4.3.3.29.3.27. pIntervalChanged

Ссылка на команду, выполняемую при изменении границ интервала.

## 1.1.4.4.3.3.29.3.28. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.29.3.29. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



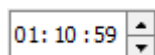
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.3.30. TimeEditor

Окно редактора времени для ввода времени в виде «часы:минуты:секунды».



Значения полей редактора можно менять:

- › с помощью ввода цифр;
- › клавишами-стрелками;
- › кнопками-стрелками в окне редактора.



Можно использовать совместно с редактором даты.

## 1.1.4.4.3.30.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта



## 1.1.4.4.3.30.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.30.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.30.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.30.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.30.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

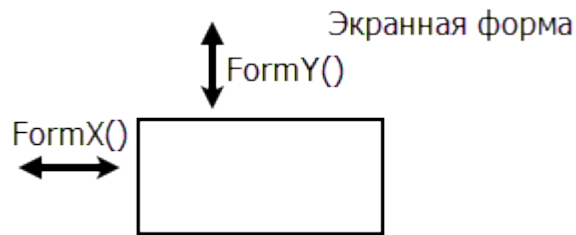
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.30.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.30.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



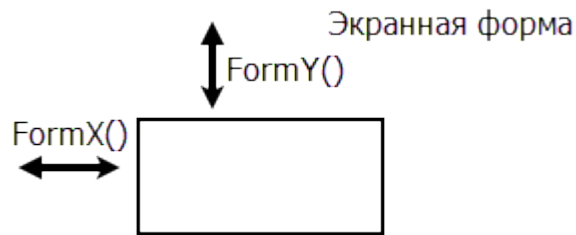
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.30.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

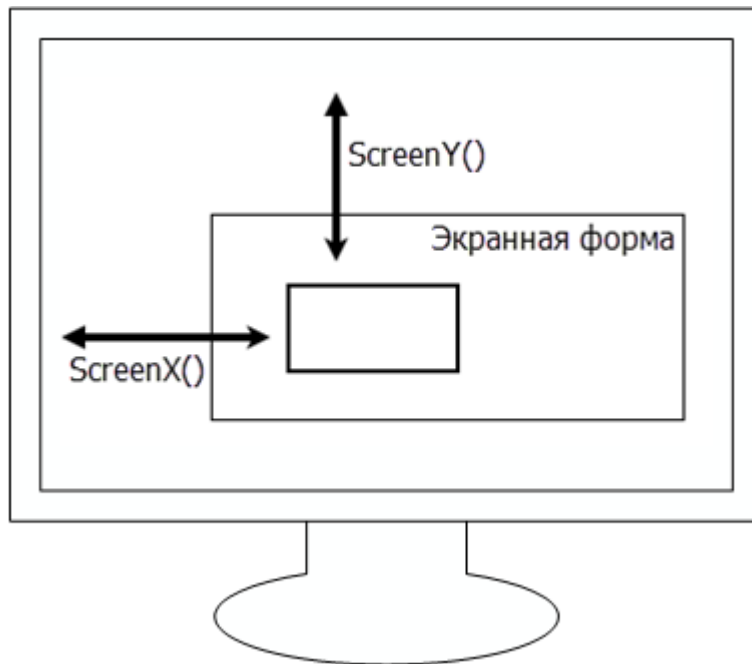
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.30.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

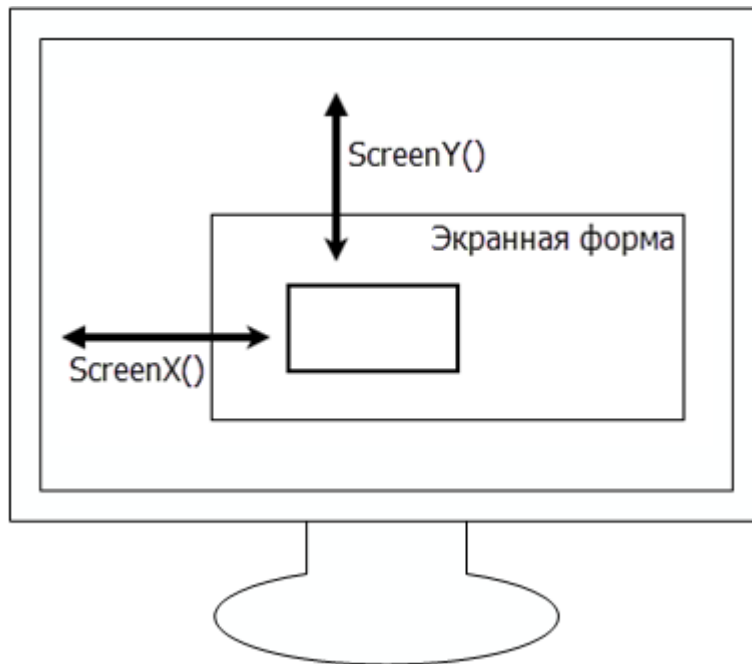
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.30.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

### 1.1.4.4.3.30.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">IsVisible</a>	Меняет видимость редактора, меняя его ширину

<a href="#">Значение</a>	Хранит введенные в окне редактора дату и время, округленное до секунд
<a href="#">pValueChanged</a>	Ссылка на команду, выполняемую при изменении даты в окне редактора
<a href="#">Описание ошибки</a>	Текст ошибки ввода даты
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.30.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.30.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

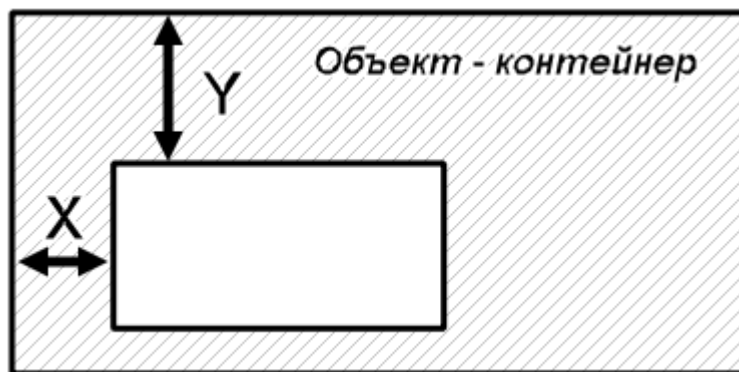
## 1.1.4.4.3.30.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.30.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.30.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.3.30.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.30.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.30.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.30.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

## 1.1.4.4.3.3.30.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.30.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.30.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




//Задать всплывающую подсказку для объекта:

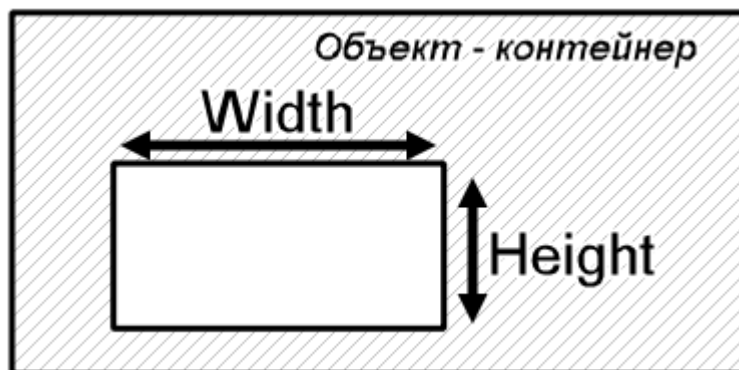
```
Object.Tooltip = "Подача команды MNA_start = true";
```



# 1.1.4.4.3.3.30.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

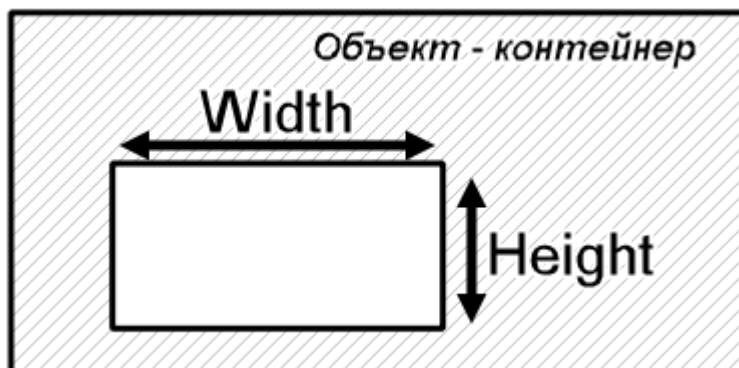


## 1.1.4.4.3.3.30.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.30.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

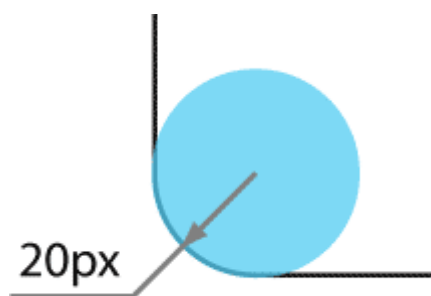
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.30.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.30.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.30.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.30.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.30.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





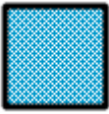

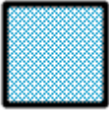

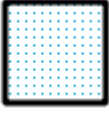
## 1.1.4.4.3.30.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.30.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.30.3.23. IsVisible

Меняет видимость редактора, меняя его ширину:

- › «true» – редактор виден, ширина редактора обычная;
- › «false» – редактор скрывается, так как его ширина становится нулевой, благодаря чему выравниваются другие элементы.



bool IsVisible

## 1.1.4.4.3.3.30.3.24. Значение

Хранит введенные в окне редактора дату и время, округленное до секунд. При редактировании меняет значение даты, а значение времени оставляет без изменений.



timestamp Value

## 1.1.4.4.3.3.30.3.25. pValueChanged

Ссылка на команду, выполняемую при изменении даты в окне редактора.

## 1.1.4.4.3.3.30.3.26. Описание ошибки

Текст ошибки ввода даты.



string Error

## 1.1.4.4.3.30.3.27. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.4.3.3.30.3.28. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



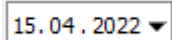
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.31. DateEditor

Окно редактора даты для ввода даты в формате «день:месяц:год».



15.04.2022 ▼

Значения полей редактора можно менять:

- › с помощью ввода цифр;
- › клавишами-стрелками;
- › выбрав дату из выпадающего календаря.

Можно использовать совместно с редактором времени.

## 1.1.4.4.3.31.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.31.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.31.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.31.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.3.31.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.31.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.31.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

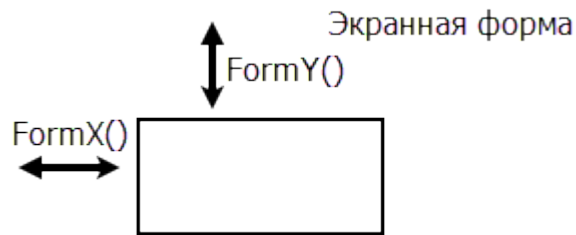


## 1.1.4.4.3.31.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.31.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



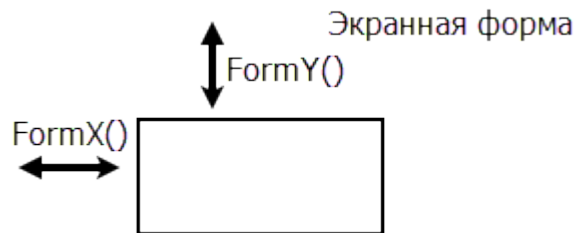
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.31.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



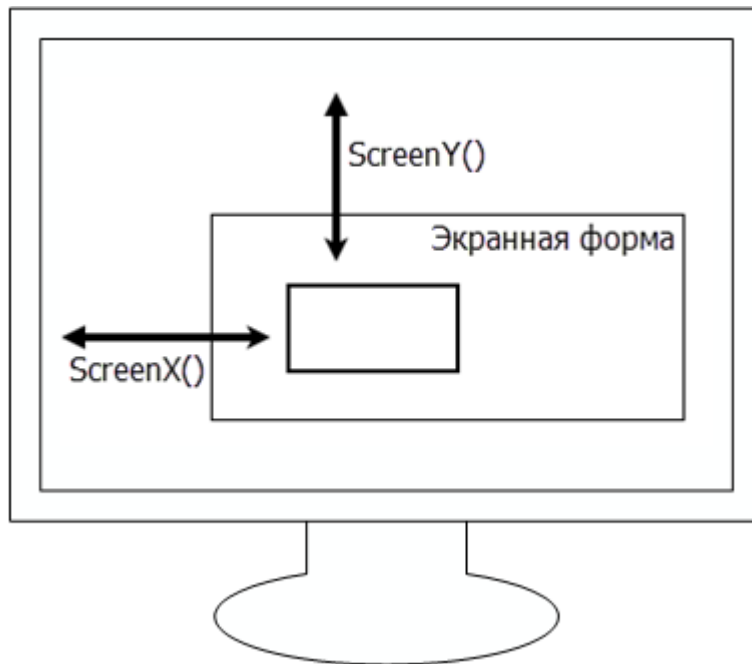
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.31.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

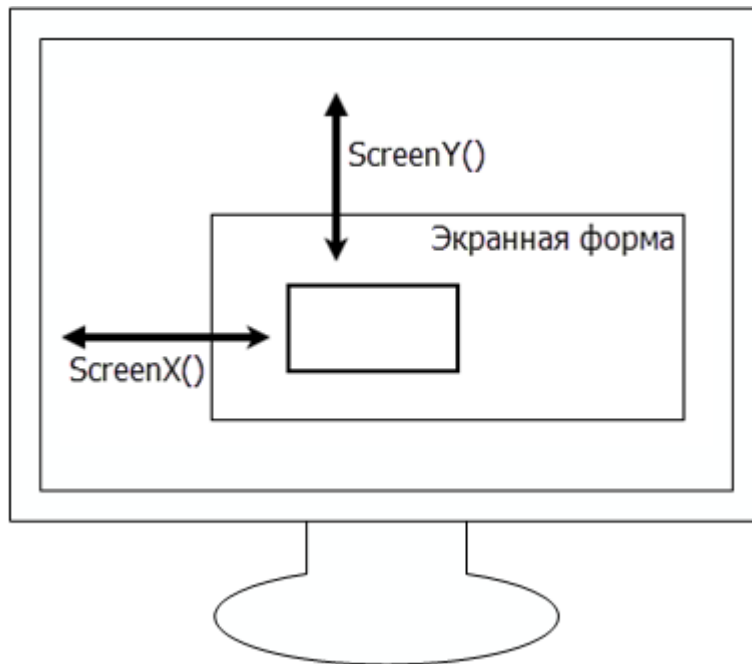
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.31.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.31.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов
<a href="#">IsVisible</a>	Меняет видимость редактора, меняя его ширину

<a href="#">Значение</a>	Хранит введенные в окне редактора дату и время, округленное до секунд
<a href="#">pValueChanged</a>	Ссылка на команду, выполняемую при изменении даты в окне редактора
<a href="#">Описание ошибки</a>	Текст ошибки ввода даты
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.31.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.4.3.3.31.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.31.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

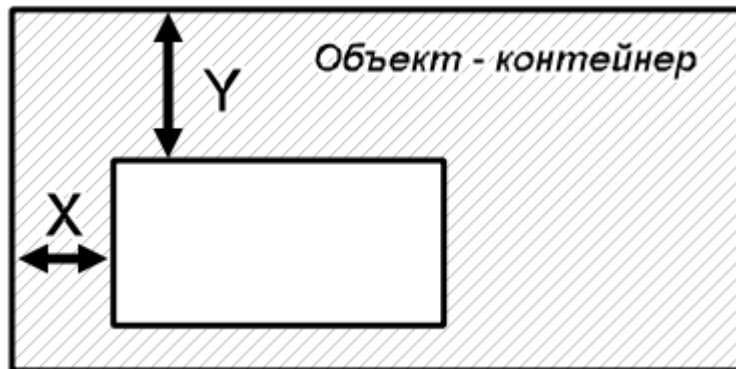
## 1.1.4.4.3.31.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.31.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.31.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.31.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.31.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.31.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт



# 1.1.4.4.3.3.31.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.31.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.3.31.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры



//Задать всплывающую подсказку для объекта:

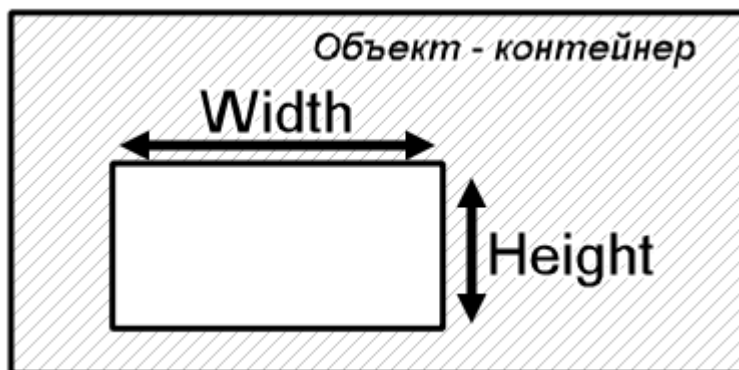
```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.31.3.13. Ширина

Ширина объекта в пикселях.




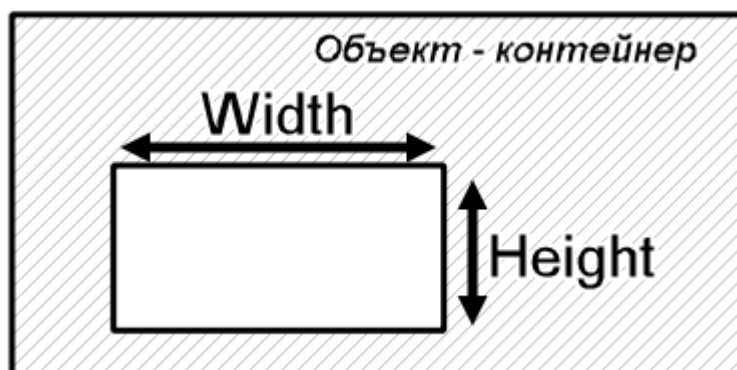
float8 Width



# 1.1.4.4.3.3.31.3.14. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.3.31.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

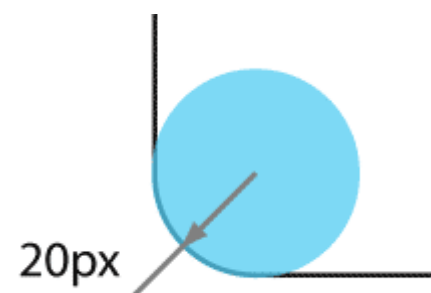
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.31.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.31.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```



## 1.1.4.4.3.31.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.31.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.31.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;






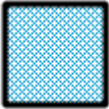

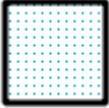
## 1.1.4.4.3.31.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

# 1.1.4.4.3.3.31.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.31.3.23. IsVisible

Меняет видимость редактора, меняя его ширину:

- › «true» – редактор виден, ширина редактора обычная;
- › «false» – редактор скрывается, так как его ширина становится нулевой, благодаря чему выравниваются другие элементы.



bool IsVisible

## 1.1.4.4.3.3.31.3.24. Значение

Хранит введенные в окне редактора дату и время, округленное до секунд. При редактировании меняет значение даты, а значение времени оставляет без изменений.



timestamp Value



## 1.1.4.4.3.3.31.3.25. pValueChanged

Ссылка на команду, выполняемую при изменении даты в окне редактора.

## 1.1.4.4.3.3.31.3.26. Описание ошибки

Текст ошибки ввода даты.



string Error



Свойство не предназначено для проверки введенной даты. Проверку даты можно описать в команде, на которую ссылается свойство [pValueChanged](#).

## 1.1.4.4.3.31.3.27. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.31.3.28. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



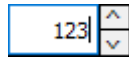
```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.32. IntegerUpDown

Компонент, позволяющий менять значения целочисленных переменных с указанным шагом.



## 1.1.4.4.3.32.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.32.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.32.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.32.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```



## 1.1.4.4.3.3.32.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.32.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;
```

```
Ellipse_1.Y = 341;
```

```
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.3.32.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

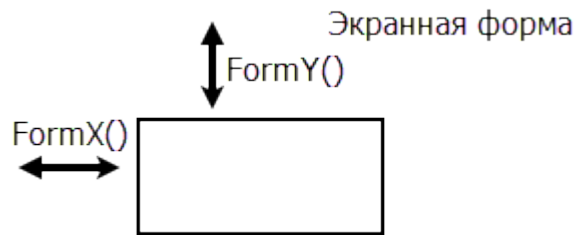
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.32.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y
<a href="#">Increase</a>	Увеличить значение
<a href="#">Decrease</a>	Уменьшить значение
<a href="#">SetValue</a>	Установить значение

## 1.1.4.4.3.32.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



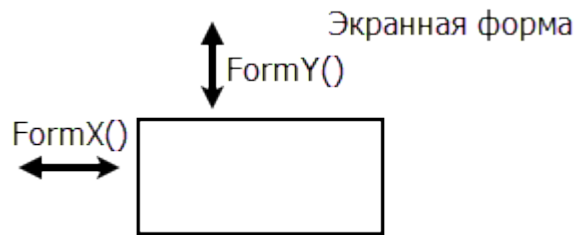
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.32.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



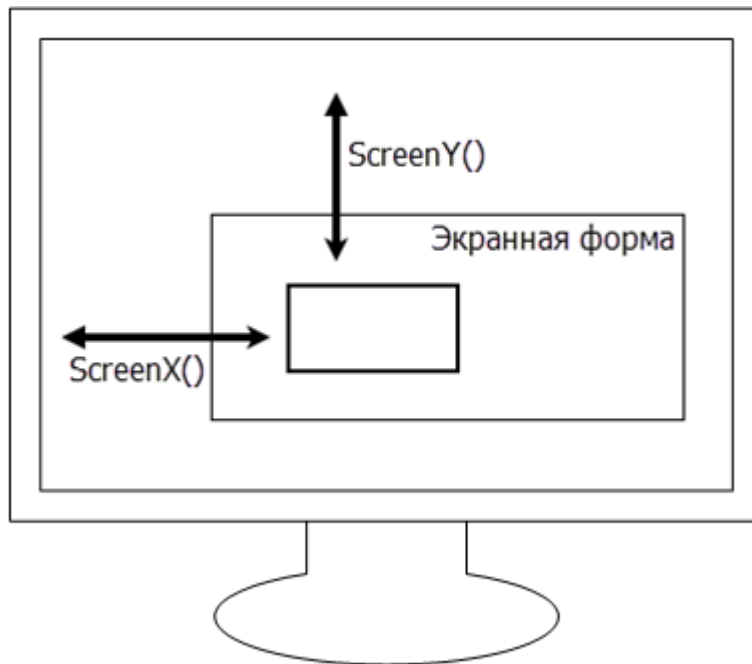
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.32.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

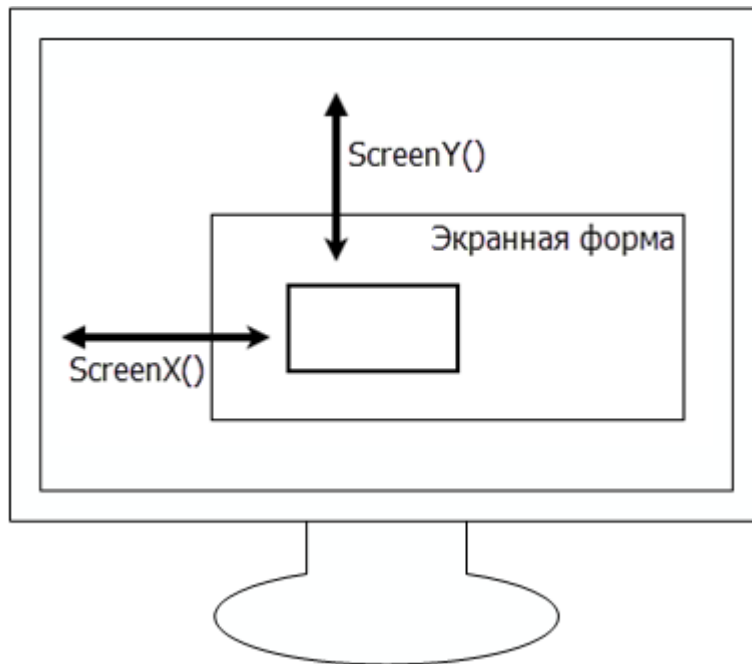
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.32.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```



## 1.1.4.4.3.3.32.2.5. Increase

Увеличить значение.



void Increase()

## 1.1.4.4.3.32.2.6. Decrease

Уменьшить значение.



void Decrease()

## 1.1.4.4.3.32.2.7. SetValue

Установить значение.



```
void SetValue(int4 NewValue)
```

### Параметры

Параметр	Тип	Описание
NewValue	int4	Новое значение

## 1.1.4.4.3.32.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Значение</a>	Изменяемое значение

<a href="#">Максимум</a>	Максимум, до которого можно инкрементировать изменяемое значение
<a href="#">Минимум</a>	Минимум, до которого можно декрементировать изменяемое значение
<a href="#">Шаг изменения</a>	Шаг инкремента / декремента
<a href="#">ValueChanged</a>	Ссылка на команду, выполняемую при изменении значения
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.3.32.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.32.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

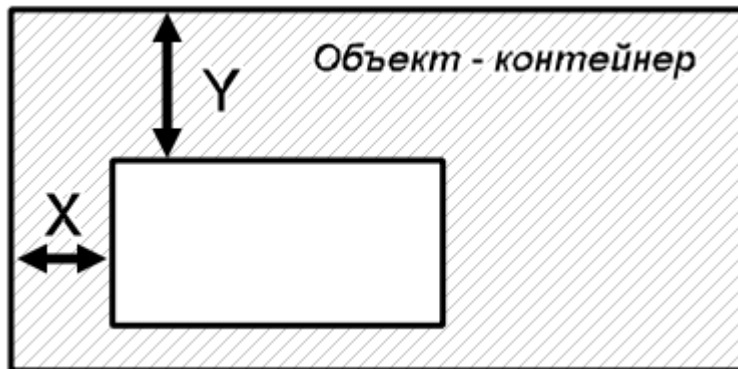
## 1.1.4.4.3.32.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```



## 1.1.4.4.3.32.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.32.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.32.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.32.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.32.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.32.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.32.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.32.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.4.3.32.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




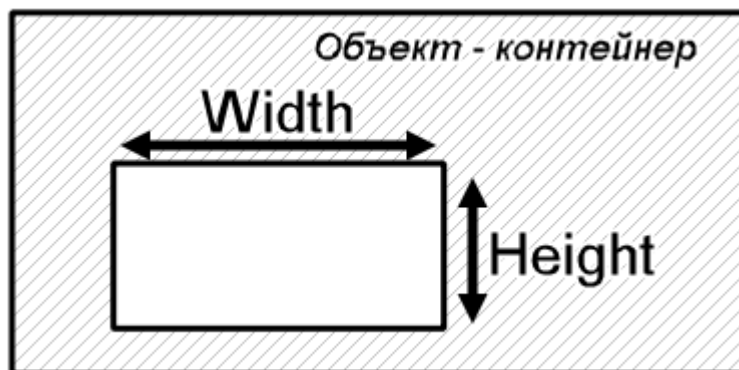
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.32.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

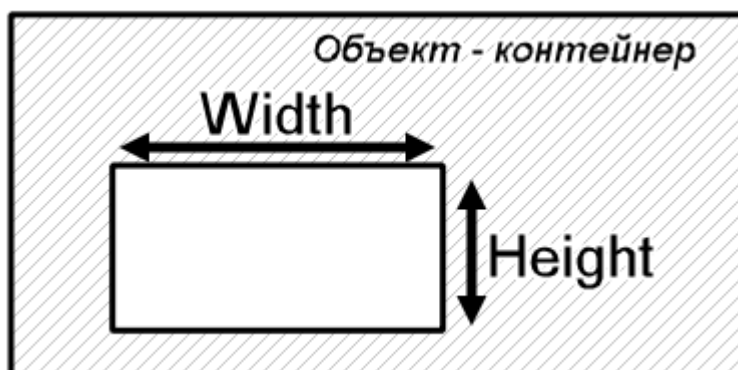


## 1.1.4.4.3.3.32.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.32.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

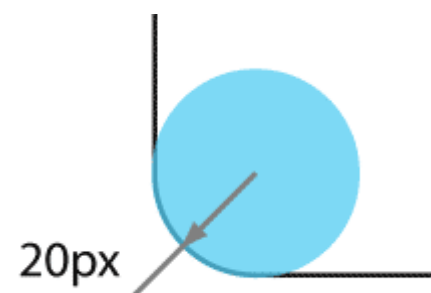
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.32.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.32.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.32.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.32.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```



## 1.1.4.4.3.3.32.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





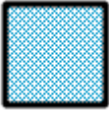

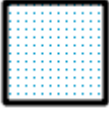
## 1.1.4.4.3.32.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.32.3.22. Значение

Изменяемое значение.



int8 Value

## 1.1.4.4.3.3.32.3.23. Максимум

Максимум, до которого можно инкрементировать изменяемое значение.



int8 Maximum

## 1.1.4.4.3.3.32.3.24. Минимум

Минимум, до которого можно декрементировать изменяемое значение.



int8 Minimum

## 1.1.4.4.3.3.32.3.25. Шаг изменения

Шаг инкремента / декремента.



int8 Step

## 1.1.4.4.3.3.32.3.26. ValueChanged

Ссылка на команду, выполняемую при изменении значения.



## 1.1.4.4.3.32.3.27. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.32.3.28. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

### 1.1.4.4.3.33. EventArgs

Вспомогательный тип. Представляет собой буфер, предназначенный для передачи аргументов события, реализованного с помощью компонента Команда.

Запись и извлечение данных происходит по принципу FIFO (first-in-first-out). При переполнении буфера новые аргументы вытесняют старые также по принципу очереди.

## 1.1.4.4.3.33.1. Функции

Функция	Описание
<a href="#">Push</a>	Записывает аргумент в буфер
<a href="#">Get</a>	Извлекает первый в очереди аргумент из буфера

## 1.1.4.4.3.33.1.1. Push

Записывает аргумент в буфер. Запись помещается в конец очереди.



void Push(string args)

### Параметры

Параметр	Тип	Описание
args	string	Значение аргумента

## 1.1.4.4.3.3.33.1.2. Get

Извлекает первый в очереди аргумент из буфера.



string Get()

## 1.1.4.4.3.33.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">BufferSize</a>	Размер буфера, измеряемый количеством записей
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.33.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.



## 1.1.4.4.3.33.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.3.33.2.3. BufferSize

Размер буфера, измеряемый количеством записей.



int4 BufferSize

## 1.1.4.4.3.33.2.4. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.33.2.5. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.34. WindowUtils

Вспомогательный тип для работы с окнами в режиме исполнения.

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

## 1.1.4.4.3.34.1. Функции

Функция	Описание
<a href="#">CorrectPosition</a>	Функция корректирует положение окна так, чтобы оно отобразилось на первом мониторе
<a href="#">FitToMonitor</a>	Размещает окно на указанном мониторе
<a href="#">GetMonitor</a>	Возвращает номер монитора, на котором отображается большая часть окна
<a href="#">CorrectPositionAboutPoint</a>	Корректирует положение окна относительно указанной точки так, чтобы окно располагалось в пределах монитора

## 1.1.4.4.3.34.1.1. CorrectPosition

Функция корректирует положение окна так, чтобы оно отобразилось на первом мониторе, если:

- › окно не отображается ни на одном из имеющихся мониторов;
- › окно невозможно переместить из-за выхода заголовка окна за пределы монитора.



```
void CorrectPosition()
```

Применяется при открытии окна с определенными координатами, которые могут выходить за область текущего монитора.

### Примеры



```
WindowUtils.CorrectPosition();
```

## 1.1.4.4.3.34.1.2. FitToMonitor

Размещает окно на указанном мониторе.



```
void FitToMonitor(int4 Monitor)
```

### Параметры

Параметр	Тип	Описание
Monitor	int4	Номер монитора

### Примеры



```
WindowUtils.FitToMonitor (1);
```



## 1.1.4.4.3.34.1.3. GetMonitor

Возвращает номер монитора, на котором отображается большая часть окна.



int4 GetMonitor ()

Можно использовать совместно с функцией [FitToMonitor](#), чтобы расположить окно целиком на том мониторе, где открывается его большая часть.

### Возвращаемое значение

Значение	Описание
-1	Не отображается ни на одном из имеющихся мониторов
>-1	Номер монитора

### Примеры



```
NumMonitor: int4 = WindowUtils.GetMonitor();
```

## 1.1.4.4.3.34.1.4. CorrectPositionAboutPoint

Корректирует положение окна относительно указанной точки так, чтобы окно располагалось в пределах монитора.

Применяется при открытии контекстного меню и других всплывающих окон.



```
void CorrectPositionAboutPoint (int4 X, int4 Y, int4 PointHeight)
```

### Параметры

Параметр	Тип	Описание
X	int4	Координата точки
Y	int4	Координата точки
PointHeight	int4	Смещение вверх относительно низа монитора

### Примеры



```
WindowUtils.CorrectPositionAboutPoint (0, 0, 10);
```

## 1.1.4.4.3.34.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Ссылка на форму</a>	Ссылка на форму, поведение окна которой настраивается с помощью функций этого компонента
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.34.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.34.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.34.2.3. Ссылка на форму

Ссылка на форму, поведение окна которой настраивается с помощью функций этого компонента.

## 1.1.4.4.3.34.2.4. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.34.2.5. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.4.3.35. StringUtils

Компонент, использующийся для различных преобразований строковых значений.

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

Чтобы выполнить преобразование, вызовите функцию компонента: `StringUtils.Function()`.

## 1.1.4.4.3.35.1. Функции

Компонент	Описание
<a href="#">GetDeclension</a>	Склоняет указанное во входных параметрах слово в зависимости от указанного числа (количества)
<a href="#">CalcRowHeight</a>	Вычисляет примерную высоту строки текста для указанного шрифта font

## 1.1.4.4.3.35.1.1. GetDeclension

Склоняет указанное во входных параметрах слово в зависимости от указанного числа (количества).



```
string GetDeclension(int8 Number, string Nominative, string Genetive, string Plural)
```

### Параметры

Параметр	Тип	Описание
Number	int8	Число (количество), в зависимости от которого склоняется слово
Nominative	string	Склоняемое слово в именительном падеже
Genetive	string	Склоняемое слово в родительном падеже
Plural	string	Склоняемое слово во множественное числе

Возвращает слово в склонении, подходящем указанному числу.

### Примеры



```
Предположим, некая переменная hour хранит количество часов, оставшихся до конца сессии пользователя, а в переменную word нужно записывать склоняемое слово "час". Тогда:  
word: string = StringUtils.GetDeclenxesion(hour, "час", "часа", "часов");  
Message.Text = "Осталось " + hour + word;
```

## 1.1.4.4.3.35.1.2. CalcRowHeight

Вычисляет примерную высоту строки текста для указанного шрифта font. Применяется для определения высоты заголовка в таблице при изменении шрифта темы [Theme](#).



int4 CalcRowHeight(string Font)

### Параметры

Параметр	Тип	Описание
Font	string	Строка, в которой перечислены параметры шрифта

### Примеры



Пример строки входного значения:  
Tahoma,8,-1,5,50,0,0,0,0,Обычный

## 1.1.4.4.3.35.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.35.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.35.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.35.2.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```



## 1.1.4.4.3.35.2.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.36. Converter

Компонент, использующийся для различных преобразований значений.

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

Чтобы выполнить преобразование, вызовите функцию компонента: `Converter.Function()`.

# Функции

Компонент, использующийся для различных преобразований значений. Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.

Функция	Описание
<a href="#">Convert</a>	Конвертирует значение value типа variant в значение типа DataType
<a href="#">StringToVariant</a>	Приводит указанное значение к универсальному типу данных variant
<a href="#">VariantToString</a>	Преобразовывает указанное значение из типа variant в string
<a href="#">VariantToLocalizedString</a>	Конвертирует переменную типа variant, хранящую значение value типа dataType, в значение типа string с учетом региональных настроек
<a href="#">ValidateValue</a>	Проверяет значение, записанное в виде строки, на соответствие указанному типу данных
<a href="#">ValidateRange</a>	Проверяет, входит ли указанное значение в допустимый диапазон от «min» до «max»
<a href="#">ValidateValueRange</a>	Проверяет значение, указанное в виде строки, на соответствие указанному типу данных и на принадлежность допустимому диапазону от «min» до «max»
<a href="#">CanonicalTypeToDataType</a>	Конвертирует канонический тип данных, заданный в виде числа, в название этого типа в виде строки

# Convert

Конвертирует значение `value` типа `variant` в значение типа `DataType`. Результат функции хранится в переменной типа `variant`.



```
variant Convert(variant Value, string DataType)
```

В качестве значения `DataType` можно указывать только названия типов данных языка `Astra.Om`. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
Value	variant	Преобразуемое значение
DataType	string	Тип преобразуемого значения

Преобразование выполняется с учетом ограничений пространства имен `TypeConvert`. Подробнее об ограничениях – в разделе на [Astra.Om](#).

## Примеры



```
param: variant = Converter.Convert(52, "uint4");
```

# StringToVariant

Приводит указанное значение к универсальному типу данных variant.



```
variant StringToVariant(string Value, string DataType)
```

В качестве значения dataType можно указывать только названия типов данных языка Astra.OM. Типы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
Value	string	Преобразуемое значение
DataType	string	Тип преобразуемого значения

## Примеры



```
param: variant = Converter.StringToVariant("123", "uint4");
```

# VariantToString

Преобразовывает указанное значение из типа variant в string.



```
string VariantToString(variant Value, string DataType)
```

В качестве значения DataType можно указывать только названия типов данных языка Astra.OM. ипы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
Value	variant	Преобразуемое значение
DataType	string	Тип преобразуемого значения

## Примеры



```
param: string = Converter.StringToVariant(InputValue, "uint4");
```

# VariantToLocalizedString

Конвертирует переменную типа `variant`, хранящую значение `value` типа `dataType`, в значение типа `string` с учетом региональных настроек. В региональных настройках могут быть заданы: точность (количество знаков после разделителя) и знак разделителя.



```
string VariantToLocalizedString(variant Value, string DataType)
```

В качестве значения `DataType` можно указывать только названия типов данных языка `Astra.Om`. ипы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
<code>Value</code>	<code>variant</code>	Преобразуемое значение
<code>DataType</code>	<code>string</code>	Тип преобразуемого значения

## Примеры



```
Text_1 = String.VariantToLocalizedString(a, "double");
```

# ValidateValue

Проверяет значение, записанное в виде строки, на соответствие указанному типу данных.



```
string ValidateValue(string Value, string DataType)
```

В качестве значения `DataType` можно указывать только названия типов данных языка `Astra.OM`. ипы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
Value	string	Проверяемое значение.
DataType	string	Тип данных.

## Возвращаемое значение

- › текст ошибки в случае ее возникновения;
- › пустую строку, если ошибки не возникает.

## Примеры



```
result: string = Converter.ValidateValue("1234", "uint4");
```



# ValidateRange

Проверяет, входит ли указанное значение в допустимый диапазон от «min» до «max».



```
string ValidateRange(variant Value, string DataType, variant Min, variant Max)
```

В качестве значения `DataType` можно указывать только названия типов данных языка Astra.OM. ипы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
Value	variant	Проверяемое значение
DataType	string	Тип данных
Min	variant	Нижняя граница диапазона «min»
Max	variant	Верхняя граница диапазона «max»



Если не указать значения «min» и/или «max», будут использоваться границы диапазона указанного типа данных.

## Возвращаемое значение

- › текст ошибки в случае ее возникновения;
- › пустую строку, если ошибки не возникает.

## Примеры



```
result: string = Converter.ValidateRange(123, "uint8", 1, 500);
```

# ValidateValueRange

Проверяет значение, указанное в виде строки, на соответствие указанному типу данных и на принадлежность допустимому диапазону от «min» до «max».



```
string ValidateValueRange(string Value, string DataType, variant Min, variant Max)
```

В качестве значения `DataType` можно указывать только названия типов данных языка Astra.OM. ипы можно найти в документе на соответствующий продукт, либо в таблице, приведенной в описании функции [CanonicalTypeToDataType](#). Также можно использовать значение, возвращаемое этой функцией.

## Параметры

Параметр	Тип	Описание
Value	string	Проверяемое значение
DataType	string	Проверяемый тип данных
Min	variant	Нижняя граница диапазона «min»
Max	variant	Верхняя граница диапазона «max»

## Возвращаемое значение

- › текст ошибки в случае ее возникновения;
- › пустую строку, если ошибки не возникает.

## Примеры



```
result: string = Converter.ValidateValueRange(123, "uint8", 1, 500);
```

# CanonicalTypeToDataType

Вспомогательная функция. Возвращает название канонического типа данных, указанного в числовом виде. Возвращаемое значение можно использовать в качестве входного параметра dataType остальных функции Convert.



```
string CanonicalTypeToDataType(uint1 CanonicalType)
```

## Параметры

Параметр	Тип	Описание
CanonicalType	uint1	Число, соответствующее каноническому типу данных.

## Возвращаемое значение

Функция возвращает название типа данных в виде string.

Типе	Название типа
0	"void"
1	"bool"
2	"int1"
3	"uint1"
4	"int2"
5	"uint2"
6	"int4"
7	"uint4"
8	"int8"

9	"uint8"
10	"float"
11	"double"
12	"timestamp"
14	"string"

## Примеры



Вызов: `a = Converter.CanonicalTypeToDataType(3);`  
Результат: переменная `a` типа `string`, значение – «uint1».

## 1.1.4.4.3.36.1. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.36.1.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.36.1.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



## 1.1.4.4.3.36.1.3. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.36.1.4. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



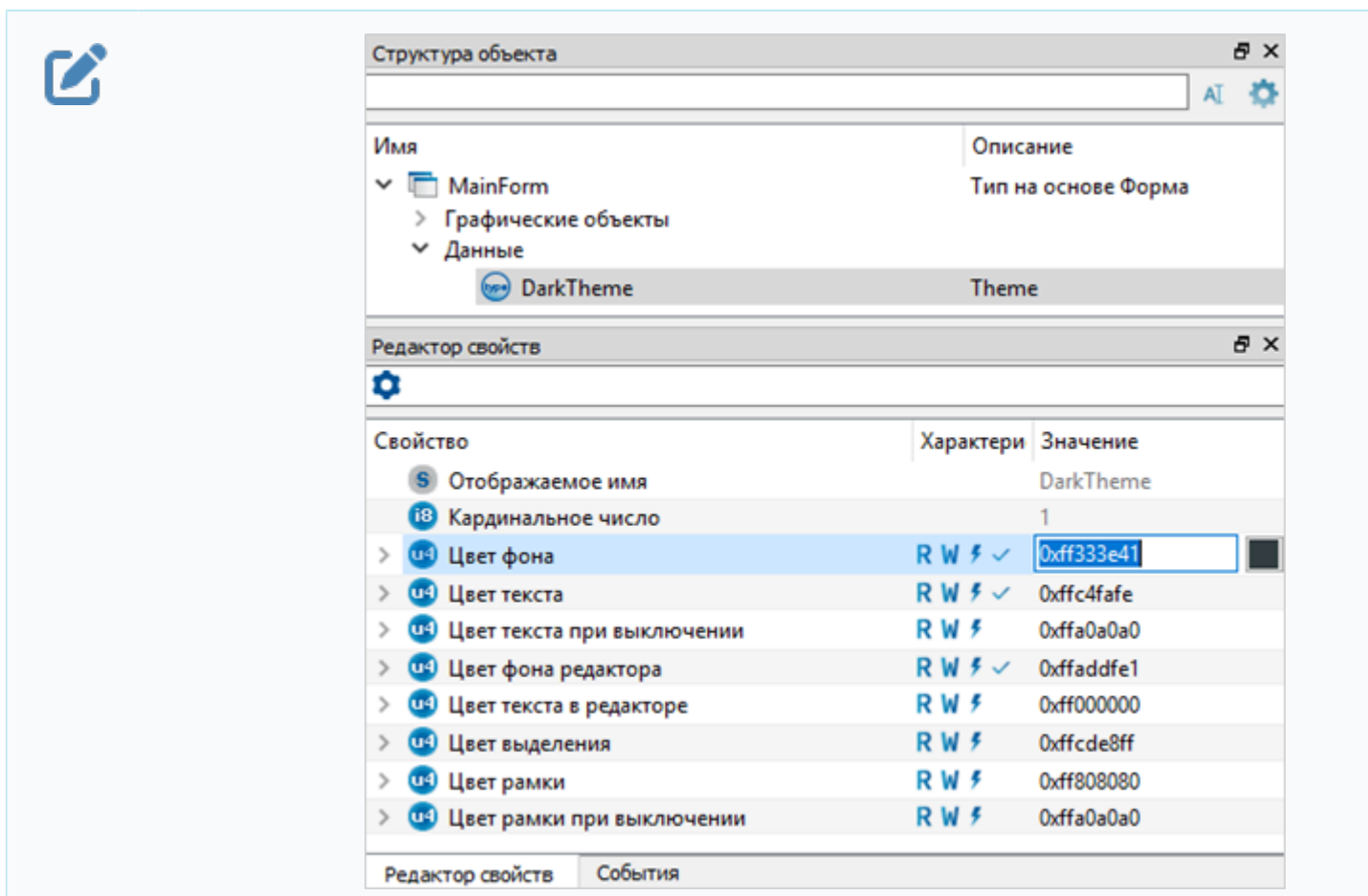
//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.4.3.37. Theme

Компонент, позволяющий менять тему оформления визуальных компонентов Astra.HMI.CommonLib:

- › шрифт основного текста;
- › шрифт заголовков;
- › цвет фона;
- › цвет текста;
- › цвет текста при выключении;
- › цвет фона редактора;
- › цвет текста в редакторе;
- › цвет выделения;
- › цвет рамки;
- › цвет рамки при выключении.

Экземпляр этого типа не отображается на форме и виден только в области Структура объекта.



Скриншот интерфейса разработки, показывающий структуру объекта и редактор свойств.

**Структура объекта**

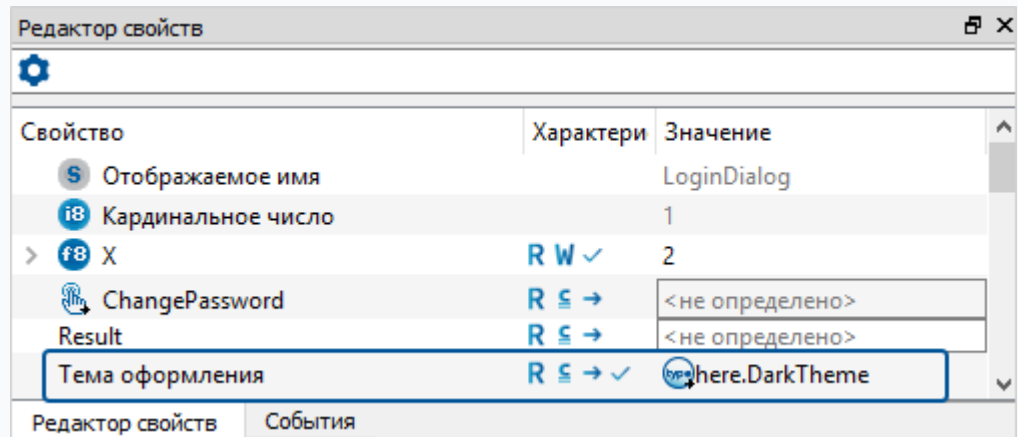
Имя	Описание
MainForm	Тип на основе Форма
Графические объекты	
Данные	
DarkTheme	Theme

**Редактор свойств**

Свойство	Характери	Значение
Отображаемое имя		DarkTheme
Кардинальное число		1
Цвет фона	R W ✓	0xff333e41
Цвет текста	R W ✓	0xffc4f4fe
Цвет текста при выключении	R W ✓	0xffa0a0a0
Цвет фона редактора	R W ✓	0xffaddfe1
Цвет текста в редакторе	R W ✓	0xff000000
Цвет выделения	R W ✓	0xffcde8ff
Цвет рамки	R W ✓	0xff808080
Цвет рамки при выключении	R W ✓	0xffa0a0a0

DarkTheme – экземпляр типа Theme.

Чтобы применить тему DarkTheme к объекту, укажите в свойстве Тема оформления этого объекта ссылку на экземпляр DarkTheme.



На экземпляр может быть сделано сколько угодно ссылок.

## 1.1.4.4.3.37.1. Функции

Компонент	Описание
GetViewSettings	Возвращает настройки текущей темы в виде json-строки
UpdateView	Меняет внешний вид компонентов, у которых в свойстве Тема оформления указана текущая тема, в соответствии с настройками этой темы

## 1.1.4.4.3.37.1.1. GetViewSettings

Возвращает настройки текущей темы в виде json-строки. Эти настройки могут использоваться для восстановления внешнего вида компонентов с помощью функции [UpdateView](#) при перезапуске проекта.



string GetViewSettings()

### Примеры



Пример возвращаемого значения:

```
{  
  "Font": "Times New Roman,12,-1,5,50,0,0,0,0,Обычный",  
  "TitleFont": "Times New Roman,12,-1,5,50,0,0,0,0,Обычный",  
  "Background": "0xff00344a",  
  "Foreground": "0xffff0f0f0",  
  "DisabledFont": "0xffa0a0a0",  
  "EditorBrush": "0xff416075",  
  "EditorFont": "0xffffffff",  
  "Selection": "0xff557e99",  
  "Border": "0xffb0b0b0",  
  "DisabledBorder": "0xffa0a0a0"  
}
```

## 1.1.4.4.3.37.1.2. UpdateView

Меняет внешний вид компонентов, у которых в свойстве Тема оформления указана текущая тема, в соответствии с настройками этой темы. Настройки могут быть получены в результате вызова функции [GetViewSettings](#).



```
void UpdateView(string Json)
```

### Параметры

Параметр	Тип	Описание
Json	string	Настройки текущей темы в виде json-строки

## 1.1.4.4.3.37.2. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">Шрифт</a>	Стиль шрифта
<a href="#">Шрифт заголовка</a>	Шрифт текста заголовка столбца
<a href="#">Цвет фона</a>	Цвет фона
<a href="#">Цвет текста</a>	Цвет текста в столбце
<a href="#">Цвет текста при выключении</a>	Цвет текста при выключении
<a href="#">Цвет фона редактора</a>	Цвет фона редактора
<a href="#">Цвет текста в редакторе</a>	Цвет текста в редакторе
<a href="#">Цвет выделения</a>	Цвет выделения
<a href="#">Цвет рамки</a>	Цвет рамки
<a href="#">Цвет рамки при выключении</a>	Цвет рамки при выключении
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве



## 1.1.4.4.3.37.2.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.37.2.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.37.2.3. Шрифт

Стиль шрифта. Указывается через системное окно Выбор шрифта. После выбора автоматически преобразуется в строку.



string Font

## 1.1.4.4.3.37.2.4. Шрифт заголовка

Шрифт текста заголовка столбца.



uint4 TitleFont

## 1.1.4.4.3.37.2.5. Цвет фона

Цвет фона.



uint4 BackgroundColor

## 1.1.4.4.3.3.37.2.6. Цвет текста

Цвет текста в столбце.



uint4 ForegroundColor

## 1.1.4.4.3.37.2.7. Цвет текста при выключении

Цвет текста при выключении.



uint4 DisabledFontColor

## 1.1.4.4.3.3.37.2.8. Цвет фона редактора

Цвет фона редактора.



uint4 EditorBrushColor



## 1.1.4.4.3.3.37.2.9. Цвет текста в редакторе

Цвет текста в редакторе.



uint4 EditorFontColor

## 1.1.4.4.3.3.37.2.10. Цвет выделения

Цвет выделения.



uint4 SelectionColor

## 1.1.4.4.3.3.37.2.11. Цвет рамки

Цвет рамки.



uint4 BorderColor

## 1.1.4.4.3.37.2.12. Цвет рамки при выключении

Цвет рамки при выключении.



uint4 DisabledBorderColor

## 1.1.4.4.3.37.2.13. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.37.2.14. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

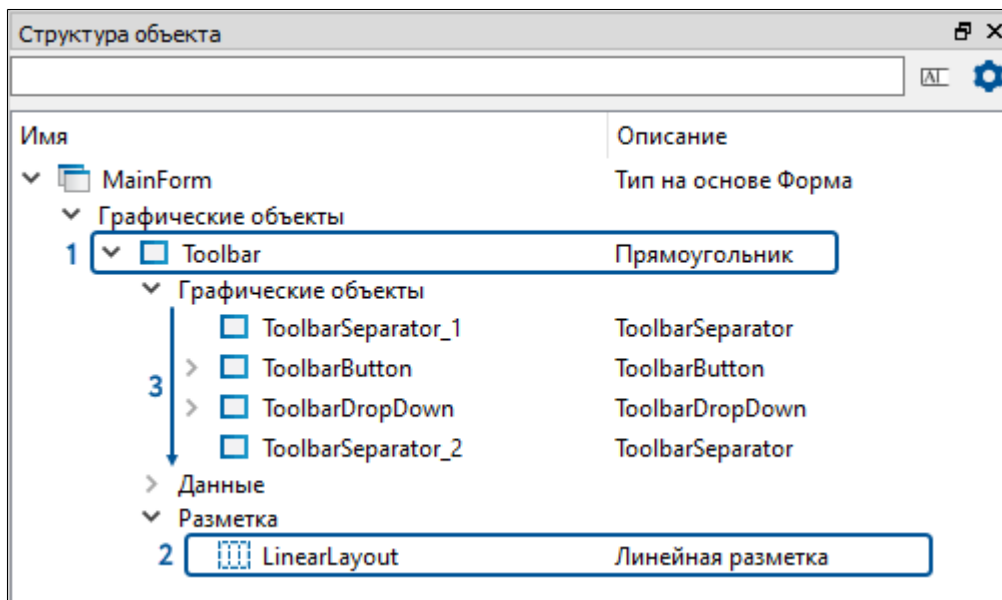
## 1.1.4.4.3.38. Панель инструментов

Панель инструментов в проекте Astra.HMI можно построить из экземпляров компонентов:

- › [ToolBarButton](#)
- › [ToolBarDropDown](#)
- › [SearchBar](#)
- › [ToolBarSeparator](#)

Чтобы создать панель инструментов:

1. Создайте графический объект типа Прямоугольник.
2. Примените к нему инструмент Линейная разметка. Это необходимо для корректного поведения компонентов панели.
3. Экземпляры компонентов панели размещайте внутри созданного прямоугольника. Учитывайте порядок отображения экземпляров в структуре объекта: в таком же порядке они будут расположены на панели инструментов.



Пример построения панели инструментов приведен в Выбор цвета на панели инструментов из палитры или из меню.

## 1.1.4.4.3.38.1. ToolbarButton

Кнопка для панели инструментов.



Внешний вид и функции кнопки можно менять с помощью свойств и событий. Кнопку можно использовать совместно с [ToolbarDropDown](#). Для визуального разделения панели инструментов можно использовать [ToolbarSeparator](#).



Для корректного поведения размещайте экземпляр компонента внутри [Панели инструментов](#).



## 1.1.4.4.3.38.1.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.38.1.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.38.1.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.1.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.38.1.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyBoardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.38.1.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.38.1.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

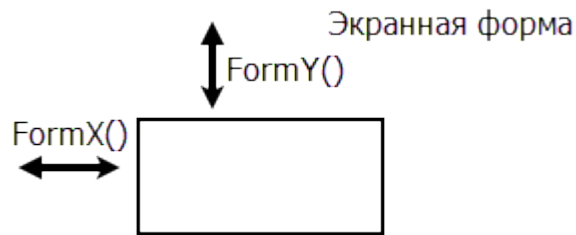
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.38.1.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.38.1.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

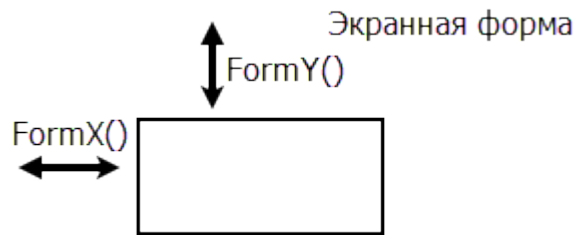
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.38.1.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



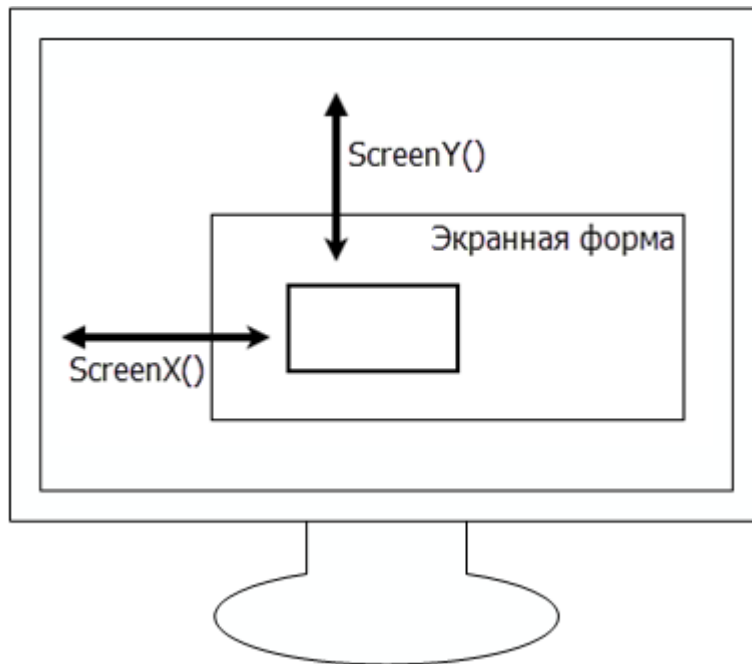
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.1.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

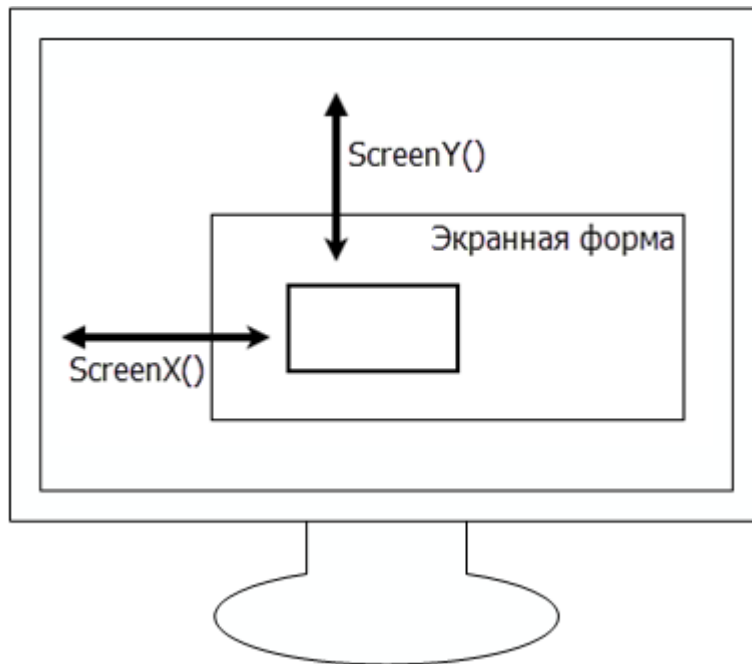
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.38.1.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.1.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов

<a href="#">IconFile</a>	Путь к файлу иконки для текущего экземпляра кнопки
<a href="#">TitleText</a>	Текст кнопки
<a href="#">Checked</a>	Позволяет использовать кнопку как удерживаемую
<a href="#">Active</a>	Позволяет изменить активность кнопки
<a href="#">IsVisible</a>	Позволяет изменить видимость кнопки
<a href="#">pDropDown</a>	Ссылка на определенный экземпляр <a href="#">ToolbarDropDown</a>
<a href="#">Margin</a>	Устанавливает размер отступов для расположения иконки относительно кнопки
<a href="#">IconSize</a>	Размер иконки для текущего экземпляра кнопки
<a href="#">ButtonWidth</a>	Ширина кнопки
<a href="#">ButtonHeight</a>	Высота кнопки
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.38.1.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.38.1.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

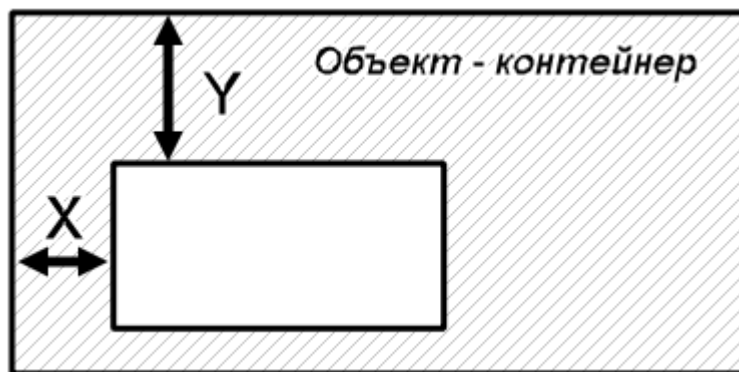
## 1.1.4.4.3.38.1.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```



## 1.1.4.4.3.38.1.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.38.1.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.38.1.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

# 1.1.4.4.3.38.1.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

## Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.38.1.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.38.1.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.38.1.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.38.1.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```



## 1.1.4.4.3.38.1.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




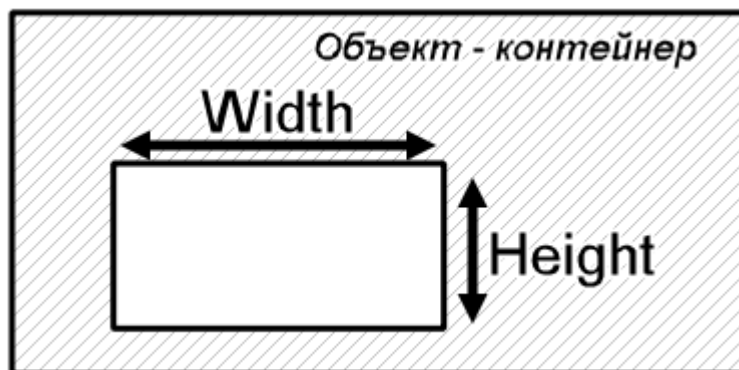
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.38.1.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width

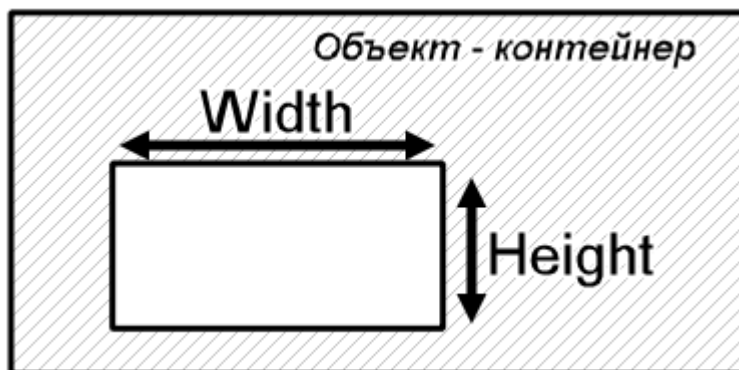


# 1.1.4.4.3.3.38.1.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.38.1.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

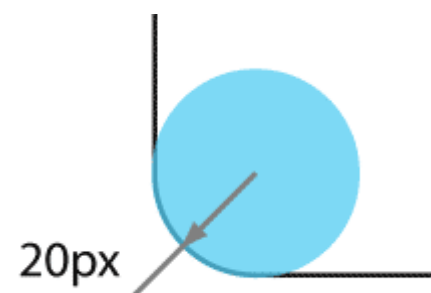
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.38.1.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.38.1.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.38.1.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.38.1.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```



## 1.1.4.4.3.38.1.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





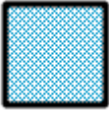

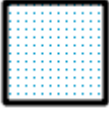
## 1.1.4.4.3.38.1.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

# 1.1.4.4.3.3.38.1.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.38.1.3.23. IconFile

Путь к файлу иконки для текущего экземпляра кнопки.



string IconFile

## 1.1.4.4.3.38.1.3.24. TitleText

Текст кнопки.

Для корректного отображения текста необходимо вручную подобрать ширину кнопки с помощью свойства `ButtonWidth`.



`string TitleText`

## 1.1.4.4.3.38.1.3.25. Checked

Позволяет использовать кнопку как удерживаемую.



bool Checked

### Значение

Значение	Описание
true	Кнопка удерживаемая, выглядит как нажатая
false	Кнопка отпущена, выглядит как обычно

## 1.1.4.4.3.3.38.1.3.26. Active

Позволяет изменить активность кнопки.



bool Active

### Значение

Значение	Описание
true	Кнопка активна и выглядит как обычно, при наведении мышкой подсвечивается
false	Кнопка неактивна: цвета иконки бледные, при наведении мышкой кнопка не подсвечивается



## 1.1.4.4.3.3.38.1.3.27. IsVisible

Позволяет изменить видимость кнопки.



bool IsVisible

### Значение

Значение	Описание
true	Кнопка отображается и выглядит как обычно
false	Кнопка не отображается, так как ее ширина становится нулевой, благодаря чему выравниваются остальные элементы панели

## 1.1.4.4.3.3.38.1.3.28. pDropDown

Ссылка на определенный экземпляр `ToolBarDropDown`.

Если указать значение, то в рантайме при наведении мышкой на текущий экземпляр `ToolBarButton` подсвечиваться также будет указанный экземпляр `ToolBarDropDown`.

## 1.1.4.4.3.3.38.1.3.29. Margin

Устанавливает размер отступов для расположения иконки относительно кнопки.



uint1 Margin

## 1.1.4.4.3.3.38.1.3.30. IconSize

Размер иконки для текущего экземпляра кнопки.



uint1 IconSize

# 1.1.4.4.3.3.38.1.3.31. ButtonWidth

Ширина кнопки.



uint2 ButtonWidth

# 1.1.4.4.3.38.1.3.32. ButtonHeight

Высота кнопки.



uint2 ButtonHeight

## 1.1.4.4.3.38.1.3.33. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.38.1.3.34. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле индекс элемента массива, у которого  
сработало событие изменения значения:  
TextEdit_2.Text = String.ToString(Index);
```



```
//Записать в текстовое поле текущее значение элемента массива, у  
которого сработало событие изменения значения:  
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



```
//В зависимости от индекса элемента в массиве вывести в  
текстовое поле запись о включении соответствующего режима  
работы задвижки:  
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```



## 1.1.4.4.3.38.2. ToolbarDropDown

Кнопка открытия выпадающего элемента для панели инструментов.



Внешний вид и функции кнопки можно менять с помощью свойств и событий. Кнопку можно использовать совместно с [ToolbarButton](#). Для визуального разделения панели инструментов можно использовать [ToolbarSeparator](#).



Для корректного поведения размещайте экземпляр компонента внутри [Панели инструментов](#). Пример использования приведен в Выбор цвета на панели инструментов из палитры или из меню.

## 1.1.4.4.3.38.2.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.38.2.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.38.2.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.2.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.38.2.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.38.2.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.38.2.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

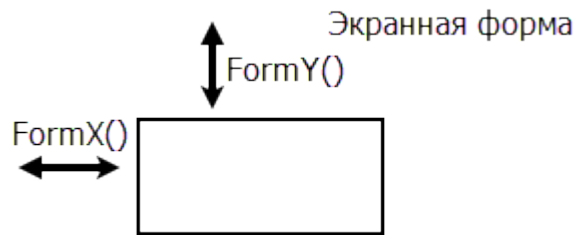
## 1.1.4.4.3.38.2.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y



## 1.1.4.4.3.38.2.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



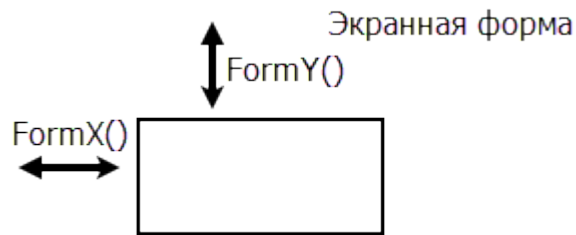
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.2.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



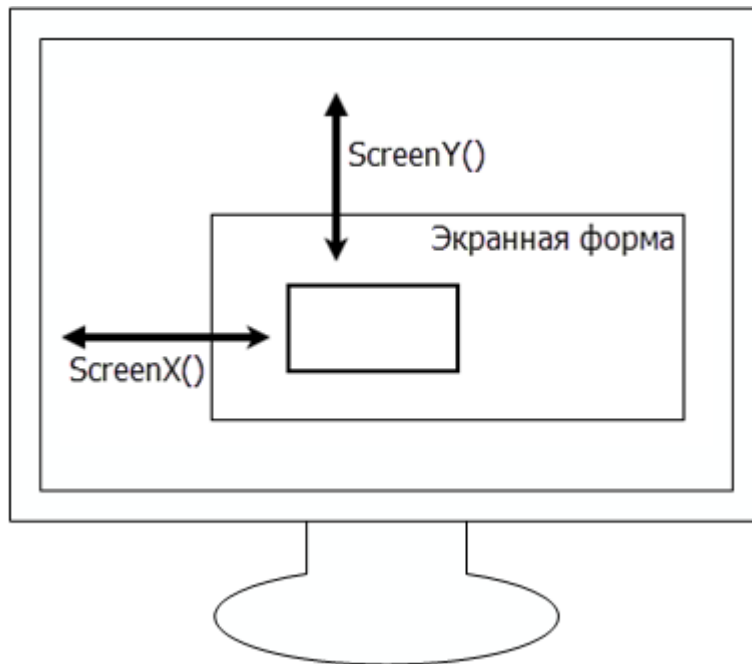
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.2.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

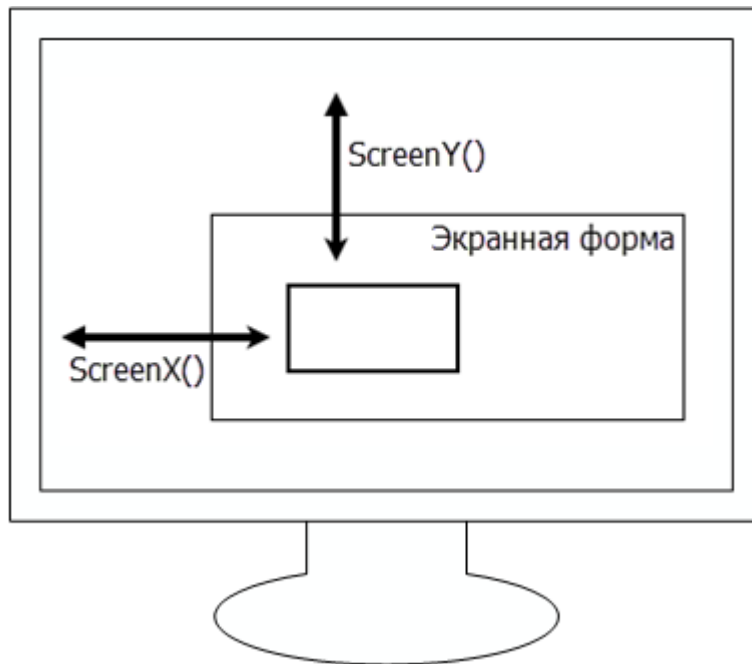
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.38.2.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.2.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов

<a href="#">Checked</a>	Позволяет использовать кнопку как удерживаемую
<a href="#">Active</a>	Позволяет изменить активность кнопки
<a href="#">IsVisible</a>	Позволяет изменить видимость кнопки
<a href="#">pButton</a>	Ссылка на определенный экземпляр ToolbarDropDown
<a href="#">Margin</a>	Устанавливает размер отступов для расположения иконки относительно кнопки
<a href="#">ButtonHeight</a>	Высота кнопки
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.38.2.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.38.2.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности



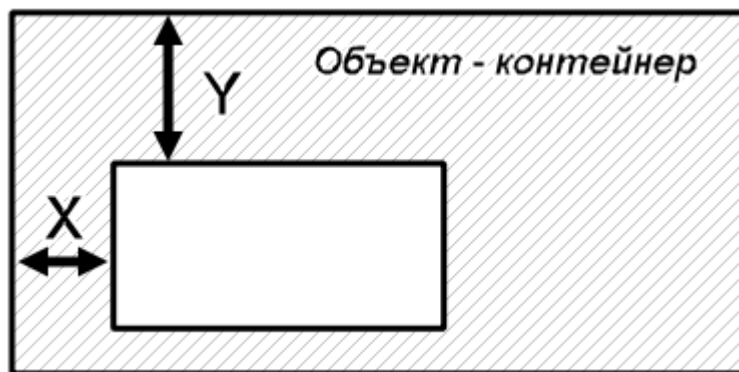
## 1.1.4.4.3.38.2.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

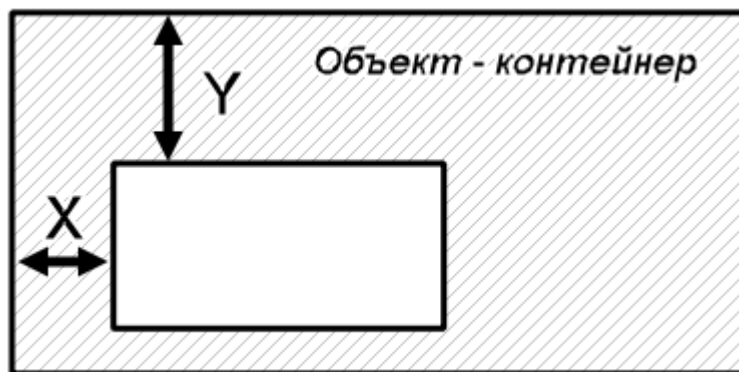
## 1.1.4.4.3.38.2.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.38.2.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.38.2.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.38.2.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.38.2.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.38.2.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.38.2.3.10. Непрозрачность

Непрозрачность.



float8 Opacity



# 1.1.4.4.3.38.2.3.11. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.38.2.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




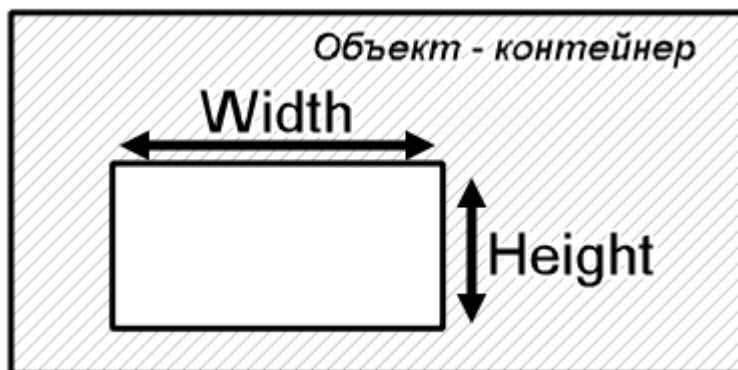
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.38.2.3.13. Ширина


Ширина объекта в пикселях.

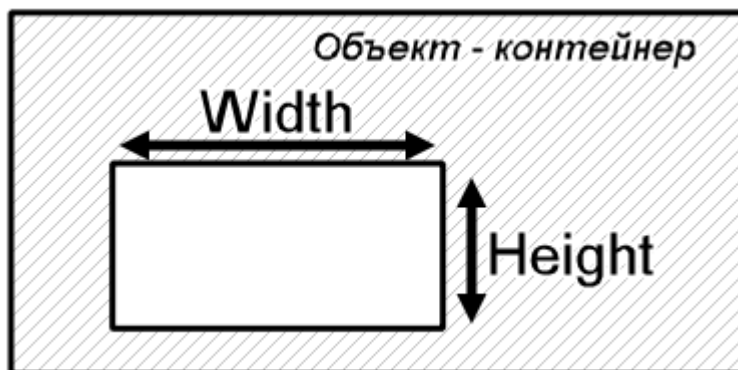
 float8 Width



# 1.1.4.4.3.38.2.3.14. Высота

Высота объекта в пикселях.

 float8 Height



## 1.1.4.4.3.38.2.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

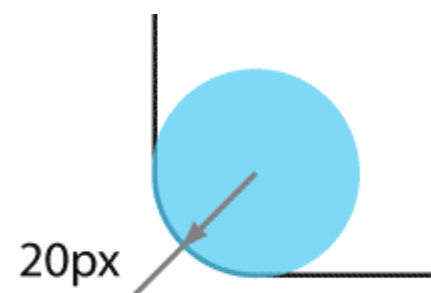
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.38.2.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.38.2.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.38.2.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;



## 1.1.4.4.3.38.2.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.38.2.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;

## 1.1.4.4.3.38.2.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

# 1.1.4.4.3.3.38.2.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.38.2.3.23. Checked

Позволяет использовать кнопку как удерживаемую.



bool Checked

### Значение

Значение	Описание
true	Кнопка удерживаемая, выглядит как нажатая
false	Кнопка отпущена, выглядит как обычно

## 1.1.4.4.3.3.38.2.3.24. Active

Позволяет изменить активность кнопки.



bool Active

### Значение

Значение	Описание
true	Кнопка активна и выглядит как обычно, при наведении мышкой подсвечивается
false	Кнопка неактивна: цвета иконки бледные, при наведении мышкой кнопка не подсвечивается

## 1.1.4.4.3.3.38.2.3.25. isVisible

Позволяет изменить видимость кнопки.



bool isVisible

### Значение

Значение	Описание
true	Кнопка отображается и выглядит как обычно
false	Кнопка не отображается, так как ее ширина становится нулевой, благодаря чему выравниваются остальные элементы панели



## 1.1.4.4.3.3.38.2.3.26. pButton

Ссылка на определенный экземпляр `ToolBarButton`.

Если указать значение, то в рантайме при наведении мышкой на текущий экземпляр `ToolBarDropDown` подсвечиваться также будет указанный экземпляр `ToolBarButton`.

## 1.1.4.4.3.3.38.2.3.27. Margin

Устанавливает размер отступов для расположения иконки относительно кнопки.



uint1 Margin

# 1.1.4.4.3.3.38.2.3.28. ButtonHeight

Высота кнопки.



uint2 ButtonHeight

## 1.1.4.4.3.38.2.3.29. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.38.2.3.30. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:

```
TextEdit_2.Text = String.ToString(Index);
```



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:

```
TextEdit_3.Text = String.ToString(penwidth[Index]);
```



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:

```
if (SW_1.Mode.Index == 0) {  
    TextEdit_1.Text = "Включен режим 1";  
}  
else  
    {TextEdit_1.Text = "Включен режим 2";}
```

## 1.1.4.4.3.38.3. ToolbarSeparator

Разделитель элементов для панелей инструментов.



Для корректного поведения размещайте экземпляр компонента внутри [Панели инструментов](#).



После изменения размеров компонента для обновления разметки необходимо вызвать метод `UpdateLayout()`.

## 1.1.4.4.3.38.3.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта

## 1.1.4.4.3.38.3.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.38.3.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.3.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.3.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.3.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.3.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.3.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.3.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```



## 1.1.4.4.3.38.3.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.38.3.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.38.3.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.38.3.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

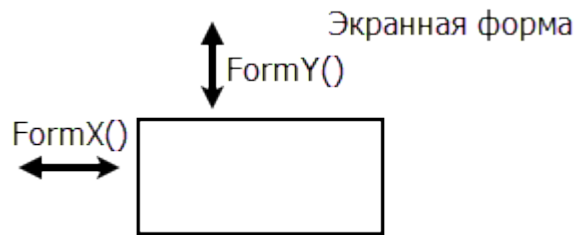
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.3.38.3.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.38.3.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



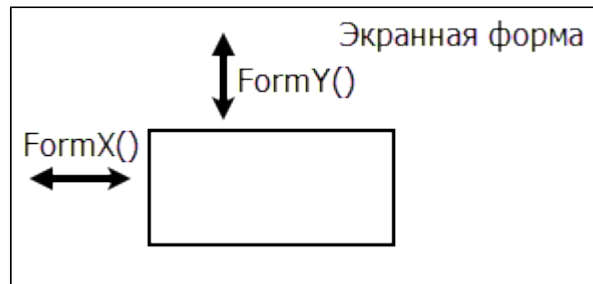
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.3.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X, Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



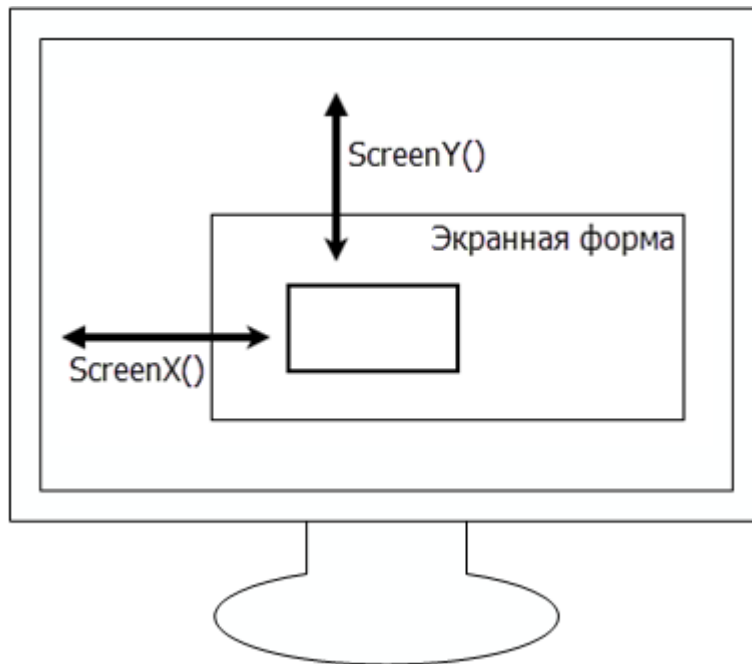
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.3.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры

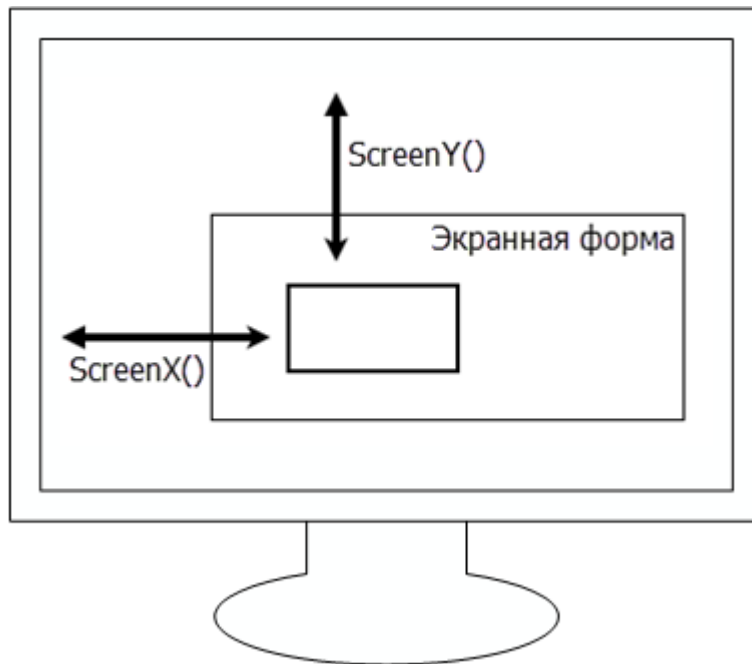


```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```



## 1.1.4.4.3.38.3.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.3.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">Тема оформления</a>	Тема оформления визуальных компонентов

<a href="#"><u>IsVisible</u></a>	Позволяет изменить видимость кнопки
<a href="#"><u>ButtonHeight</u></a>	Высота кнопки
<a href="#"><u>Length</u></a>	Размер массива (количество элементов в массиве)
<a href="#"><u>Index</u></a>	Индекс элемента в массиве

## 1.1.4.4.3.3.38.3.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.3.38.3.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

## 1.1.4.4.3.38.3.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



//Установить позицию элемента (100; 0):

```
Object.X = 100;
```



//Сдвинуть объект на 50 пикселей правее его текущей позиции:

```
Object.X += 50;
```

## 1.1.4.4.3.38.3.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```

# 1.1.4.4.3.3.38.3.3.5. Z-значение

Z-значение.



float8 ZValue



## 1.1.4.4.3.3.38.3.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.38.3.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.38.3.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.3.38.3.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.38.3.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

## 1.1.4.4.3.38.3.3.11. Включено

Активность объекта.



bool Enabled

### Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

### Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.38.3.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




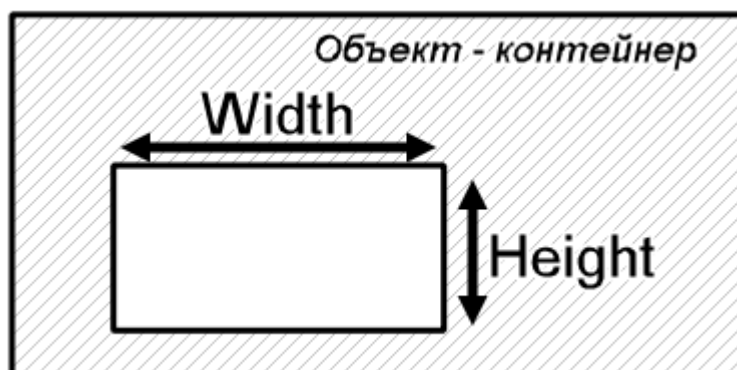
//Задать всплывающую подсказку для объекта:

```
Object.Tooltip = "Подача команды MNA_start = true";
```

# 1.1.4.4.3.3.38.3.3.13. Ширина

Ширина объекта в пикселях.

 float8 Width



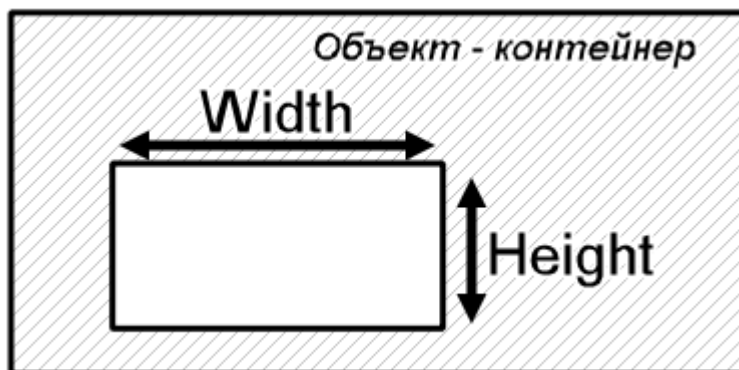


# 1.1.4.4.3.3.38.3.3.14. Высота

Высота объекта в пикселях.



float8 Height



## 1.1.4.4.3.3.38.3.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

### Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

### Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

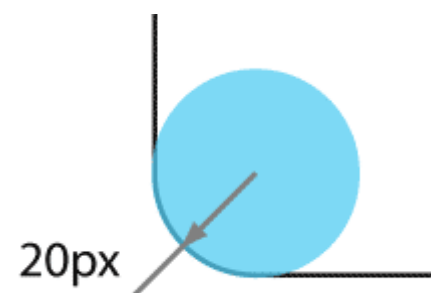
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.38.3.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.38.3.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.3.38.3.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.3.38.3.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.3.38.3.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;




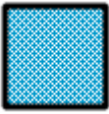

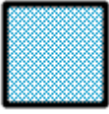

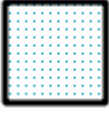
## 1.1.4.4.3.38.3.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	



Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

# 1.1.4.4.3.3.38.3.3.22. Тема оформления

Тема оформления визуального компонента.



theme pTheme

## 1.1.4.4.3.3.38.3.3.23. isVisible

Позволяет изменить видимость кнопки.



bool isVisible

### Значение

Значение	Описание
true	Кнопка отображается и выглядит как обычно
false	Кнопка не отображается, так как ее ширина становится нулевой, благодаря чему выравниваются остальные элементы панели

# 1.1.4.4.3.3.38.3.3.24. ButtonHeight

Высота кнопки.



uint2 ButtonHeight

## 1.1.4.4.3.38.3.3.25. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.3.38.3.3.26. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



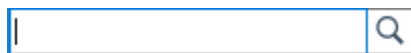
//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

## 1.1.4.4.3.38.4. SearchBar

Панель поиска.



Представляет собой окно ввода текста для поиска (шаблона) и кнопку запуска процесса поиска.



После ввода шаблона кнопка меняет свое назначение и позволяет очистить результаты поиска.



## 1.1.4.4.3.38.4.1. События

Событие	Описание
<a href="#">MouseMove</a>	Перемещение курсора внутри области объекта
<a href="#">MouseClicked</a>	Щелчок левой кнопкой мыши по объекту
<a href="#">MouseRightClick</a>	Щелчок правой кнопкой мыши по объекту
<a href="#">MouseDoubleClick</a>	Двойной щелчок левой кнопкой мыши по объекту
<a href="#">MousePress</a>	Нажатие кнопки мыши в области объекта
<a href="#">MouseRelease</a>	Нажатая кнопка мыши отпущена в области объекта
<a href="#">MouseEnter</a>	Ввод курсора в область объекта
<a href="#">MouseLeave</a>	Вывод курсора за пределы области объекта
<a href="#">TouchMove</a>	Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану
<a href="#">TouchPress</a>	Нажатие на объект
<a href="#">TouchRelease</a>	Событие срабатывает, если после нажатия на объект отпустить перо в любом месте экрана
<a href="#">FocusChanged</a>	Изменение фокуса у объекта



## 1.1.4.4.3.38.4.1.1. MouseMove

Перемещение курсора внутри области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.2. MouseClick

Щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;  
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.3. MouseRightClick

Щелчок правой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.4. MouseDoubleClick

Двойной щелчок левой кнопкой мыши по объекту.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.5. MousePress

Нажатие кнопки мыши в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.6. MouseRelease

Нажатая кнопка мыши отпущена в области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.7. MouseEnter

Ввод курсора в область объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```



//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.8. MouseLeave

Вывод курсора за пределы области объекта.

### Параметры

Параметр	Тип	Описание
X	float	Локальная координата объекта по оси X, в которой возникло событие
Y	float	Локальная координата объекта по оси Y, в которой возникло событие
Buttons	int4	Код кнопок мыши, которые были нажаты при возникновении события
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события



Для доступа к параметрам используйте структуру event.

### Примеры



//Разместить кнопку в месте срабатывания события:

```
Button.X = event.X;
```

```
Button.Y = event.Y;
```



//Записать в текстовое поле код кнопок мыши, которые были зажаты при возникновении события:

```
TextEdit_1.Text = String.ToString(event.KeyboardModifiers);
```





//Записать в текстовое поле код клавиш клавиатуры, которые были нажаты при возникновении события:

```
TextEdit_2.Text = String.ToString(event.Buttons);
```

## 1.1.4.4.3.38.4.1.9. TouchMove

Нажатие на объект и дальнейшее ведение пальцем (пером) по экрану.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//Выводить в текстовые поля координаты точки нажатия экрана по оси X и по оси Y:

```
TextEdit_1.Text = String.ToString(event.PointX);
```

```
TextEdit_2.Text = String.ToString(event.PointY);
```



//При перетягивании объекта Tank на таблицу заполнять ячейки таблицы текущими данными по резервуару (название, уровень кислорода, температура жидкости и т.д.). В обработчике события TouchMove объекта Tank пропишите передаваемые значения:

```
Tank_1.DragNDrop_1.setData("name",Str.toString(rez345.Name .Значение));
```

```
Tank_1.DragNDrop_1.setData("h2o",Str.toString(rez345.H2O.Value));
```

```
Tank_1.DragNDrop_1.setData("temp",Str.toString(rez345.Temp.Value));
```

```
Танк_1.DragNDrop_1.DoDragDrop();
```

## 1.1.4.4.3.38.4.1.10. TouchPress

Нажатие на объект.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При нажатии на объект SW и зажатии клавиши Ctrl увеличивать масштаб объекта. В обработчике события TouchPress объекта SW пропишите:

```
if (event.KeyboardModifiers == 67108864)
{
    SW_1.Scale = 5;
}
```

## 1.1.4.4.3.38.4.1.11. TouchRelease

Событие срабатывает, если после нажатия на объект отпустить палец (перо) в любом месте экрана.

Параметр	Тип	Описание
PointX	float	Функция, получающая координату точки нажатия по оси X
PointY	float	Функция, получающая координату точки нажатия по оси Y
PointsCount	uint4	Количество нажатий
KeyboardModifiers	int4	Коды клавиш клавиатуры, которые были зажаты при возникновении события

### Примеры



//При поднятии пальца (пера) после нажатия на объект SW нужно сместить кнопку и присвоить сигналу OpenSW значение 1. В обработчике события TouchRelease объекта типа SW пропишите:

```
Ellipse_1.X = 52;  
Ellipse_1.Y = 341;  
SW_1.OpenSW = 1;
```

## 1.1.4.4.3.38.4.1.12. FocusChanged

Изменение фокуса у объекта.

### Параметры

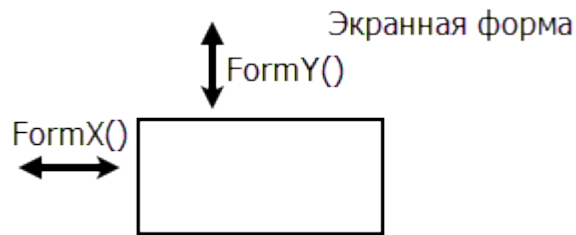
Параметр	Тип	Описание
hasFocus	bool	Флаг наличия фокуса на объекте

## 1.1.4.4.3.38.4.2. Функции

Функция	Описание
<a href="#">FormX</a>	Получение координаты X на форме
<a href="#">FormY</a>	Получение координаты Y на форме
<a href="#">ScreenX</a>	Получение оконной координаты X
<a href="#">ScreenY</a>	Получение оконной координаты Y

## 1.1.4.4.3.38.4.2.1. FormX

Получение координаты X на форме. Координаты измеряются в пикселях.



```
double FormX(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



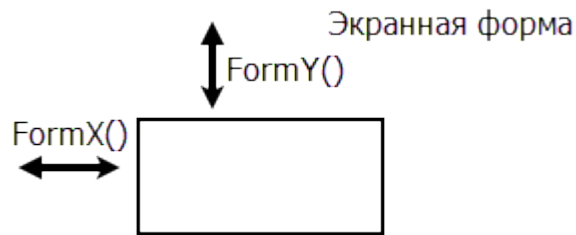
//Записать координаты X и Y объекта относительно экрана в поля ввода:

```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.4.2.2. FormY

Получение координаты Y на форме. Координаты измеряются в пикселях.



```
double FormY(Object.X,Object.Y)
```

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



//Записать координаты X и Y объекта относительно экрана в поля ввода:

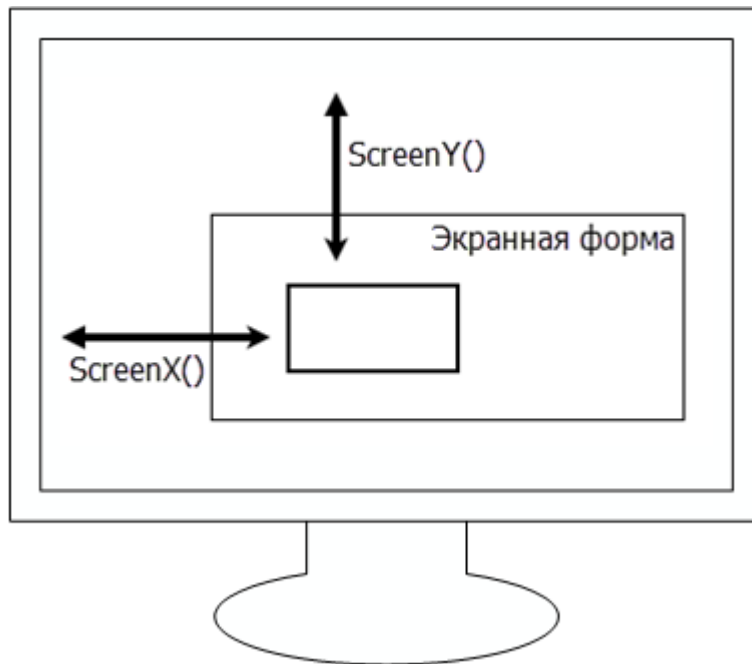
```
TextEdit1.Text = Object.FormX(Object.X, Object.Y);
```

```
TextEdit2.Text = Object.FormY(Object.X, Object.Y);
```



## 1.1.4.4.3.38.4.2.3. ScreenX

Получение оконной координаты X. Координаты измеряются в пикселях.



int4 ScreenX(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

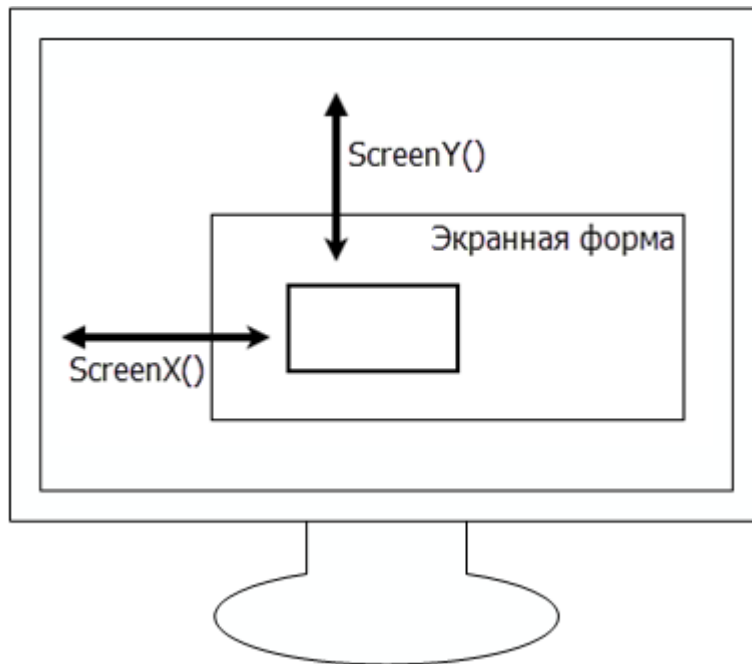
### Примеры



```
//Записать координату X объекта относительно экрана в поле ввода  
TextEdit1.Text = Object.ScreenX(Object.X, Object.Y);
```

## 1.1.4.4.3.38.4.2.4. ScreenY

Получение оконной координаты Y. Координаты измеряются в пикселях.



int4 ScreenY(Object.X,Object.Y)

### Параметры

Параметр	Тип	Описание
Object.X	float	Значение свойства X объекта
Object.Y	float	Значение свойства Y объекта

### Примеры



```
//Записать координату Y объекта относительно экрана в поле ввода:  
TextEdit2.Text = Object.ScreenY(Object.X, Object.Y);
```

## 1.1.4.4.3.38.4.3. Свойства

Свойство	Описание
<a href="#">Отображаемое имя</a>	Описание объекта
<a href="#">Кардинальное число</a>	Преобразует объект в массив и задает размер массива
<a href="#">X</a>	Позиция объекта по горизонтали
<a href="#">Y</a>	Позиция объекта по вертикали
<a href="#">Z-значение</a>	Z-значение
<a href="#">Угол поворота</a>	Угол поворота объекта
<a href="#">Масштаб</a>	Коэффициент масштабирования объекта
<a href="#">Отражение</a>	Отражение объекта
<a href="#">Видимость</a>	Видимость объекта на форме
<a href="#">Непрозрачность</a>	Непрозрачность объекта
<a href="#">Включено</a>	Активность объекта
<a href="#">Всплывающая подсказка</a>	Подсказка, всплывающая при наведении курсора на объект
<a href="#">Ширина</a>	Ширина объекта в пикселях
<a href="#">Высота</a>	Высота объекта в пикселях
<a href="#">Фокус ввода</a>	Переводит фокус на объект/снимает фокус с объекта
<a href="#">Радиус скругления</a>	Радиус скругления всех углов прямоугольника
<a href="#">Цвет пера</a>	Цвет внешней границы объекта
<a href="#">Стиль пера</a>	Стиль внешней границы объекта
<a href="#">Толщина пера</a>	Толщина внешней границы объекта
<a href="#">Цвет заливки</a>	Цвет внутренней заливки объекта
<a href="#">Стиль заливки</a>	Заливает объект одним из стилей заливки
<a href="#">IsVisible</a>	Позволяет изменить видимость кнопки
<a href="#">Текст заполнителя</a>	Текст, отображаемый в поле ввода по умолчанию

<a href="#">Шаблон поиска</a>	Меняет видимость панели поиска, меняя ее ширину
<a href="#">pPatternChanged</a>	Хранит введенный текст для поиска (шаблон)
<a href="#">Length</a>	Размер массива (количество элементов в массиве)
<a href="#">Index</a>	Индекс элемента в массиве

## 1.1.4.4.3.38.4.3.1. Отображаемое имя

Описание объекта (поля объекта). Позволяет указать произвольное (например, русскоязычное) имя для узнаваемости объекта (поля объекта) в проекте. Не мешает использованию базового англоязычного имени объекта (поля объекта) для обращения в коде.

## 1.1.4.4.3.38.4.3.2. Кардинальное число

Преобразует объект в массив и задает размер массива (количество элементов в массиве).

### Значение

Значение	Описание
1	Одиночный объект
>1	Массив соответствующей размерности

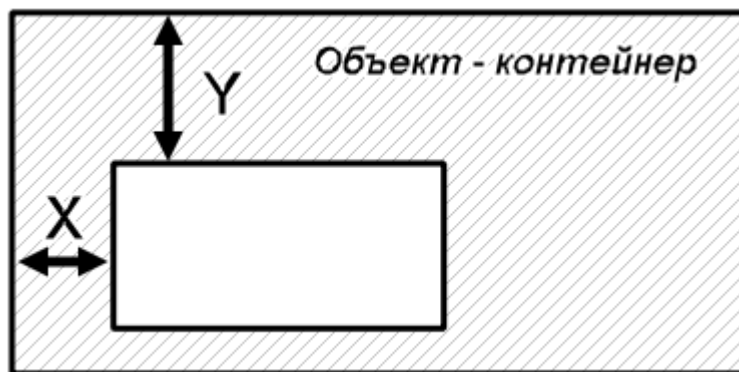
## 1.1.4.4.3.38.4.3.3. X

Позиция объекта по горизонтали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 X



## Примеры



```
//Установить позицию элемента (100; 0):  
Object.X = 100;
```



```
//Сдвинуть объект на 50 пикселей правее его текущей позиции:  
Object.X += 50;
```

## 1.1.4.4.3.38.4.3.4. Y

Позиция объекта по вертикали. Измеряется в пикселях, как сдвиг относительно левого верхнего угла родительского объекта (объекта-контейнера).

Ссылка на модель данных.



float8 Y



## Примеры



```
//Установить позицию элемента (0; 100):  
Object.Y = 100;
```



```
//Сдвинуть объект на 50 пикселей ниже его текущей позиции:  
Object.Y += 50;
```



# 1.1.4.4.3.3.38.4.3.5. Z-значение

Z-значение.



float8 ZValue

## 1.1.4.4.3.38.4.3.6. Угол поворота

Угол поворота объекта. Измеряется в градусах. При указании положительных значений объект поворачивается по часовой стрелке. При указании отрицательных значений объект поворачивается против часовой стрелки.



float8 Rotation

### Примеры



```
//Задать угол поворота объекта - 45 градусов:  
Object.Rotation = 45;
```



```
//Увеличить угол поворот объекта на 1 градус относительно текущего  
значения:  
Object.Rotation += 1;
```

## 1.1.4.4.3.38.4.3.7. Масштаб

Коэффициент масштабирования объекта.



float8 Scale

### Примеры



```
//Установить полуторный масштаб объекта:  
Object.Scale = 1.5;
```



```
//Увеличить текущий масштаб объекта в 3 раза:  
Object.Scale *= 3;
```



```
//Уменьшить текущий масштаб объекта в 2 раза:  
Object.Scale *= 0.5;
```

## 1.1.4.4.3.3.38.4.3.8. Отражение

Отражение объекта.



uint1 Flip

## 1.1.4.4.3.38.4.3.9. Видимость

Видимость объекта на форме.



bool Visible

### Значение

Значение	Описание
true	Объект виден
false	Объект скрыт

# 1.1.4.4.3.3.38.4.3.10. Непрозрачность

Непрозрачность.



float8 Opacity

# 1.1.4.4.3.38.4.3.11. Включено

Активность объекта.



bool Enabled

## Значение

Значение	Описание
true	Объект активен
false	Объект заблокирован

## Примеры



```
//Активировать объект в зависимости от выполнения условия:  
if (Signal1.value == Signal2.value){  
    Object.Enabled = true;  
}  
else{  
    Object.Enabled = false;  
}
```

## 1.1.4.4.3.38.4.3.12. Всплывающая подсказка

Подсказка, всплывающая при наведении курсора на объект.



string Tooltip

Запуск насоса

Подача команды MNA\_start = true

### Примеры




//Задать всплывающую подсказку для объекта:

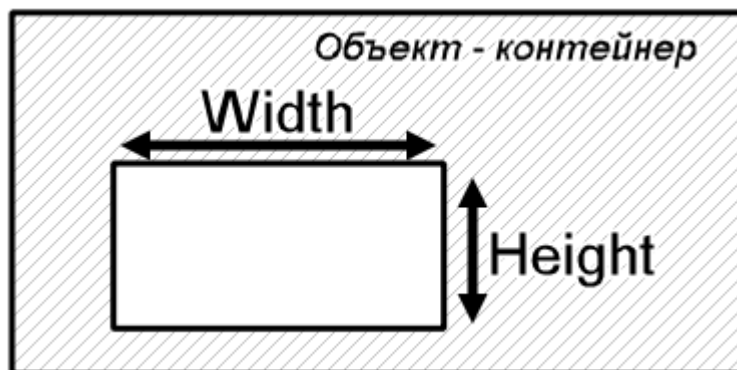
```
Object.Tooltip = "Подача команды MNA_start = true";
```



# 1.1.4.4.3.38.4.3.13. Ширина


Ширина объекта в пикселях.

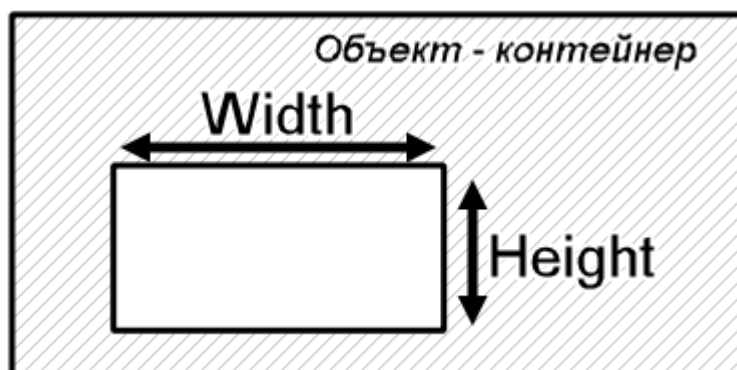
 float8 Width



# 1.1.4.4.3.38.4.3.14. Высота

Высота объекта в пикселях.

 float8 Height



# 1.1.4.4.3.38.4.3.15. Фокус ввода

Переводит фокус на объект/снимает фокус с объекта.



bool Focus

## Значение

Значение	Описание
true	Фокус на объекте
false	Фокус вне объекта

## Примеры



//Записать значение из поля ввода в сигнал при снятии фокуса с объекта:

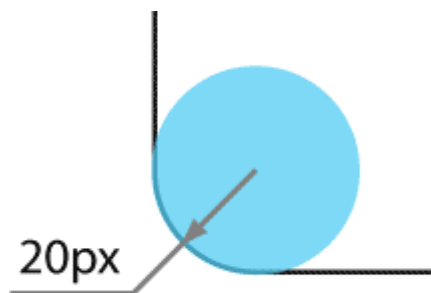
```
if (object.Focus == false)
    {
        pressure.Value = String.ToDouble(TextEdit_1.Text, 0);
    }
```

## 1.1.4.4.3.3.38.4.3.16. Радиус скругления

Радиус скругления всех углов прямоугольника. Измеряется в пикселях.



float8 RoundingRadius



## 1.1.4.4.3.3.38.4.3.17. Цвет пера

Цвет внешней границы объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 PenColor

### Примеры



```
//Задать черный цвет пера в шестнадцатеричной системе  
счисления:  
Object.PenColor = 0xff000000;
```



```
//Задать черный цвет пера в десятичной системе счисления:  
Object.PenColor = 4278190080;
```

## 1.1.4.4.3.38.4.3.18. Стиль пера

Стиль внешней границы объекта.



uint2 PenStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет пера	0	
Сплошная линия	1	
Штриховая линия	2	
Пунктирная линия	3	
Штрих-пунктирная линия	4	
Штрих-пунктирная линия с двумя точками	5	

### Примеры



//Применить стиль границы (короткий пунктир) для объекта:  
Object.PenStyle = 3;

## 1.1.4.4.3.38.4.3.19. Толщина пера

Толщина внешней границы объекта. Измеряется в пикселях.



float8 PenWidth

### Примеры



```
//Задать толщину пера 5 пикселей:  
Object.PenWidth = 5;
```

## 1.1.4.4.3.38.4.3.20. Цвет заливки

Цвет внутренней заливки объекта. Указывается через системное окно Выбор цвета. После выбора автоматически переводится в шестнадцатеричную систему счисления.



uint4 BrushColor

### Примеры



//Залить объект красным цветом (цвет в шестнадцатеричной системе счисления):  
Object.BrushColor = 0xffff0000;



//Залить объект красным цветом (цвет в десятичной системе счисления):  
Object.BrushColor = 4294901760;





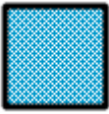

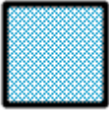

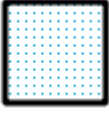
## 1.1.4.4.3.38.4.3.21. Стиль заливки

Заливает объект одним из стилей заливки.



uint2 BrushStyle

### Значение

Значение в дизайнера	Значение в коде	Вид границы
Нет заливки	0	
Сплошная заливка	1	
Точечная 1	2	
Точечная 2	3	
Точечная 3	4	
Точечная 4	5	
Точечная 5	6	
Точечная 6	7	
Точечная 7	8	

Горизонтальные линии	9	
----------------------	---	---

## Примеры



//Залить объект сплошным цветом:  
Object.BrushStyle = 1;

## 1.1.4.4.3.3.38.4.3.22. IsVisible

Позволяет изменить видимость кнопки.



bool IsVisible

### Значение

Значение	Описание
true	Кнопка отображается и выглядит как обычно
false	Кнопка не отображается, так как ее ширина становится нулевой, благодаря чему выравниваются остальные элементы панели

## 1.1.4.4.3.3.38.4.3.23. Текст заполнителя

Текст, отображаемый в поле ввода по умолчанию.



string PlaceholderText

## 1.1.4.4.3.38.4.3.24. Шаблон поиска

Хранит введенный текст для поиска (шаблон).



string Pattern

- › Изменяется при нажатии клавиши Enter или кнопки запуска процесса поиска.
- › Сбрасывается при нажатии клавиши Esc или кнопки очистки результатов поиска.

## 1.1.4.4.3.38.4.3.25. pPatternChanged

Ссылка на команду, выполняемую при изменении шаблона в Pattern.

## 1.1.4.4.3.38.4.3.26. Length

Размер массива (количество элементов в массиве).



int8 Length



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



```
//Записать в текстовое поле длину массива:  
TextEdit_1.Text = String.ToString (SW_1.Length);
```

## 1.1.4.4.3.38.4.3.27. Index

Индекс элемента в массиве. Позволяет отследить у какого элемента сработало то или иное событие.



int8 Index



Есть только у массивов и доступно только для чтения в режиме рантайма.

### Примеры



//Записать в текстовое поле индекс элемента массива, у которого сработало событие изменения значения:  
`TextEdit_2.Text = String.ToString(Index);`



//Записать в текстовое поле текущее значение элемента массива, у которого сработало событие изменения значения:  
`TextEdit_3.Text = String.ToString(penwidth[Index]);`



//В зависимости от индекса элемента в массиве вывести в текстовое поле запись о включении соответствующего режима работы задвижки:  
`if (SW_1.Mode.Index == 0) {  
 TextEdit_1.Text = "Включен режим 1";  
else  
 {TextEdit_1.Text = "Включен режим 2";}`

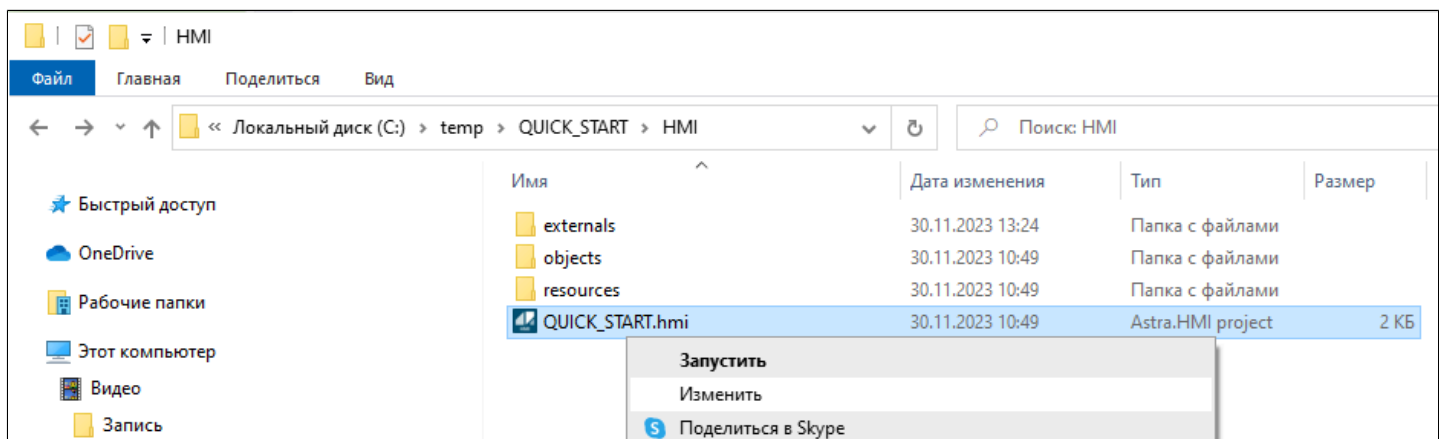


## 1.1.4.5. Разработка проекта

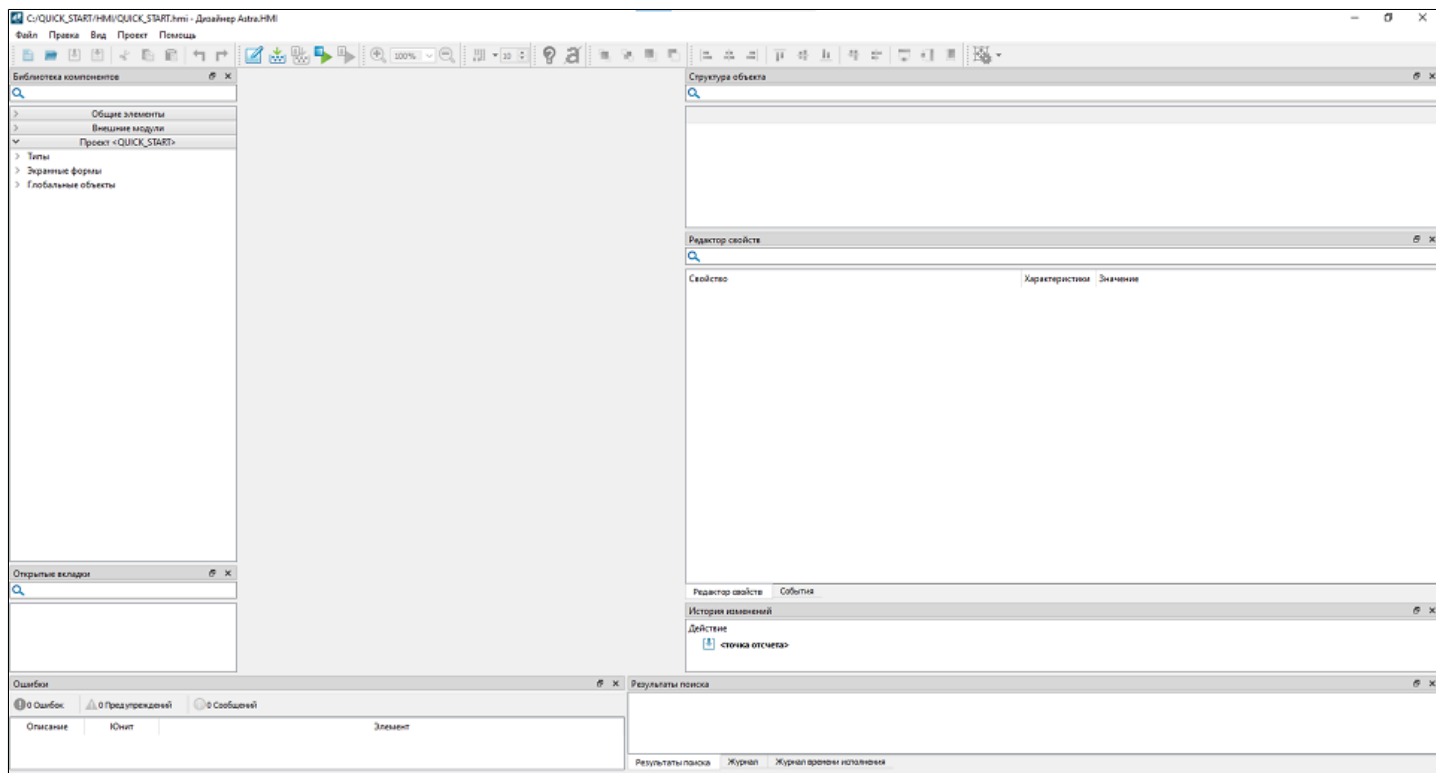
При выполнении [экспорта конфигурационных файлов](#) в папке с проектом Astra.IDE автоматически создается папка "HMI" с проектом Astra.HMI.

В папке "externals" расположены папки со стандартными библиотеками, подключенными к проекту Astra.HMI

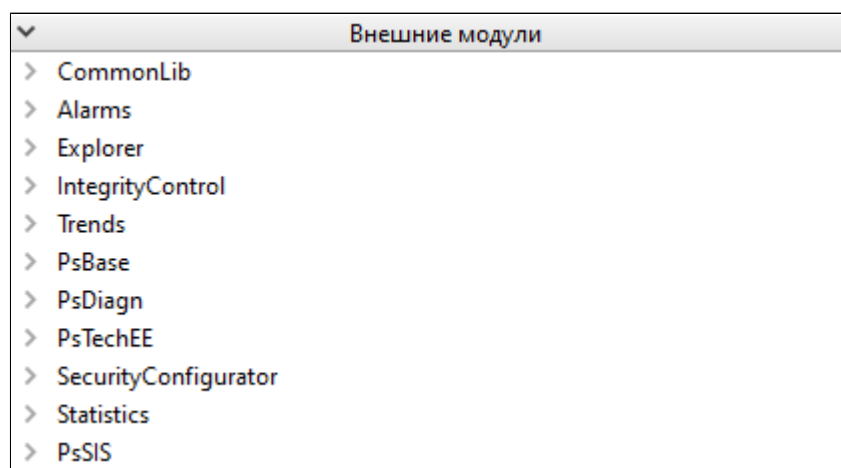
1. Запустите проект "QUICK\_START" в дизайнера Astra.HMI. Для запуска проекта перейдите в папку с проектом "QUICK\_START" → "HMI", нажмите правой кнопкой мыши по файлу "QUICK\_START", и выберите команду "Изменить".



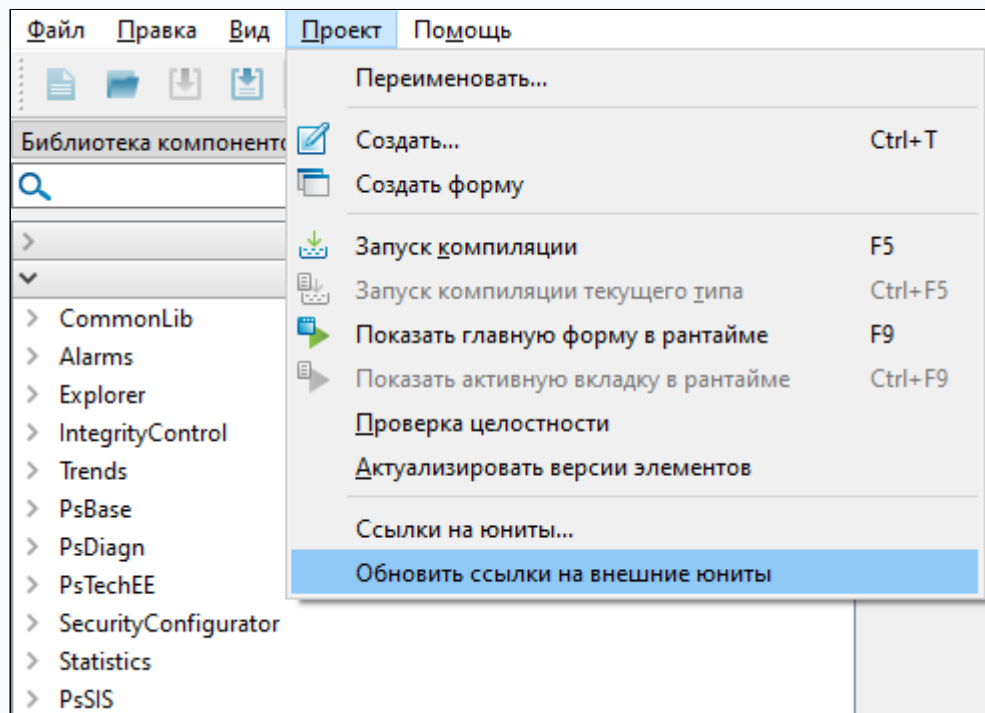
2. В дизайнера Astra.HMI откроется проект "QUICK\_START".



3. Во вкладке "Внешние модули" отображаются все библиотеки, подключенные к проекту.



Если файл с внешней библиотекой расположен в папке "externals", но на вкладке внешние модули не отображается, в главном меню выберите "Проект" → "Обновить ссылки на внешние юниты".



## 1.1.4.5.1. Источники данных

Источники данных в проекте создаются автоматически при выполнении [экспорта конфигурационных файлов](#).

1. Откройте вкладку "Глобальные объекты", в которой находятся три объекта:

- Connections – элемент, содержащий источники данных.
- Settings – элемент, содержащий настройки проекта.
- Security – элемент, содержащий права доступа.

2. Откройте глобальный объект "Connections" двойным кликом левой мыши. В окне "Структура объекта" содержится источник данных "MAIN\_SOURCE".

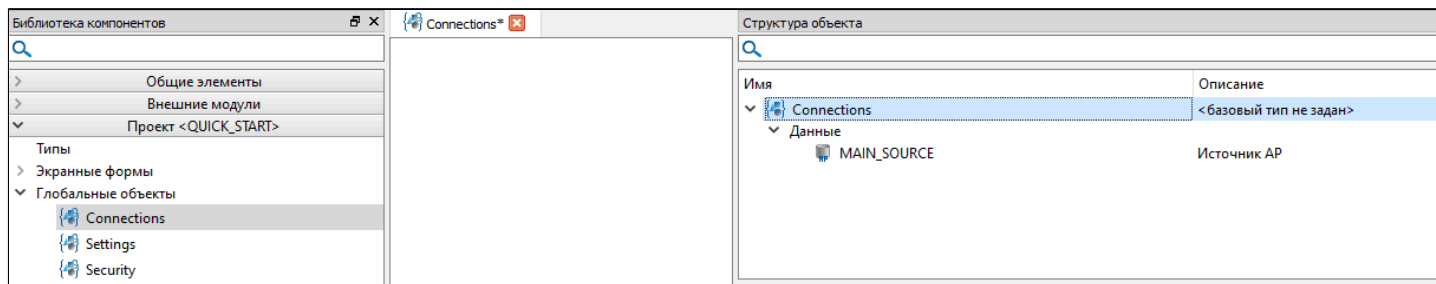
The screenshot displays the software interface with three main panels:

- Библиотека компонентов (Component Library):** Shows a tree view with 'Глобальные объекты' (Global Objects) expanded to show 'Connections', 'Settings', and 'Security'.
- Структура объекта (Object Structure):** Shows a tree view where 'Connections' is expanded to 'Данные' (Data), which contains the 'MAIN\_SOURCE' object.
- Редактор свойств (Property Editor):** Displays a table of properties for the 'MAIN\_SOURCE' object.


Свойство	Харак	Значение
Отображаемое имя		MAIN_SOURCE
Кардинальное число		1
Родительский источник	R W →	< не определено >
Источник учетных данных	R W →	< не определено >
Таймаут операций с БД	R ⚡ ✓	10000
Таймаут установления соединения с БД	R ⚡ ✓	10000
Максимальный размер пула буферов	R ⚡ ✓	10000000
<b>Хост</b>	R W	
Начальное значение	✓	127.0.0.1
Вычисляемое значение		< не определено >
Порт	R W	4388
Порт истории	R W	4950
Путь	R W	
Активность	R W	true
Период переподписки	R W	1000
Имя клиента	R W	HMI
Идентификатор клиента	R W	HMI

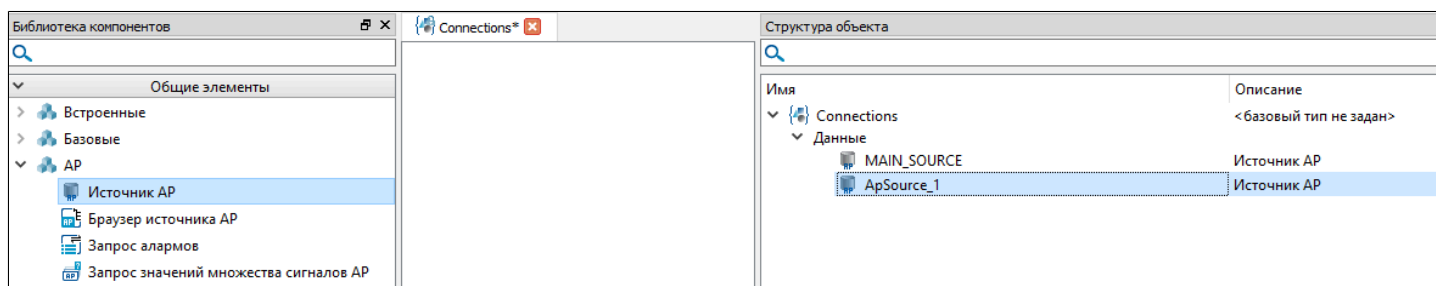
Чтобы настроить источник данных для серверов в/в (для основного сервера "SERVER1": AP\_SERVER1\_SOURCE\_LINE1 и AP\_SERVER1\_SOURCE\_LINE2; и для резервного сервера "SERVER2": AP\_SERVER2\_SOURCE\_LINE1 и AP\_SERVER2\_SOURCE\_LINE2), выполните следующие действия:

1. Откройте глобальный объект "Connections" двойным кликом левой кнопки мыши.



2. Добавьте на рабочую область "Connections" элемент "Источник AP" из вкладки "Общие элементы".

 Чтобы добавить элемент, перетащите его из панели элементов в рабочую область.



3. Задайте имя элемента "AP\_SERVER1\_SOURCE\_LINE1". Выберите источник в окне "Структура объектов" и задайте следующие свойства в редакторе свойств:

Свойство	Значение
Таймаут операций с БД	10000
Таймаут установления соединения с БД	10000
Максимальный размер пула буферов	10000000
Порт истории	4950
Активность	true
Период переподписки	1000
Хост (Начальное значение)	192.168.1.201

Структура объекта

Имя	Описание
Connections	<базовый тип не задан>
Данные	
MAIN_SOURCE	Источник AP
AP_SERVER1_SOURCE_LINE1	Источник AP

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		AP_SERVER1_SOURCE_LINE1
Кардинальное число		1
Родительский источник	R W → 0	<не определено>
Источник учетных данных	R W → 0	<не определено>
Таймаут операций с БД	R ≤ ✓	10000
Таймаут установления соединения с БД	R ≤ ✓	10000
Максимальный размер пула буферов	R ≤ ✓	10000000
Общее соединение	R ≤	<не определено>
<b>Хост</b>	R W	
Начальное значение	✓	192.168.1.201
Вычисляемое значение		<не определено>
Порт	R W	4388
Порт истории	R W	4950
Путь	R W	
Активность	R W	true
Период переподписки	R W	1000
Имя клиента	R W	HMI
Идентификатор клиента	R W	HMI

4. Аналогичным образом добавьте источник AP "AP\_SERVER1\_SOURCE\_LINE2", который будет обладать следующими свойствами:

Свойство	Значение
Таймаут операций с БД	10000
Таймаут установления соединения с БД	10000
Максимальный размер пула буферов	10000000
Порт истории	4950

Активность	true
Период переподписки	1000
Хост (Начальное значение)	192.168.2.201

Структура объекта 🔍 ✕

🔍

Имя	Описание
<ul style="list-style-type: none"> <li>▼  Connections &lt;базовый тип не задан&gt;</li> <li>▼ Данные <ul style="list-style-type: none"> <li> MAIN_SOURCE Источник AP</li> <li> AP_SERVER1_SOURCE_LINE1 Источник AP</li> <li style="background-color: #e0e0e0;"> AP_SERVER1_SOURCE_LINE2 Источник AP</li> </ul> </li> </ul>	

Редактор свойств 🔍 ✕

🔍

Свойство	Характеристики	Значение
Отображаемое имя		AP_SERVER1_SOURCE_LINE2
Кардинальное число		1
Родительский источник	R W → 0	<не определено>
Источник учетных данных	R W → 0	<не определено>
Таймаут операций с БД	R ≤ ✓	10000
Таймаут установления соединения с БД	R ≤ ✓	10000
Максимальный размер пула буферов	R ≤ ✓	10000000
Общее соединение	R ≤	<не определено>
▼ <b>Хост</b>	R W	
Начальное значение	✓	192.168.2.201
Вычисляемое значение		<не определено>
>  Порт	R W	4388
>  Порт истории	R W	4950
>  Путь	R W	
>  Активность	R W	true
>  Период переподписки	R W	1000
>  Имя клиента	R W	HMI
>  Идентификатор клиента	R W	HMI

5. Аналогичным образом добавьте источник данных для резервного сервера "SERVER2": AP\_SERVER2\_SOURCE\_LINE1 и AP\_SERVER2\_SOURCE\_LINE2, задав свойству "Хост" значения 192.168.1.202 и 192.168.2.202 соответственно.



## 1.1.4.5.2. Главная экранная форма

Главная экранная форма открывается при запуске проекта в Runtime.

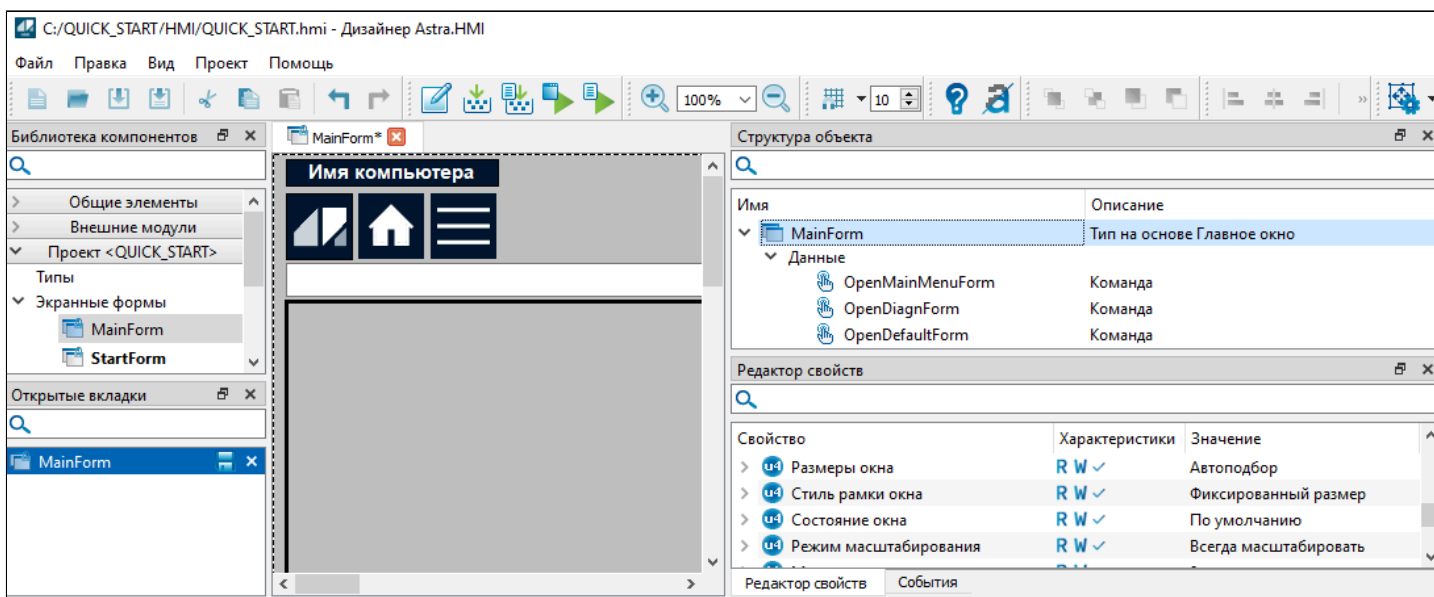
1. Главная экранная форма в проекте создается автоматически при выполнении [экспорта конфигурационных файлов](#). На главную экранную форму добавлено три команды:

- › OpenMainMenuForm – команда для открытия главного меню;
- › OpenDiagnForm – команда для открытия формы диагностики;
- › OpenDefaultForm – команда для открытия формы по умолчанию.

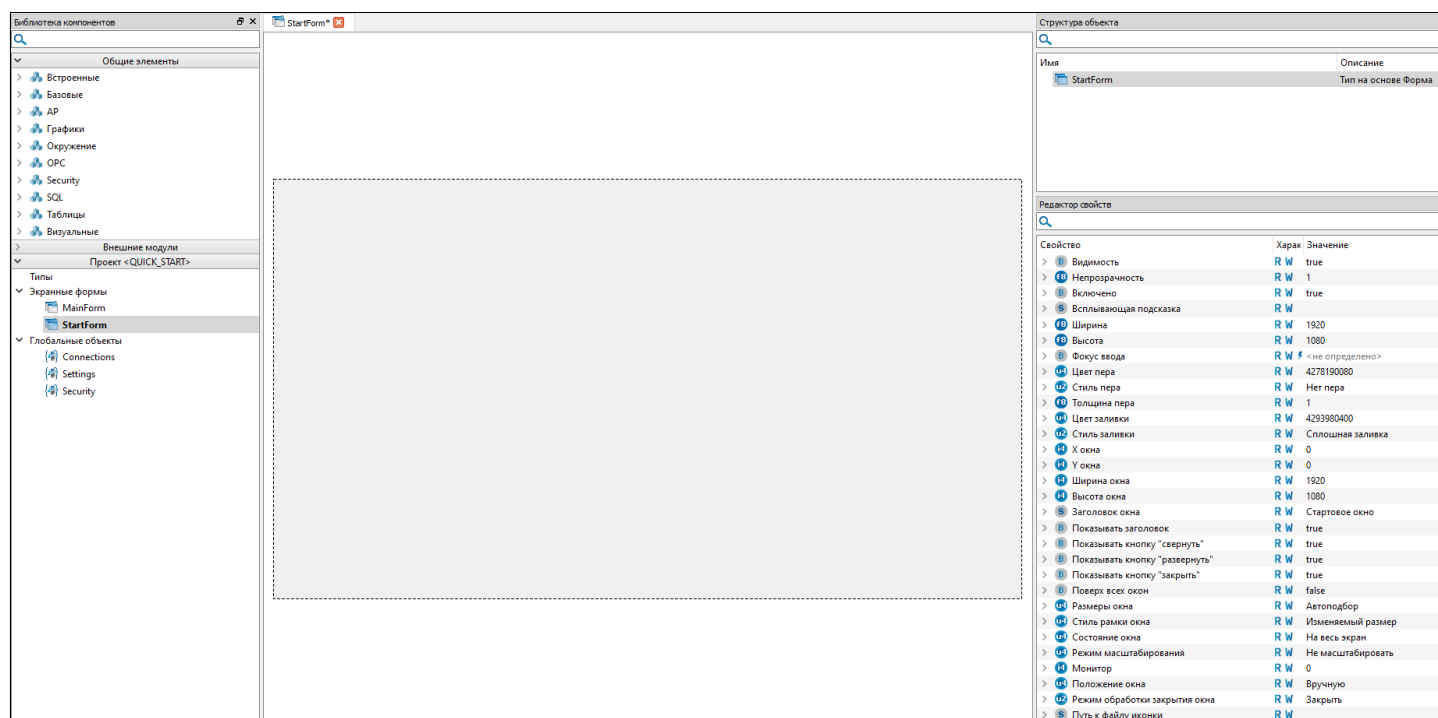


Чтобы главная экранная форма открывалась на весь экран, откройте редактор свойств экранной формы "MainForm" и задайте следующие свойства:

- › Размеры окна – Автоподбор;
- › Стиль рамки окна – фиксированный размер;
- › Состояние окна – по умолчанию;
- › Режим масштабирования – всегда масштабировать.



2. Для открытия нескольких главных экранных форм (используется при работе в многомониторной системе) при выполнении [экспорта конфигурационных файлов](#) автоматически создается экранная форма "StartForm".



3. Для экранной формы "StartForm" добавлены команды и обработчики событий для открытия до четырех главных экранных форм при работе в многомониторном режиме.

Структура объекта

Имя	Описание
StartForm	Тип на основе Форма
Данные	
OpenForm1	Команда
OpenForm2	Команда
OpenForm3	Команда
OpenForm4	Команда

События

Имя	Характеристики	Обработчик
MouseMove		
MouseClick		
MouseRightClick		
MouseDoubleClick		
MousePress		
MouseRelease		
MouseEnter		
MouseLeave		
TouchMove		
TouchPress		
TouchRelease		
FocusChanged		
CloseRequested		
AboutToOpen		
Opened		
Handler_3		Выполнить код: OpenForm1.Invoke();
Handler_4		Выполнить код: OpenForm2.Invoke();
Handler_5		Выполнить код: OpenForm3.Invoke();
Handler_6		Выполнить код: OpenForm4.Invoke();

4. Каждый обработчик события открывает Главную экранную форму в новом окне. При использовании одномониторной системы оставьте только один обработчик событий, удалив оставшиеся через контекстное меню.

Структура объекта

Имя Описание

- StartForm Тип на основе Форма
  - Данные
    - OpenForm1 Команда
    - OpenForm2 Команда
    - OpenForm3 Команда
    - OpenForm4 Команда

События

Имя Характеристики Обработчик

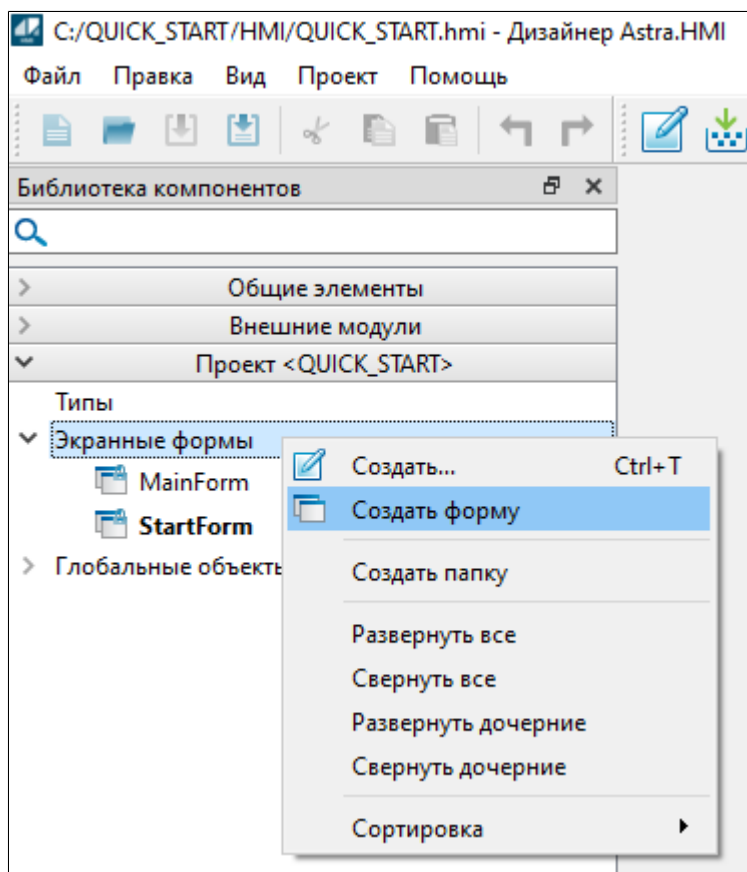
- MouseMove
- MouseClicked
- MouseRightClick
- MouseDoubleClick
- MousePress
- MouseRelease
- MouseEnter
- MouseLeave
- TouchMove
- TouchPress
- TouchRelease
- FocusChanged
- CloseRequested
- AboutToOpen
- Opened
  - Handler\_3 Выполнить код: OpenForm1.Invoke();
  - Handler\_4 Выполнить код: OpenForm2.Invoke();
  - Handler\_5 Выполнить код: OpenForm3.Invoke();
  - Handler\_6 Выполнить код: OpenForm4.Invoke();
- Closed
- WindowActivationCh
- WindowSizeChanged
- WindowPosChanged
- KeyPress
- KeyRelease

Контекстное меню:

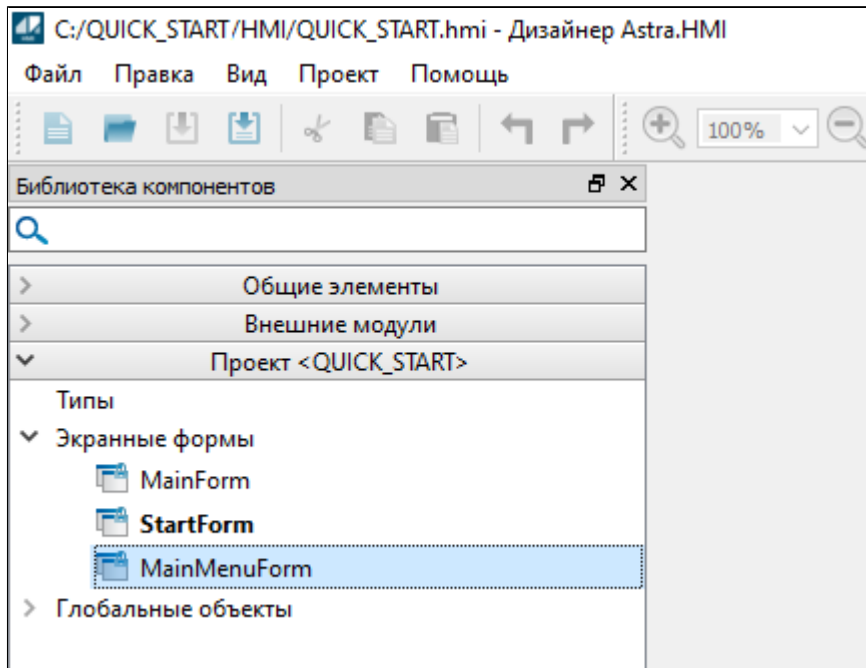
- Вырезать Ctrl+X
- Копировать Ctrl+C
- Удалить Del
- Развернуть все
- Свернуть все
- Сортировка

### 1.1.4.5.3. Создание экранной формы главного меню

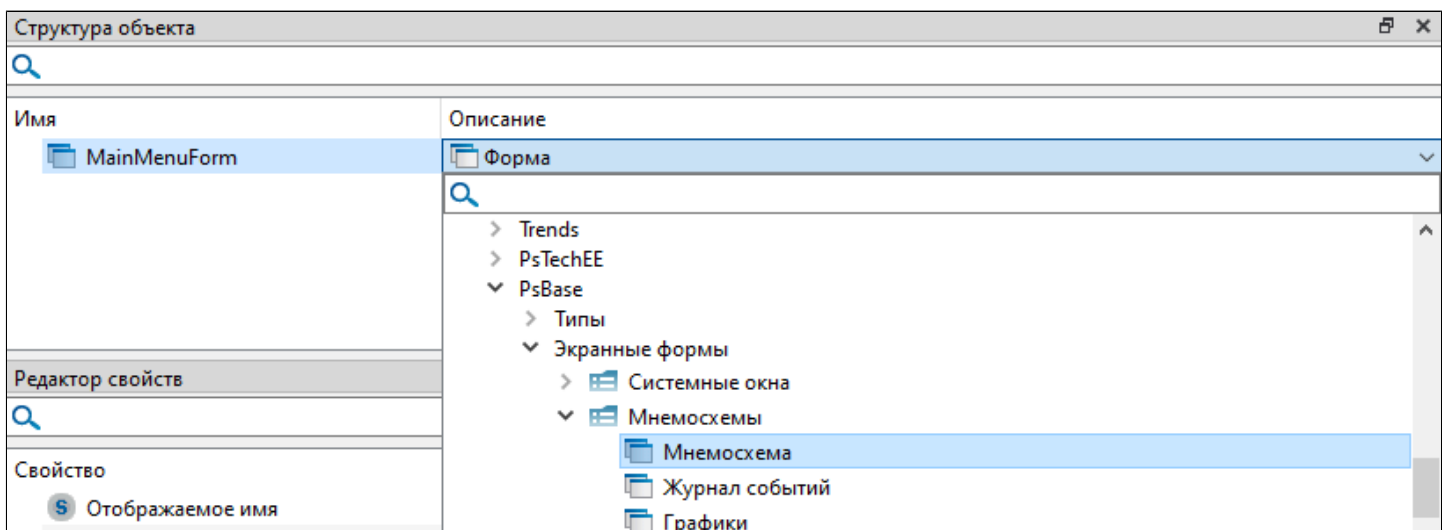
1. Кликните правой кнопкой мыши по вкладке "Экранные формы" и в контекстном меню выберите команду "Создать форму".



2. Во вкладке "Экранные формы" переименуйте созданную форму в "MainMenuForm".



3. Измените тип экранной формы двойным кликом левой кнопки мыши по полю "Описание". В выпадающем списке выберите "Внешние модули" → "PsBase" → "Экранные формы" → "Мнемосхемы" → "Мнемосхема" и нажмите клавишу "Enter".



4. В редакторе свойств формы задайте следующие свойства:

Структура объекта

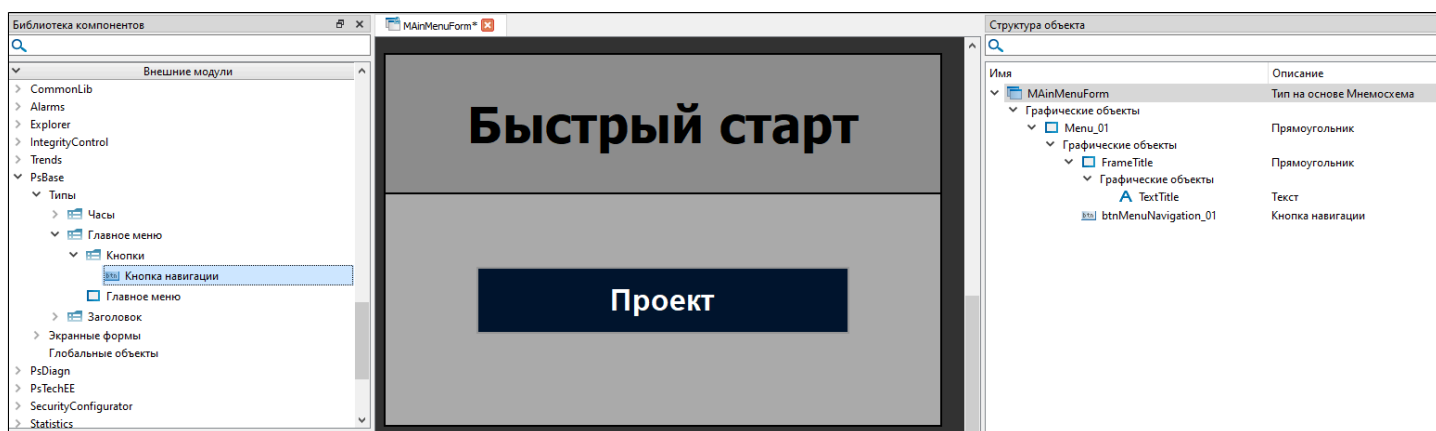
Имя: MainMenuForm | Описание: Тип на основе Мнемосхема

---

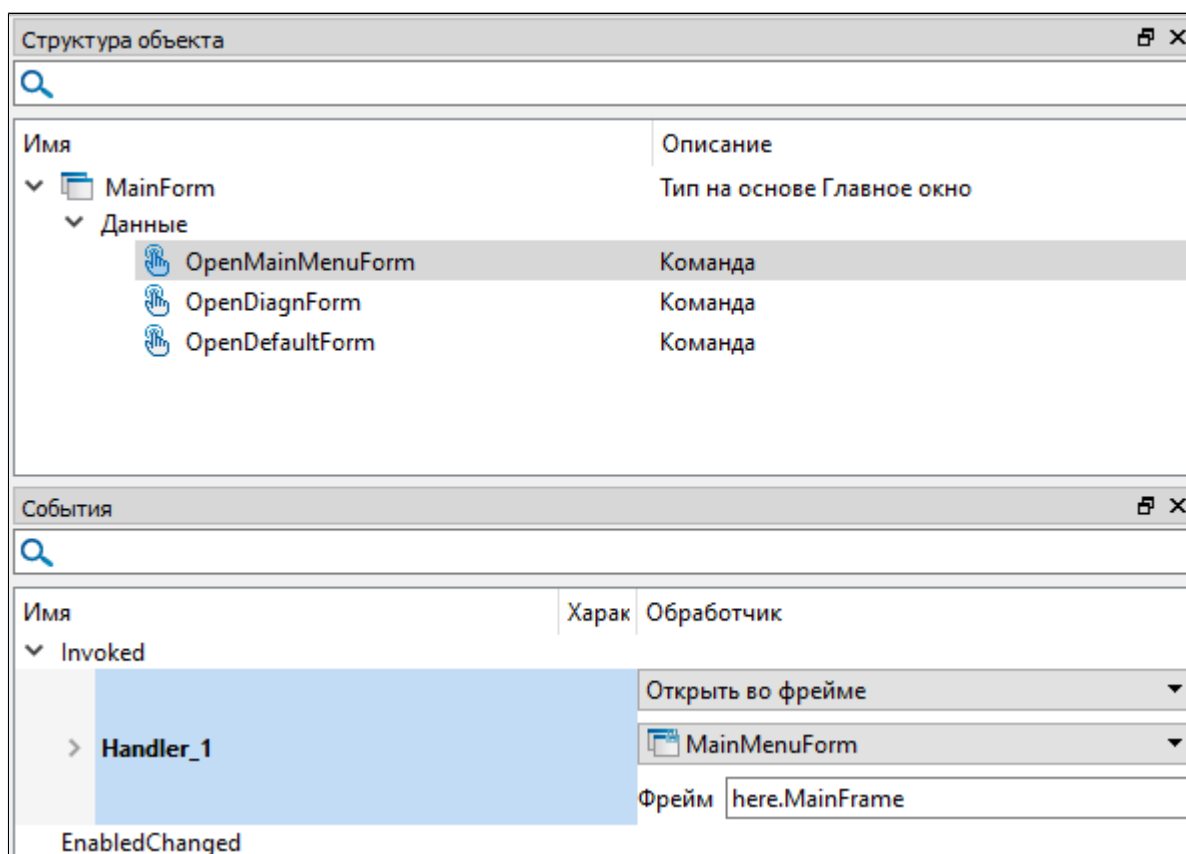
Редактор свойств

Свойство	Характеристи	Значение
> Включено	R W ✓	true
> Всплывающая подсказка	R W ✓	
> f8 Ширина	R W ✓	1900
> f8 Высота	R W ✓	960
> В Фокус ввода	R W ⚡	< не определено >
> u4 Цвет пера	R W ✓	4278190080
> u2 Стиль пера	R W ✓	Нет пера
> f8 Толщина пера	R W ✓	1
> u4 Цвет заливки	R W ✓	0xff323232
> u2 Стиль заливки	R W ✓	Сплошная заливка
> i4 X окна	R W ✓	0
> i4 Y окна	R W ✓	0
> i4 Ширина окна	R W ✓	1920
> i4 Высота окна	R W ✓	1080
> S Заголовок окна	R W ✓	Главное меню
> В Показывать заголовок	R W ✓	true
> В Показывать кнопку "свернуть"	R W ✓	true
> В Показывать кнопку "развернуть"	R W ✓	true
> В Показывать кнопку "закреть"	R W ✓	true
> В Поверх всех окон	R W ✓	false
> u4 Размеры окна	R W ✓	Вручную
> u4 Стиль рамки окна	R W ✓	Изменяемый размер
> u4 Состояние окна	R W ✓	По умолчанию
> u4 Режим масштабирования	R W ✓	Не масштабировать
> i4 Монитор	R W ✓	0
> u4 Положение окна	R W ✓	Вручную
> u2 Режим обработки закрытия окна	R W ✓	Закреть
> S Путь к файлу иконки	R W ✓	
Команда Назад	R ⌨ →	< не определено >
Команда Вперед	R ⌨ →	< не определено >
Команда На уровень выше	R ⌨ →	< не определено >
> В Кнопка Назад доступна	R W ⚡ ✓	false
> В Кнопка Вперед доступна	R W ⚡ ✓	false
> В Кнопка На уровень выше доступна	R W ⚡	false
Ссылка на главную форму	R ⌨ →	< не определено >

5. С помощью стандартных элементов "Прямоугольник", "Текст" и библиотечного типа кнопки "Кнопка навигации" из библиотеки "PsBase" создайте меню навигации.



6. Перейдите на экранную форму "MainForm". С помощью контекстного меню добавьте обработчик события "Invoked" для команды "OpenMainMenuForm".





7. Раскройте обработчик и задайте свойству "Ссылка на главную форму" значение "here". Нажмите клавишу Enter. Задайте для обработчика следующие свойства:

<b>Свойство</b>	<b>Значение</b>
Команда назад доступна	false
Команда вперед доступна	false
Команда на уровень выше доступна	false

Структура объекта

Имя Описание

- MainForm
  - Данные
    - OpenMainMenuForm Команда
    - OpenDiagnForm Команда
    - OpenDefaultForm Команда

---

События

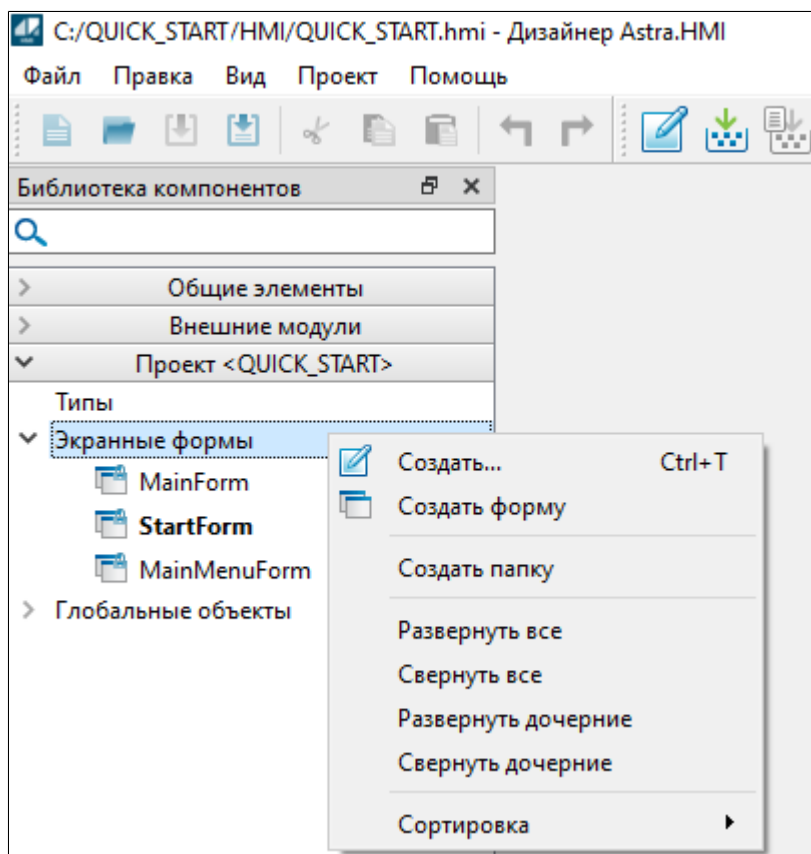
Имя Харак Обработчик

Имя	Харак	Обработчик
> u2 Стилль пера	R W	Нет пера
> f8 Толщина пера	R W	1
> u4 Цвет заливки	R W	0xff323232
> u2 Стилль заливки	R W	Сплошная заливка
> i4 X окна	R W	0
> i4 Y окна	R W	0
> i4 Ширина окна	R W	1920
> i4 Высота окна	R W	1080
> S Заголовок окна	R W	Главное меню
> B Показывать заголовок	R W	true
> B Показывать кнопку "свернуть"	R W	true
> B Показывать кнопку "развернуть"	R W	true
> B Показывать кнопку "закреть"	R W	true
> B Поверх всех окон	R W	false
> u4 Размеры окна	R W	Вручную
> u4 Стилль рамки окна	R W	Изменяемый размер
> u4 Состояние окна	R W	По умолчанию
> u4 Режим масштабирования	R W	Не масштабировать
> i4 Монитор	R W	0
> u4 Положение окна	R W	Вручную
> u2 Режим обработки закрытия окна	R W	Закреть
> S Путь к файлу иконки	R W	
Команда Назад	R ↵ →	< не определено >
Команда Вперед	R ↵ →	< не определено >
Команда На уровень выше	R ↵ →	< не определено >
> B Кнопка Назад доступна	R W ⚡	false
> B Кнопка Вперед доступна	R W ⚡	false
> B Кнопка На уровень выше досту...	R W ⚡	false
Ссылка на главную форму	R ↵ →	here

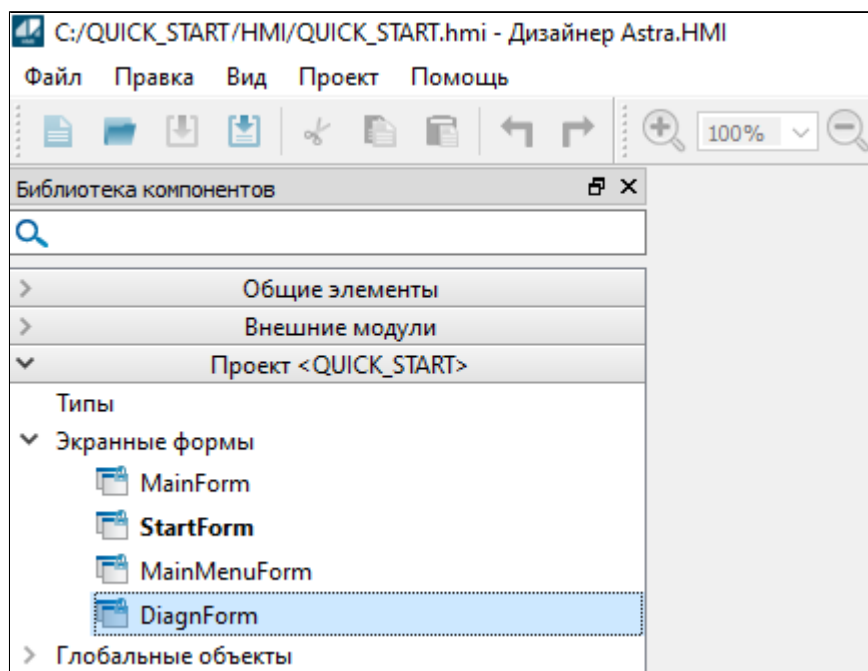
EnabledChanged

## 1.1.4.5.4. Создание экранной формы диагностики

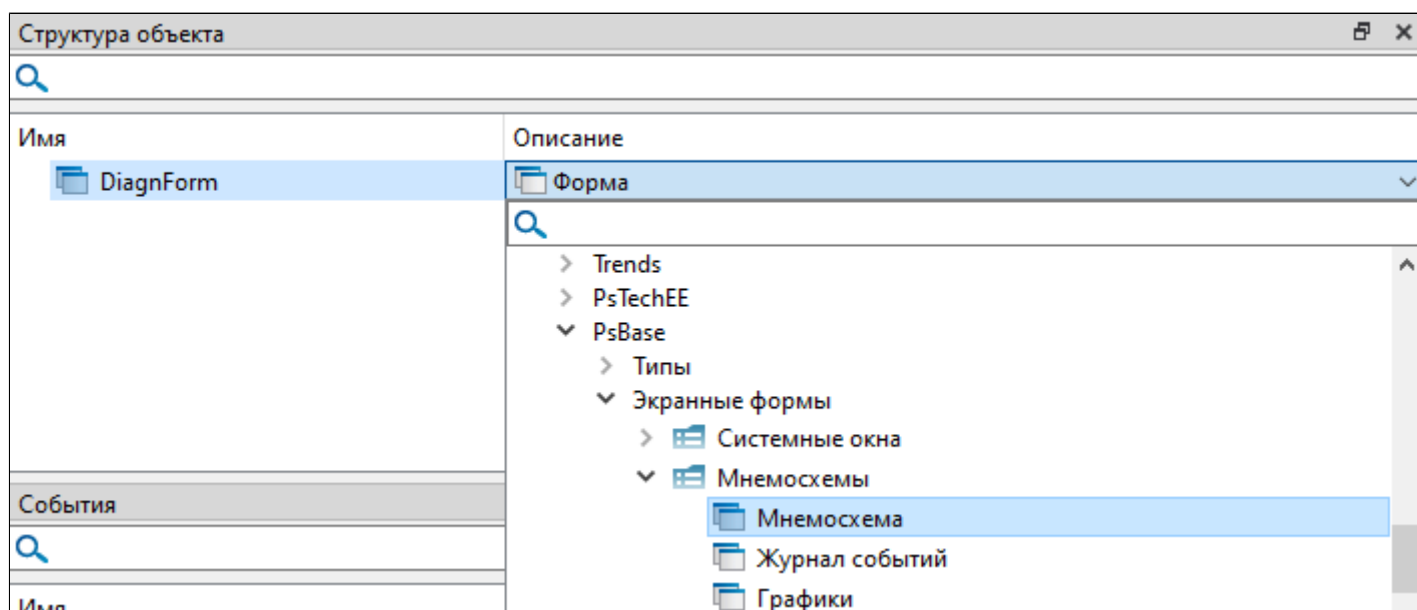
1. Кликните правой кнопкой мыши по вкладке "Экранные формы" и в контекстном меню выберите команду "Создать форму".



2. Во вкладке "Экранные формы" переименуйте созданную форму в "DiagnForm".



3. Измените тип экранной формы двойным кликом левой кнопки мыши по полю "Описание". В выпадающем списке выберите "Внешние модули" → "PsBase" → "Экранные формы" → "Мнемосхемы" → "Мнемосхема" и нажмите клавишу "Enter".



4. В редакторе свойств формы задайте следующие свойства:

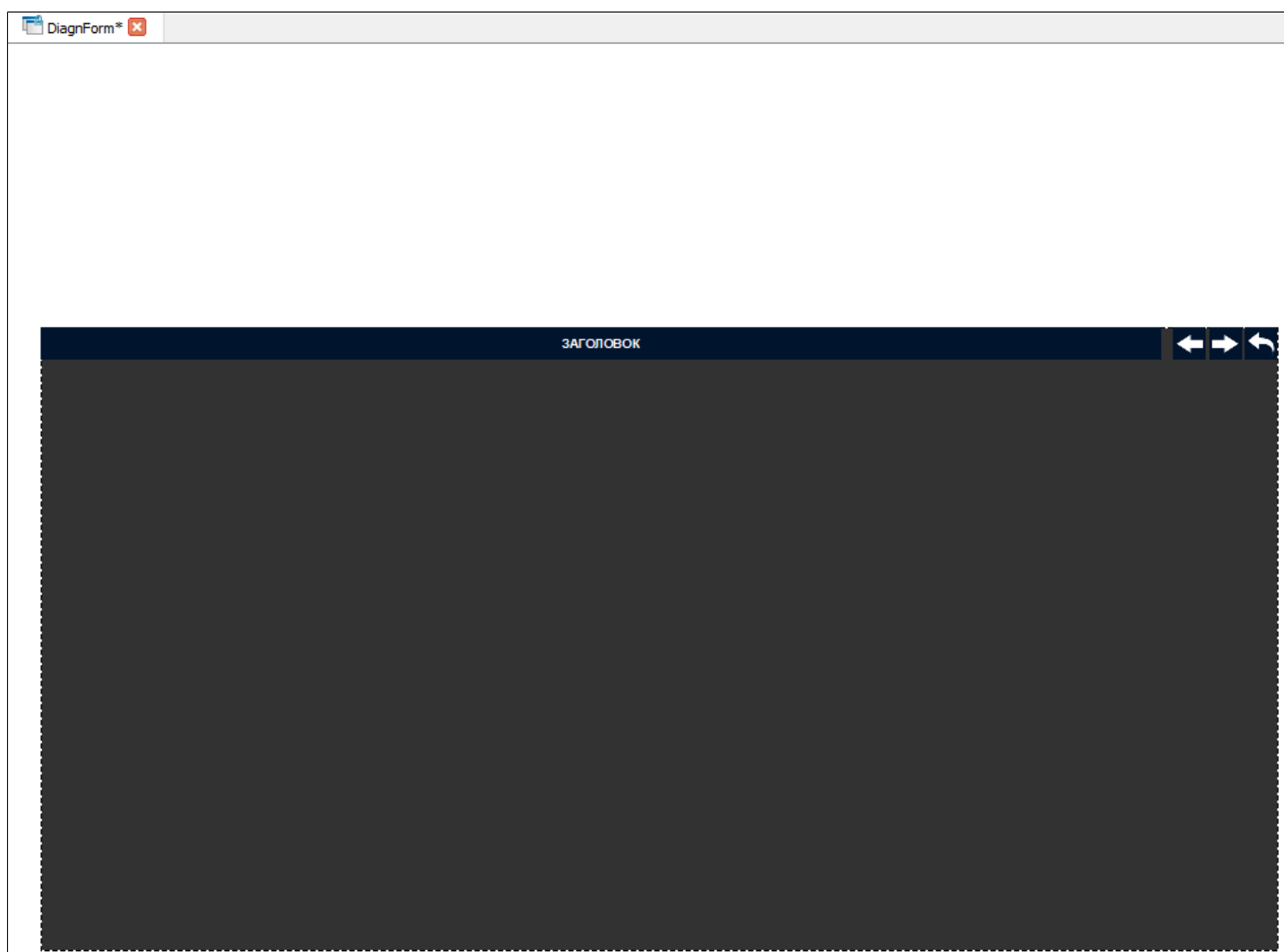
✖
Структура объекта

Имя	Описание
DiagnForm	Тип на основе Мнемосхема

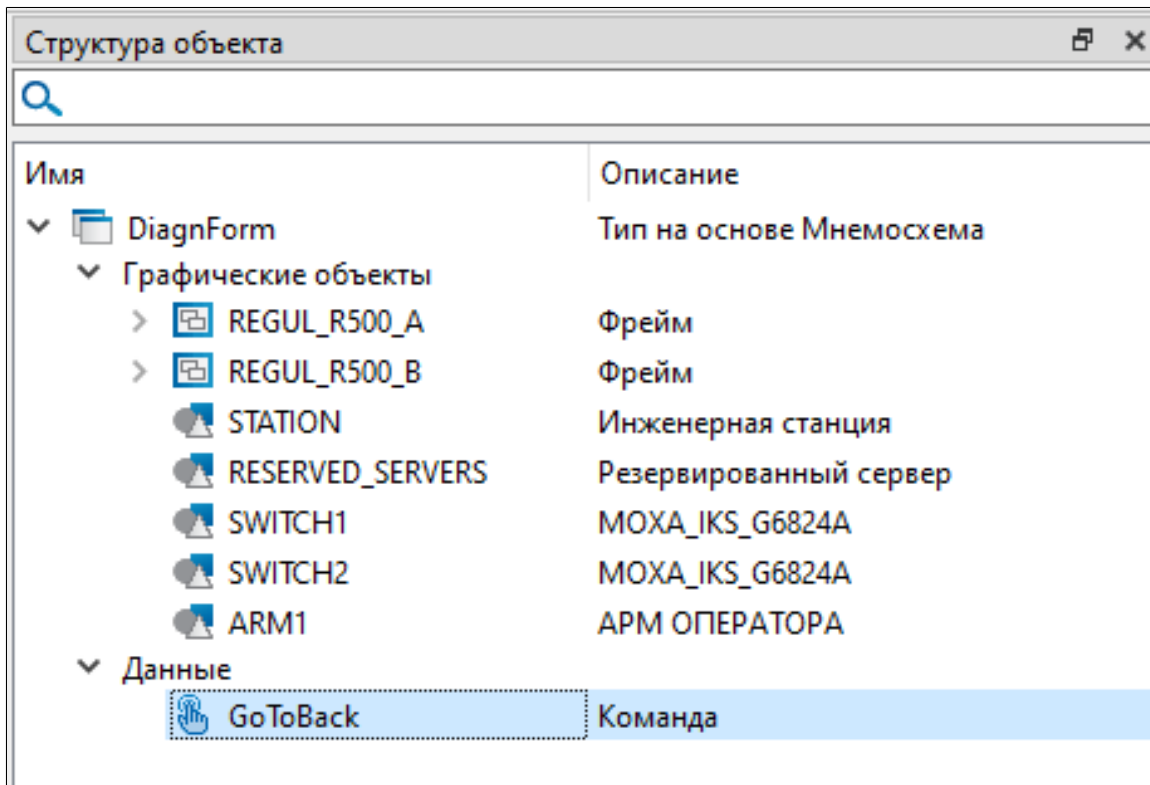
✖
Редактор свойств

Свойство	Характеристи	Значение
>  Всплывающая подсказка	R W ✓	
>  Ширина	R W ✓	1900
>  Высота	R W ✓	960
>  Фокус ввода	R W ✗	< не определено >
>  Цвет пера	R W ✓	4278190080
>  Стиль пера	R W ✓	Нет пера
>  Толщина пера	R W ✓	1
>  Цвет заливки	R W ✓	0xff323232
>  Стиль заливки	R W ✓	Сплошная заливка
>  X окна	R W ✓	0
>  Y окна	R W ✓	0
>  Ширина окна	R W ✓	1920
>  Высота окна	R W ✓	1080
>  Заголовок окна	R W ✓	Экран диагностики
>  Показывать заголовок	R W ✓	true
>  Показывать кнопку "свернуть"	R W ✓	true
>  Показывать кнопку "развернуть"	R W ✓	true
>  Показывать кнопку "закрыть"	R W ✓	true
>  Поверх всех окон	R W ✓	false
>  Размеры окна	R W ✓	Вручную
>  Стиль рамки окна	R W ✓	Изменяемый размер
>  Состояние окна	R W ✓	По умолчанию
>  Режим масштабирования	R W ✓	Не масштабировать
>  Монитор	R W ✓	0
>  Положение окна	R W ✓	Вручную
>  Режим обработки закрытия окна	R W ✓	Закрыть
>  Путь к файлу иконки	R W ✓	
Команда Назад	R ⚡ →	< не определено >
Команда Вперед	R ⚡ →	< не определено >
Команда На уровень выше	R ⚡ →	< не определено >
>  Кнопка Назад доступна	R W ✗ ✓	false
>  Кнопка Вперед доступна	R W ✗ ✓	false
>  Кнопка На уровень выше доступна	R W ✗	true
>  Ссылка на главную форму	R ⚡ →	< не определено >

После задания свойств, форма диагностики будет выглядеть следующим образом:



5. Добавьте на экранную форму "DiagnForm" элемент "Команда" из вкладки "Встроенные" панели элементов. Назовите команду "GoToBack".



6. В редакторе свойств команды "GoToBack" задайте следующие свойства:

Свойство	Значение
Включено	true
Триггер	false

7. С помощью контекстного меню добавьте обработчик события "Invoked" для команды "GoToBack".

Структура объекта

Имя Описание

- DiagnForm
  - Графические объекты
    - Данные
      - GoToBack Команда

---

События

Имя Харак Обработчик

- Invoked
  - Handler\_1
    - Открыть во фрейме
      - MainMenuForm
        - Фрейм refMainForm.MainFrame

EnabledChanged



8. Раскройте обработчик и задайте в свойстве "Ссылка на главную форму" значение "here.refMainForm".

Структура объекта			
<input type="text" value=""/>			
Имя	Описание		
<ul style="list-style-type: none"> <li>▼  DiagnForm <ul style="list-style-type: none"> <li>&gt; Графические объекты</li> <li>▼ Данные <ul style="list-style-type: none"> <li> GoToBack</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Тип на основе Мнемосхема</li> </ul>		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"></td> <td style="width: 50%; text-align: center;">Команда</td> </tr> </table>			Команда
	Команда		
События			
<input type="text" value=""/>			
Имя	Харак	Обработчик	
>  Ширина	R W	1900	
>  Высота	R W	960	
>  Фокус ввода	R W ⚡	<не определено>	
>  Цвет пера	R W	4278190080	
>  Стиль пера	R W	Нет пера	
>  Толщина пера	R W	1	
>  Цвет заливки	R W	0xff323232	
>  Стиль заливки	R W	Сплошная заливка	
>  X окна	R W	0	
>  Y окна	R W	0	
>  Ширина окна	R W	1920	
>  Высота окна	R W	1080	
>  Заголовок окна	R W	Главное меню	
>  Показывать заголовок	R W	true	
>  Показывать кнопку "свернуть"	R W	true	
>  Показывать кнопку "развернуть"	R W	true	
>  Показывать кнопку "закреть"	R W	true	
>  Поверх всех окон	R W	false	
>  Размеры окна	R W	Вручную	
>  Стиль рамки окна	R W	Изменяемый размер	
>  Состояние окна	R W	По умолчанию	
>  Режим масштабирования	R W	Не масштабировать	
>  Монитор	R W	0	
>  Положение окна	R W	Вручную	
>  Режим обработки закрытия окна	R W	Закреть	
>  Путь к файлу иконки	R W		
Команда Назад	R ⇐ →	<не определено>	
Команда Вперед	R ⇐ →	<не определено>	
Команда На уровень выше	R ⇐ →	<не определено>	
>  Кнопка Назад доступна	R W ⚡	false	
>  Кнопка Вперед доступна	R W ⚡	false	
>  Кнопка На уровень выше досту...	R W ⚡	false	
Ссылка на главную форму	R ⇐ →	here.refMainForm	
EnabledChanged			

9. Откройте свойства экранной формы "DiagnForm" и укажите ссылку у свойства "Команда на уровень выше" на созданную команду "GoToBack".

Структура объекта

Имя	Описание
DiagnForm	Тип на основе Мнемосхема
> Графические объекты	
> Данные	
GoToBack	Команда

Редактор свойств

Свойство	Характеристика	Значение
> f8 Ширина	R W ✓	1900
> f8 Высота	R W ✓	960
> B Фокус ввода	R W ✗	< не определено >
> u4 Цвет пера	R W ✓	4278190080
> u2 Стил пера	R W ✓	Нет пера
> f8 Толщина пера	R W ✓	1
> u4 Цвет заливки	R W ✓	0xff323232
> u2 Стил заливки	R W ✓	Сплошная заливка
> i4 X окна	R W ✓	0
> i4 Y окна	R W ✓	0
> i4 Ширина окна	R W ✓	1920
> i4 Высота окна	R W ✓	1080
> S Заголовок окна	R W ✓	Экран диагностики
> B Показывать заголовок	R W ✓	true
> B Показывать кнопку "свернуть"	R W ✓	true
> B Показывать кнопку "развернуть"	R W ✓	true
> B Показывать кнопку "закреть"	R W ✓	true
> B Поверх всех окон	R W ✓	false
> u4 Размеры окна	R W ✓	Вручную
> u4 Стил рамки окна	R W ✓	Изменяемый размер
> u4 Состояние окна	R W ✓	По умолчанию
> u4 Режим масштабирования	R W ✓	Не масштабировать
> i4 Монитор	R W ✓	0
> u4 Положение окна	R W ✓	Вручную
> u2 Режим обработки закрытия окна	R W ✓	Закреть
> S Путь к файлу иконки	R W ✓	
Команда Назад	R ↵ →	< не определено >
Команда Вперед	R ↵ →	< не определено >
Команда На уровень выше	R ↵ → ✓	GoToBack
> B Кнопка Назад доступна	R W ✗ ✓	false
> B Кнопка Вперед доступна	R W ✗ ✓	false
> B Кнопка На уровень выше доступна	R W ✗ ✓	true
Ссылка на главную форму	R ↵ →	< не определено >

10. Перейдите на экранную форму "MainForm" и с помощью контекстного меню добавьте обработчик события "Invoked" для команды "OpenDiagnForm".

The screenshot displays two windows from the Visual Studio IDE:

- Структура объекта (Object Structure):** Shows a tree view for the 'MainForm' project. Under the 'Данные' (Data) folder, three command items are listed: 'OpenMainMenuForm', 'OpenDiagnForm', and 'OpenDefaultForm', all of which are of type 'Команда' (Command). The 'OpenDiagnForm' item is currently selected.
- События (Events):** Shows the event handling configuration for the selected 'OpenDiagnForm' command. The event 'Invoked' is expanded, and a context menu is open over the 'Handler\_1' entry. The menu options are:
  - Открыть во фрейме (Open in frame) - currently selected
  - DiagnFormThe 'Фрейм' (Frame) property is set to 'here.MainFrame'.

11. Раскройте обработчик и задайте свойству "Ссылка на главную форму" значение "here".

Структура объекта

Имя	Описание
MainForm	Тип на основе Главное окно
Данные	
OpenMainMenuForm	Команда
OpenDiagnForm	Команда

События

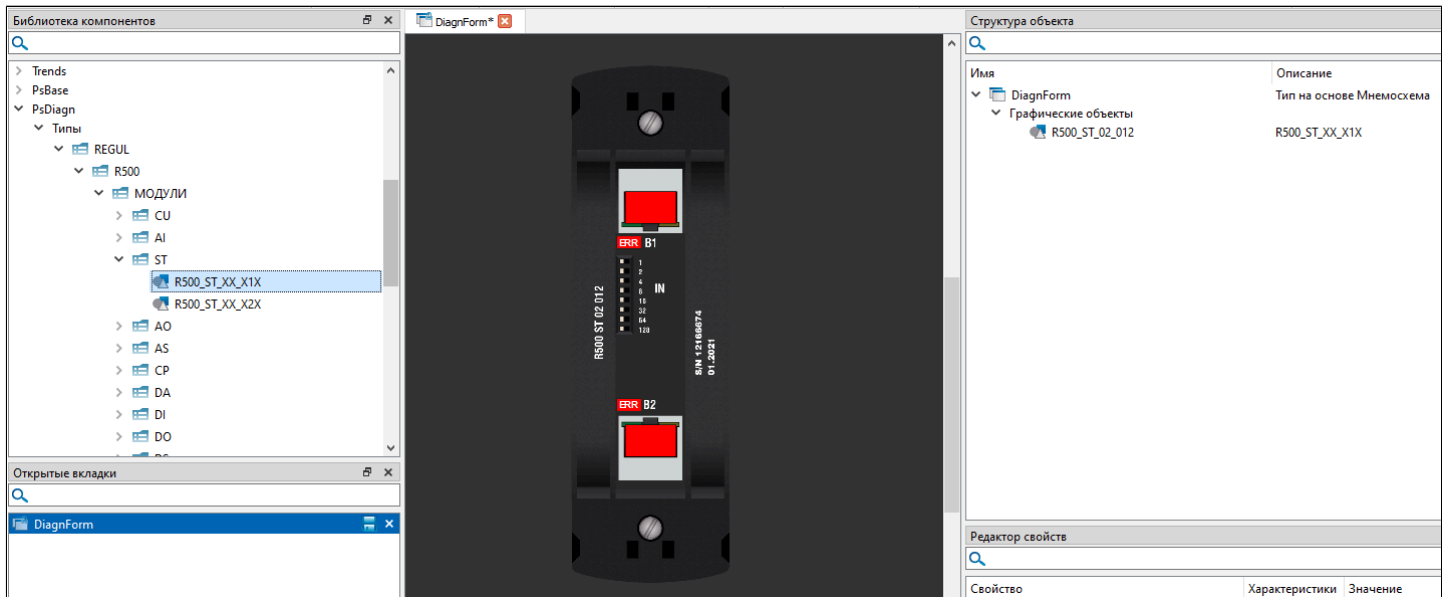
Имя	Харак	Обработчик
> f8 Высота	R W	960
> B Фокус ввода	R W ⚡	< не определено >
> u4 Цвет пера	R W	4278190080
> u2 Стил пера	R W	Нет пера
> f8 Толщина пера	R W	1
> u4 Цвет заливки	R W	0xff323232
> u2 Стил заливки	R W	Сплошная заливка
> i4 X окна	R W	0
> i4 Y окна	R W	0
> i4 Ширина окна	R W	1920
> i4 Высота окна	R W	1080
> S Заголовок окна	R W	Экран диагностики
> B Показывать заголовок	R W	true
> B Показывать кнопку "свернуть"	R W	true
> B Показывать кнопку "развернуть"	R W	true
> B Показывать кнопку "закреть"	R W	true
> B Поверх всех окон	R W	false
> u4 Размеры окна	R W	Вручную
> u4 Стил рамки окна	R W	Изменяемый размер
> u4 Состояние окна	R W	По умолчанию
> u4 Режим масштабирования	R W	Не масштабировать
> i4 Монитор	R W	0
> u4 Положение окна	R W	Вручную
> u2 Режим обработки закрытия окна	R W	Закреть
> S Путь к файлу иконки	R W	
Команда Назад	R ↵ →	< не определено >
Команда Вперед	R ↵ →	< не определено >
Команда На уровень выше	R ↵ →	GoToBack
> B Кнопка Назад доступна	R W ⚡	false
> B Кнопка Вперед доступна	R W ⚡	false
> B Кнопка На уровень выше доступна	R W ⚡	true
Ссылка на главную форму	R ↵ →	here

EnabledChanged

## 1.1.4.5.4.1. Диагностика ПЛК REGUL R500

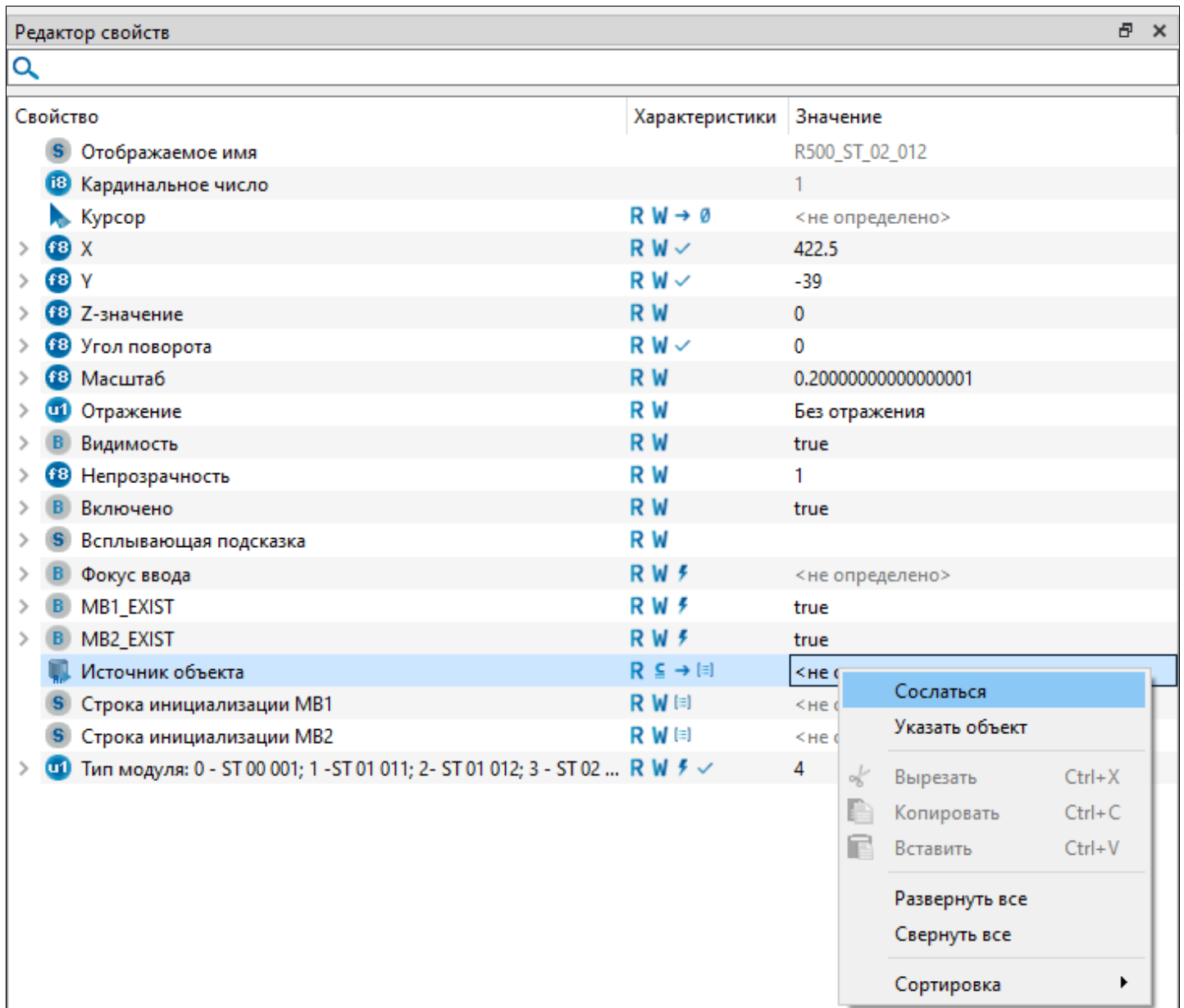
Создайте конфигурацию крейта "REGUL\_R500\_A".

1. Добавьте на экранную форму тип "R500\_ST\_XX\_X1X" из библиотеки "PsDiagn". Во вкладке "Структура объекта" переименуйте его в "R500\_ST\_02\_012".



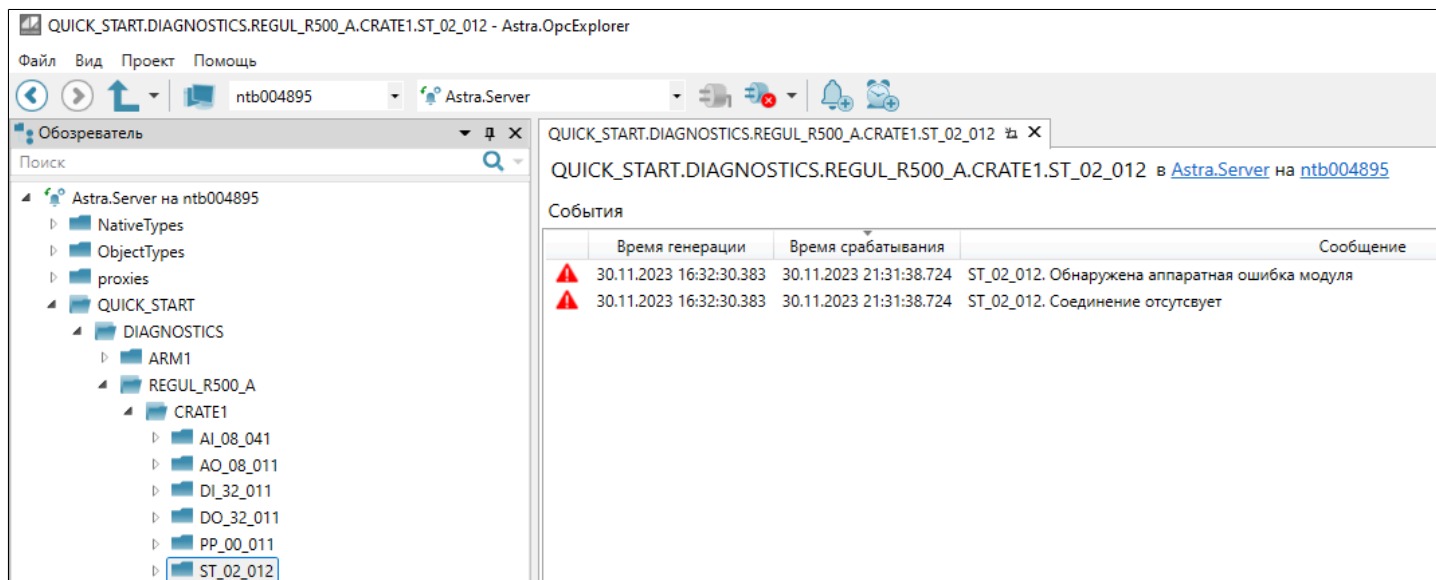
2. Настройте следующие свойства объекта:

➤ Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections":  
unit.Connections.MAIN\_SOURCE.



› Строка инициализации MB1 – строка инициализации первой шины.

Полный путь до объекта можно посмотреть в утилите Op Explorer: "QUICK\_START" → "DIAGNOSTICS" → "REGUL\_R500\_A" → "CRATE1" → "ST\_02\_012".



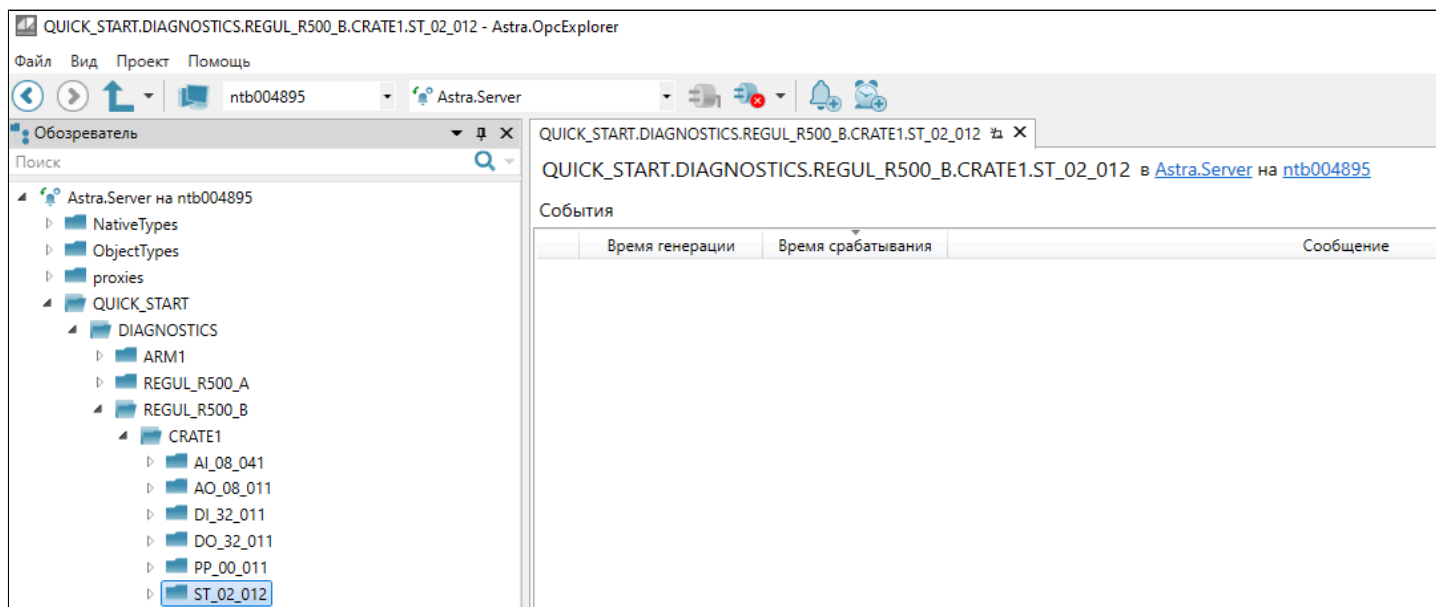
Выделите весь полный путь и скопируйте в свойство "Строка инициализации MB1".

Источник объекта	R $\leq$ $\rightarrow$ (=) $\checkmark$	unit.Connections.MAIN_SOURCE
Строка инициализации MB1	R W (=) $\checkmark$	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.ST_02_012
Строка инициализации MB2	R W (=)	<не определено>



› Строка инициализации MB2 – строка инициализации второй шины.

Полный путь до объекта можно посмотреть в утилите OpсExplorer:  
"QUICK\_START" → "DIAGNOSTICS" → "REGUL\_R500\_B" → "CRATE1" →  
"ST\_02\_012".



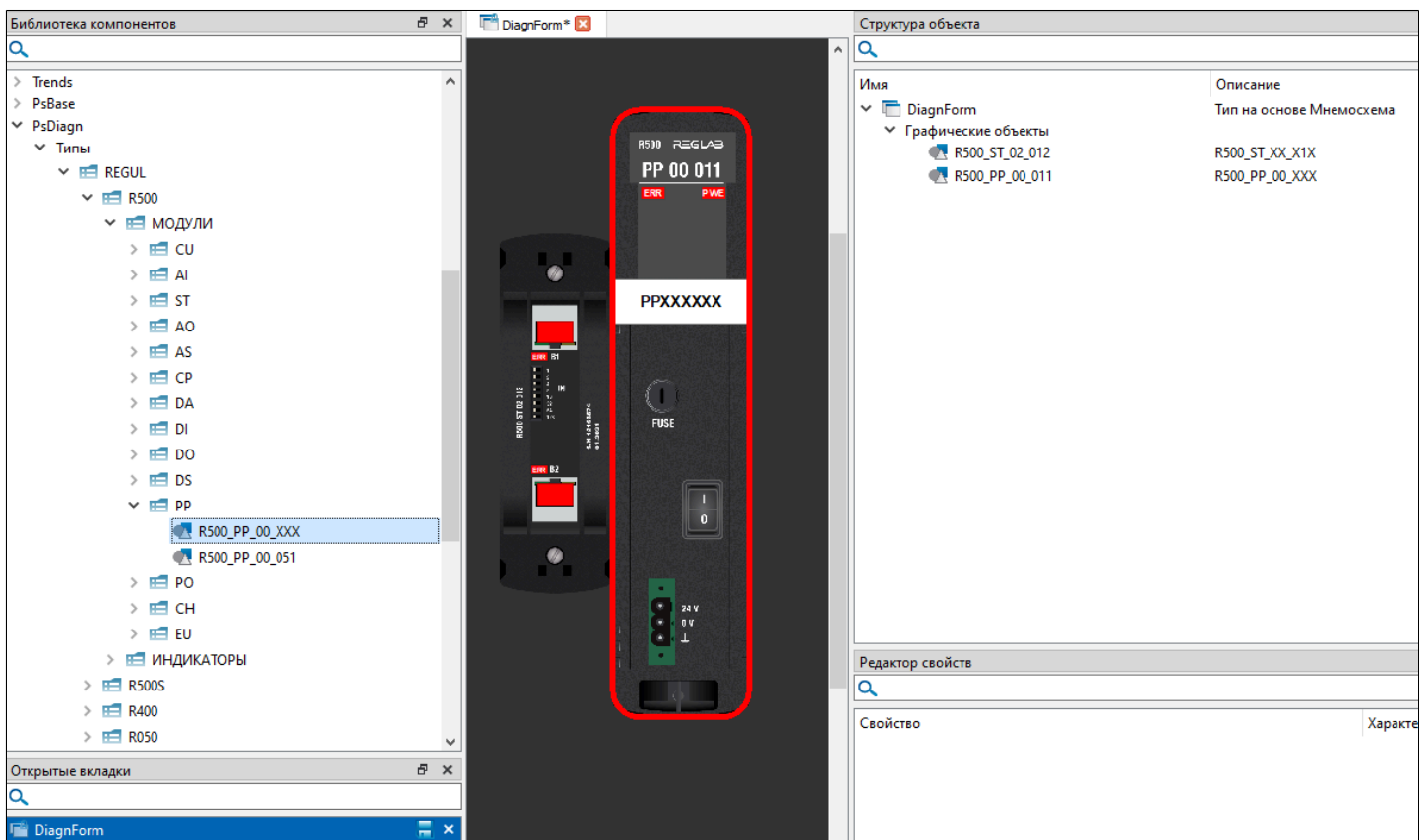
Выделите полный путь и скопируйте в свойство " Строка инициализации MB2".

Источник объекта	R ↵ → ✓	unit.Connections.MAIN_SOURCE
Строка инициализации MB1	R W ✓	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.ST_02_012
Строка инициализации MB2	R W ✓	QUICK_START.DIAGNOSTICS.REGUL_R500_B.CRATE1.ST_02_012

> Тип модуля - выбор необходимого типа модуля для отображения. Укажите 4 тип модуля.

Строка инициализации MB2	R W ✓	QUICK_START.DIAGNOSTICS...
Тип модуля: 0 - ST 00 001; 1 -ST 01 011; 2- ST 01 012; 3 - ST 02 011; 4- ST 02 012; 5-...	R W ⚡ ✓	4

3. Добавьте на экранную форму "DiagnForm" модуль "R500\_PP\_00\_XXX" из библиотеки "PsDiagn". Во вкладке "Структура объекта" переименуйте его в "R500\_PP\_00\_011".



4. Настройте следующие свойства:

➤ Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Затем укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": unit.Connections.MAIN\_SOURCE.

Структура объекта

Имя Описание

- DiagnForm Тип на основе Мнемосхема
- Графические объекты
  - R500\_ST\_02\_012 R500\_ST\_XX\_X1X
  - R500\_PP\_00\_011 R500\_PP\_00\_XXX

---

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		R500_PP_00_011
Кардинальное число		1
Курсор	R W → ∅	< не определено >
X	R W ✓	370.666666666666862
Y	R W ✓	-127.299999999999998
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	0.32300000000000001
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ⚡	< не определено >
<b>Источник объекта</b>	<b>R ↵ → [=]</b>	<b>&lt; не определено &gt;</b>
Строка инициализации объекта	R W [=]	< не определено >
mX	R W ⚡	< не определено >
mY	R W ⚡	< не определено >
FormId	R W ⚡	< не определено >
Работа в дублированном режиме	R W ⚡	false
init_Path_MB1	R W [=]	< не определено >
init_Path_MB2	R W [=]	< не определено >
MB	R W ⚡	1
Тип модуля: 0 - PP 00 011; 1 - PP 00 021; 2 - PP 00 031	R W ⚡ ✓	0

Сослаться

Указать объект

✂ Вырезать Ctrl+X

📄 Копировать Ctrl+C

📄 Вставить Ctrl+V

Развернуть все

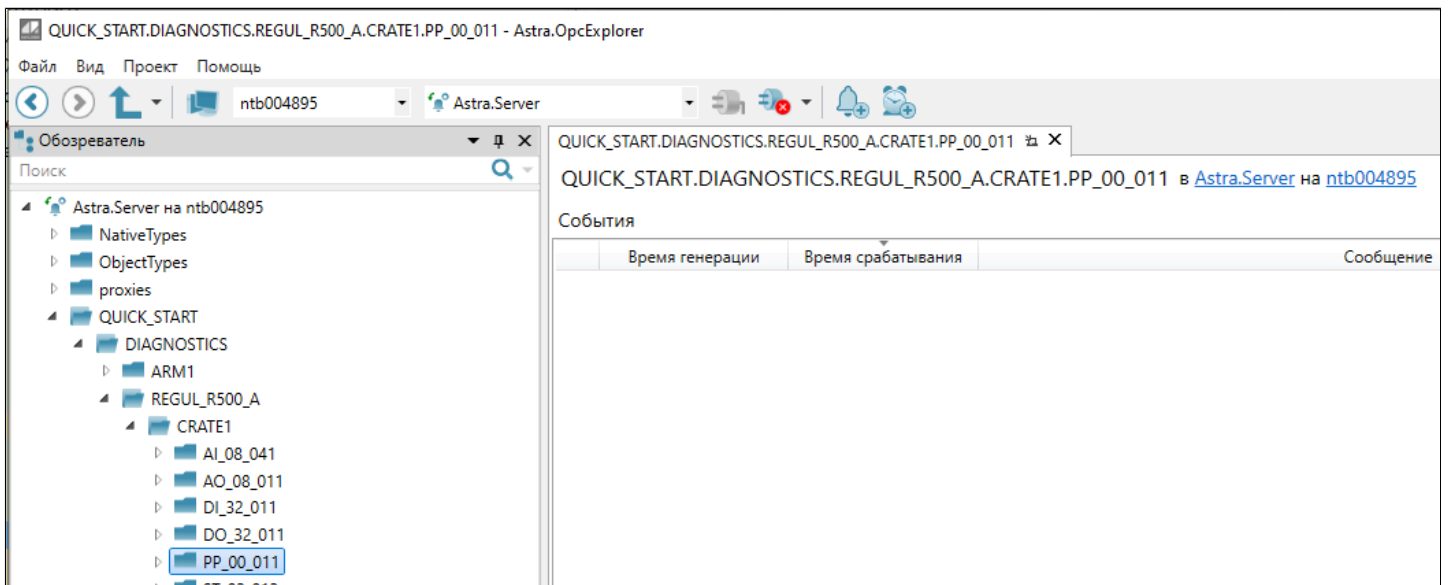
Свернуть все

Сортировка ▶

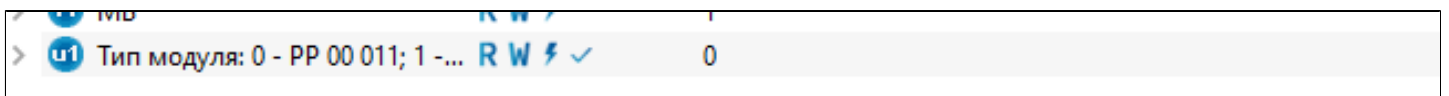
➤ Строка инициализации объекта – здесь необходимо прописать путь до логического объекта модуля для REGUL\_R500\_A на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpсExplorer: "QUICK\_START" → "DIAGNOSTICS" → "REGUL\_R500\_A" → "CRATE1" → "PP\_00\_011".

Скопируйте данный путь и вставьте в свойство "Строка инициализации объекта".



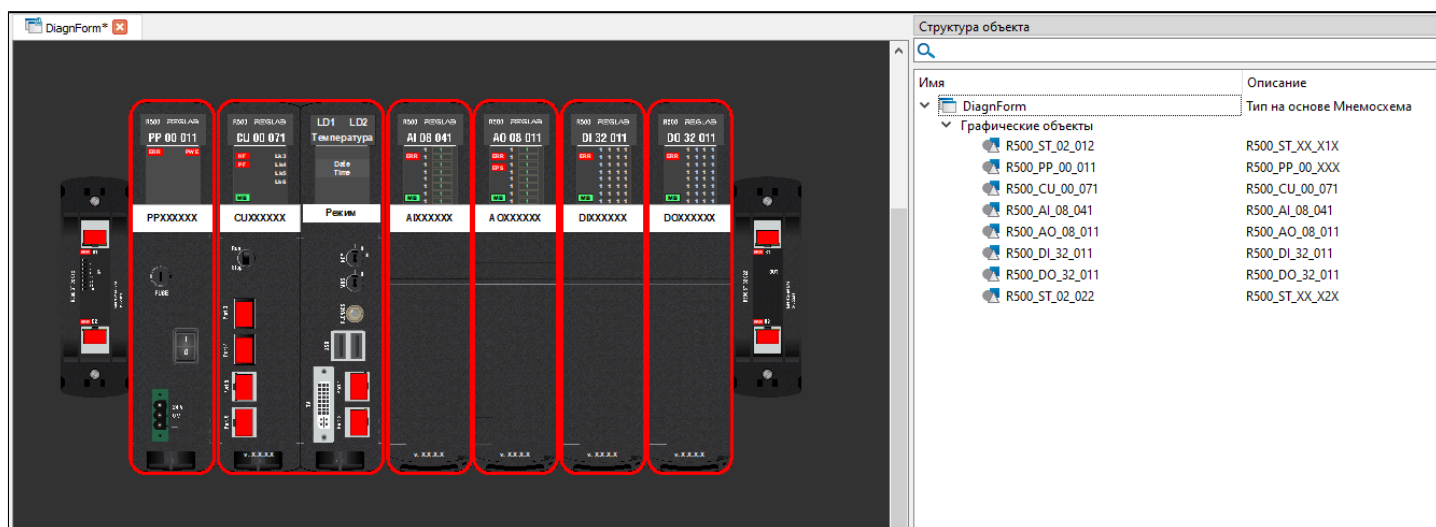
➤ Тип модуля - выбор необходимого типа модуля для отображения. Укажите 0 тип модуля.



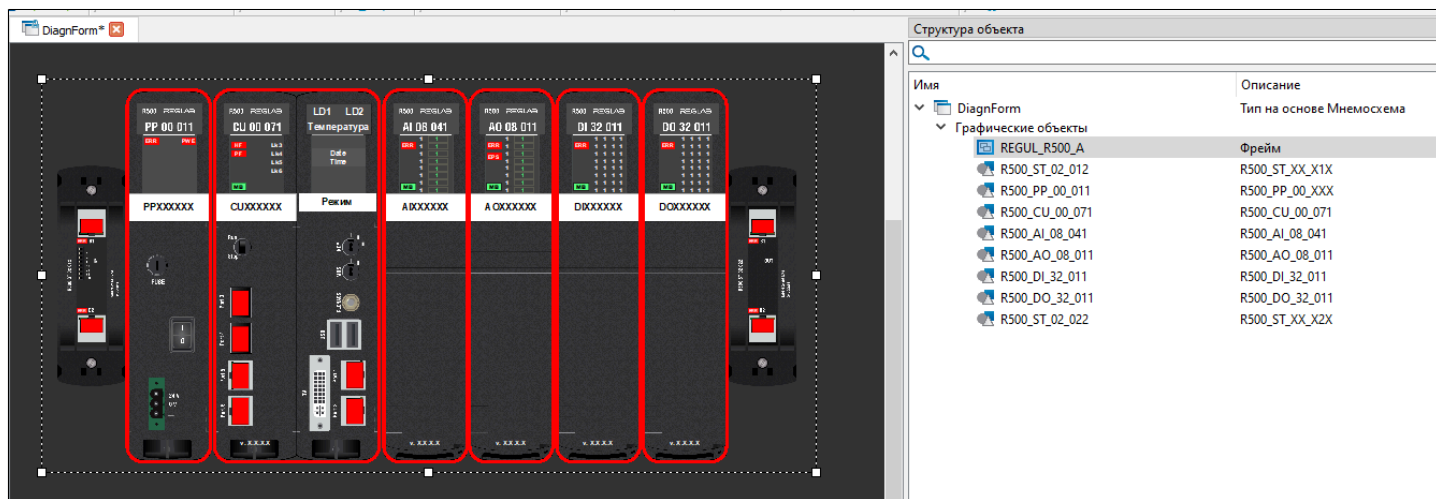
В результате должны получиться следующие свойства для модуля:

Редактор свойств		
🔍		
Свойство	Характеристики	Значение
<b>S</b> Отображаемое имя		R500_PP_00_011
<b>i8</b> Кардинальное число		1
<b>🖱</b> Курсор	R W → 0	<не определено>
> <b>f8</b> X	R W ✓	370.666666666666862
> <b>f8</b> Y	R W ✓	-127.29999999999998
> <b>f8</b> Z-значение	R W	0
> <b>f8</b> Угол поворота	R W ✓	0
> <b>f8</b> Масштаб	R W	0.32300000000000001
> <b>u1</b> Отражение	R W	Без отражения
> <b>B</b> Видимость	R W	true
> <b>f8</b> Непрозрачность	R W	1
> <b>B</b> Включено	R W	true
> <b>S</b> Всплывающая подсказка	R W	
> <b>B</b> Фокус ввода	R W ⚡	<не определено>
<b>🖱</b> Источник объекта	R ≤ → [=] ✓	unit.Connections.MAIN_SOURCE
<b>S</b> Строка инициализации объекта	R W [=] ✓	QUICK_START.DIAGNOSTICS.REGUL_R500_A.CRATE1.PP_00_011
> <b>i4</b> mX	R W ⚡	<не определено>
> <b>i4</b> mY	R W ⚡	<не определено>
> <b>S</b> FormId	R W ⚡	<не определено>
> <b>B</b> Работа в дублированном режиме	R W ⚡	false
<b>S</b> init_Path_MB1	R W [=]	<не определено>
<b>S</b> init_Path_MB2	R W [=]	<не определено>
> <b>i1</b> MB	R W ⚡	1
> <b>u1</b> Тип модуля: 0 - PP 00 011; 1 - PP 00 021; 2 - PP 00 031	R W ⚡ ✓	0

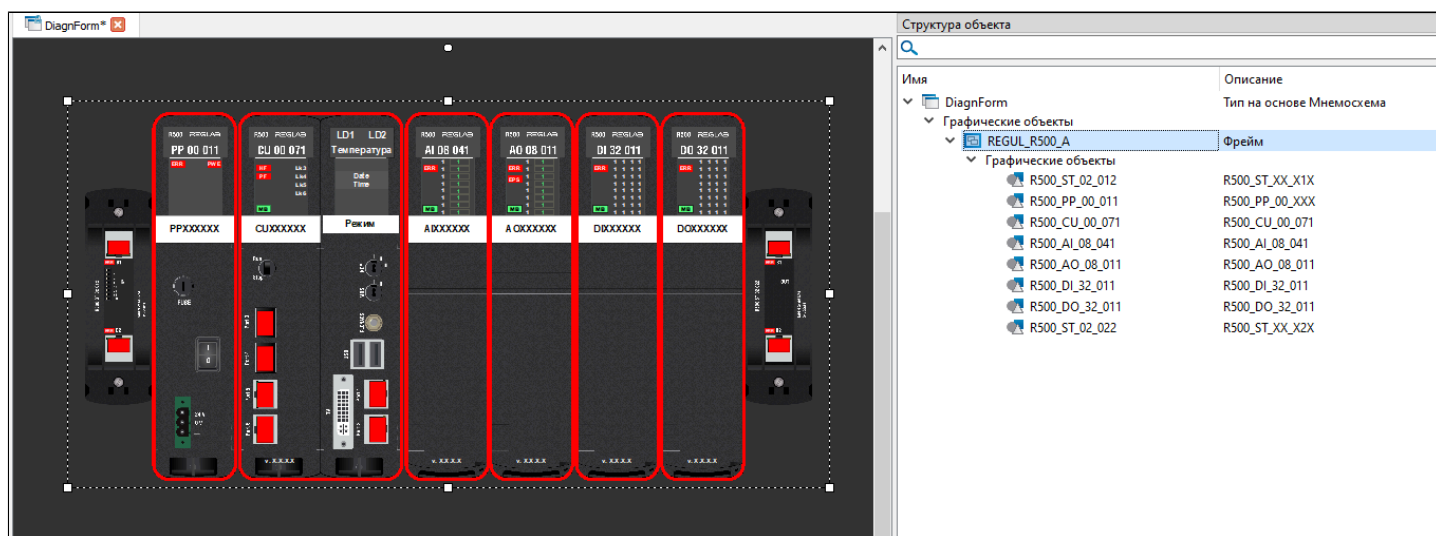
5. Аналогичным образом добавьте и настройте все модули крейта "REGUL\_R500\_A". Уберите из названий типов хвосты "\_1".



6. Добавьте элемент "Фрейм" из вкладки общих элементов "Визуальные" панели элементов на форму "DiagnForm". Задайте размер фрейма, чтобы в нем поместились все, добавленные ранее, модули. В редакторе свойств фрейма задайте имя "REGUL\_R500\_A".



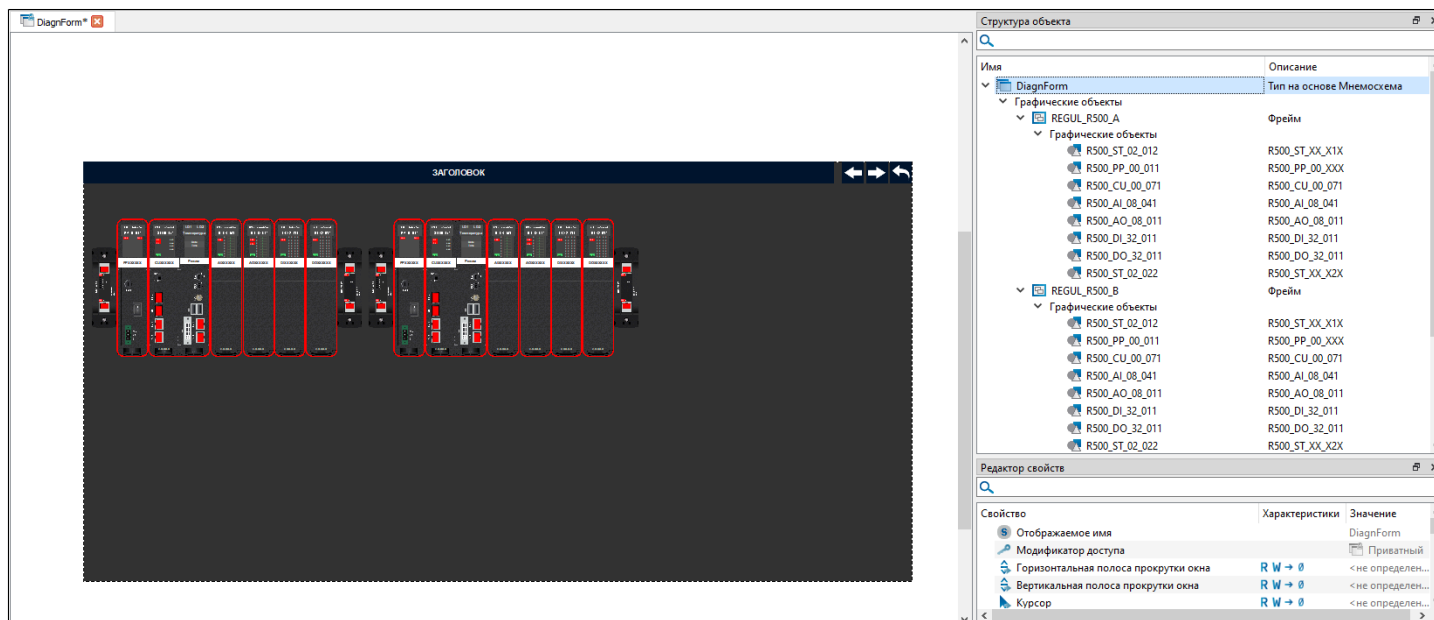
7. Перетащите в дереве объектов все созданные модули в крейт "REGUL\_R500\_A".



Конфигурация крейта ПЛК REGUL\_R500\_A завершена.

8. Так как в проекте используется полное резервирование ПЛК, то их конфигурации одинаковы.

Выделите в дереве объектов фрейм "REGUL\_R500\_A" и сделайте его копию. В редакторе свойств нового фрейма задайте имя "REGUL\_R500\_B".



9. Измените свойство "Строка инициализации" (для всех модулей, кроме оконечных) для каждого модуля крейта ПЛК В – замените значение "REGUL\_R500\_A" на "REGUL\_R500\_B". Ниже приведен пример для модуля "PP 00 011".



Структура объекта

Имя Описание

- DiagnForm
  - Графические объекты
    - REGUL\_R500\_A Фрейм
    - REGUL\_R500\_B Фрейм
      - Графические объекты
        - R500\_ST\_02\_012 R500\_ST\_XX\_X1X
        - R500\_PP\_00\_011 R500\_PP\_00\_XXX
        - R500\_CU\_00\_071 R500\_CU\_00\_071
        - R500\_AL\_00\_041 R500\_AL\_00\_041

Редактор свойств

Свойство	Характеристики	Значение
Кардинальное число		1
Курсор	R W → 0	< не определено >
>  X	R W ✓	0.666666666666853303
>  Y	R W ✓	-351.30000000000001
>  Z-значение	R W	0
>  Угол поворота	R W ✓	0
>  Масштаб	R W	0.32300000000000001
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Фокус ввода	R W ⚡	< не определено >
Источник объекта	R ⊆ → (=) ✓	unit.Connections.MAIN_SOURCE
Строка инициализации объекта	R W (=) ✓	QUICK_START.DIAGNOSTICS.REGUL_R500_B.CRATE1.PP_00_011
>  mX	R W ⚡	< не определено >
>  mY	R W ⚡	< не определено >
>  FormId	R W ⚡	< не определено >
>  Работа в дублированном режиме	R W ⚡	false
init_Path_MB1	R W (=)	< не определено >
init_Path_MB2	R W (=)	< не определено >
>  MB	R W ⚡	1
>  Тип модуля: 0 - PP 00 011; 1 - PP 0...	R W ⚡ ✓	0

10. В редакторе свойств фреймов "REGUL\_R500\_A" и "REGUL\_R500\_B" задайте свойствам "Стиль пера" значение "Нет пера" и "Масштаб" – "0.9".

Структура объекта

Имя Описание

- DiagnForm Тип на основе Мнемосхема
  - Графические объекты
    - REGUL\_R500\_A Фрейм

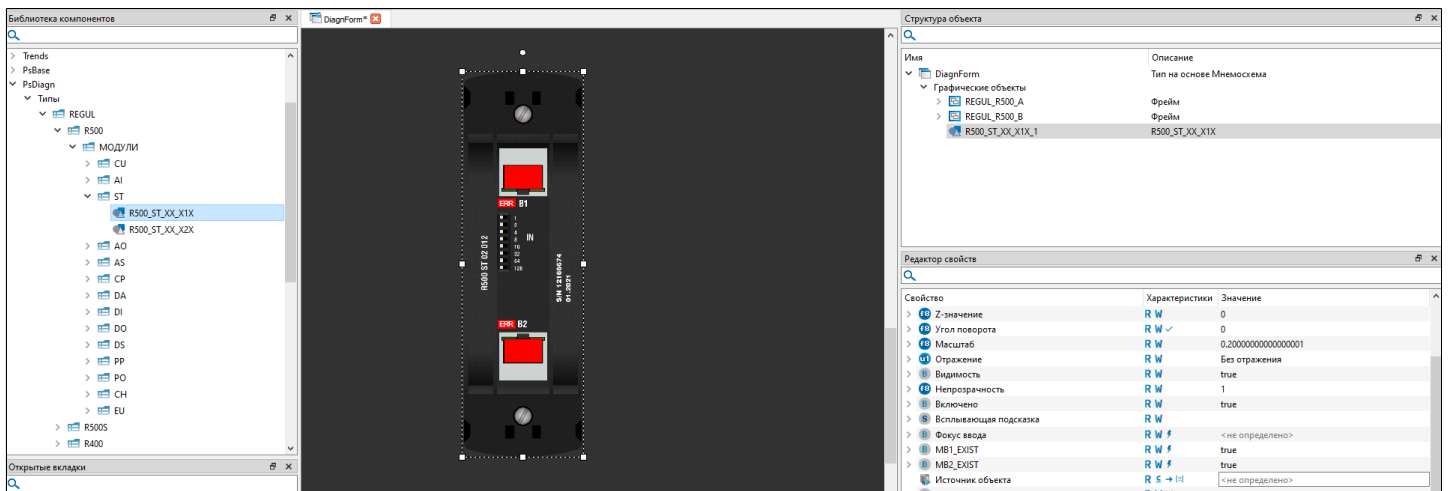
Редактор свойств

Свойство	Характеристи	Значение
<b>S</b> Отображаемое имя		REGUL_R500_A
<b>i8</b> Кардинальное число		1
Горизонтальная полоса прокрутки	R W → 0	< не определен...
Вертикальная полоса прокрутки	R W → 0	< не определен...
> <b>f8</b> X	R W	108.2857142857...
> <b>f8</b> Y	R W	497.3571428571...
> <b>f8</b> Z-значение	R W	0
> <b>f8</b> Угол поворота	R W	0
> <b>f8</b> Масштаб	R W	0.900000000000...
> <b>u1</b> Отражение	R W	Без отражения
> <b>B</b> Видимость	R W	true
> <b>f8</b> Непрозрачность	R W	1
> <b>B</b> Включено	R W	true
> <b>S</b> Всплывающая подсказка	R W	
> <b>f8</b> Ширина	R W	704
> <b>f8</b> Высота	R W	382
> <b>B</b> Фокус ввода	R W ⚡ ⓘ	< не определен...
> <b>u4</b> Цвет пера	R W	4278190080
> <b>u2</b> Стиль пера	R W	Нет пера
> <b>f8</b> Толщина пера	R W	1
> <b>u4</b> Цвет заливки	R W	4278190080
> <b>u2</b> Стиль заливки	R W	Нет заливки
> <b>B</b> Ручное управление масштабом	R W	false
> <b>B</b> Отображать полосы прокрутки	R W	true
> <b>B</b> Перемещение мышью	R W	false

## 1.1.4.5.4.2. Диагностика ПЛК SAFETY REGUL R500S

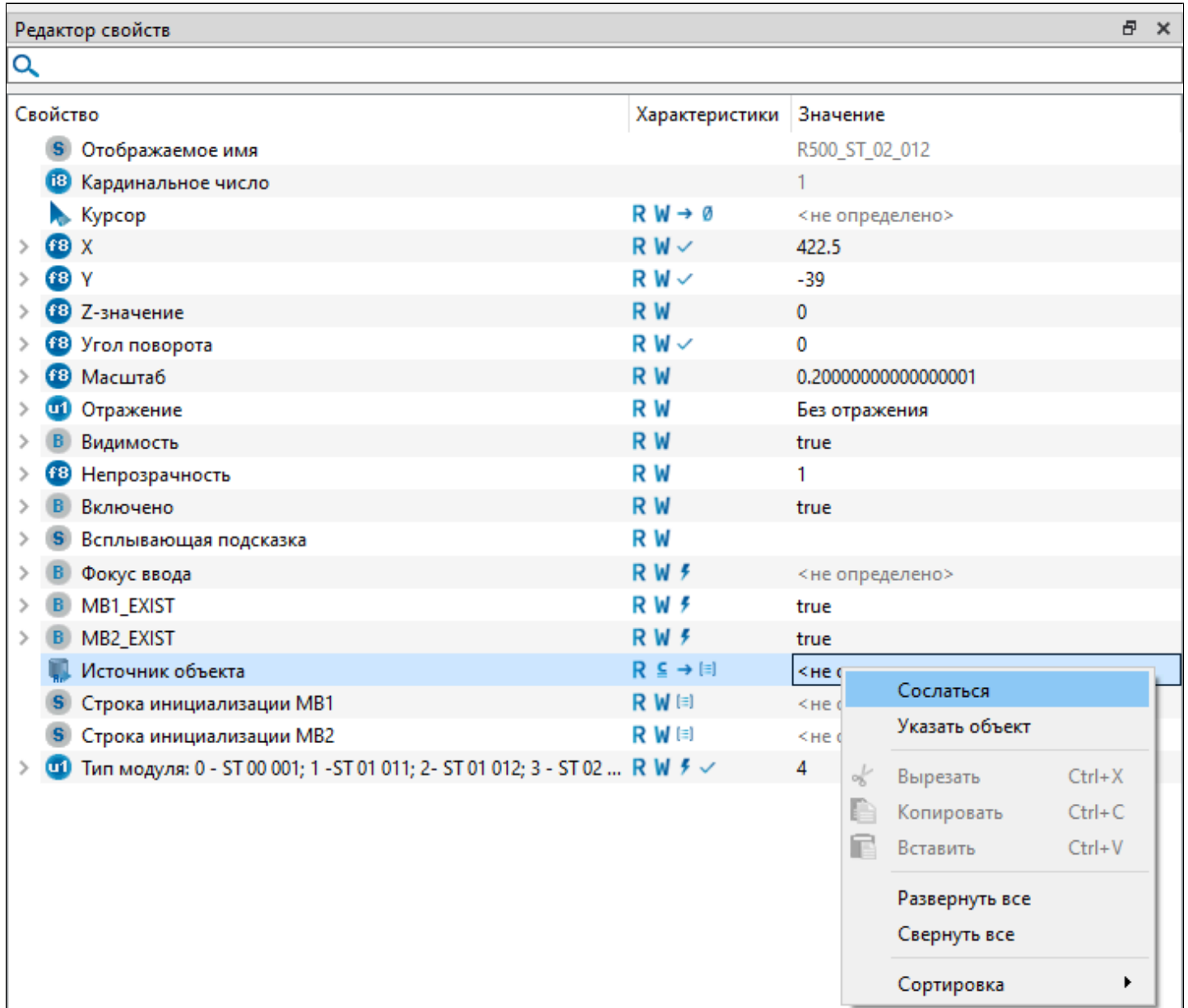
Создайте конфигурацию крейта "SAFETY\_REGUL\_R500S".

1. Добавьте на экранную форму тип "R500\_ST\_XX\_X1X" из библиотеки "PsDiagn". Во вкладке "Структура объекта" переименуйте его в "R500\_ST\_02\_012".



## 2. Настройте следующие свойства объекта:

› Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": unit.Connections.MAIN\_SOURCE.

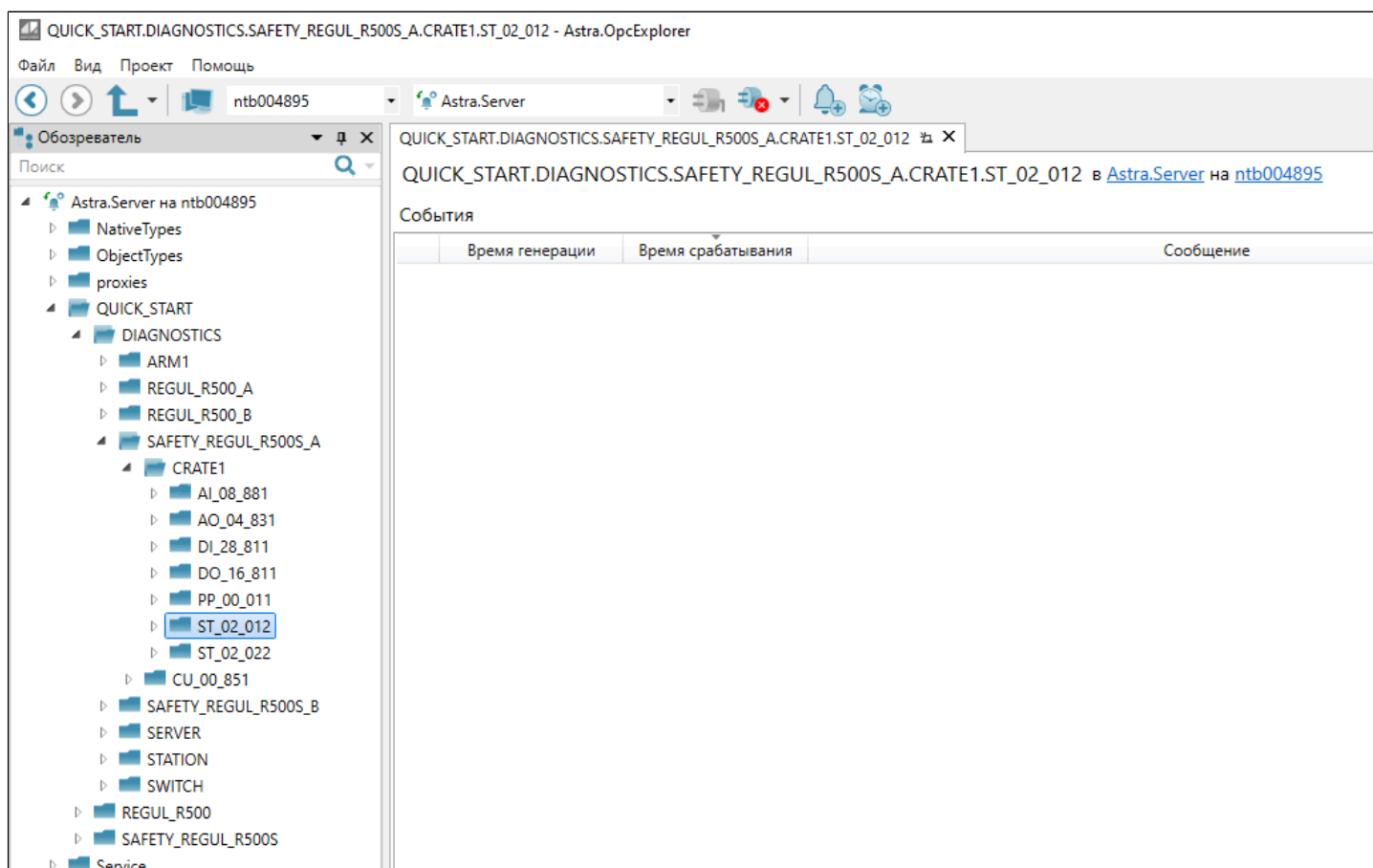


The screenshot shows the 'Properties Editor' window with a table of properties. The 'Object Source' property is selected, and a context menu is open over it. The context menu includes options like 'Link' (Сослаться), 'Specify Object' (Указать объект), 'Cut' (Вырезать), 'Copy' (Копировать), 'Paste' (Вставить), 'Expand All' (Развернуть все), 'Collapse All' (Свернуть все), and 'Sort' (Сортировка).

Свойство	Характеристики	Значение
Отображаемое имя		R500_ST_02_012
Кардинальное число		1
Курсор	R W → ∅	<не определено>
X	R W ✓	422.5
Y	R W ✓	-39
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	0.20000000000000001
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ✗	<не определено>
MB1_EXIST	R W ✗	true
MB2_EXIST	R W ✗	true
Источник объекта	R ≤ → (=)	<не определено>
Строка инициализации MB1	R W (=)	<не определено>
Строка инициализации MB2	R W (=)	<не определено>
Тип модуля: 0 - ST 00 001; 1 -ST 01 011; 2- ST 01 012; 3 - ST 02 ...	R W ✗ ✓	4

› Строка инициализации MB1 – строка инициализации первой шины.

Полный путь до объекта можно посмотреть в утилите OpсExplorer: "QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_A" → "CRATE1" → "ST\_02\_012".

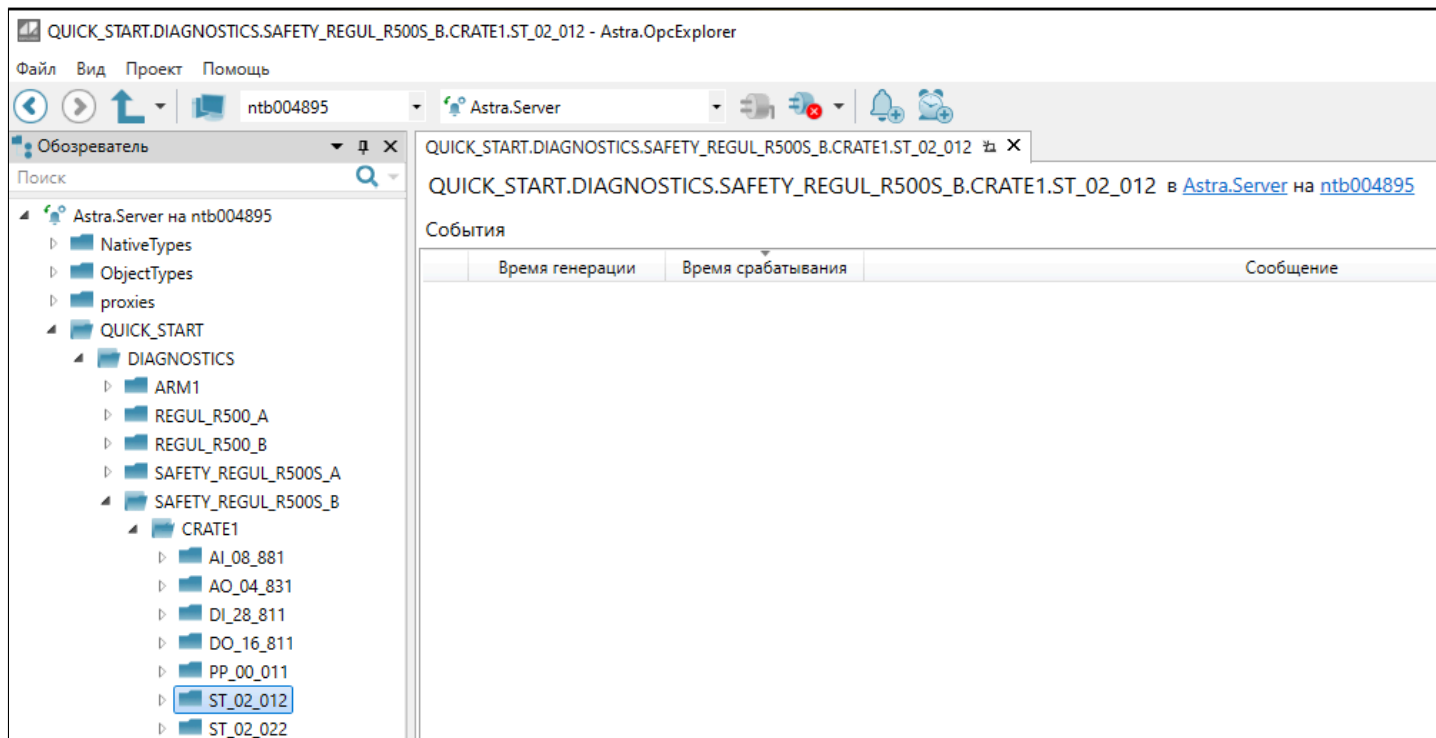


Выделите весь полный путь и скопируйте в свойство "Строка инициализации MB1".

Свойство	Характеристики	Значение
Отображаемое имя		R500_ST_02_012
Кардинальное число		1
Курсор	R W → 0	< не определено >
X	R W ✓	917
Y	R W ✓	180
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	0.20000000000000001
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ⚡	< не определено >
MB1_EXIST	R W ⚡	true
MB2_EXIST	R W ⚡	true
Источник объекта	R ⊆ → [≡] ✓	unit.Connections.MAIN_SOURCE
Строка инициализации MB1	R W [≡] ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CRATE1.ST_02_012
Строка инициализации MB2	R W [≡]	< не определено >
Тип модуля: 0 - ST 00 001; 1 -ST 01 011; 2- ST 01 01...	R W ⚡	4

› Строка инициализации MB2 – строка инициализации второй шины.

Полный путь до объекта можно посмотреть в утилите OpcExplorer:  
"QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_B" → "CRATE1" →  
"ST\_02\_012".



Выделите полный путь и скопируйте в свойство " Строка инициализации MB2".

Источник объекта	R ↵ → ✓	unit.Connections.MAIN_SOURCE
С строка инициализации MB1	R W ↵ ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CRATE1.ST_02_012
С строка инициализации MB2	R W ↵ ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_B.CRATE1.ST_02_012

➤ Тип модуля - выбор необходимого типа модуля для отображения. Укажите 4 тип модуля.

С строка инициализации MB2	R W ↵ ✓	QUICK_START.DIAGNOSTICS...
➤ u1 Тип модуля: 0 - ST 00 001; 1 -ST 01 011; 2- ST 01 012; 3 - ST 02 011; 4- ST 02 012; 5-...	R W ↵ ✓	4

3. Добавьте на экранную форму "DiagnForm" модуль "R500\_PP\_00\_XXX" из библиотеки "PsDiagn". Во вкладке "Структура объекта" переименуйте его в "R500\_PP\_00\_011".

The screenshot shows the software interface with three main panels:

- Библиотека компонентов (Component Library):** A tree view on the left showing various modules under 'МОДУЛИ'. The 'R500\_PP\_00\_XXX' module is highlighted.
- DiagnForm:** A central window displaying a diagnostic form with a red dashed box around a section labeled 'R500 PP 00 011' and 'PPXXXXXX'.
- Структура объекта (Object Structure):** A panel on the right showing the hierarchy of objects. The 'R500\_PP\_00\_011' object is selected and highlighted in blue.

4. Настройте следующие свойства:

➤ Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Затем укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": "unit.Connections.MAIN\_SOURCE".

Структура объекта



Имя	Описание
DiagnForm	Тип на основе Мнемосхема
Графические объекты	
> REGUL_R500_A	Фрейм
> REGUL_R500_B	Фрейм
R500_ST_02_012	R500_ST_XX_X1X
R500_PP_00_011	R500_PP_00_XXX

Редактор свойств



Свойство	Характеристики	Значение
<b>i8</b> Кардинальное число		1
Курсор	R W → 0	<не определено>
> <b>f8</b> X	R W ✓	1026.3333333333326
> <b>f8</b> Y	R W ✓	145
> <b>f8</b> Z-значение	R W	0
> <b>f8</b> Угол поворота	R W ✓	0
> <b>f8</b> Масштаб	R W	0.32300000000000001
> <b>u1</b> Отражение	R W	Без отражения
> <b>B</b> Видимость	R W	true
> <b>f8</b> Непрозрачность	R W	1
> <b>B</b> Включено	R W	true
> <b>S</b> Всплывающая подсказка	R W	
> <b>B</b> Фокус ввода	R W ✗	<не определено>
Источник объекта	R ≤ → (=)	<не определено>
> <b>S</b> Строка инициализации объекта	R W (=)	<не определенс
> <b>i4</b> mX	R W ✗	<не определенс

- Сослаться
- Указать объект





Параметр "Строка инициализации объекта" не заполняется для модулей, находящихся в конфигурации дублированного крейта ПЛК SAFETY REGUL R500S, кроме модулей центрального процессора.

➤ Работа в дублированном режиме - установите значение данного параметра в "true", для работы модуля в дублированном режиме.

➤ init\_Path\_MB1 и init\_Path\_MB2 – для данных параметрах необходимо прописать путь до логических объектов модуля в SAFETY\_REGUL\_R500S\_A и SAFETY\_REGUL\_R500S\_B на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpcExplorer:

➤ "QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_A" → "CRATE1" → "PP\_00\_011"

The screenshot shows the OpcExplorer interface. The left pane displays a tree view of the object structure. The right pane shows the selected object's details and events.

Tree View Structure:

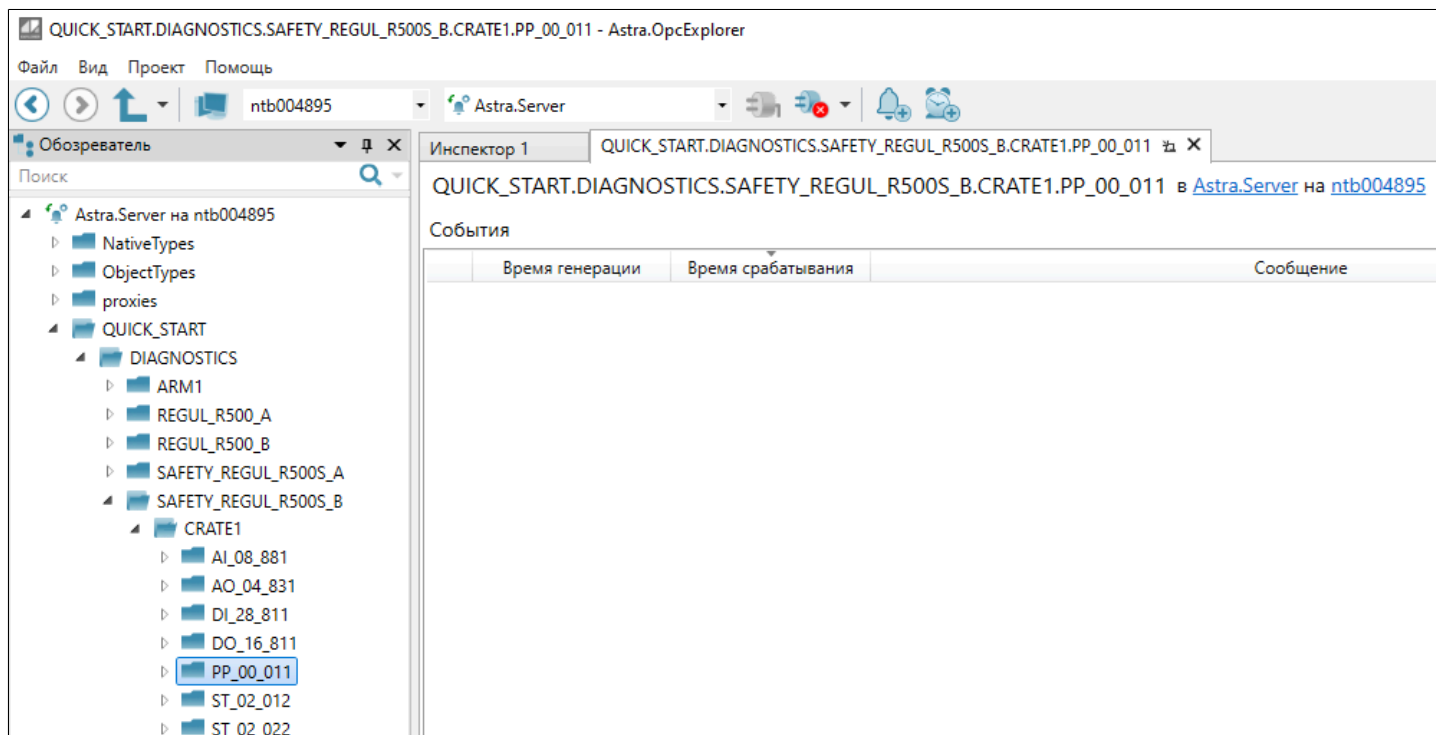
- Astra.Server на ntb004895
  - NativeTypes
  - ObjectTypes
  - proxies
  - QUICK\_START
    - DIAGNOSTICS
      - ARM1
      - REGUL\_R500\_A
      - REGUL\_R500\_B
      - SAFETY\_REGUL\_R500S\_A
        - CRATE1
          - AI\_08\_881
          - AO\_04\_831
          - DI\_28\_811
          - DO\_16\_811
          - PP\_00\_011
          - ST\_02\_012
          - ST\_02\_022

Right Pane Details:

QUICK\_START.DIAGNOSTICS.SAFETY\_REGUL\_R500S\_A.CRATE1.PP\_00\_011 в Astra.Server на ntb004895

Время генерации	Время срабатывания	Сообщение
-----------------	--------------------	-----------

➤ "QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_B" → "CRATE1" → "PP\_00\_011"



Скопируйте данные пути и вставьте в свойства "init\_Path\_MB1" и "init\_Path\_MB2".

>	B	Работа в дублированном режиме	R W $\neq$ ✓	true
	S	init_Path_MB1	R W (=) ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CRATE1.PP_00_011
	S	init_Path_MB2	R W (=) ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_B.CRATE1.PP_00_011

> Тип модуля - выбор необходимого типа модуля для отображения. Укажите 0 тип модуля.

>	u1	Тип модуля: 0 - PP 00 011; 1 -...	R W $\neq$ ✓	0
---	----	-----------------------------------	--------------	---

В результате должны получиться следующие свойства для модуля:

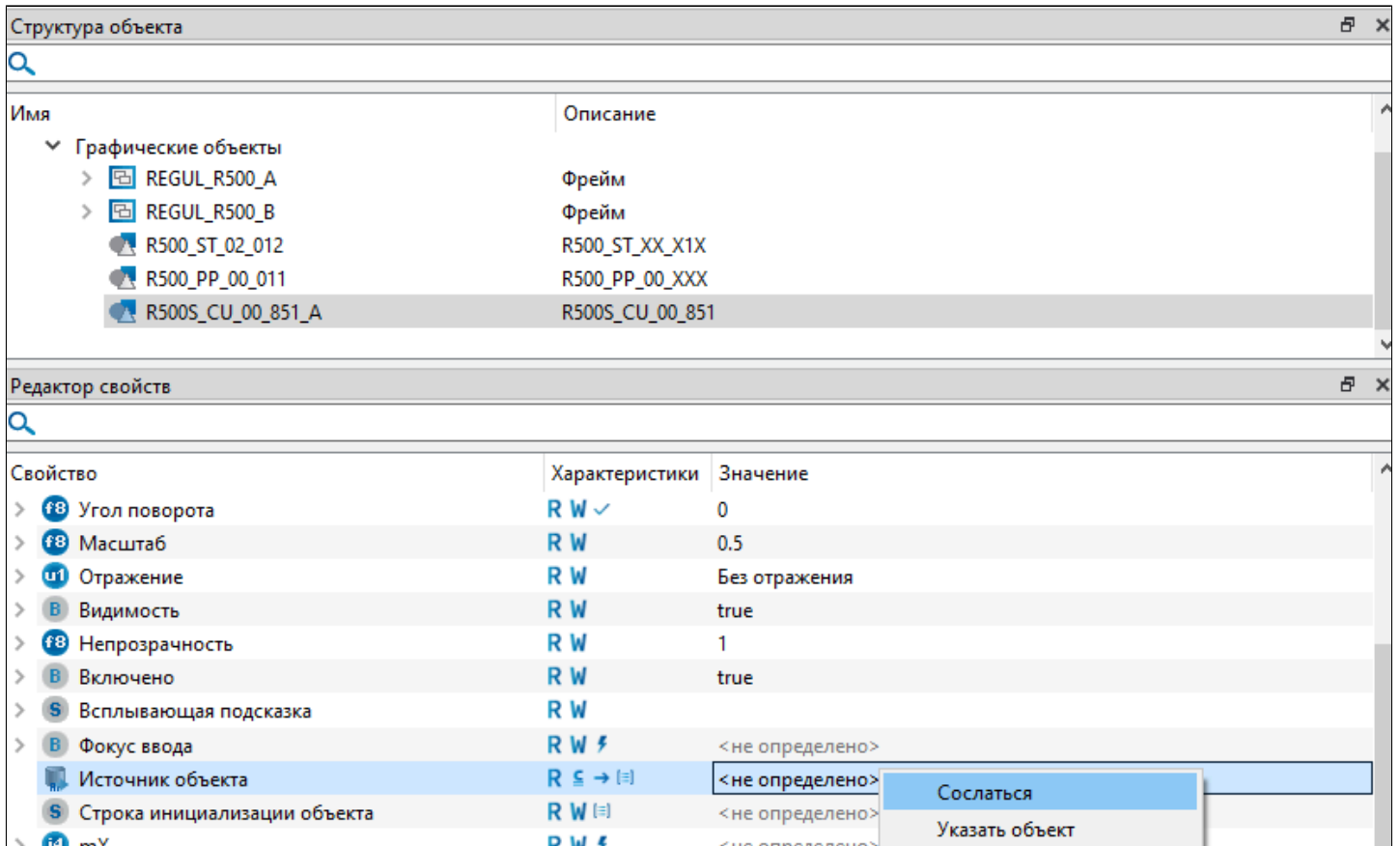
Свойство	Характеристики	Значение
i8 Кардинальное число		1
Курсор	R W → 0	< не определено >
f8 X	R W ✓	1026.3333333333326
f8 Y	R W ✓	145
f8 Z-значение	R W	0
f8 Угол поворота	R W ✓	0
f8 Масштаб	R W	0.32300000000000001
u1 Отражение	R W	Без отражения
B Видимость	R W	true
f8 Непрозрачность	R W	1
B Включено	R W	true
S Всплывающая подсказка	R W	
B Фокус ввода	R W f	< не определено >
Источник объекта	R ↵ → [i] ✓	unit.Connections.MAIN_SOURCE
S Строка инициализации объекта	R W [i]	< не определено >
i4 mX	R W f	< не определено >
i4 mY	R W f	< не определено >
S FormId	R W f	< не определено >
B Работа в дублированном режиме	R W f ✓	true
S init_Path_MB1	R W [i] ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CRATE1.PP_00_011
S init_Path_MB2	R W [i] ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_B.CRATE1.PP_00_011
i1 MB	R W f ✓	1
u1 Тип модуля: 0 - PP 00 011; 1 - PP 00 021; 2 - PP 00 ...	R W f	0

5. Добавьте на экранную форму "DiagnForm" модуль "R500S\_CU\_00\_851" из библиотеки "PsDiagn". Во вкладке "Структура объекта" переименуйте его в "R500S\_CU\_00\_851\_A".

The screenshot displays the software interface for configuring a diagnostic form. On the left, the 'Библиотека компонентов' (Component Library) shows a tree structure under 'PsDiagn' with 'R500S' and 'CU' sub-categories. The 'R500S\_CU\_00\_851' module is highlighted. The main window shows a graphical representation of the diagnostic form with a red dashed box highlighting the 'R500S\_CU\_00\_851' module. On the right, the 'Структура объекта' (Object Structure) panel shows the object hierarchy, and the 'Редактор свойств' (Properties Editor) panel shows the properties for the selected object, including 'R500S\_CU\_00\_851\_A'.

## 6. Настройте следующие свойства:

› Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Затем укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections":  
unit.Connections.MAIN\_SOURCE.



Структура объекта

Имя	Описание
Графические объекты	
REGUL_R500_A	Фрейм
REGUL_R500_B	Фрейм
R500_ST_02_012	R500_ST_XX_X1X
R500_PP_00_011	R500_PP_00_XXX
R500S_CU_00_851_A	R500S_CU_00_851

Редактор свойств

Свойство	Характеристики	Значение
Угол поворота	R W ✓	0
Масштаб	R W	0.5
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W f	< не определено >
Источник объекта	R ↵ → [⇨]	< не определено >
Строка инициализации объекта	R W [⇨]	< не определено >
mX	R W f	< не определено >

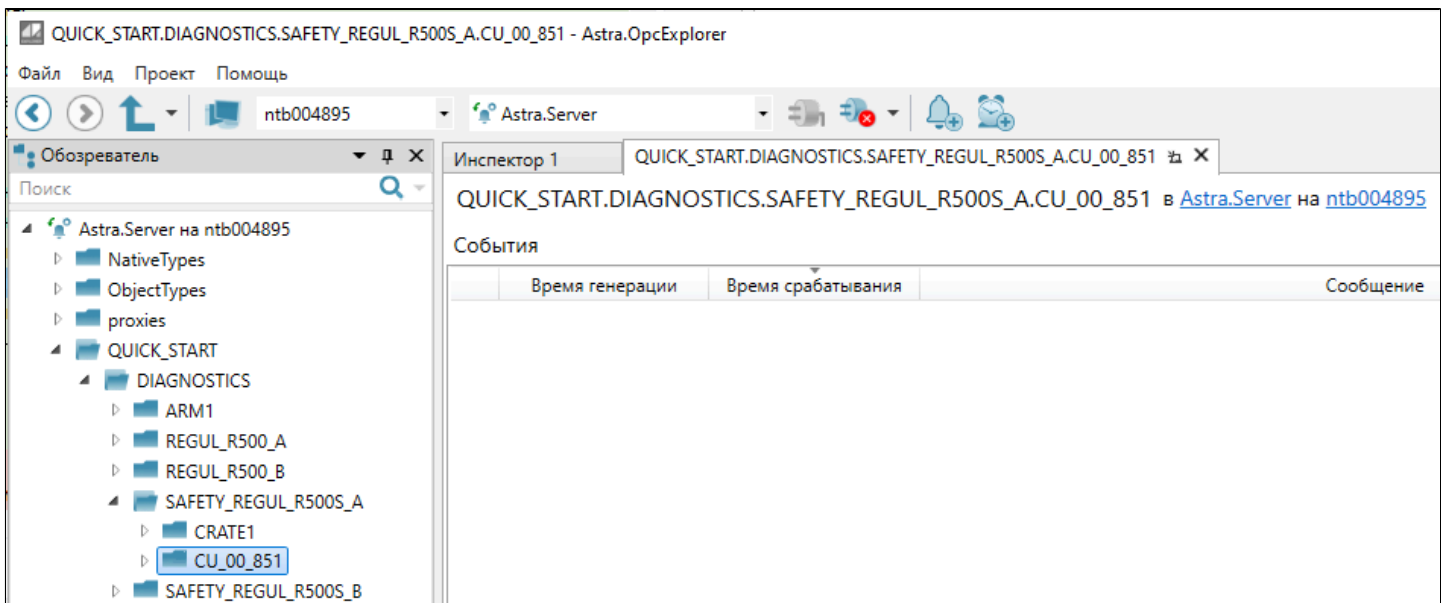
Сослаться  
Указать объект

➤ Строка инициализации объекта – здесь необходимо прописать путь до логического объекта центрального процессора для SAFETY\_REGUL\_R500S\_A на сервере AstraServer.



Для модулей центрального процессора параметр "Работа в дублированном режиме" необходимо оставить в значение "false", так как данные модули работают только по одной шине данных. Для отображение работы второй шины необходимо использовать дополнительный модуль центрального процессора из библиотеки "PsDiagn".

Полный путь до объекта можно посмотреть в утилите OpсExplorer: "QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_A" → "CU\_00\_851".



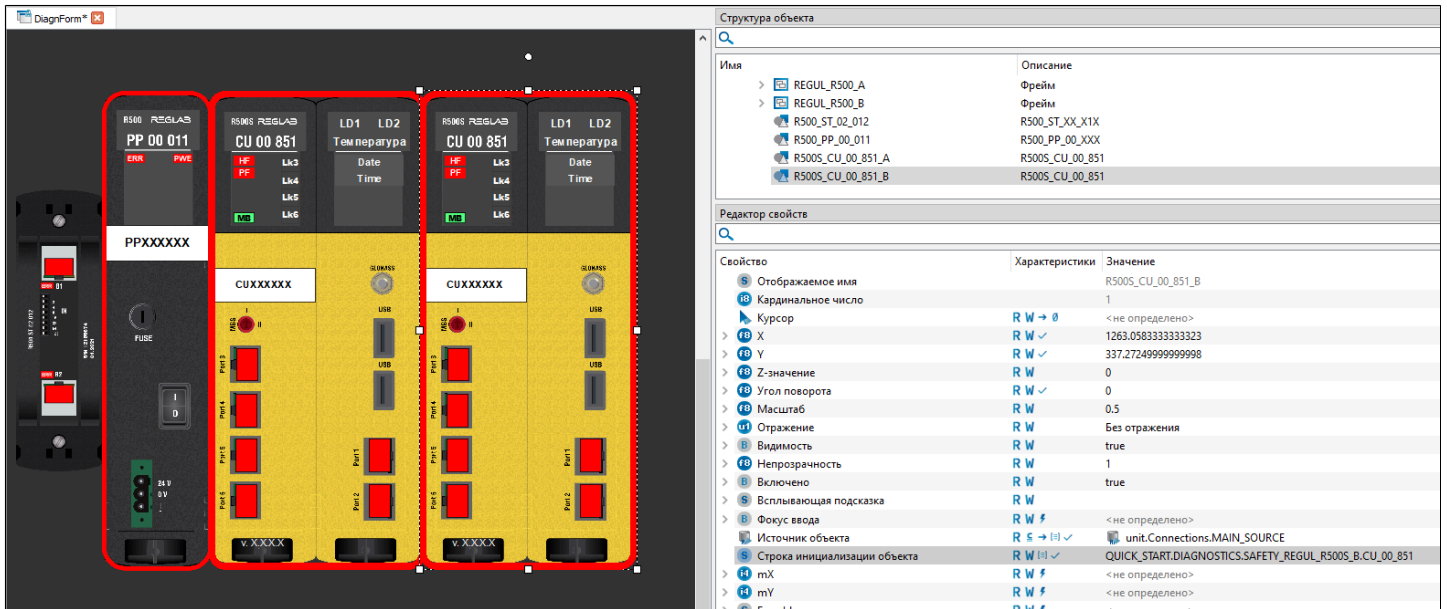
Скопируйте данный путь и вставьте в свойство "Строка инициализации объекта".



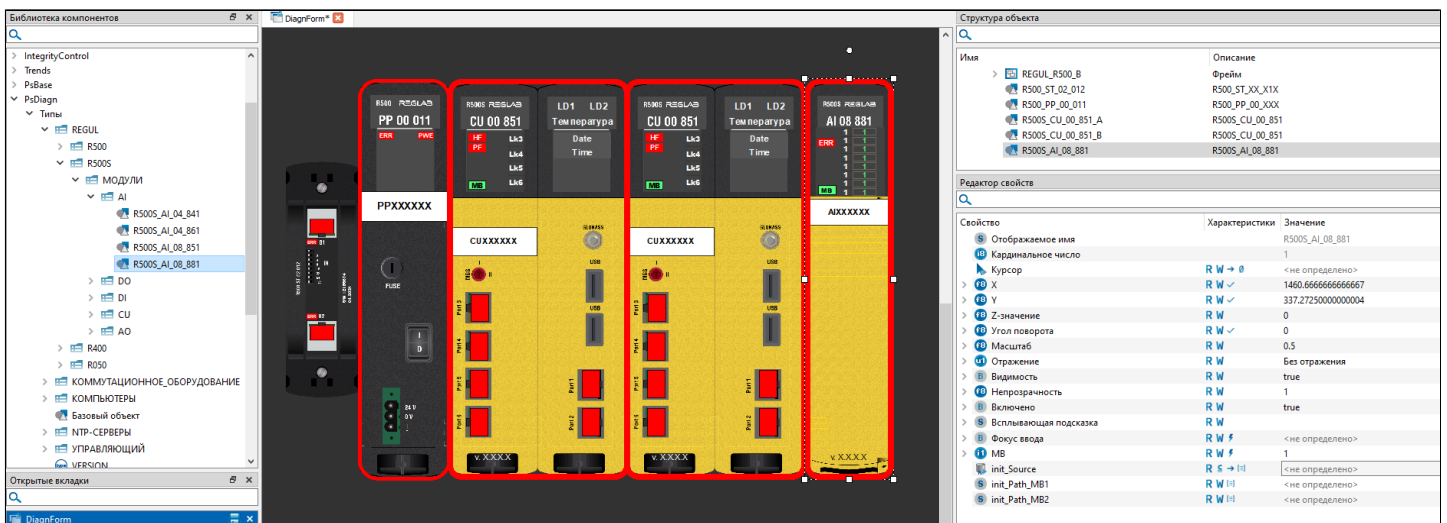
В результате должны получиться следующие свойства для модуля:

Редактор свойств		
🔍		
Свойство	Характеристики	Значение
Кардинальное число		1
Курсор	R W → 0	< не определено >
>  X	R W ✓	79.058333333332257
>  Y	R W ✓	-146.72750000000008
>  Z-значение	R W	0
>  Угол поворота	R W ✓	0
>  Масштаб	R W	0.5
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Фокус ввода	R W ✗	< не определено >
Источник объекта	R  →  ✓	unit.Connections.MAIN_SOURCE
Строка инициализации объекта	R W  ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CU_00_851
>  mX	R W ✗	< не определено >
>  mY	R W ✗	< не определено >
>  FormId	R W ✗	< не определено >
>  IP_PORT1	R W ✗	< не определено >
>  IP_PORT2	R W ✗	< не определено >
>  IP_PORT3	R W ✗	< не определено >
>  IP_PORT4	R W ✗	< не определено >
>  IP_PORT5	R W ✗	< не определено >
>  IP_PORT6	R W ✗	< не определено >
>  Работа в дублированном режиме	R W ✗ ✓	false
>  MB	R W ✗	1
init_Path_MB1	R W	< не определено >
init_Path_MB2	R W	< не определено >

7. Аналогичным образом добавьте и настройте второй (дублирующий) модуль центрального процессора. Во вкладке "Структура объекта" переименуйте его в "R500S\_CU\_00\_851\_B".



8. Добавьте на экранную форму "DiagnForm" модуль "R500S\_AI\_08\_881" из библиотеки "PsDiagn". Во вкладке "Структура объекта" переименуйте его в "R500S\_AI\_08\_881".



9. Настройте следующие свойства:

> Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Затем укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": "unit.Connections.MAIN\_SOURCE".

Структура объекта

Имя	Описание
> REGUL_R500_B	Фрейм
R500_ST_02_012	R500_ST_XX_X1X
R500_PP_00_011	R500_PP_00_XXX
R500S_CU_00_851_A	R500S_CU_00_851
R500S_CU_00_851_B	R500S_CU_00_851
R500S_AI_08_881	R500S_AI_08_881

Редактор свойств

Свойство	Характеристики	Значение
S Отображаемое имя		R500S_AI_08_881
i8 Кардинальное число		1
Курсор	R W → 0	<не определено>
> f8 X	R W ✓	1460.6666666666667
> f8 Y	R W ✓	337.27250000000004
> f8 Z-значение	R W	0
> f8 Угол поворота	R W ✓	0
> f8 Масштаб	R W	0.5
> u1 Отражение	R W	Без отражения
> B Видимость	R W	true
> f8 Непрозрачность	R W	1
> B Включено	R W	true
> S Всплывающая подсказка	R W	
> B Фокус ввода	R W ⚡	<не определено>
> i1 MB	R W ⚡	1
init_Source	R ⚡ → (=)	<не определено>
S init_Path_MB1	R W (=)	<не определено>
S init_Path_MB2	R W (=)	<не определено>

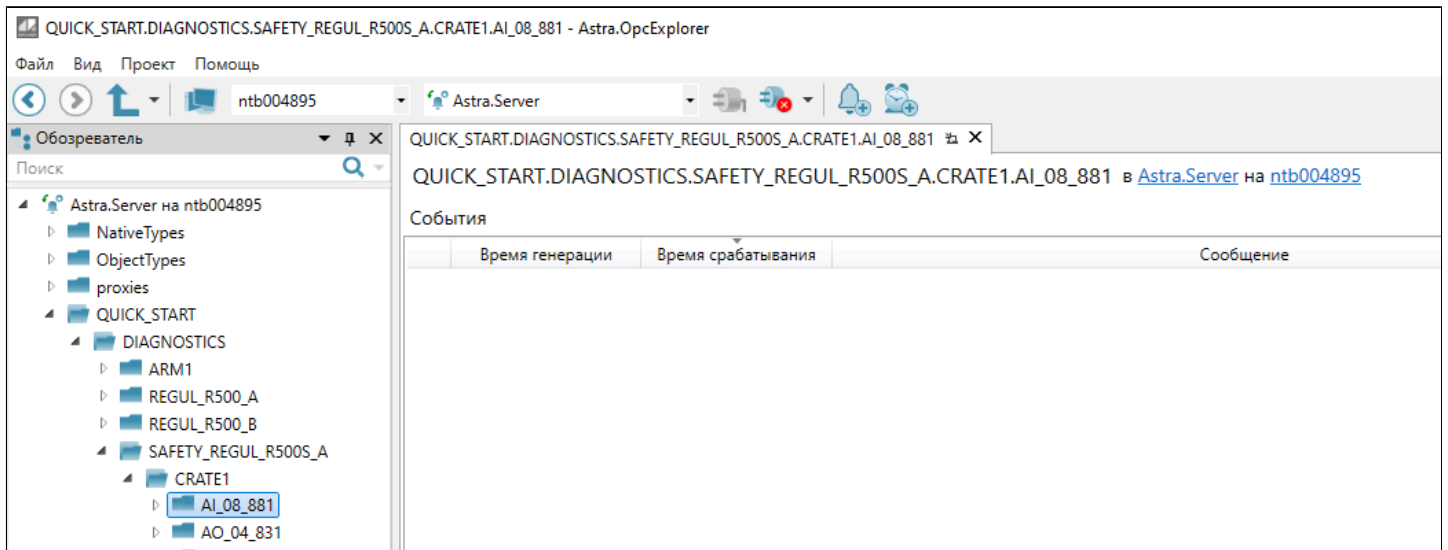
Сослаться  
Указать объект



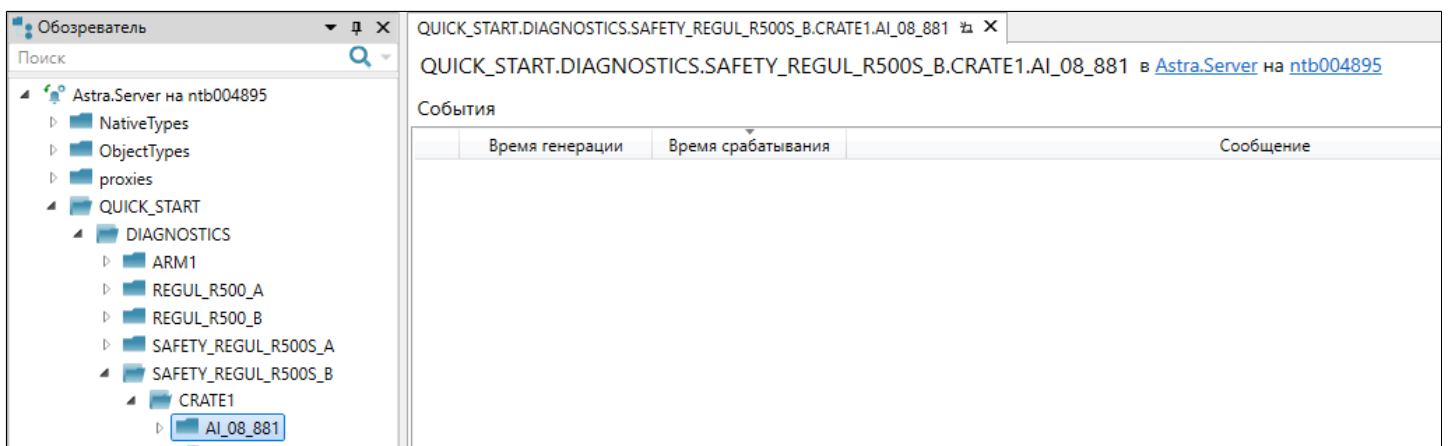
› init\_Path\_MB1 и init\_Path\_MB2 – для данных параметрах необходимо прописать путь до логических объектов модуля в SAFETY\_REGUL\_R500S\_A и SAFETY\_REGUL\_R500S\_B на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpcExplorer:











› "QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_A" → "CRATE1" → "AI\_08\_881"




























› "QUICK\_START" → "DIAGNOSTICS" → "SAFETY\_REGUL\_R500S\_B" → "CRATE1" → "AI\_08\_881"



Скопируйте данные пути и вставьте в свойства "init\_Path\_MB1" и "init\_Path\_MB2".

 init_Source	R  	 unit.Connections.MAIN_SOURCE
 init_Path_MB1	R W  	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CRATE1.AI_08_881
 init_Path_MB2	R W  	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_B.CRATE1.AI_08_881

В результате должны получиться следующие свойства для модуля:

Редактор свойств		
Q		
Свойство	Характеристики	Значение
 Отображаемое имя		R500S_AI_08_881
 Кардинальное число		1
 Курсор	R W → 0	< не определено >
>  X	R W ✓	430.66666666666652
>  Y	R W ✓	-146.72749999999996
>  Z-значение	R W	0
>  Угол поворота	R W ✓	0
>  Масштаб	R W	0.5
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Фокус ввода	R W ✗	< не определено >
>  MB	R W ✗	1
 init_Source	R  →  ✓	 unit.Connections.MAIN_SOURCE
 init_Path_MB1	R W   ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_A.CRATE1.AI_08_881
 init_Path_MB2	R W   ✓	QUICK_START.DIAGNOSTICS.SAFETY_REGUL_R500S_B.CRATE1.AI_08_881

11. По аналогии модуля добавьте и настройте все модули крейта "SAFETY\_REGUL\_R500S". Уберите из названий типов хвосты "\_1".

The screenshot displays the DiagnForm software interface. The main window shows a rack of modules with the following labels from left to right: PP000000, CU000000, CU000000, AI080801, AO040801, DI280801, and DO160801. The right-hand side of the interface contains two panels: 'Структура объекта' (Object Structure) and 'Редактор свойств' (Property Editor).

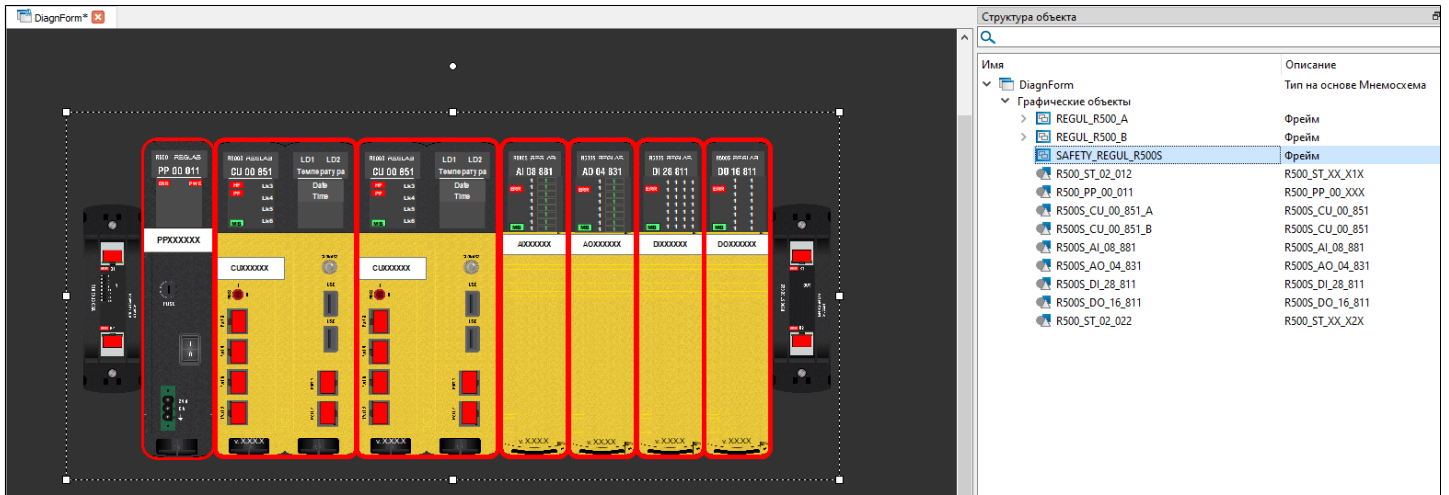
**Структура объекта (Object Structure):**

Имя	Описание
R500_ST_02_012	R500_ST_XX_X1X
R500_PP_00_011	R500_PP_00_XXX
R500S_CU_00_851_A	R500S_CU_00_851
R500S_CU_00_851_B	R500S_CU_00_851
R500S_AI_08_881	R500S_AI_08_881
R500S_AO_04_831	R500S_AO_04_831
R500S_DI_28_811	R500S_DI_28_811
R500S_DO_16_811	R500S_DO_16_811
R500_ST_02_022	R500_ST_XX_X2X

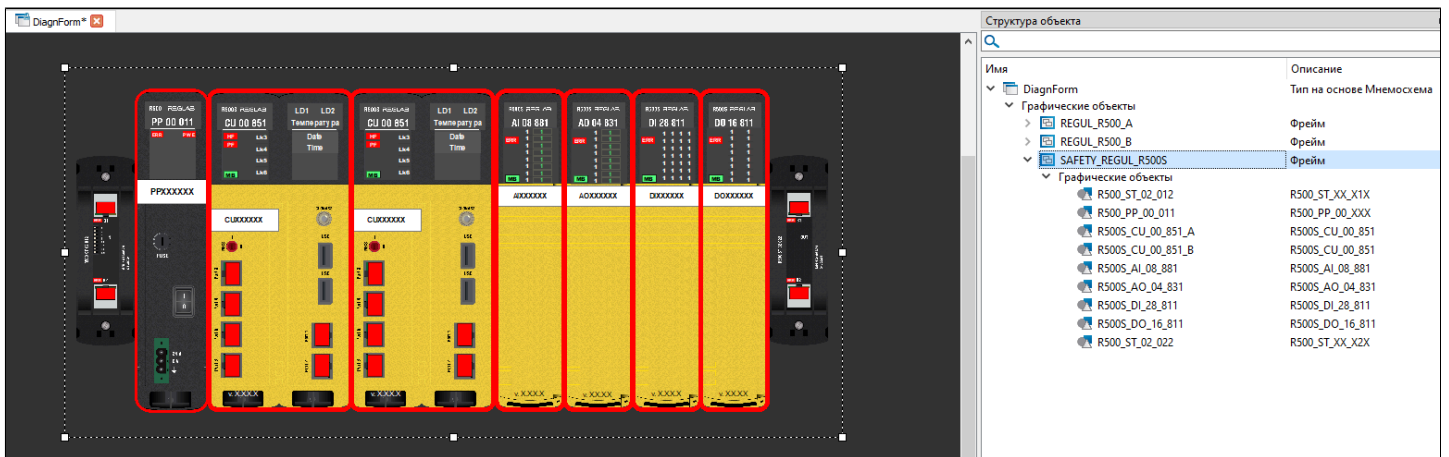
**Редактор свойств (Property Editor):**

Свойство	Характеристики	Знач
Отображаемое имя		Diagn
Модификатор доступа		
Горизонтальная полоса прокрутки окна	R W → 0	< не с
Вертикальная полоса прокрутки окна	R W → 0	< не с
Курсор	R W → 0	< не с
	R W	

12. Добавьте элемент "Фрейм" из вкладки общих элементов "Визуальные" панели элементов на форму "DiagnForm". Задайте размер фрейма, чтобы в нем поместились все, добавленные ранее, модули. В редакторе свойств фрейма задайте имя "SAFETY\_REGUL\_R500S".

































13. Перетащите в дереве объектов все созданные модули в крейт "SAFETY\_REGUL\_R500S".



Конфигурация крейта ПЛК Safety завершена.

14. В редакторе свойств фрейма "SAFETY\_REGUL\_R500S" задайте свойствам "Стиль пера" значение "Нет пера" и "Масштаб" – "0.9".

Структура объекта		
		
Имя	Описание	
<ul style="list-style-type: none"> <li>▼  DiagnForm <ul style="list-style-type: none"> <li>▼ Графические объекты <ul style="list-style-type: none"> <li>&gt;  REGUL_R500_A      Фрейм</li> <li>&gt;  REGUL_R500_B      Фрейм</li> <li>▼  SAFETY_REGUL_R500S      Фрейм <ul style="list-style-type: none"> <li>▼ Графические объекты</li> </ul> </li> </ul> </li> </ul> </li> </ul>	Тип на основе Мнемосхема	
Редактор свойств		
		
Свойство	Характеристики	Значение
 Горизонтальная полоса прокрутки	R W → 0	< не определено >
 Вертикальная полоса прокрутки	R W → 0	< не определено >
 Курсор	R W → 0	< не определено >
>  X	R W	932
>  Y	R W	193
>  Z-значение	R W	0
>  Угол поворота	R W	0
>  Масштаб	R W	0.900000000000000002
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Ширина	R W	840.0000000000000091
>  Высота	R W	400
>  Фокус ввода	R W ⚡ ⓘ	< не определено >
>  Цвет пера	R W	4278190080
>  Стиль пера	R W	Нет пера
>  Толщина пера	R W	1
>  Цвет заливки	R W	4278190080
>  Стиль заливки	R W	Нет заливки
>  Ручное управление масштабом	R W	false
>  Отображать полосы прокрутки	R W	true
>  Перемещение мышью	R W	false

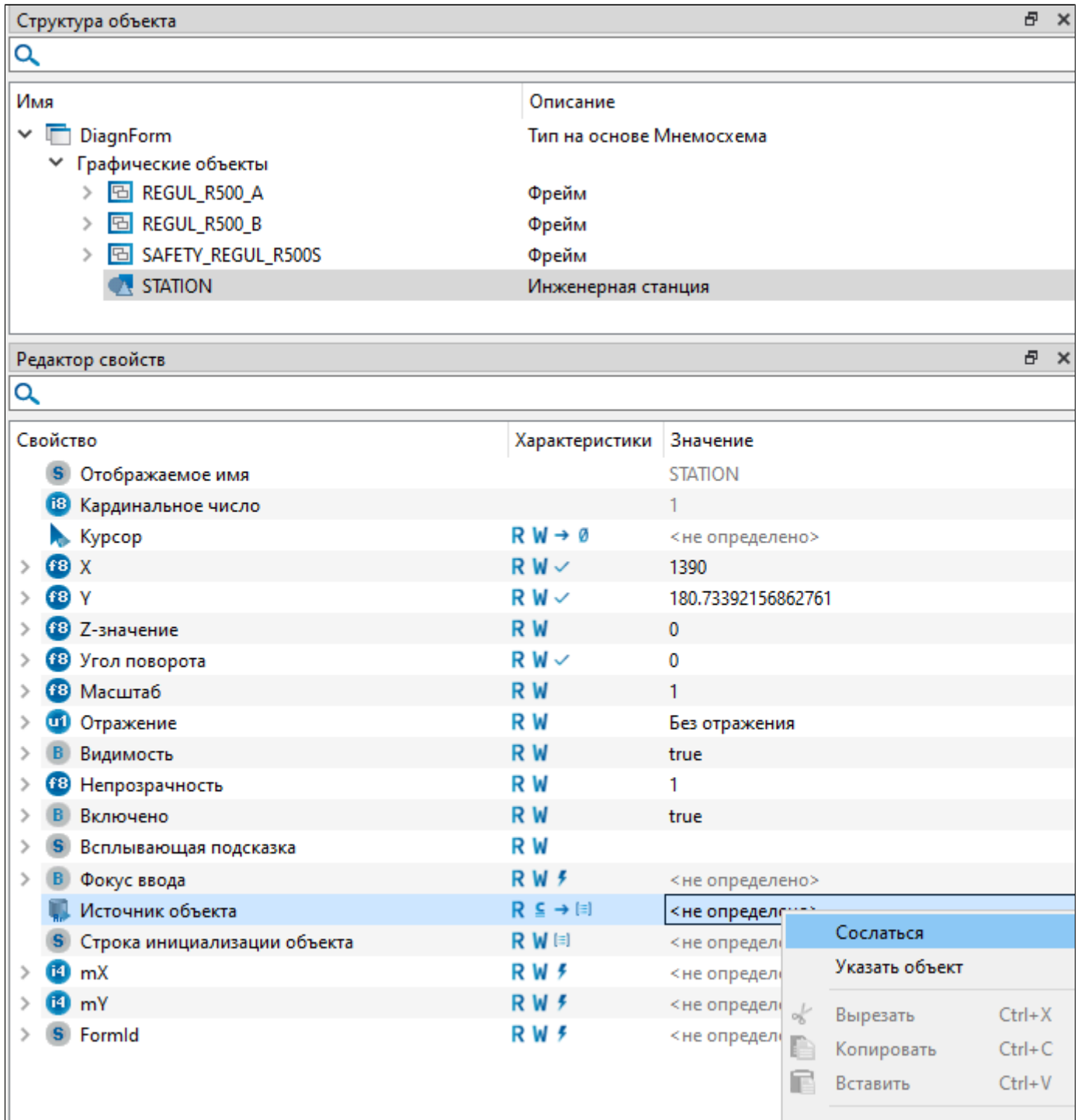
# 1.1.4.5.4.3. Диагностика инженерной станции

1. Добавьте на экранную форму "DiagnForm" тип "Инженерная станция" из библиотеки "PsDiagn", задайте имя "STATION".



## 2. Настройте следующие свойства:

➤ Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": unit.Connections.MAIN\_SOURCE.



The screenshot displays two windows from a software application:

- Структура объекта (Object Structure):** A tree view showing the hierarchy of objects. The 'STATION' object is selected, which is an 'Инженерная станция' (Engineering Station).
- Редактор свойств (Properties Editor):** A table listing the properties of the selected object. The 'Источник объекта' (Object Source) property is highlighted, and a context menu is open over it, showing the 'Сослаться' (Link) option.

Свойство	Характеристики	Значение
Отображаемое имя		STATION
Кардинальное число		1
Курсор	R W → 0	<не определено>
X	R W ✓	1390
Y	R W ✓	180.73392156862761
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ⚡	<не определено>
Источник объекта	R ⊆ → (=)	<не определено>
Строка инициализации объекта	R W (=)	<не определено>
mX	R W ⚡	<не определено>
mY	R W ⚡	<не определено>
FormId	R W ⚡	<не определено>

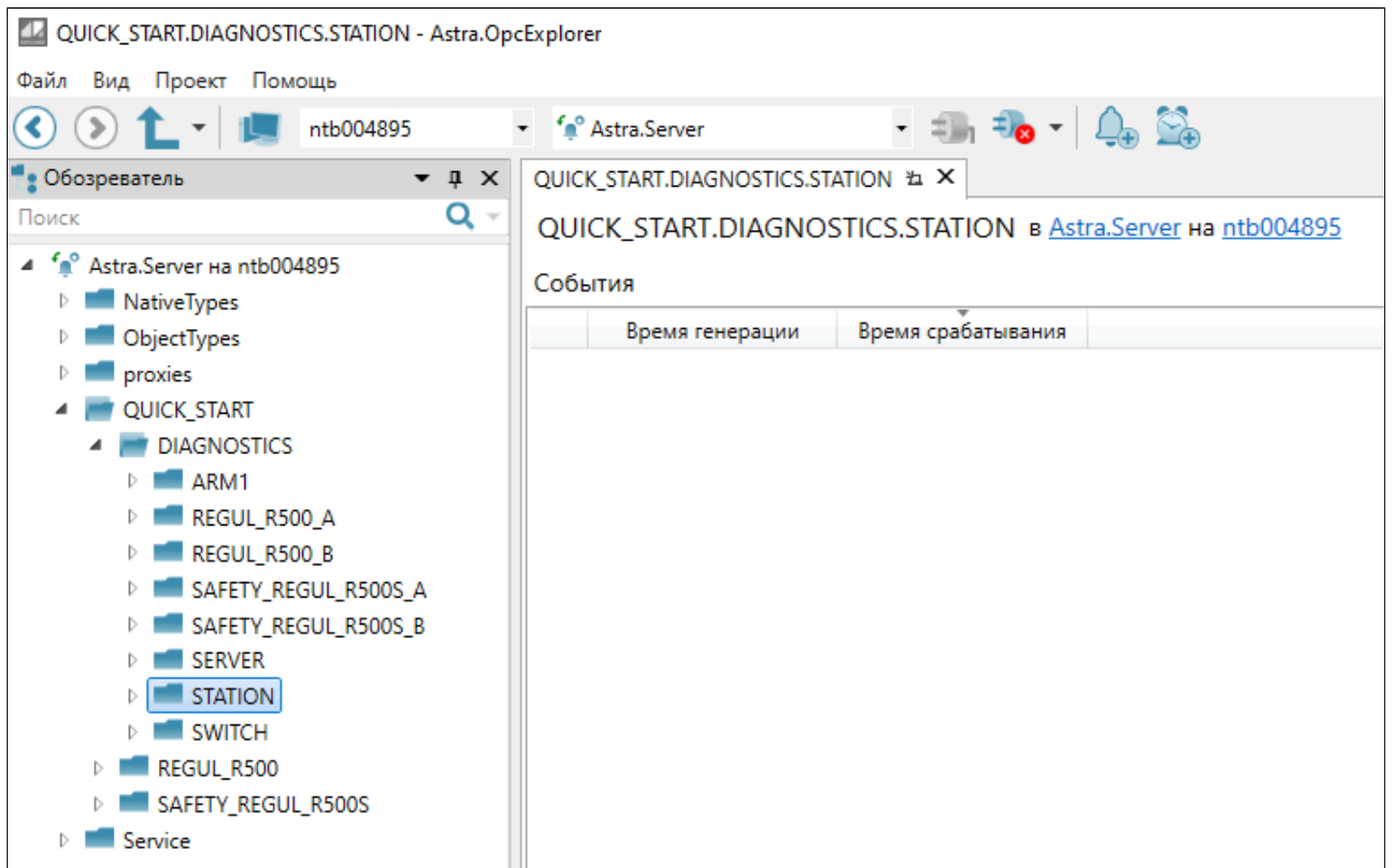
The context menu for the 'Источник объекта' property includes the following options:

- Сослаться (Link)
- Указать объект (Point to Object)
- Вырезать (Cut) - Ctrl+X
- Копировать (Copy) - Ctrl+C
- Вставить (Paste) - Ctrl+V



➤ Строка инициализации – здесь необходимо прописать путь до логического объекта инженерной станции "STATION" на сервере AstraServer.







Полный путь до объекта можно посмотреть в утилите OpсExplorer: "QUICK\_START" → "DIAGNOSTICS" → "STATION".



























Скопируйте данный путь и вставьте в свойство "Строка инициализации".



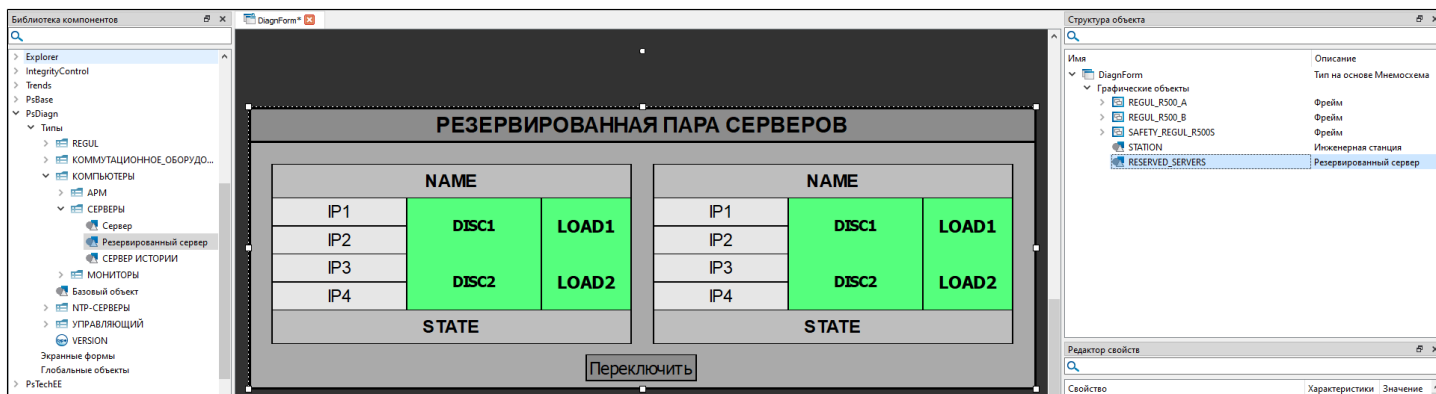
В результате должны получиться следующие свойства для инженерной станции:

Структура объекта	
	
Имя	Описание
<ul style="list-style-type: none"> <li>▼  DiagnForm           <ul style="list-style-type: none"> <li>▼ Графические объекты               <ul style="list-style-type: none"> <li>&gt;  REGUL_R500_A      Фрейм</li> <li>&gt;  REGUL_R500_B      Фрейм</li> <li>&gt;  SAFETY_REGUL_R500S      Фрейм</li> <li> STATION      Инженерная станция</li> </ul> </li> </ul> </li> </ul>	Тип на основе Мнемосхема

Редактор свойств		
		
Свойство	Характеристики	Значение
 Отображаемое имя		STATION
 Кардинальное число		1
 Курсор	R W → 0	<не определено>
>  X	R W ✓	1390
>  Y	R W ✓	180.73392156862761
>  Z-значение	R W	0
>  Угол поворота	R W ✓	0
>  Масштаб	R W	1
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Фокус ввода	R W ⚡	<не определено>
 Источник объекта	R  →  ✓	 unit.Connections.MAIN_SOURCE
 Строка инициализации объекта	R W  ✓	QUICK_START.DIAGNOSTICS.STATION
>  mX	R W ⚡	<не определено>
>  mY	R W ⚡	<не определено>
>  FormId	R W ⚡	<не определено>

## 1.1.4.5.4.4. Диагностика серверов ввода/вывода

1. Добавьте на экранную форму "DiagnForm" тип "Резервированный сервер" из библиотеки "PsDiagn", задайте имя "RESERVED\_SERVERS".



## 2. Настройте следующие свойства:

› Источник сервера 1 основной – здесь необходимо сослаться на основной источник AP для основного сервера. Кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "AP\_SERVER1\_SOURCE\_LINE1" из глобального объекта "Connections":  
`unit.Connections.AP_SERVER1_SOURCE_LINE1`.

› Источник сервера 2 основной – здесь необходимо сослаться на основной источник AP для резервного сервера. Кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "AP\_SERVER2\_SOURCE\_LINE1" из глобального объекта "Connections":  
`unit.Connections.AP_SERVER2_SOURCE_LINE1`.

› Источник сервера 1 резервный – здесь необходимо сослаться на резервный источник AP для основного сервера. Кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "AP\_SERVER1\_SOURCE\_LINE2" из глобального объекта "Connections":  
`unit.Connections.AP_SERVER1_SOURCE_LINE2`.

› Источник сервера 2 резервный – здесь необходимо сослаться на резервный источник AP для резервного сервера. Кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "AP\_SERVER2\_SOURCE\_LINE2" из глобального объекта "Connections":  
`unit.Connections.AP_SERVER2_SOURCE_LINE2`.

Редактор свойств		
🔍		
Свойство	Характеристики	Значение
🅈 Отображаемое имя		RESERVED_SERVERS
🅈 Кардинальное число		1
🖱️ Курсор	R W → 0	<не определено>
> 🅈 X	R W ✓	1136
> 🅈 Y	R W ✓	532.94524999999999
> 🅈 Z-значение	R W	0
> 🅈 Угол поворота	R W ✓	0
> 🅈 Масштаб	R W	1
> 🅈 Отражение	R W	Без отражения
> 🅈 Видимость	R W	true
> 🅈 Непрозрачность	R W	1
> 🅈 Включено	R W	true
> 🅈 Всплывающая подсказка	R W	
> 🅈 Фокус ввода	R W ⚡	<не определено>
🖱️ Источник сервера 1 основной	R ⚡ → [=]	<не определено>
🖱️ Источник сервера 2 основной	R ⚡ → [=]	<не определено>
> 🅈 Имя сервера	R W ⚡	Резервированные сервера
🖱️ Источник сервера 1 резервный	R ⚡ → [=]	<не определено>
🖱️ Источник сервера 2 резервный	R ⚡ → [=]	<не определено>
🅈 Строка инициализации сервера 1	R W [=]	<не определено>
🅈 Строка инициализации сервера 2	R W [=]	<не определено>

Сослаться

Указать объект

---

✂ Вырезать Ctrl+X

📄 Копировать Ctrl+C

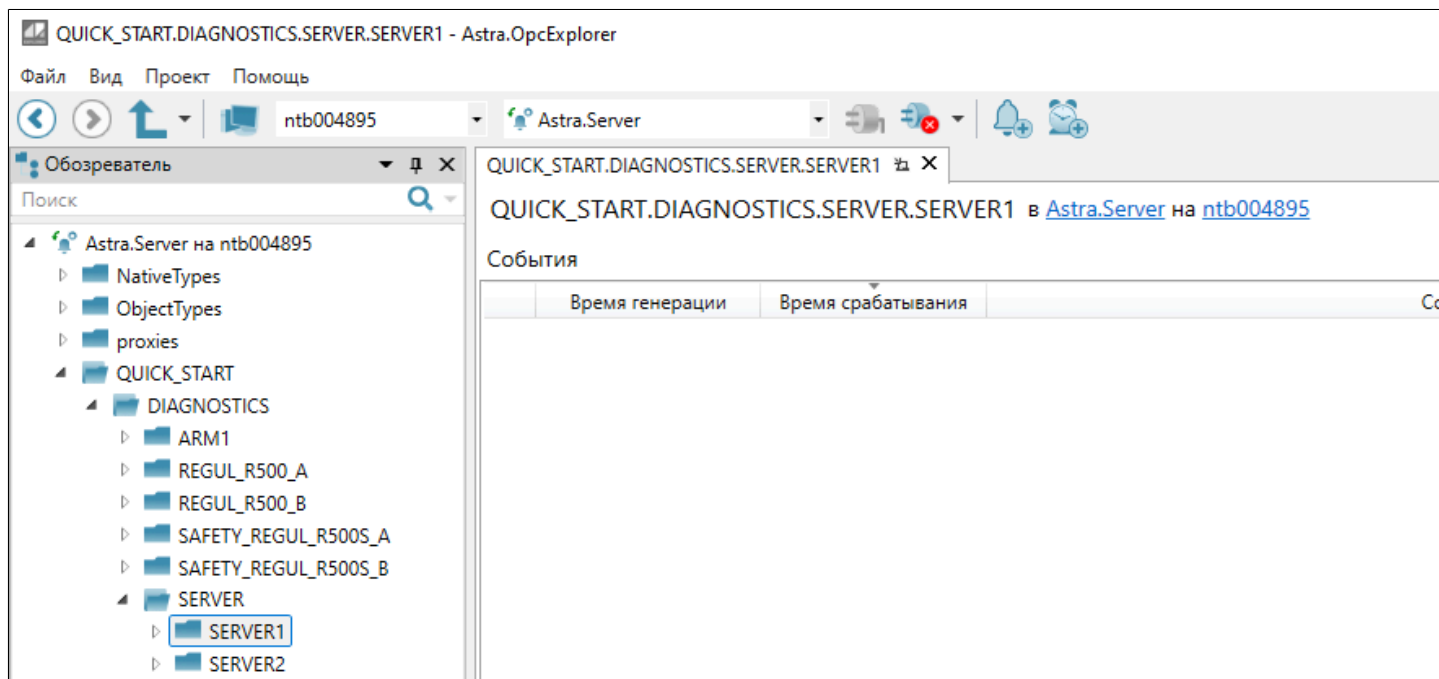
📄 Вставить Ctrl+V

🖱️ Источник сервера 1 основной	R ⚡ → [=] ✓	🖱️ unit.Connections.AP_SERVER1_SOURCE_LINE1
🖱️ Источник сервера 2 основной	R ⚡ → [=] ✓	🖱️ unit.Connections.AP_SERVER2_SOURCE_LINE1
> 🅈 Имя сервера	R W ⚡	Резервированные сервера
🖱️ Источник сервера 1 резервный	R ⚡ → [=] ✓	🖱️ unit.Connections.AP_SERVER1_SOURCE_LINE2
🖱️ Источник сервера 2 резервный	R ⚡ → [=] ✓	🖱️ unit.Connections.AP_SERVER2_SOURCE_LINE2

➤ Строка инициализации сервера 1 – здесь необходимо прописать путь до логического объекта сервера "SERVER1" на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OrcExplorer: "QUICK\_START" → "DIAGNOSTICS" → "SERVER" → "SERVER1".

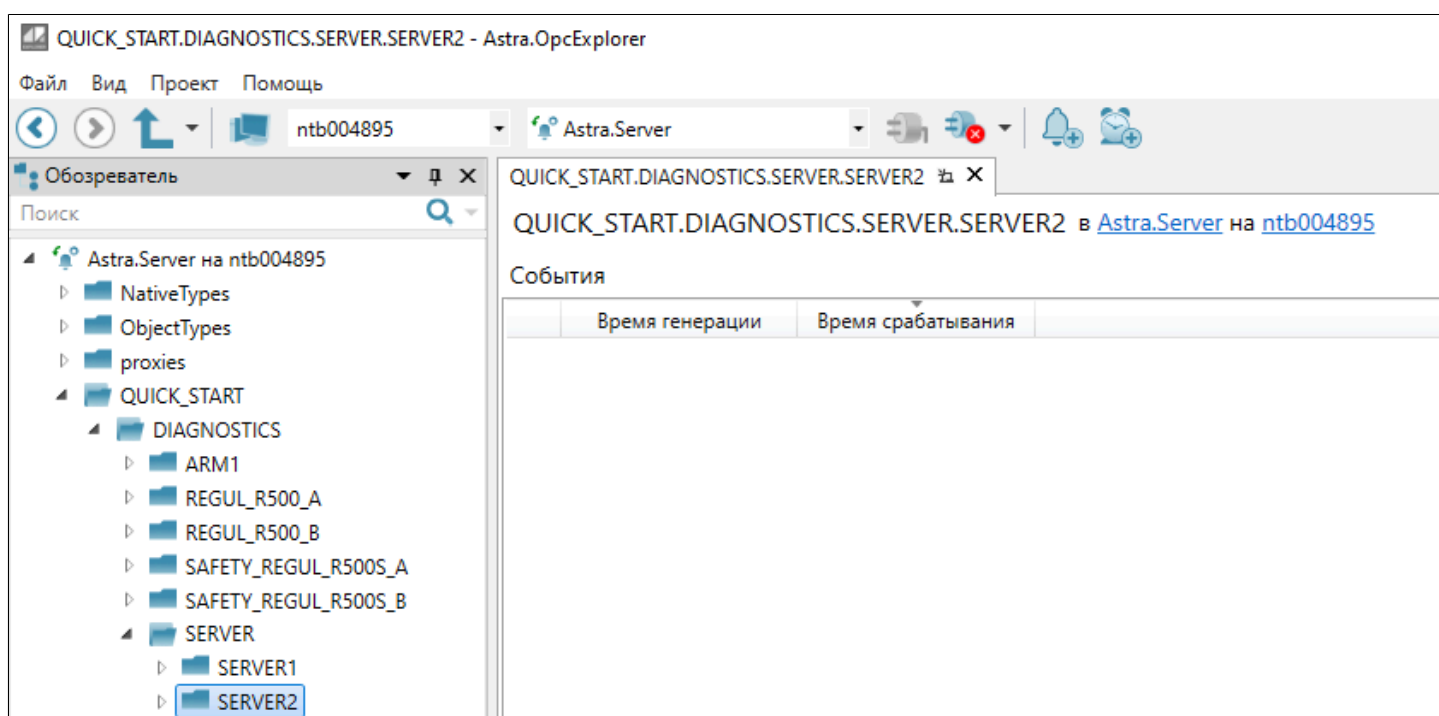
Скопируйте данный путь и вставьте в свойство "Строка инициализации сервера 1".



➤ Строка инициализации сервера 2 – здесь необходимо прописать путь до логического объекта сервера "SERVER2" на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpcExplorer: QUICK\_START → DIAGNOSTICS → SERVER → SERVER2.

Скопируйте данный путь и вставьте в свойство "Строка инициализации сервера 2".



В результате должны получиться следующие свойства для резервированных серверов:

Структура объекта

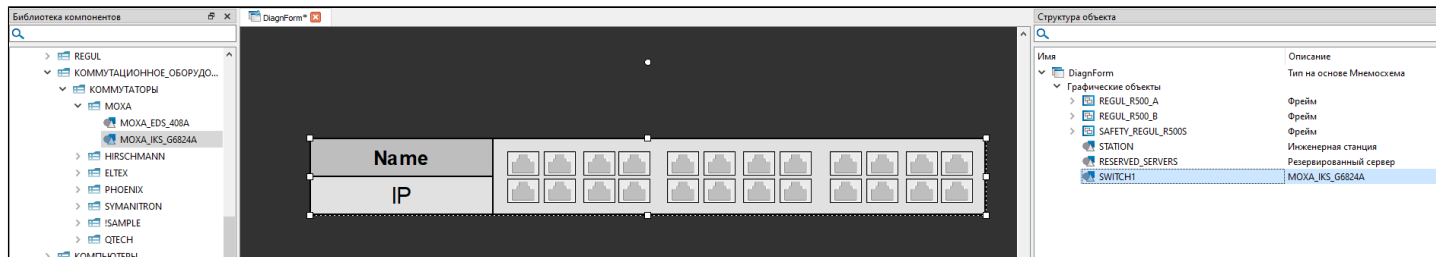
Имя	Описание
DiagnForm	Тип на основе Мнемосхема
Графические объекты	
> REGUL_R500_A	Фрейм
> REGUL_R500_B	Фрейм
> SAFETY_REGUL_R500S	Фрейм
STATION	Инженерная станция
RESERVED_SERVERS	Резервированный сервер

Редактор свойств

Свойство	Характеристики	Значение
<b>S</b> Отображаемое имя		RESERVED_SERVERS
<b>i8</b> Кардинальное число		1
<b>⊞</b> Курсор	R W → 0	<не определено>
> <b>f8</b> X	R W ✓	1136
> <b>f8</b> Y	R W ✓	532.9452499999999
> <b>f8</b> Z-значение	R W	0
> <b>f8</b> Угол поворота	R W ✓	0
> <b>f8</b> Масштаб	R W	1
> <b>u1</b> Отражение	R W	Без отражения
> <b>B</b> Видимость	R W	true
> <b>f8</b> Непрозрачность	R W	1
> <b>B</b> Включено	R W	true
> <b>S</b> Всплывающая подсказка	R W	
> <b>B</b> Фокус ввода	R W ⚡	<не определено>
<b>⊞</b> Источник сервера 1 основной	R ⊞ → ⊞ ✓	<b>⊞</b> unit.Connections.AP_SERVER1_SOURCE_LINE1
<b>⊞</b> Источник сервера 2 основной	R ⊞ → ⊞ ✓	<b>⊞</b> unit.Connections.AP_SERVER2_SOURCE_LINE1
> <b>S</b> Имя сервера	R W ⚡	Резервированные сервера
<b>⊞</b> Источник сервера 1 резервный	R ⊞ → ⊞ ✓	<b>⊞</b> unit.Connections.AP_SERVER1_SOURCE_LINE2
<b>⊞</b> Источник сервера 2 резервный	R ⊞ → ⊞ ✓	<b>⊞</b> unit.Connections.AP_SERVER2_SOURCE_LINE2
<b>S</b> Строка инициализации сервера 1	R W ⊞ ✓	QUICK_START.DIAGNOSTICS.SERVER.SERVER1
<b>S</b> Строка инициализации сервера 2	R W ⊞ ✓	QUICK_START.DIAGNOSTICS.SERVER.SERVER2

## 1.1.4.5.4.5. Диагностика коммутаторов

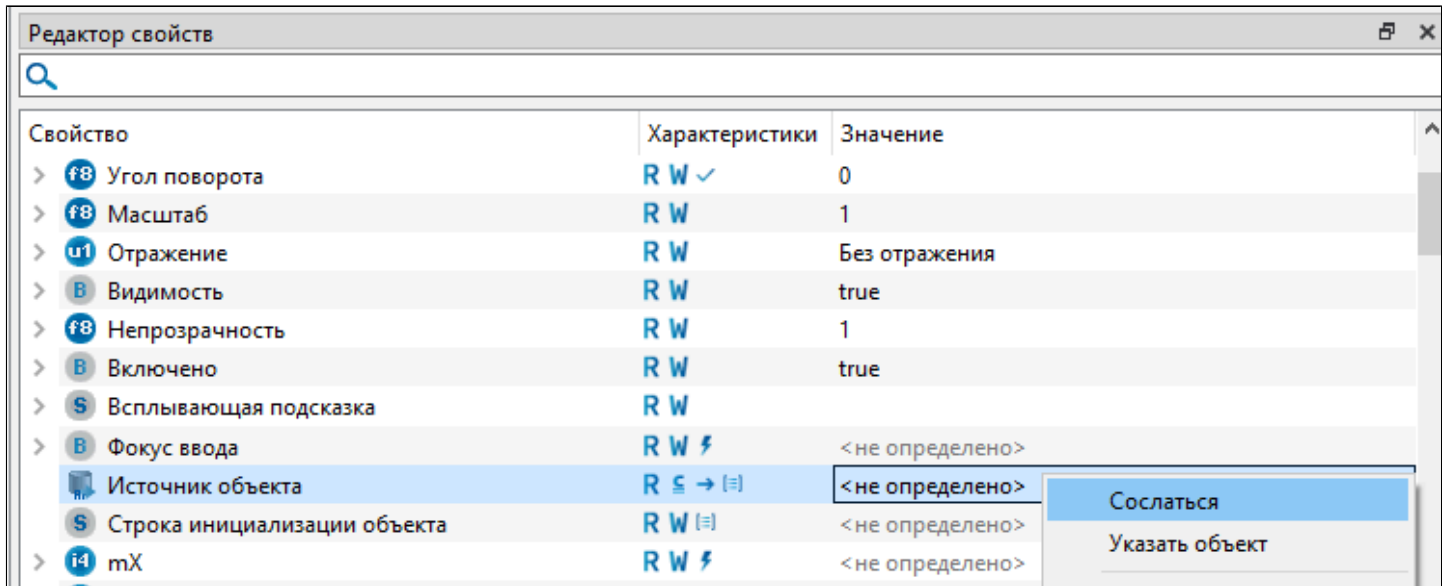
1. Добавьте на экранную форму "DiagnForm" тип "MOXA\_IKS\_G6824A" из библиотеки "PsDiagn", задайте имя "SWITCH1".





## 2. Настройте следующие свойства:

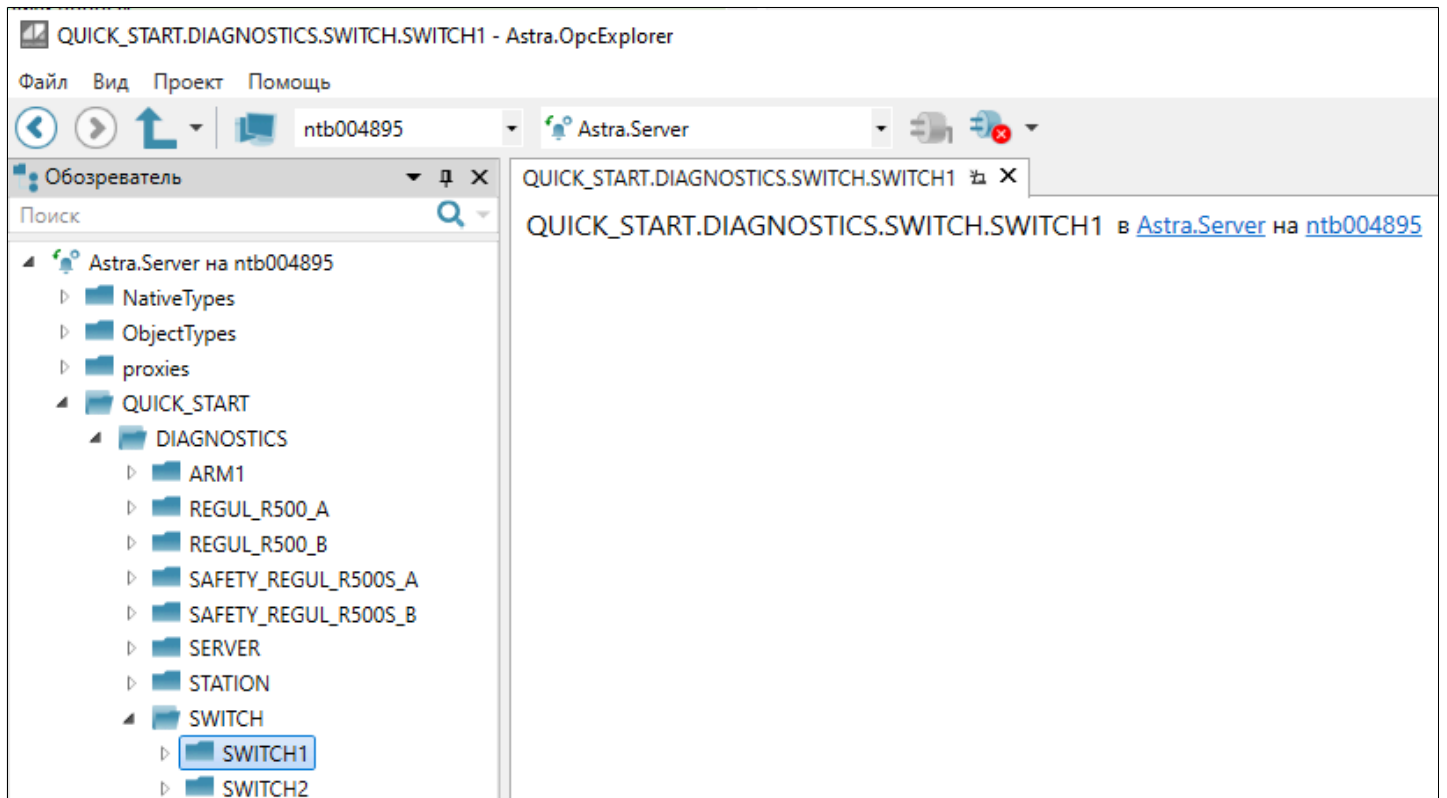
› Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": unit.Connections.MAIN\_SOURCE.



› Строка инициализации объекта – здесь необходимо прописать путь до логического объекта коммутатора "SWITCH1" на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OrcExplorer: "QUICK\_START" → "DIAGNOSTICS" → "SWITCH" → "SWITCH1".

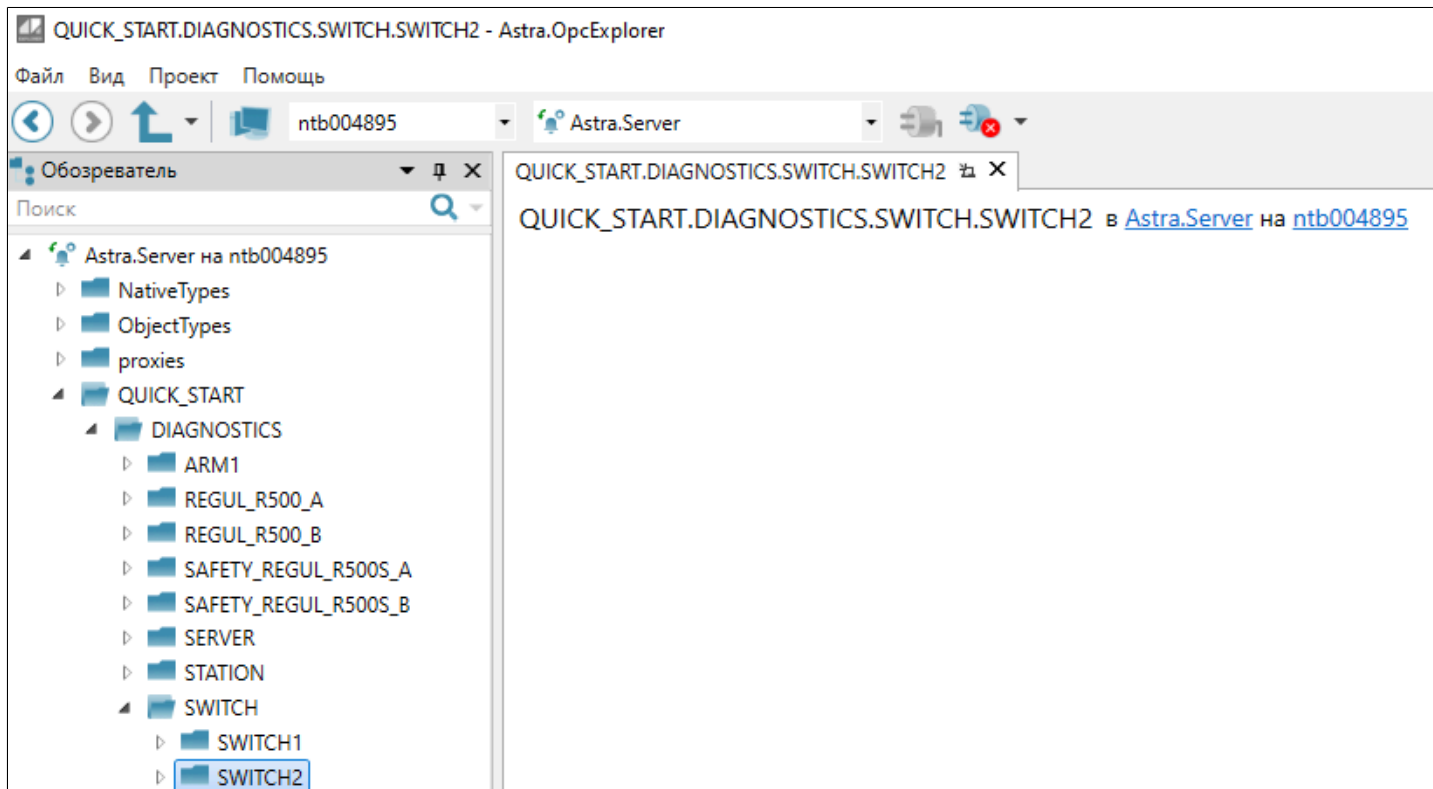
Скопируйте данный путь и вставьте в свойство "Строка инициализации объекта".



Редактор свойств

Свойство	Характеристики	Значение
<b>S</b> Отображаемое имя		SWITCH1
<b>i8</b> Кардинальное число		1
> <b>f8</b> X	R W ✓	988
> <b>f8</b> Y	R W ✓	371
> <b>f8</b> Z-значение	R W	0
> <b>f8</b> Угол поворота	R W ✓	0
> <b>f8</b> Масштаб	R W	1
> <b>u1</b> Отражение	R W	Без отражения
> <b>B</b> Видимость	R W	true
> <b>f8</b> Непрозрачность	R W	1
> <b>B</b> Включено	R W	true
> <b>S</b> Всплывающая подсказка	R W	
> <b>B</b> Фокус ввода	R W ⚡	<не определено>
<b>🖨</b> Источник объекта	R ⚡ → [≡] ✓	unit.Connections.MAIN_SOURCE
<b>S</b> Строка инициализации объекта	R W [≡] ✓	QUICK_START.DIAGNOSTICS.SWITCH.SWITCH1

3. Аналогичным образом необходимо добавить на экранную форму "DiagnForm" второй коммутатор "SWITCH2".



## 1.1.4.5.4.6. Диагностика АРМ

1. Добавьте на экранную форму "DiagnForm" тип "АРМ ОПЕРАТОРА" из библиотеки "PsDiagn", задайте имя "ARM1".

The screenshot displays a software development interface for creating a diagnostic form. The central workspace shows a monitor icon with a green screen displaying the text "NAME" and "CPU LOAD". Below the monitor are two rectangular boxes labeled "IP1" and "IP2".

The left pane, titled "Библиотека компонентов", shows a tree view of components. Under "PsDiagn" > "АРМ", the "АРМ ОПЕРАТОРА" component is selected.

The right pane, titled "Структура объекта", shows a tree view of the form's structure. Under "DiagnForm", the "АРМ1" component is listed.

The bottom right pane, titled "Редактор свойств", shows the properties of the selected "АРМ1" component. The table below represents the data shown in this editor:

Свойство	Характеристики	Значение
Отображаемое имя		DiagnForm
Модификатор доступа		Приватный
Горизонтальная полоса прокрутки окна	RW → 0	< не определено >
Вертикальная полоса прокрутки окна	RW → 0	< не определено >
Курсор	RW → 0	< не определено >
X	RW	0
Y	RW	0
Z-значение	RW ✓	0
Угол поворота	RW	0
Масштаб	RW ✓	1

## 2. Настройте следующие свойства:

➤ Источник объекта – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": unit.Connections.MAIN\_SOURCE.

The screenshot shows two windows: 'Структура объекта' (Object Structure) and 'Редактор свойств' (Properties Editor).

**Структура объекта:**

Имя	Описание
DiagnForm	Тип на основе Мнемосхема
Графические объекты	
REGUL_R500_A	Фрейм
REGUL_R500_B	Фрейм
SAFETY_REGUL_R500S	Фрейм
STATION	Инженерная станция
RESERVED_SERVERS	Резервированный сервер
SWITCH1	МОХА_IKS_G6824A
SWITCH2	МОХА_IKS_G6824A
ARM1	АРМ ОПЕРАТОРА

**Редактор свойств:**

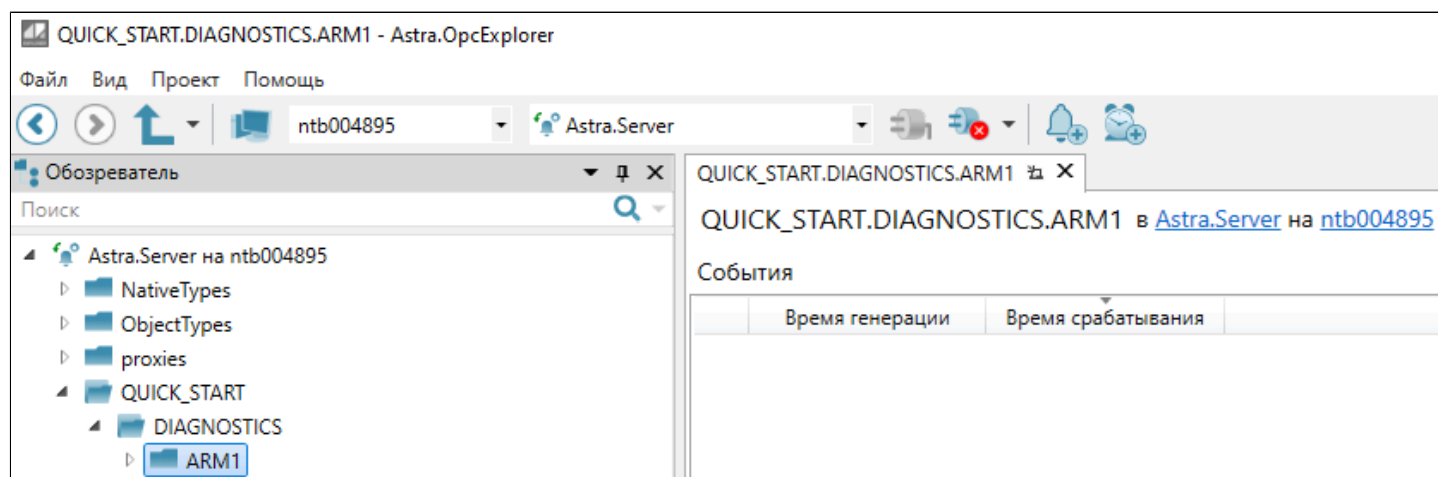
Свойство	Характеристики	Значение
Отображаемое имя		ARM1
Кардинальное число		1
Курсор	R W → 0	<не определено>
X	R W ✓	1717.5
Y	R W ✓	214.98392156862758
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ⚡	<не определено>
Источник объекта	R ↗ → (=)	<не определено>
Строка инициализации объекта	R W (=)	<не определено>
mX	R W ⚡	<не определено>

The context menu for the 'Источник объекта' property is open, showing options: 'Сослаться' (Link) and 'Указать объект' (Specify object).

➤ Строка инициализации объекта – здесь необходимо прописать путь до логического объекта "ARM1" на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpсExplorer: "QUICK\_START" → "DIAGNOSTICS" → "ARM1".

Скопируйте данный путь и вставьте в свойство "Строка инициализации объекта".



В результате должны получиться следующие свойства для резервированных серверов:

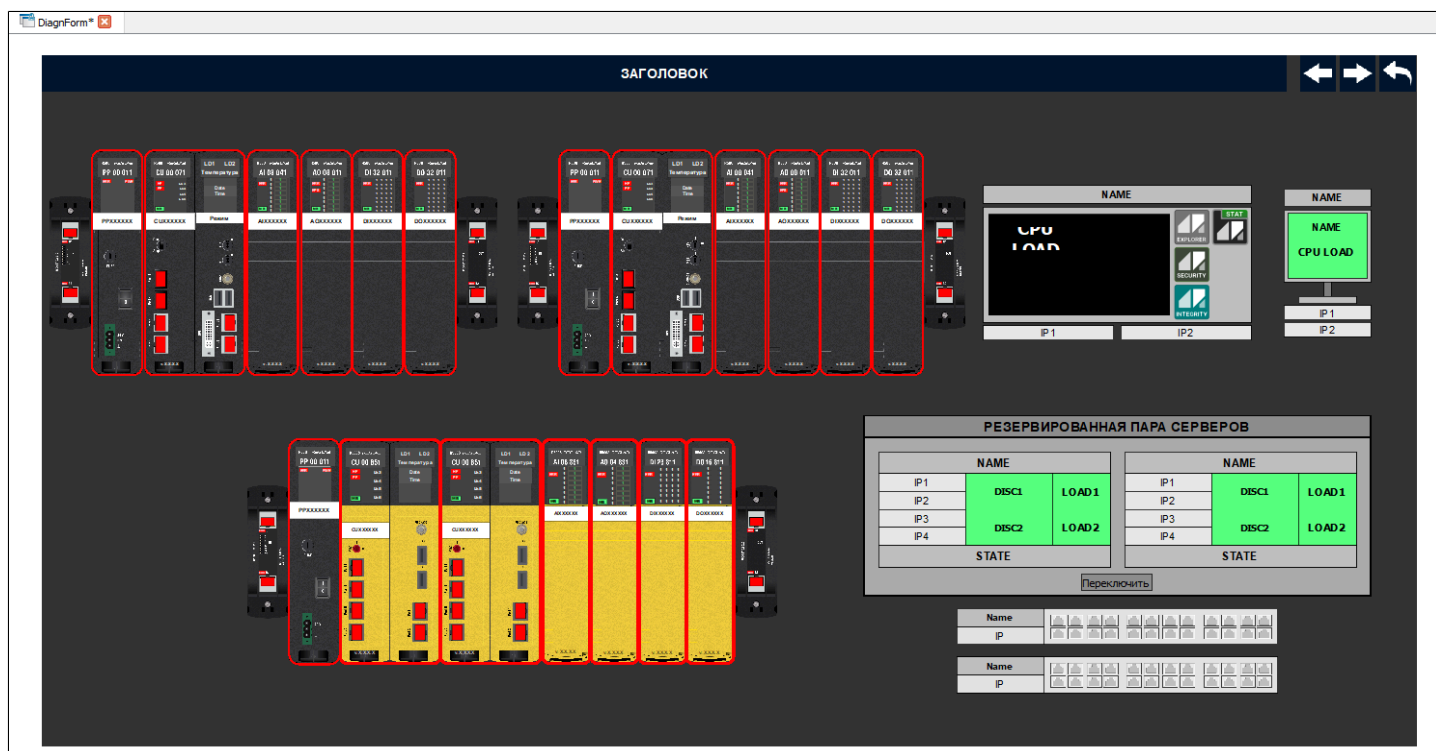
Редактор свойств



Свойство	Характеристики	Значение
<b>S</b> Отображаемое имя		ARM1
<b>i8</b> Кардинальное число		1
Курсор	R W → 0	<не определено>
> <b>f8</b> X	R W ✓	1717.5
> <b>f8</b> Y	R W ✓	214.98392156862758
> <b>f8</b> Z-значение	R W	0
> <b>f8</b> Угол поворота	R W ✓	0
> <b>f8</b> Масштаб	R W	1
> <b>u1</b> Отражение	R W	Без отражения
> <b>B</b> Видимость	R W	true
> <b>f8</b> Непрозрачность	R W	1
> <b>B</b> Включено	R W	true
> <b>S</b> Всплывающая подсказка	R W	
> <b>B</b> Фокус ввода	R W ⚡	<не определено>
Источник объекта	R $\leq$ → (≡) ✓	unit.Connections.MAIN_SOURCE
<b>S</b> Строка инициализации объекта	R W (≡) ✓	QUICK_START.DIAGNOSTICS.ARM1
> <b>i4</b> mX	R W ⚡	<не определено>
> <b>i4</b> mY	R W ⚡	<не определено>
> <b>S</b> FormId	R W ⚡	<не определено>

# 1.1.4.5.4.7. Структурирование формы диагностики

Распределите все добавленные объекты на экранной форме. Экранная форма диагностики примет вид:

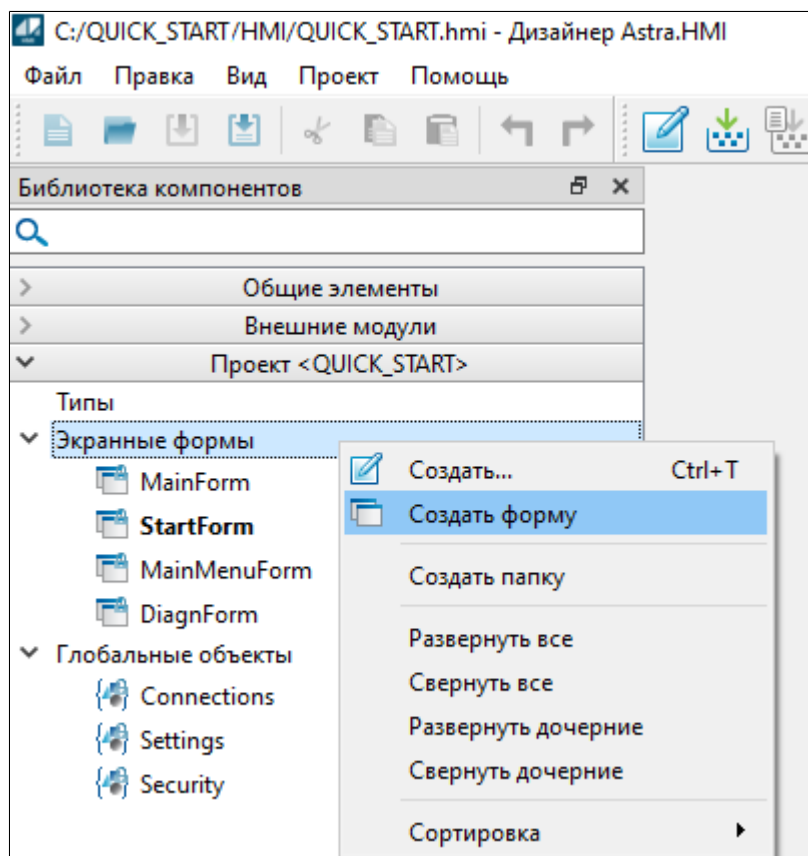


Создание экранной формы диагностики завершено.

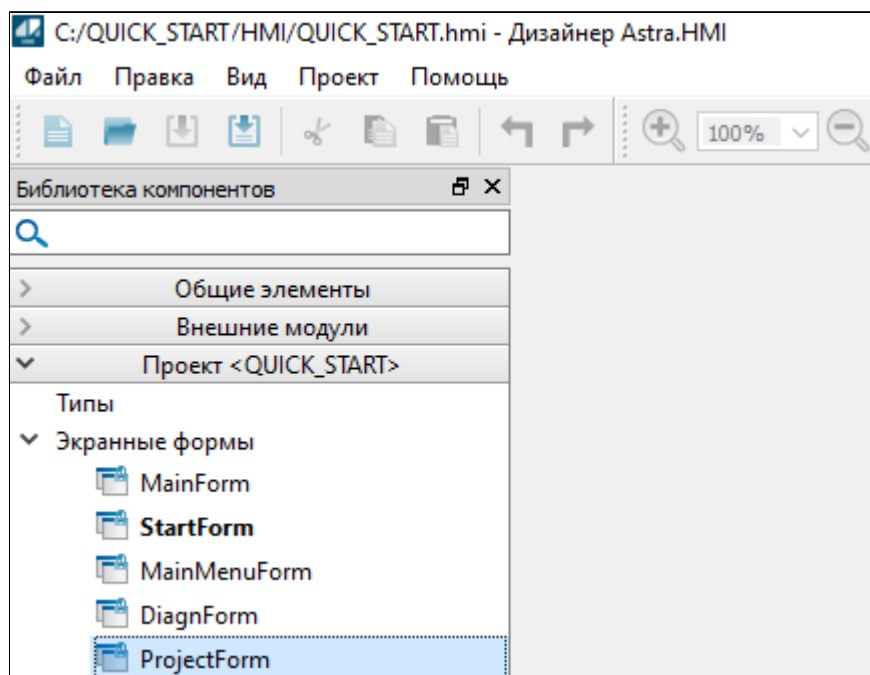


## 1.1.4.5.5. Создание экранной формы технологических параметров

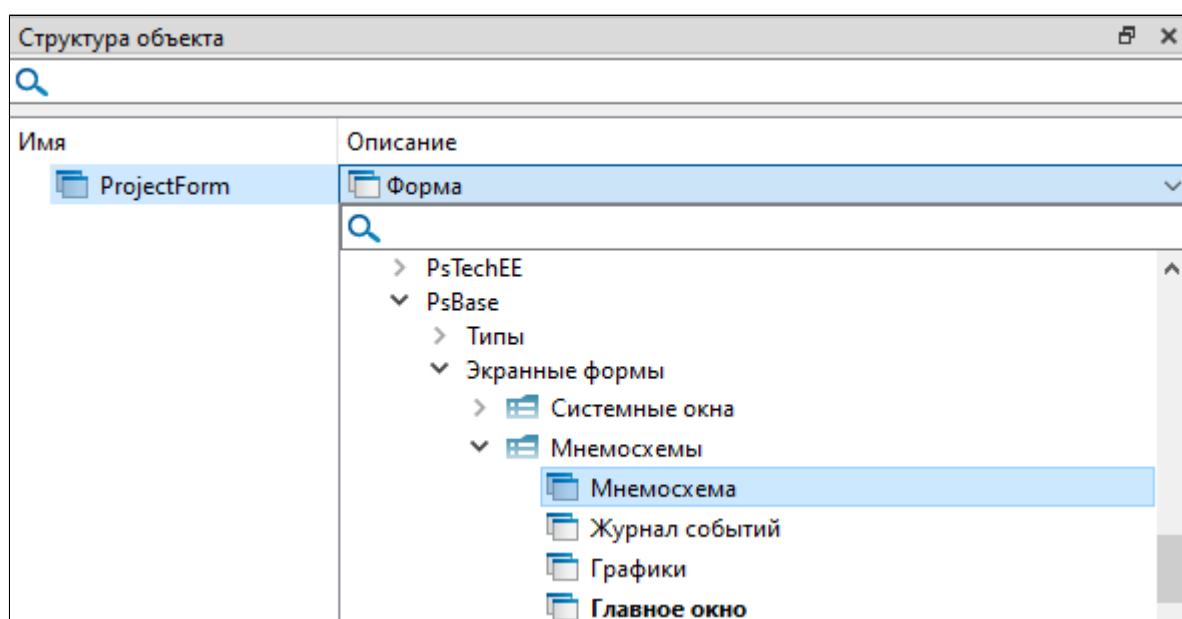
1. Кликните правой кнопкой мыши по вкладке "Экранные формы" и в контекстном меню выберите команду "Создать форму".



2. Во вкладке "Экранные формы" переименуйте созданную форму в "ProjectForm".



3. Измените тип экранной формы двойным кликом левой кнопки мыши по полю "Описание". В выпадающем списке выберите "Внешние модули" → "PsBase" → "Экранные формы" → "Мнемосхемы" → "Мнемосхема" и нажмите клавишу "Enter".



4. В редакторе свойств формы задайте следующие свойства:

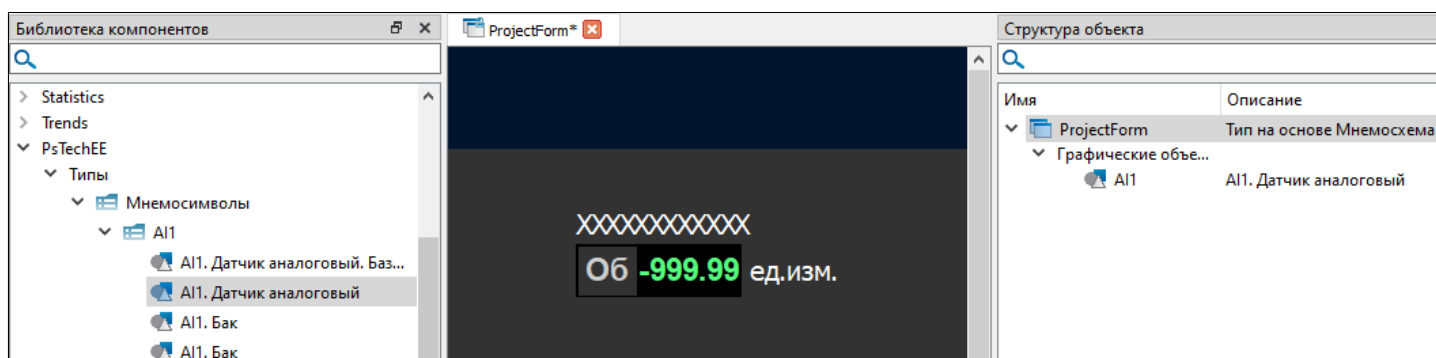
Структура объекта

Имя: ProjectForm | Описание: Тип на основе Мнемосхема

Редактор свойств

Свойство	Характерист	Значение
> <b>B</b> Включено	R W ✓	true
> <b>S</b> Всплывающая подсказка	R W ✓	
> <b>f8</b> Ширина	R W ✓	1900
> <b>f8</b> Высота	R W ✓	960
> <b>B</b> Фокус ввода	R W ✗	< не определено >
> <b>u4</b> Цвет пера	R W ✓	4278190080
> <b>u2</b> Стиль пера	R W ✓	Нет пера
> <b>f8</b> Толщина пера	R W ✓	1
> <b>u4</b> Цвет заливки	R W ✓	0xff323232
> <b>u2</b> Стиль заливки	R W ✓	Сплошная заливка
> <b>i4</b> X окна	R W ✓	0
> <b>i4</b> Y окна	R W ✓	0
> <b>i4</b> Ширина окна	R W ✓	1920
> <b>i4</b> Высота окна	R W ✓	1080
> <b>S</b> Заголовок окна	R W ✓	Технологические алгоритмы
> <b>B</b> Показывать заголовок	R W ✓	true
> <b>B</b> Показывать кнопку "свернуть"	R W ✓	true
> <b>B</b> Показывать кнопку "развернуть"	R W ✓	true
> <b>B</b> Показывать кнопку "закрыть"	R W ✓	true
> <b>B</b> Поверх всех окон	R W ✓	false
> <b>u4</b> Размеры окна	R W ✓	Вручную
> <b>u4</b> Стиль рамки окна	R W ✓	Изменяемый размер
> <b>u4</b> Состояние окна	R W ✓	По умолчанию
> <b>u4</b> Режим масштабирования	R W ✓	Не масштабировать
> <b>i4</b> Монитор	R W ✓	0
> <b>u4</b> Положение окна	R W ✓	Вручную
> <b>u2</b> Режим обработки закрытия окна	R W ✓	Закрыть
> <b>S</b> Путь к файлу иконки	R W ✓	
<b>H</b> Команда Назад	R ⚡ →	< не определено >
<b>H</b> Команда Вперед	R ⚡ →	< не определено >
<b>H</b> Команда На уровень выше	R ⚡ →	< не определено >
> <b>B</b> Кнопка Назад доступна	R W ✗ ✓	false
> <b>B</b> Кнопка Вперед доступна	R W ✗ ✓	false
> <b>B</b> Кнопка На уровень выше доступна	R W ✗	true
<b>H</b> Ссылка на главную форму	R ⚡ →	< не определено >

5. Добавьте на экранную форму "ProjectForm" библиотечный тип "AI1. Датчик аналоговый" библиотеки "PsTechEE" из вкладки "Внешние модули". Уберите хвост "\_1" в названии объекта.



6. Выберите экземпляр датчика в дереве объектов и в редакторе свойств задайте следующие свойства:

➤ Источник данных – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Затем укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections":  
unit.Connections.MAIN\_SOURCE.

Структура объекта

Имя | Описание

- ProjectForm | Тип на основе Мнемосхема
  - Графические объе...
    - AI1 | AI1. Датчик аналоговый

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		AI1
Кардинальное число		1
X	R W ✓	134.66666666666663
Y	R W ✓	77.333333333333314
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ✗	<не определено>
Источник данных	R ↵ → (=)	<не определено>
Строка инициализации	R W (=)	
mX	R W ✗	<не определено>
mY	R W ✗	<не определено>
Цвет окружения	R W ✗	f unit.ColorsEnv.COLOR_GAS
FormId	R W ✗	<не определено>
Режим открытия рабочего окна: false - двой...	R W ✗	false
Цветовая схема	R ↵ →	<не определено>
Цветовая схема графиков: 0 - светлая, 1 - те...	R W ✗	0
Отображать название	R W ✗	false
Отображать ед. изм.	R W ✗	false
Отображать обозначение	R W ✗	true

Сослаться

Указать объект

✂ Вырезать Ctrl+X

📄 Копировать Ctrl+C

📄 Вставить Ctrl+V

Развернуть все

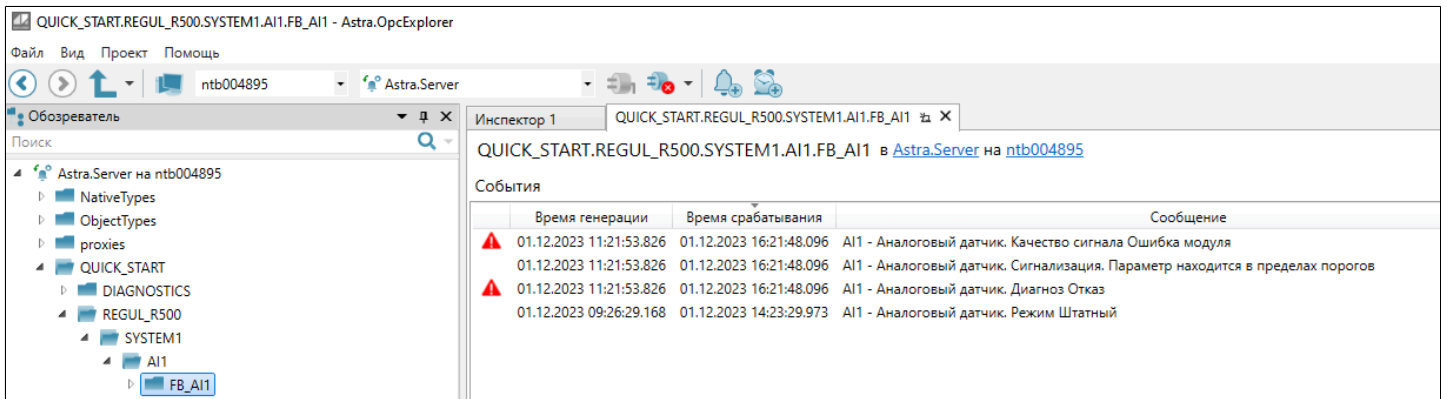
Свернуть все

Сортировка ▶

➤ Строка инициализации – здесь необходимо прописать путь до логического объекта аналогового датчика "AI1" на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpсExplorer: "QUICK\_START" → "REGUL\_R500" → "SYSTEM1" → "AI1" → "FB\_AI1".

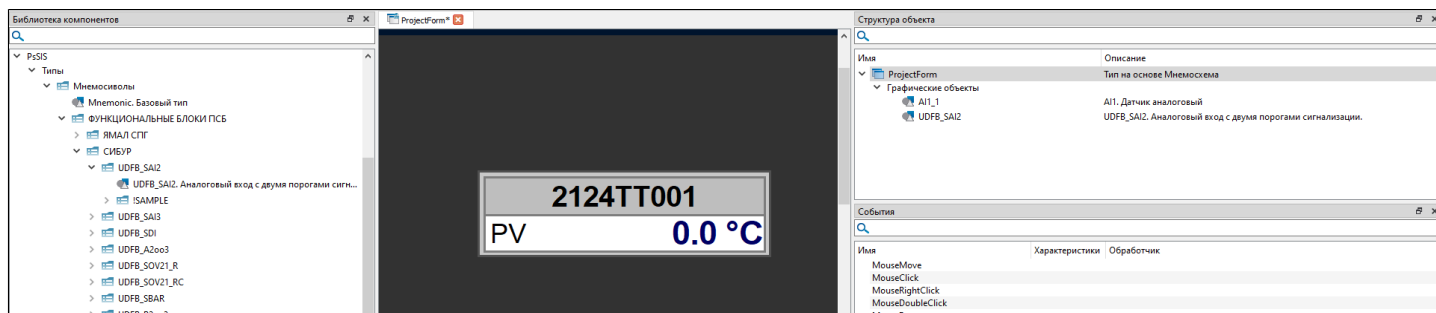
Скопируйте данный путь и вставьте в свойство "Строка инициализации".



В результате должны получиться следующие свойства для модуля:

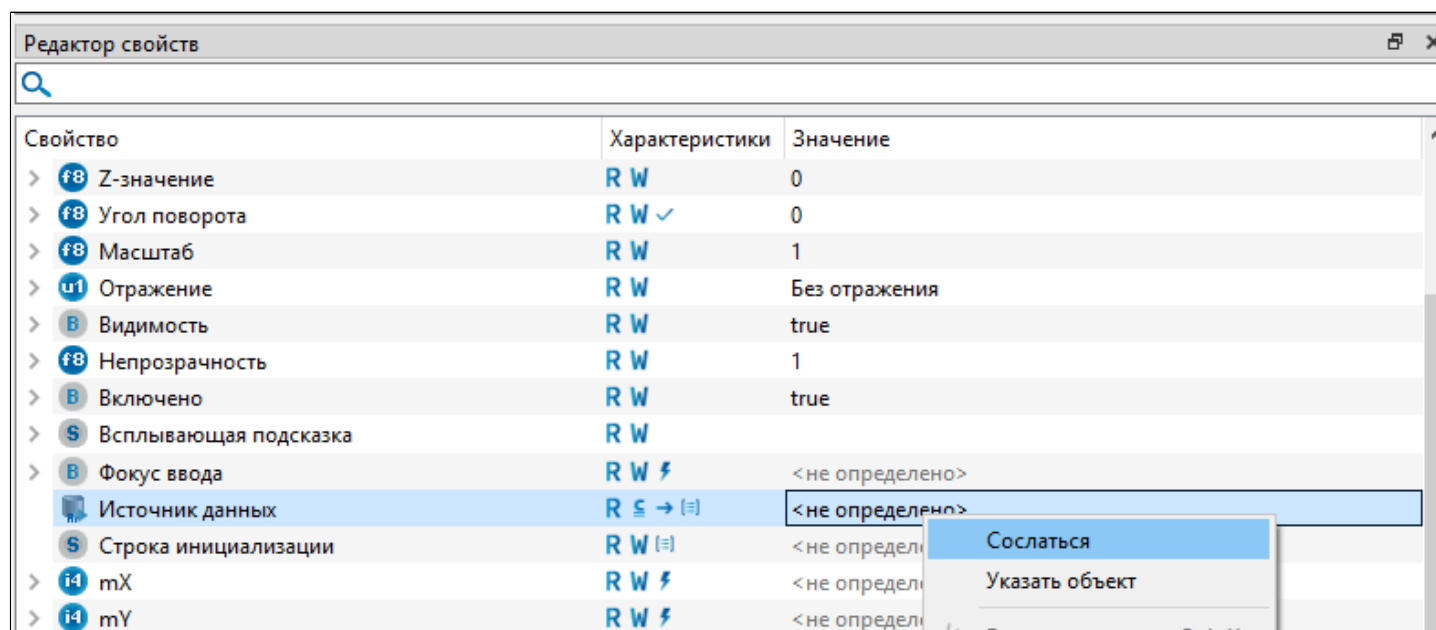
Структура объекта		
<input type="text"/>		
Имя	Описание	
<ul style="list-style-type: none"> <li>▼  ProjectForm</li> <li> <ul style="list-style-type: none"> <li>▼ Графические объе...</li> <li> <ul style="list-style-type: none"> <li> AI1</li> </ul> </li> </ul> </li> </ul>	Тип на основе Мнемосхема	
	AI1. Датчик аналоговый	
Редактор свойств		
<input type="text"/>		
Свойство	Характеристики	Значение
Отображаемое имя		AI1
Кардинальное число		1
>  X	R W ✓	134.66666666666663
>  Y	R W ✓	77.333333333333314
>  Z-значение	R W	0
>  Угол поворота	R W ✓	0
>  Масштаб	R W	1
>  Отражение	R W	Без отражения
>  Видимость	R W	true
>  Непрозрачность	R W	1
>  Включено	R W	true
>  Всплывающая подсказка	R W	
>  Фокус ввода	R W ✗	< не определено >
Источник данных	R    ✓	unit.Connections.MAIN_SOURCE
Строка инициализации	R W  ✓	QUICK_START.REGUL_R500.SYSTEM1.AI1.FB_AI1
>  mX	R W ✗	< не определено >
>  mY	R W ✗	< не определено >
>  Цвет окружения	R W ✗	unit.ColorsEnv.COLOR_GAS
>  FormId	R W ✗	< не определено >
>  Режим открытия рабочего окна: false - двойной клик, true - одиночный клик	R W ✗	false
Цветовая схема	R   →	< не определено >
>  Цветовая схема графиков: 0 - светлая, 1 - темная	R W ✗	0
>  Отображать название	R W ✗	false
>  Отображать ед. изм.	R W ✗	false
>  Отображать обозначение	R W ✗	true

7. Добавьте на экранную форму "ProjectForm" библиотечный тип "UDFB\_SAI2. Аналоговый вход с двумя порогами сигнализации" библиотеки "PsSIS" из вкладки "Внешние модули". Уберите хвост "\_1" в названии объекта.



8. Выберите экземпляр аналогового входа в дереве объектов и в редакторе свойств задайте следующие свойства:

› Источник данных – кликните правой кнопкой мыши по свойству и выберите из контекстного меню команду "Сослаться". Затем укажите путь к элементу "MAIN\_SOURCE" из глобального объекта "Connections": unit.Connections.MAIN\_SOURCE.

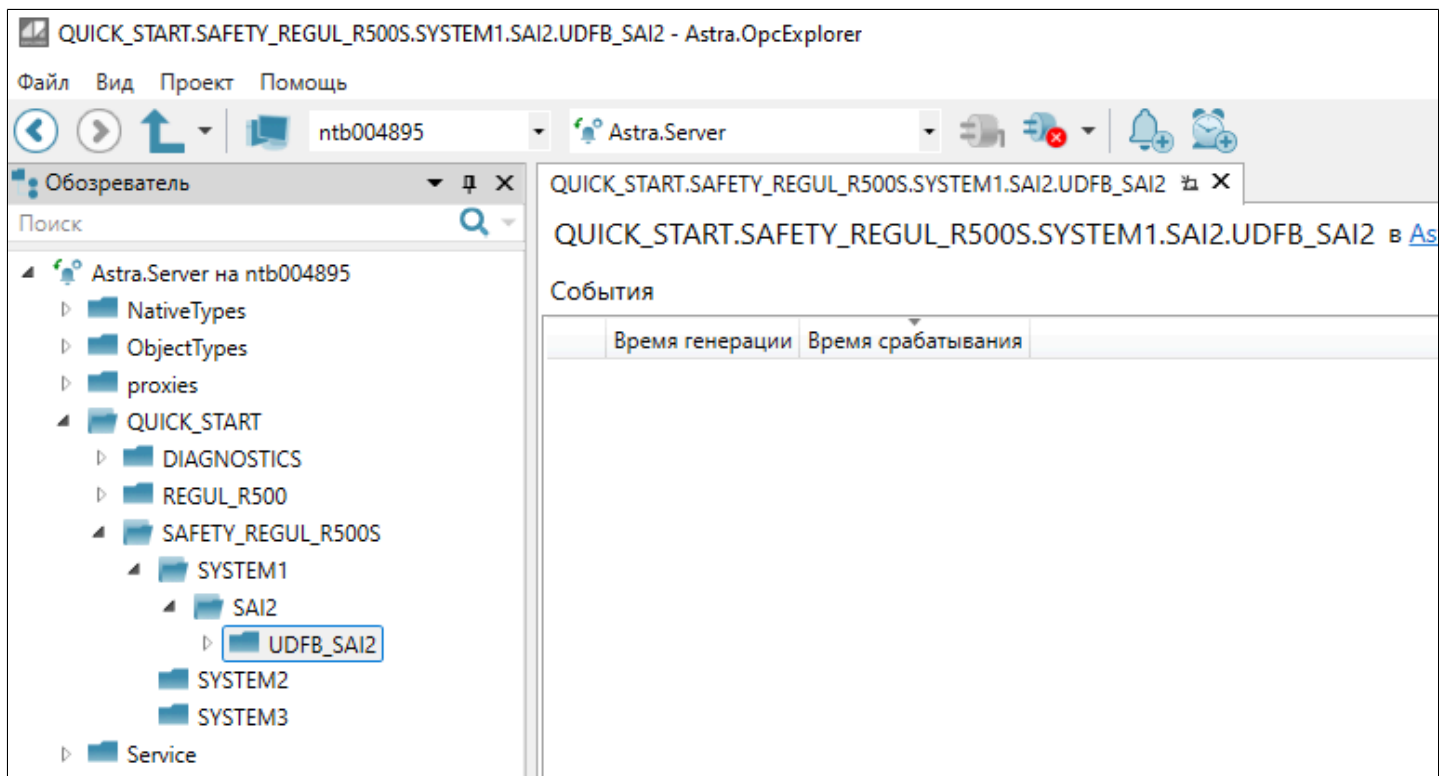




➤ Строка инициализации – здесь необходимо прописать путь до логического объекта аналогового входа "UDFB\_SAI2" на сервере AstraServer.

Полный путь до объекта можно посмотреть в утилите OpcExplorer: "QUICK\_START" → "SAFETY\_REGUL\_R500S" → "SYSTEM1" → "SAI2" → "UDFB\_SAI2".

Скопируйте данный путь и вставьте в свойство "Строка инициализации".



В результате должны получиться следующие свойства для модуля:

Структура объекта

Имя	Описание
ProjectForm	Тип на основе Мнемосхема
Графические объекты	
AI1_1	AI1. Датчик аналоговый
UDFB_SAI2	UDFB_SAI2. Аналоговый вход с двумя порогами сигнализации.

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		UDFB_SAI2
Кардинальное число		1
Курсор	R W → 0	< не определено >
X	R W ✓	706
Y	R W ✓	109.66666666666663
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ✗	< не определено >
Источник данных	R ↗ → [≡] ✓	unit.Connections.MAIN_SOURCE
Строка инициализации	R W [≡] ✓	QUICK_START.SAFETY_REGUL_R500S.SYSTEM1.SAI2.UDFB_SAI2
mX	R W ✗	< не определено >
mY	R W ✗	< не определено >

9. Добавьте на экранную форму элемент "Команда" из вкладки "Встроенные".  
 Задайте имя команды "GoToBack" и в редакторе свойств задайте следующие свойства:

Свойство	Значение
Включено	true
Триггер	false

Структура объекта

Имя	Описание
<ul style="list-style-type: none"> <li>▼ ProjectForm                             <ul style="list-style-type: none"> <li>▼ Графические объекты                                     <ul style="list-style-type: none"> <li> AI1_1</li> <li> UDFB_SAI2</li> </ul> </li> <li>▼ Данные                                     <ul style="list-style-type: none"> <li> GoToBack</li> </ul> </li> </ul> </li> </ul>	Тип на основе Мнемосхема  AI1. Датчик аналоговый UDFB_SAI2. Аналоговый вход с двумя порогоми сигнализации.  Команда

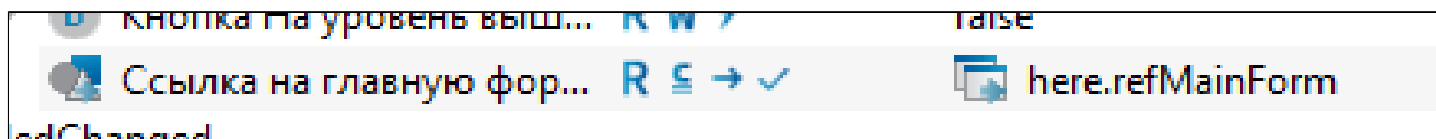
Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		GoToBack
Кардинальное число		1
>  Включено	R W	true
>  Триггер	R W	false

10. Добавьте обработчик события "Invoked" для команды "GoToBack".

События		
🔍		
Имя	Характеристики	Обработчик
▼ Invoked		Открыть во фрейме
> <b>Handler_1</b>		MainMenuForm
		Фрейм refMainForm.MainFrame
EnabledChanged		

11. Раскройте обработчик и задайте в свойстве "Ссылка на главную форму" значение "here.refMainForm".



12. Укажите в свойстве "Команда на уровень выше" экранной формы "ProjectForm" ссылку на созданную команду "GoToBack".

События		
Q		
Имя	Характеристики	Обработчик
> f8 Непрозрачность	R W	1
> B Включено	R W	true
> S Всплывающая подсказка	R W	
> f8 Ширина	R W	1900
> f8 Высота	R W	960
> B Фокус ввода	R W ⚡	< не определено >
> u4 Цвет пера	R W	4278190080
> u2 Стиль пера	R W	Нет пера
> f8 Толщина пера	R W	1
> u4 Цвет заливки	R W	0xff323232
> u2 Стиль заливки	R W	Сплошная заливка
> i4 X окна	R W	0
> i4 Y окна	R W	0
> i4 Ширина окна	R W	1920
> i4 Высота окна	R W	1080
> S Заголовок окна	R W	Главное меню
> B Показывать заголовок	R W	true
> B Показывать кнопку "све...	R W	true
> B Показывать кнопку "раз...	R W	true
> B Показывать кнопку "зак...	R W	true
> B Поверх всех окон	R W	false
> u4 Размеры окна	R W	Вручную
> u4 Стиль рамки окна	R W	Изменяемый размер
> u4 Состояние окна	R W	По умолчанию
> u4 Режим масштабирования	R W	Не масштабировать
> i4 Монитор	R W	0
> u4 Положение окна	R W	Вручную
> u2 Режим обработки закры...	R W	Закреть
> S Путь к файлу иконки	R W	
👤 Команда Назад	R ⚡ →	< не определено >
👤 Команда Вперед	R ⚡ →	< не определено >
👤 Команда На уровень выш...	R ⚡ → ✓	👤 GoToBack
> B Кнопка Назад доступна	R W ⚡	false
> B Кнопка Вперед доступна	R W ⚡	false
> B Кнопка На уровень выш...	R W ⚡	false
👤 Ссылка на главную фор...	R ⚡ → ✓	👤 here.refMainForm

EnabledChanged

13. Перейдите на экранную форму "MainMenuForm" и настройте событие "MouseClicked" у кнопки "btnMenuNavigation\_01" следующим образом:

The image shows two windows from the Visual Studio IDE. The top window, titled 'Структура объекта' (Object Structure), displays a tree view of the form's components. The selected item is 'btnMenuNavigation\_01', which is a 'btnMenuNavigation' control. The bottom window, titled 'События' (Events), shows the 'MouseClicked' event with a handler named 'Handler\_1'. The 'Обработчик' (Handler) column shows 'refMainForm.MainFrame'.

Имя	Описание
Графические объекты	
Menu_01	Прямоугольник
Графические объекты	
FrameTitle	Прямоугольник
<b>btnMenuNavigation_01</b>	<b>btnMenuNavigation</b>

Имя	Харак	Обработчик
MouseMove		
MouseClicked		
> Handler_1	Открыть во фрейме	ProjectForm
	Фрейм	refMainForm.MainFrame
MouseRightClick		

14. Раскройте обработчик и задайте свойству "Ссылка на главную форму" значение "here.refMainForm".

Структура объекта

Имя

- Графические объекты
  - FrameTitle
    - btn btnNavigation\_01

Описание

Прямоугольник

btnMenuNavigation

---

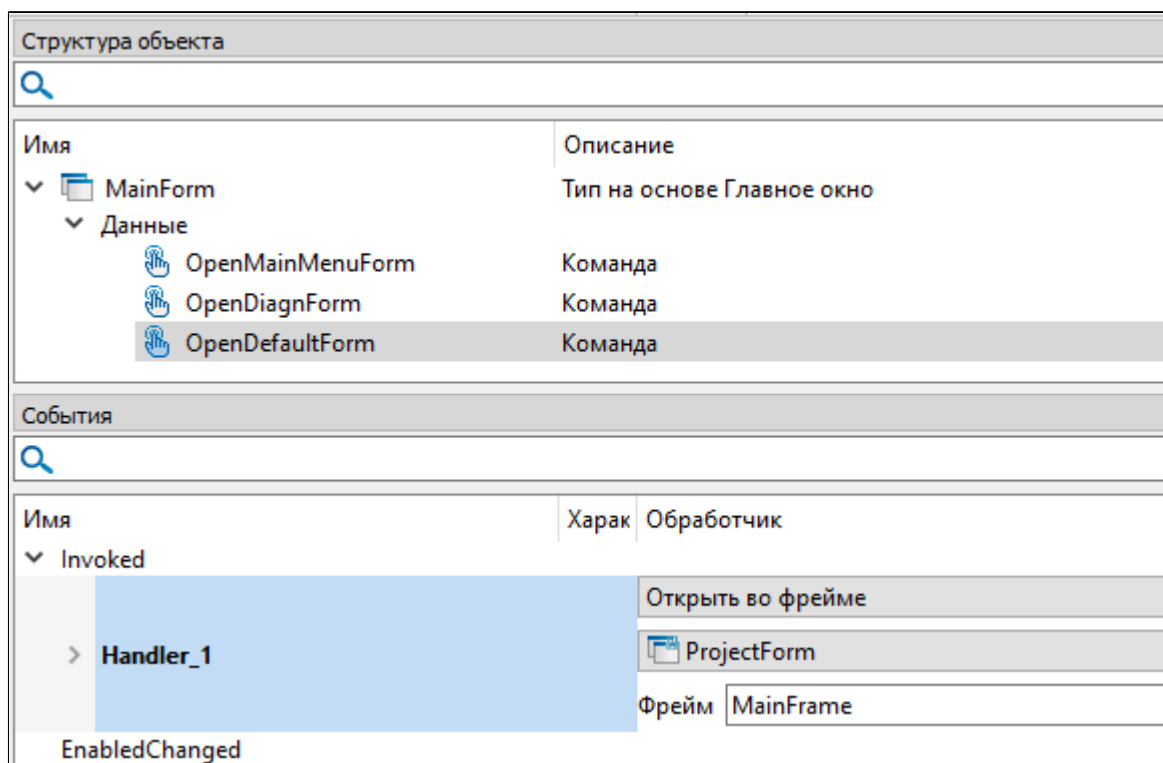
События

Имя

Имя	Харак	Обработчик
> Включено	R W	true
> Всплывающая подсказка	R W	
> f8 Ширина	R W	1900
> f8 Высота	R W	960
> В Фокус ввода	R W ⚡	<не определено>
> u4 Цвет пера	R W	4278190080
> u2 Стил пера	R W	Нет пера
> f8 Толщина пера	R W	1
> u4 Цвет заливки	R W	0xff323232
> u2 Стил заливки	R W	Сплошная заливка
> i4 X окна	R W	0
> i4 Y окна	R W	0
> i4 Ширина окна	R W	1920
> i4 Высота окна	R W	1080
> S Заголовок окна	R W	Технологические алгоритмы
> В Показывать заголовок	R W	true
> В Показывать кнопку "свернуть"	R W	true
> В Показывать кнопку "развернуть"	R W	true
> В Показывать кнопку "закрыть"	R W	true
> В Поверх всех окон	R W	false
> u4 Размеры окна	R W	Вручную
> u4 Стил рамки окна	R W	Изменяемый размер
> u4 Состояние окна	R W	По умолчанию
> u4 Режим масштабирования	R W	Не масштабировать
> i4 Монитор	R W	0
> u4 Положение окна	R W	Вручную
> u2 Режим обработки закрытия окна	R W	Закрыть
> S Путь к файлу иконки	R W	
👤 Команда Назад	R ⚡ →	<не определено>
👤 Команда Вперед	R ⚡ →	<не определено>
👤 Команда На уровень выше	R ⚡ →	👤 GoToBack
> В Кнопка Назад доступна	R W ⚡	false
> В Кнопка Вперед доступна	R W ⚡	false
> В Кнопка На уровень выше доступна	R W ⚡	true
👤 Ссылка на главную форму	R ⚡ →	👤 here.refMainForm



15. Перейдите на главную форму "MainForm" и с помощью контекстного меню добавьте обработчик события "Invoked" для команды "OpenDefaultForm".



16. Раскройте обработчик и задайте свойству "Ссылка на главную форму" значение "here". Нажмите клавишу "Enter".

Структура объекта

Имя Описание

- MainForm
  - Данные
    - OpenMainMenuForm Команда
    - OpenDiagnForm Команда
    - OpenDefaultForm Команда

---

События

Имя Харак Обработчик

Имя	Харак	Обработчик
> f8 Высота	R W	960
> B Фокус ввода	R W ⚡	<не определено>
> u4 Цвет пера	R W	4278190080
> u2 Стил ь пера	R W	Нет пера
> f8 Толщина пера	R W	1
> u4 Цвет заливки	R W	0xff323232
> u2 Стил ь заливки	R W	Сплошная заливка
> i4 X окна	R W	0
> i4 Y окна	R W	0
> i4 Ширина окна	R W	1920
> i4 Высота окна	R W	1080
> S Заголовок окна	R W	Технологические алгоритмы
> B Показывать заголовок	R W	true
> B Показывать кнопку "свернуть"	R W	true
> B Показывать кнопку "развернуть"	R W	true
> B Показывать кнопку "закр ыть"	R W	true
> B Поверх всех окон	R W	false
> u4 Размеры окна	R W	Вручную
> u4 Стил ь рамки окна	R W	Изменяемый размер
> u4 Состояние окна	R W	По умолчанию
> u4 Режим масштабирования	R W	Не масштабировать
> i4 Монитор	R W	0
> u4 Положение окна	R W	Вручную
> u2 Режим обработки закрытия окна	R W	Закр ыть
> S Путь к файлу иконки	R W	
👤 Команда Назад	R ⚡ →	<не определено>
👤 Команда Вперед	R ⚡ →	<не определено>
👤 Команда На уровень выше	R ⚡ →	👤 GoToBack
> B Кнопка Назад доступна	R W ⚡	false
> B Кнопка Вперед доступна	R W ⚡	false
> B Кнопка На уровень выше доступна	R W ⚡	true
👤 Ссылка на главную форму	R ⚡ →	📄 here

EnabledChanged

Для события "Opened" экранной формы "MainForm" добавлен вызов команды открытия экранной формы "OpenDefaultForm". При открытии окна "Главной формы" будет активироваться событие "Invoked" команды "OpenDefaultForm".

The screenshot shows the Visual Studio IDE with the 'Opened' event handler for the 'MainForm' form. The 'Events' window shows the 'Opened' event with a handler named 'HandlerOpenDefaultForm'. The 'Code' window shows the following code:

```
1 OpenDefaultForm.Invoke();
```

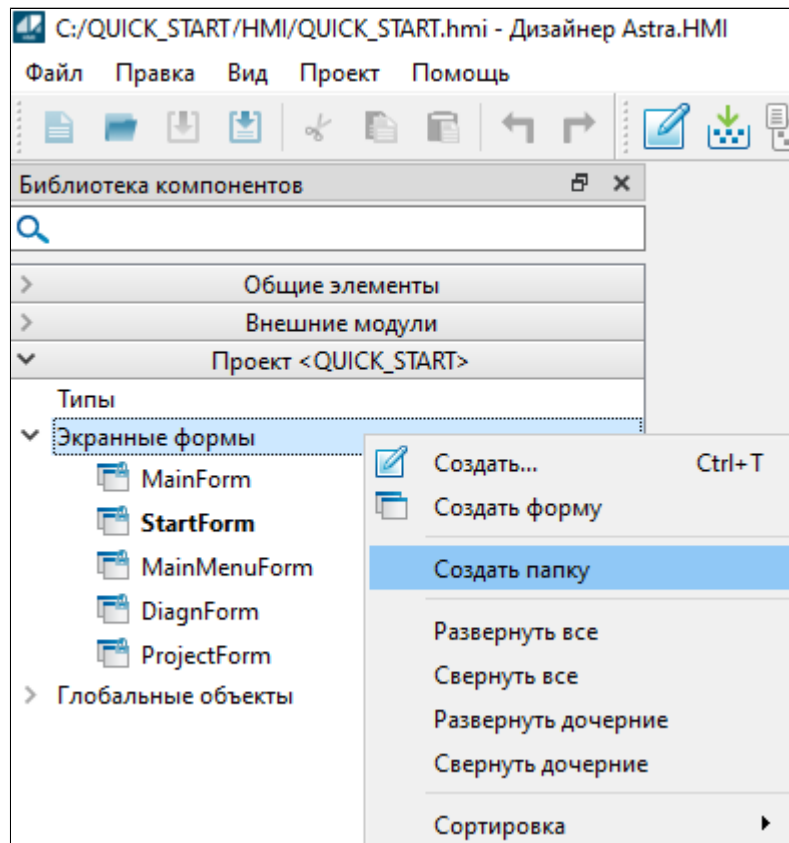
The 'Code' window also shows 0 errors, 0 warnings, and 0 messages. The language is set to 'C#', the zoom is 100%, and the line is 1:1. The 'OK' and 'Отмена' buttons are visible.

Создание экранной формы завершено.

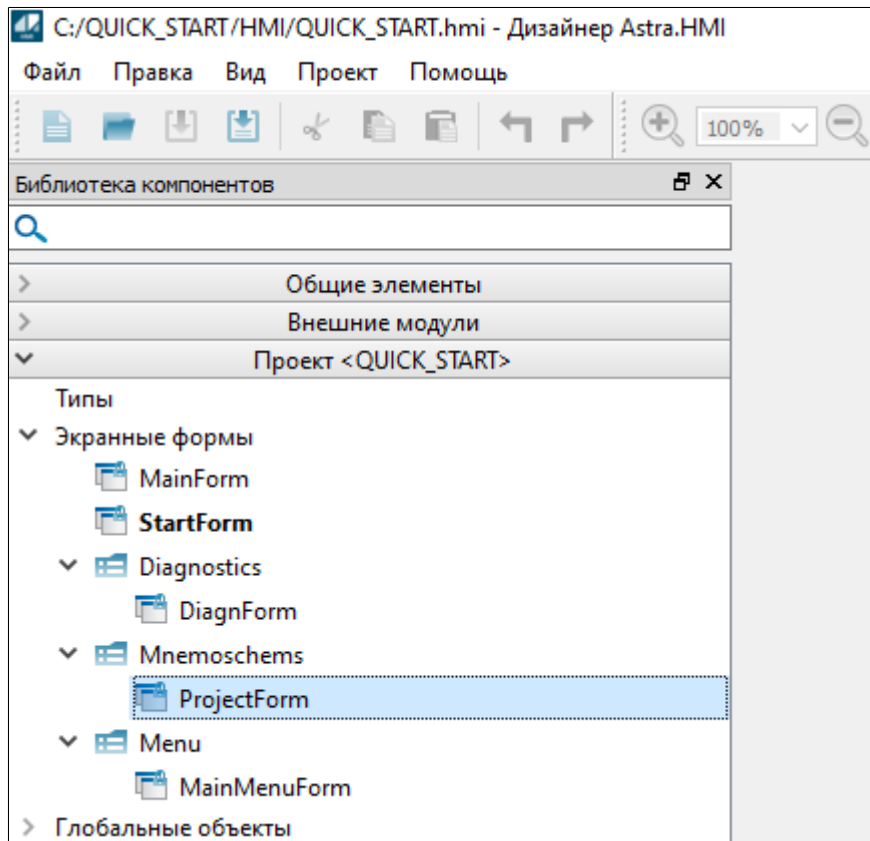
## 1.1.4.5.6. Структурирование экранных форм

Чтобы лучше ориентироваться в проекте, рекомендуется объединять объекты в папки.

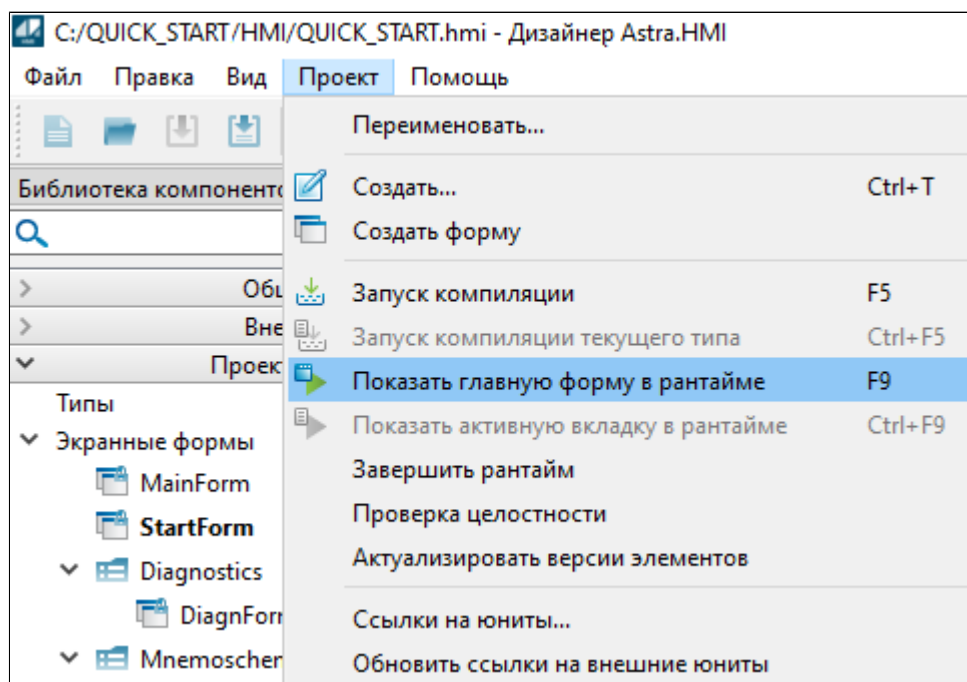
Чтобы создать папку, кликните правой кнопкой мыши по вкладке экранные формы и из контекстного меню выберите команду "Создать папку".



1. Создайте три папки и назовите их следующим образом: "Diagnostics" – для экрана диагностики, "Menu" – для мнемосхемы главного меню и "Mnemoschems" – для остальных мнемосхем. Переместите экранные формы в соответствующие папки путем перетаскивания.



2. Выберите команду "Проект" → "Показать главную форму в рантайме" или нажмите клавишу "F9".



В случае успешной компиляции откроется главная форма в окне Astra.HMI.Viewer.

## 1.2. Разработка пользовательской библиотеки

[Разработка библиотеки алгоритмов в Astra.IDE](#)

[Разработка библиотеки типов в Astra.AStudio](#)

[Разработка библиотеки мнемосимволов в Astra.HMI](#)

## 1.2.1. Правила именования тегов

1. Не используйте для именования тегов спецсимволы (пробелы, тире, точки и т.д.).



Допустимым спецсимволом является только символ подчеркивания "\_" ("земля").

2. Используйте при именвании тегов только английский алфавит.



При использовании русских букв в именвании тегов могут возникнуть проблемы при обращении к его свойствам.



## 1.2.2. Разработка библиотеки алгоритмов в Astra.IDE

[Создание проекта библиотеки](#)

[Создание функционального блока](#)

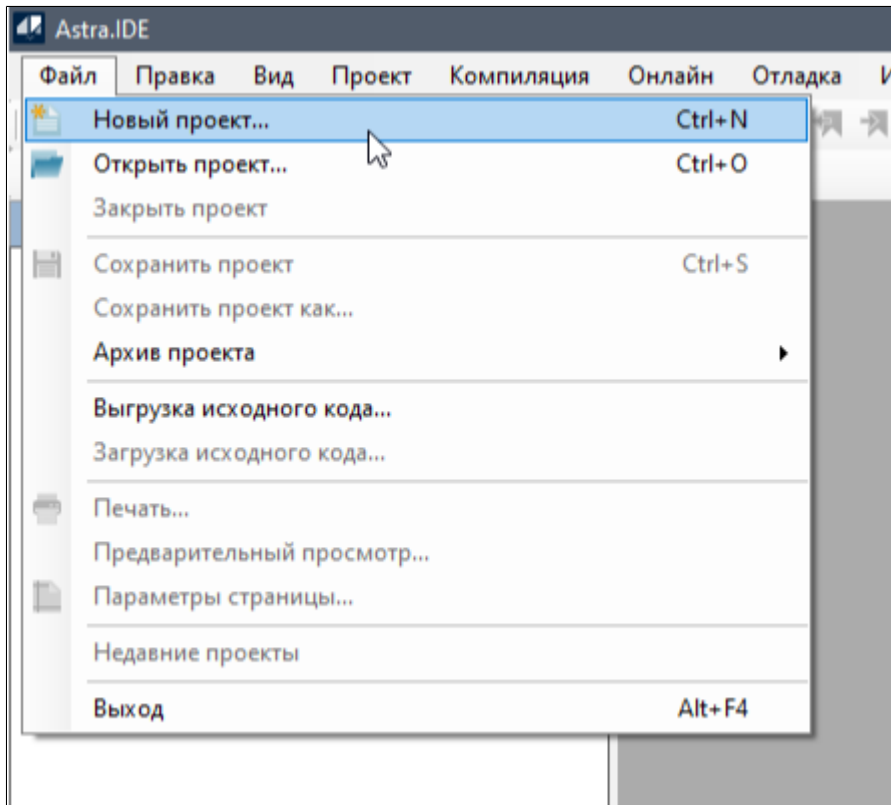
[Информация о проекте](#)

[Сохранение библиотеки](#)

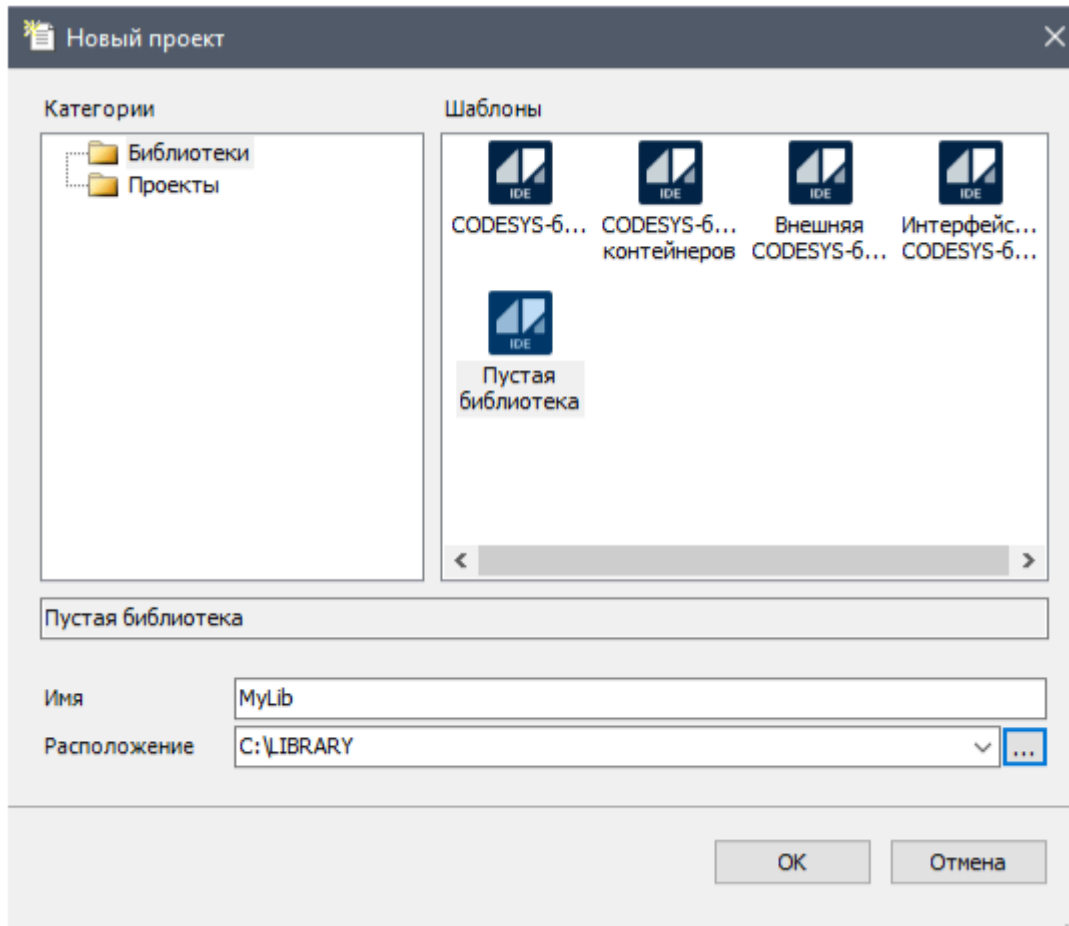
## 1.2.2.1. Создание проекта библиотеки

Чтобы создать новый проект библиотеки, выполните следующие действия:

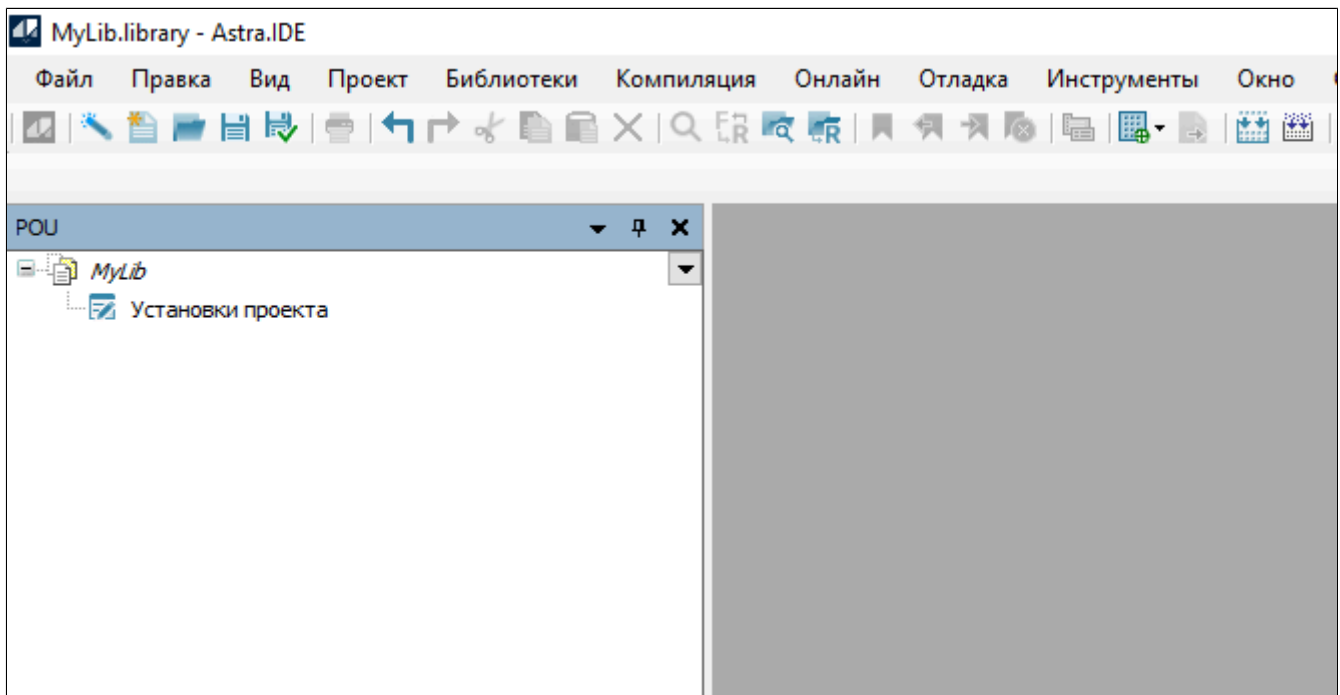
1. В главном меню выберите команду "Файл" -> "Новый проект..."



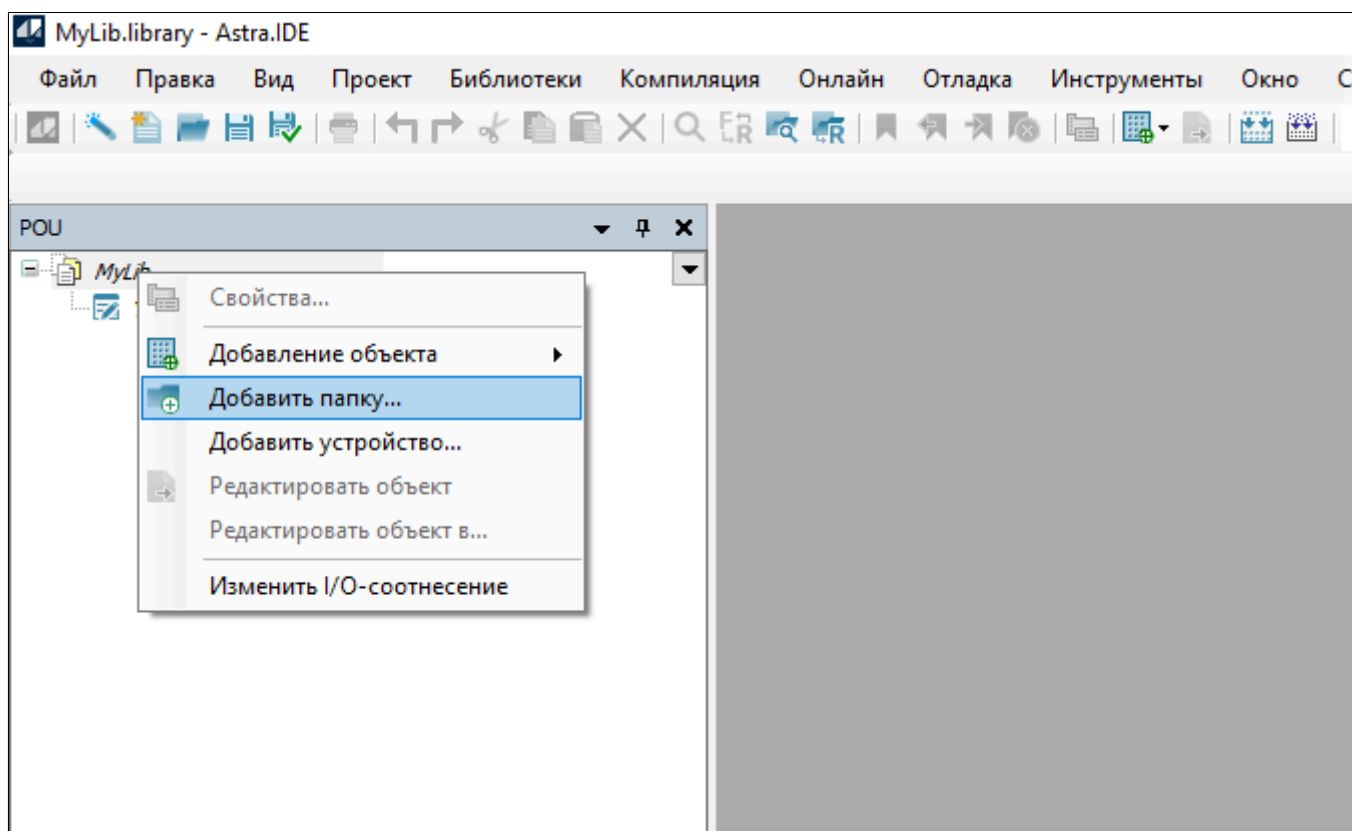
2. В открывшемся диалоговом окне выберите категорию "Библиотеки", шаблон "Пустая библиотека", задайте название и каталог. Нажмите кнопку "OK".



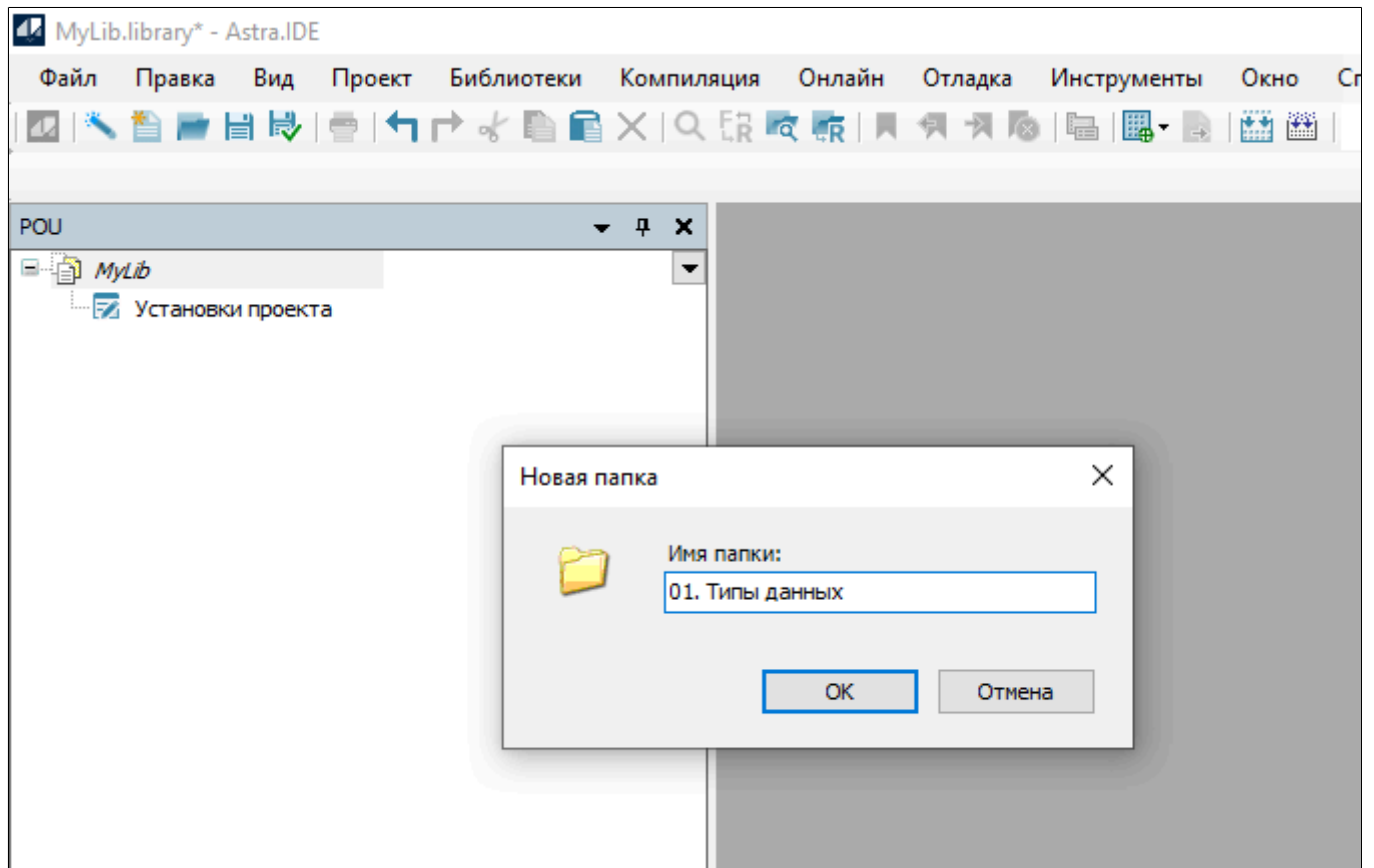
3. Будет создан пустой шаблон библиотеки. Перейдите во вкладку "POU".



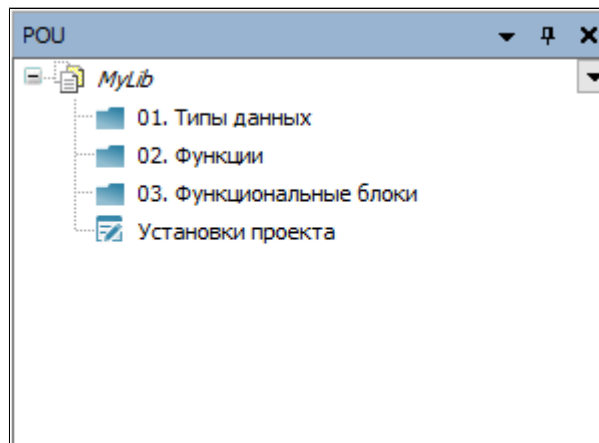
4. Чтобы легче ориентироваться в библиотеке создадим структуру объектов. Для этого выделите проект "MyLib" и в контекстном меню выберите пункт "Добавить папку...".



5. Добавьте папку "01. Типы данных" и нажмите кнопку "ОК".



6. Аналогичным образом добавьте папки "02. Функции" и "03. Функциональные блоки".

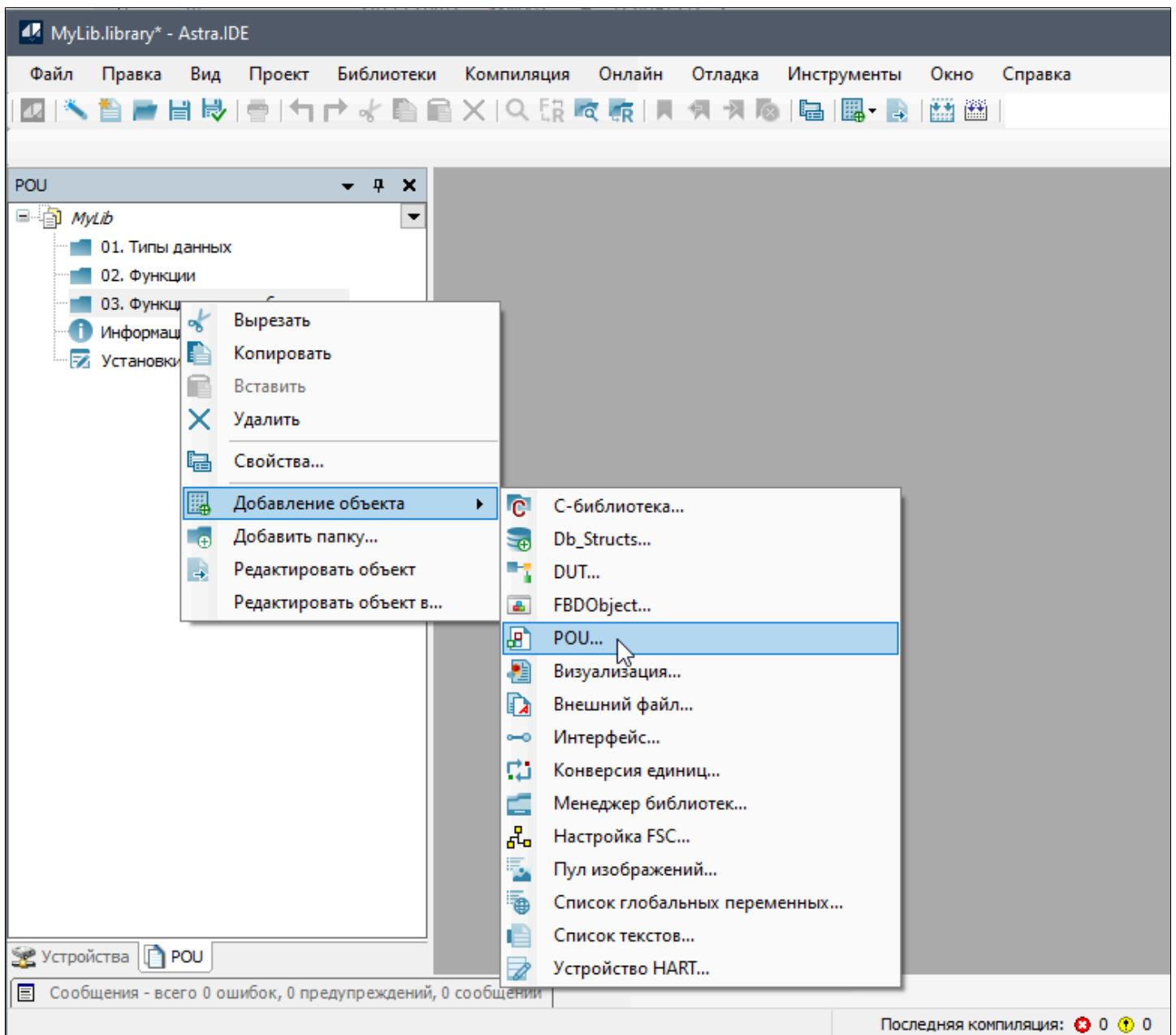


## 1.2.2.2. Создание функционального блока

Рассмотрим создание функционального блока на примере алгоритма аналогового датчика "AI1".

Выполните следующие действия:

1. Нажмите на папку правой кнопкой мыши "03. Функциональные блоки" и в контекстном меню выберите команду "Добавление объекта" -> "POU...".



2. Введите имя "AI1", выберите тип "Функциональный блок", а также предпочитаемый язык реализации. Нажмите кнопку "Добавить".

Добавить POU

Создать новый POU (компонент организации программы)

Имя  
AI1

Тип

Программа

Функциональный блок

Extends

Implements

Окончательный  Абстрактный

Спецификатор доступа

Язык реализации метода:  
Структурированный текст (ST)

Функция

Тип возвращаемого значения

Язык реализации

Структурированный текст (ST)

Sequential Function Chart (SFC)

Непрерывные функциональные схемы (CFC)

Непрерывные функциональные схемы (CFC) - постранично

Структурированный текст (ST)

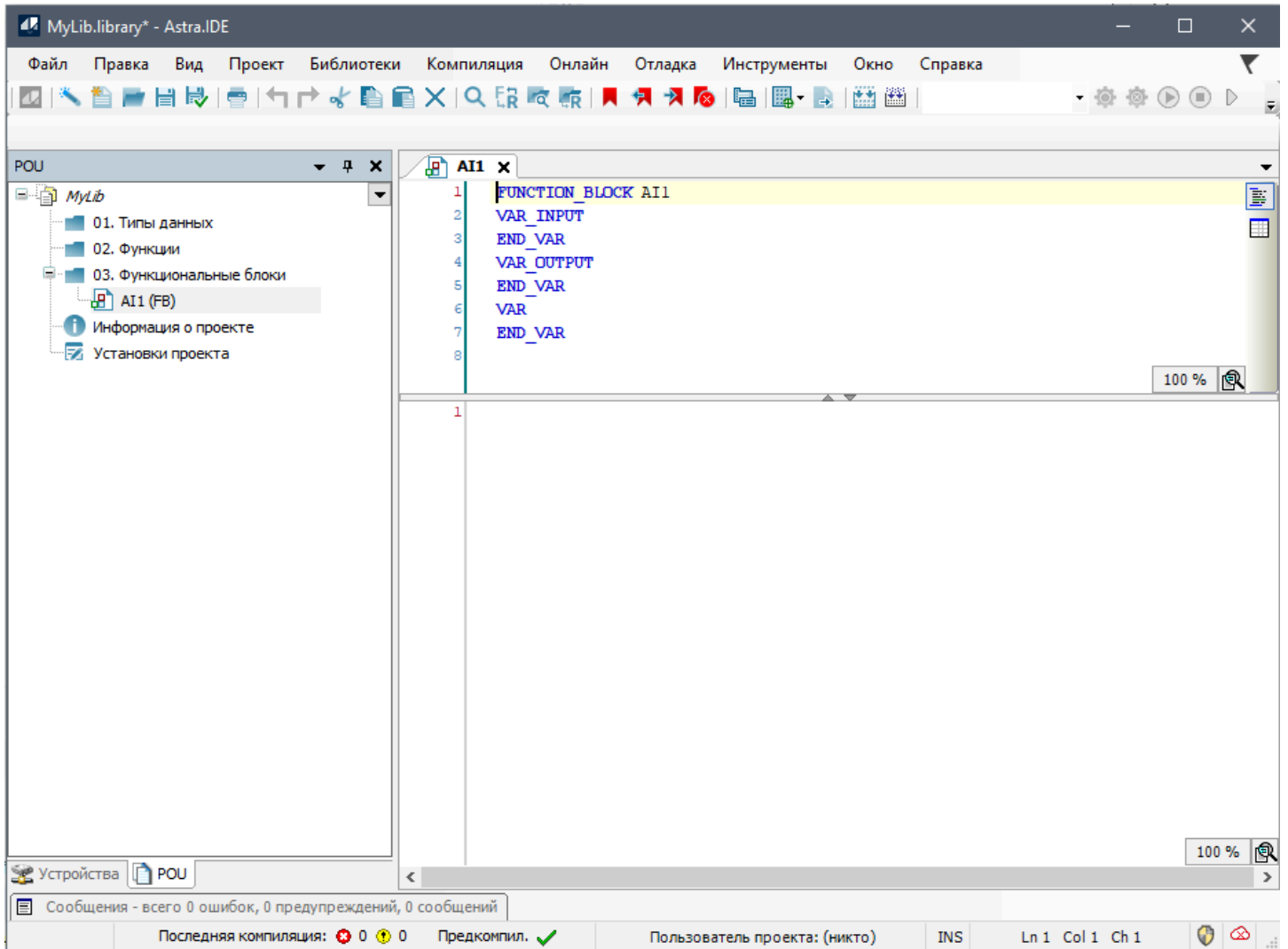
Функциональные блочные диаграммы (FBD)

Язык релейных диаграмм (LD)



Для реализации логики алгоритма рекомендуется использовать язык ST.

4. В результате в дерево объектов будет добавлен новый функциональный блок "AI1". При добавлении нового функционального блока редактор открывается автоматически.



Если редактор функционального блока закрыт, его можно открыть двойным кликом мыши по нему в дереве объектов.



# Редактор функционального блока



## 1 Область объявления переменных

В области объявления переменных задаются входные, выходные и внутренние переменные функционального блока, участвующие в алгоритме.

## 2 Редактор кода

В редакторе кода описывается алгоритм функционального блока.

Над объявлением функционального блока укажем русскоязычный комментарий к нему.

```
1 //Аналоговый датчик
2 FUNCTION_BLOCK AI1
3 VAR_INPUT
4 END_VAR
5 VAR_OUTPUT
6 END_VAR
7 VAR
8 END_VAR
9
```

100 %

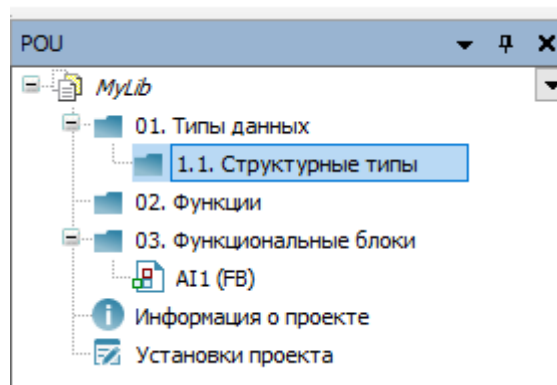
## 1.2.2.2.2. Объявление входных переменных

В области "VAR\_INPUT" необходимо объявить входные переменные функционального блока (команды, уставки, сигналы с модулей ввода/вывода и т.д.).

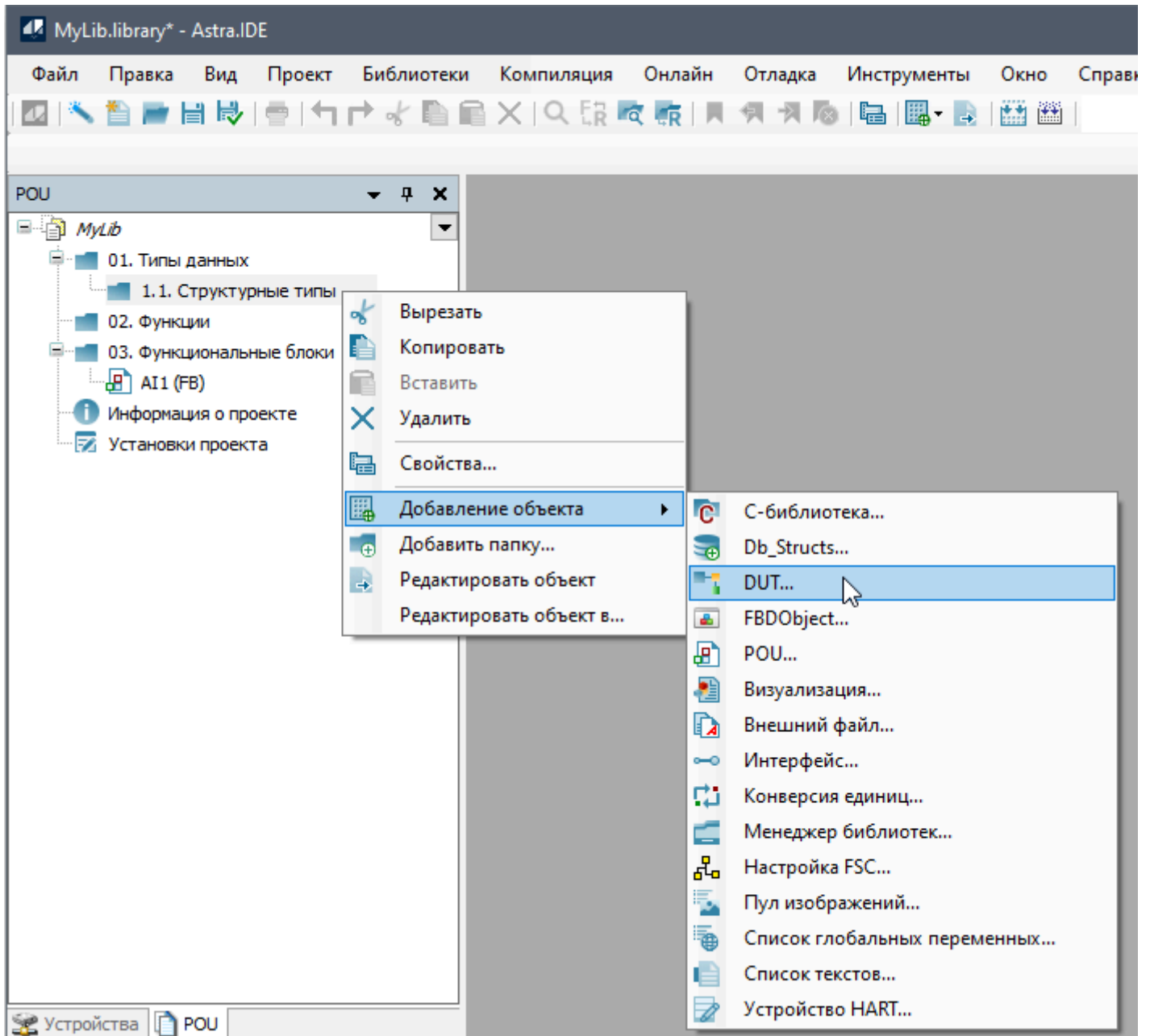
Настроечные параметры для аналогового датчика можно представить в виде структурной переменной.

Чтобы создать структурную переменную, выполните следующие действия:

1. Создайте в папке "01. Типы данных" вложенную папку "1.1. Структурные типы".




2. Выделите папку "1.1. Структурные типы" и в контекстном меню выберите команду "Добавление объекта" -> "DUT...".



3. Укажите название структуры "AI\_CONFIG" и нажмите кнопку "Добавить".

Добавить DUT X

 Создать новый объект типа данных

Имя

Тип

Структура  
 Расширение для  ...

Перечисление  
 Поддержка списков текстов

Псевдоним  
 Базовый тип  >

Объединение

4. В редакторе структурной переменной "AI\_CONFIG" объявите настроечные переменные.

```

AI_CONFIG x
1 //Настройки аналогового датчика
2 TYPE AI_CONFIG :
3 STRUCT
4     YMIN           : REAL:=0.0;           //Минимум выходной величины
5     YMAX           : REAL:=100.0;        //Максимум выходной величины
6     TF             : REAL:=0.0;          //Постоянная времени фильтра
7     AH2            : REAL:=90.0;         //Порог аварийный верхний №2
8     AH             : REAL:=80.0;         //Порог аварийный верхний
9     WH             : REAL:=75.0;         //Порог предупредительный верхний
10    WL             : REAL:=25.0;         //Порог предупредительный нижний
11    AL             : REAL:=-20.0;        //Порог аварийный нижний
12    AL2            : REAL:=10.0;         //Порог аварийный нижний №2
13    HYST           : REAL:=0.0;          //Гистерезис
14 END_STRUCT
15 END_TYPE
  
```



Не забывайте указывать комментарий для каждой переменной.

5. Чтобы параметр передавался на верхний уровень (ВУ) по протоколу OPC UA необходимо над ним указать следующий атрибут **{attribute 'symbol' := 'readwrite'}**.
6. Чтобы параметр сохранялся в энергонезависимой памяти необходимо над ним указать следующий атрибут **{attribute 'ps.setpoint':='xxxxx'}**, где xxxxx - начальное значение параметра (значения по умолчанию).
7. Чтобы опеределить данный структурный тип как безопасной для резервирования, необходимо над объявлением типа указать атрибут **{attribute 'ps.sync'}**.

```
AI_CONFIG x
1 {attribute 'ps.sync'}
2 //Настройки аналогового датчика
3 TYPE AI_CONFIG :
4 STRUCT
5     {attribute 'symbol' := 'readwrite'}
6     {attribute 'ps.setpoint':='0.0'}
7     YMIN          : REAL:=0.0;          //Минимум выходной величины
8     {attribute 'symbol' := 'readwrite'}
9     {attribute 'ps.setpoint':='100.0'}
10    YMAX         : REAL:=100.0;        //Максимум выходной величины
11    {attribute 'symbol' := 'readwrite'}
12    {attribute 'ps.setpoint':='0.0'}
13    TF           : REAL:=0.0;          //Постоянная времени фильтра
14    {attribute 'symbol' := 'readwrite'}
15    {attribute 'ps.setpoint':='90.0'}
16    AH2          : REAL:=90.0;          //Порог достоверности верхний №2
17    {attribute 'symbol' := 'readwrite'}
18    {attribute 'ps.setpoint':='90.0'}
19    AH           : REAL:=90.0;          //Порог аварийный верхний
20    {attribute 'symbol' := 'readwrite'}
21    {attribute 'ps.setpoint':='80.0'}
22    WH           : REAL:=80.0;          //Порог предупредительный верхний
23    {attribute 'symbol' := 'readwrite'}
24    {attribute 'ps.setpoint':='25.0'}
25    WL           : REAL:=25.0;          //Порог предупредительный нижний
26    {attribute 'symbol' := 'readwrite'}
27    {attribute 'ps.setpoint':='-20.0'}
28    AL           : REAL:=-20.0;         //Порог аварийный нижний
29    {attribute 'symbol' := 'readwrite'}
30    {attribute 'ps.setpoint':='10.0'}
31    AL2          : REAL:=10.0;          //Порог аварийный нижний №2
32    {attribute 'symbol' := 'readwrite'}
33    {attribute 'ps.setpoint':='0.0'}
34    HYST         : REAL:=0.0;          //Гистерезис
35 END_STRUCT
36 END_TYPE
37
```

8. Объявите переменные в области "VAR\_INPUT". Обратите внимание, что переменная "CONFIG" является переменной структурного типа.

```
AI1 x
1 //Аналоговый датчик
2 FUNCTION_BLOCK AI1
3 VAR_INPUT
4     IN           : REAL;           //Значение сигнала с канала
5     CH_FAULT    : BOOL;           //Ошибка на канале
6     MODULFAULT  : BOOL;           //Ошибка модуля
7     CONFIG      : AI_CONFIG;      //Настройки
8     SIGN_OFF    : BOOL;           //Отключение сигнализации: FALSE - ВКЛ, TRUE - ОТКЛ (из алгоритма)
9     SENSFAULT   : BOOL;           //Флаг недостоверности датчика (из алгоритма)
10    MAN_ON      : BOOL;           //Задание режима: TRUE - режим подмены, FALSE - штатный режим
11    MAN_VALUE   : REAL;           //Значение в режиме подмены
12 END_VAR
13 VAR_OUTPUT
14 END_VAR
15 VAR
16 END_VAR
17
```

9. По умолчанию все переменные ФБ передаются в адресное пространство OPC UA сервера ПЛК. Чем больше переменных передаются в адресное пространство OPC UA сервера, тем выше нагрузка на ПЛК.

Чтобы в адресное пространство OPC UA-сервера передавались только необходимые переменные:

- Над объявлением ФБ выставите атрибут **{attribute 'symbol' := 'none'}**.
- Над переменными, которые необходимо передавать на верхний уровень, выставите атрибут **{attribute 'symbol' := 'readwrite'}**.



Не используйте атрибут передачи данных на ВУ **{attribute 'symbol' := 'readwrite'}** для переменных типа STRING и WSTRING, так как это может привести к повышенной загрузке ПЛК.



```

AI1 x
1 //Аналоговый датчик
2 {attribute 'symbol' := 'none'}
3 FUNCTION_BLOCK AI1
4 VAR_INPUT
5     IN          : REAL;           //Значение сигнала с канала
6     CH_FAULT   : BOOL;           //Ошибка на канале
7     MODULFAULT : BOOL;           //Ошибка модуля
8     {attribute 'symbol' := 'readwrite'}
9     CONFIG     : AI_CONFIG;      //Настройки
10    SIGN_OFF    : BOOL;           //Отключение сигнализации: FALSE - ВКЛ, TRUE - ОТКЛ (из алгоритма)
11    SENSEFAULT  : BOOL;           //Флаг достоверности датчика (из алгоритма)
12    {attribute 'symbol' := 'readwrite'}
13    MAN_ON      : BOOL;           //Задание режима: TRUE - режим подмены, FALSE - штатный режим
14    {attribute 'symbol' := 'readwrite'}
15    MAN_VALUE   : REAL;           //Значение в режиме подмены
16 END_VAR
17 VAR_OUTPUT
18 END_VAR
19 VAR
20 END_VAR
21

```

10. Переменные, которые необходимо резервировать (счетчики, уставки, значения с предыдущих тактов и др.) помечаются атрибутом **{attribute 'ps.add\_redundancy'}**.

```

AI1 x
1 //Аналоговый датчик
2 {attribute 'symbol' := 'none'}
3 FUNCTION_BLOCK AI1
4 VAR_INPUT
5     IN          : REAL;           //Значение сигнала с канала
6     CH_FAULT   : BOOL;           //Ошибка на канале
7     MODULFAULT : BOOL;           //Ошибка модуля
8     {attribute 'symbol' := 'readwrite'}
9     {attribute 'ps.add_redundancy'}
10    CONFIG     : AI_CONFIG;      //Настройки
11    SIGN_OFF    : BOOL;           //Отключение сигнализации: FALSE - ВКЛ, TRUE - ОТКЛ (из алгоритма)
12    SENSEFAULT  : BOOL;           //Флаг достоверности датчика (из алгоритма)
13    {attribute 'symbol' := 'readwrite'}
14    {attribute 'ps.add_redundancy'}
15    MAN_ON      : BOOL;           //Задание режима: TRUE - режим подмены, FALSE - штатный режим
16    {attribute 'symbol' := 'readwrite'}
17    {attribute 'ps.add_redundancy'}
18    MAN_VALUE   : REAL;           //Значение в режиме подмены
19 END_VAR
20 VAR_OUTPUT
21 END_VAR
22 VAR
23 END_VAR
24

```

11. Переменные, которые необходимо хранить в энергонезависимой памяти (уставки) помечаются атрибутом **{attribute 'ps.setpoint':='xxxxx'}**, где xxxxx - начальное значение параметра (значения по умолчанию).

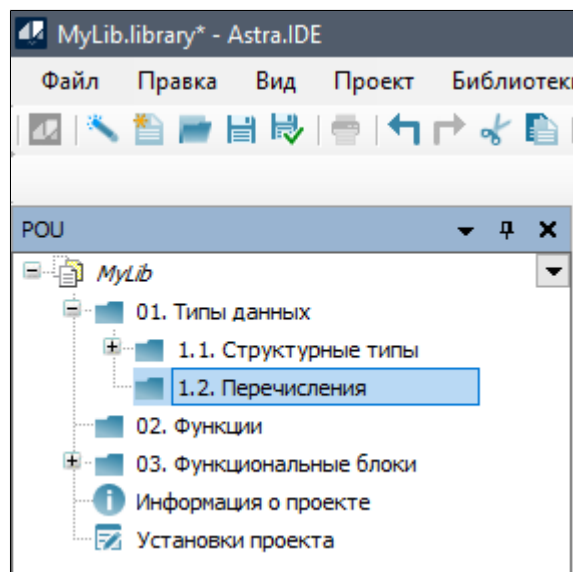
```
AI1 x
1 //Аналоговый датчик
2 {attribute 'symbol' := 'none'}
3 FUNCTION_BLOCK AI1
4 VAR_INPUT
5     IN          : REAL;           //Значение сигнала с канала
6     CH_FAULT    : BOOL;          //Ошибка на канале
7     MODULFAULT  : BOOL;          //Ошибка модуля
8     {attribute 'symbol' := 'readwrite'}
9     {attribute 'ps.add_redundancy'}
10    CONFIG      : AI_CONFIG;      //Настройки
11    SIGN_OFF    : BOOL;          //Отключение сигнализации: FALSE - ВКЛ, TRUE - ОТКЛ (из алгоритма)
12    SENSFAULT   : BOOL;          //флаг недостоверности датчика (из алгоритма)
13    {attribute 'symbol' := 'readwrite'}
14    {attribute 'ps.add_redundancy'}
15    MAN_ON      : BOOL;          //Задание режима: TRUE - режим подмены, FALSE - штатный режим
16    {attribute 'symbol' := 'readwrite'}
17    {attribute 'ps.add_redundancy'}
18    {attribute 'ps.setpoint':='0.0'}
19    MAN_VALUE   : REAL;          //Значение в режиме подмены
20 END_VAR
21 VAR_OUTPUT
22 END_VAR
23 VAR
24 END_VAR
25
```

## 1.2.2.2.3. Объявление выходных переменных

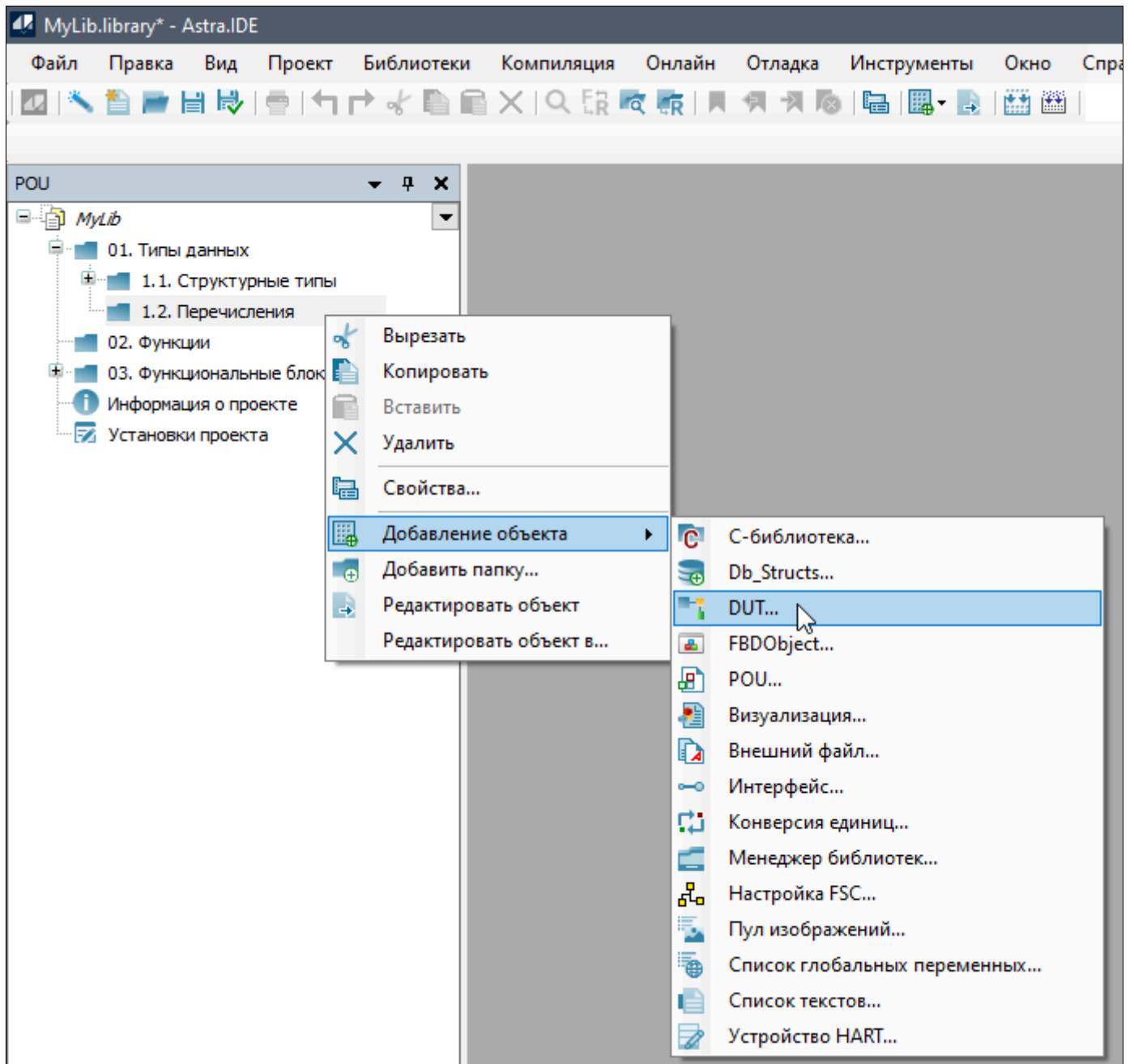
В области VAR\_OUTPUT необходимо объявить выходные переменные состояния функционального блока.

Чтобы инженеру АСУ ТП было удобнее производить отладку алгоритмов, создадим переменные для сигнализации (SIGN), диагностики (DIAGN) и качества аналогового сигнала (QUALITY) в виде перечислений.

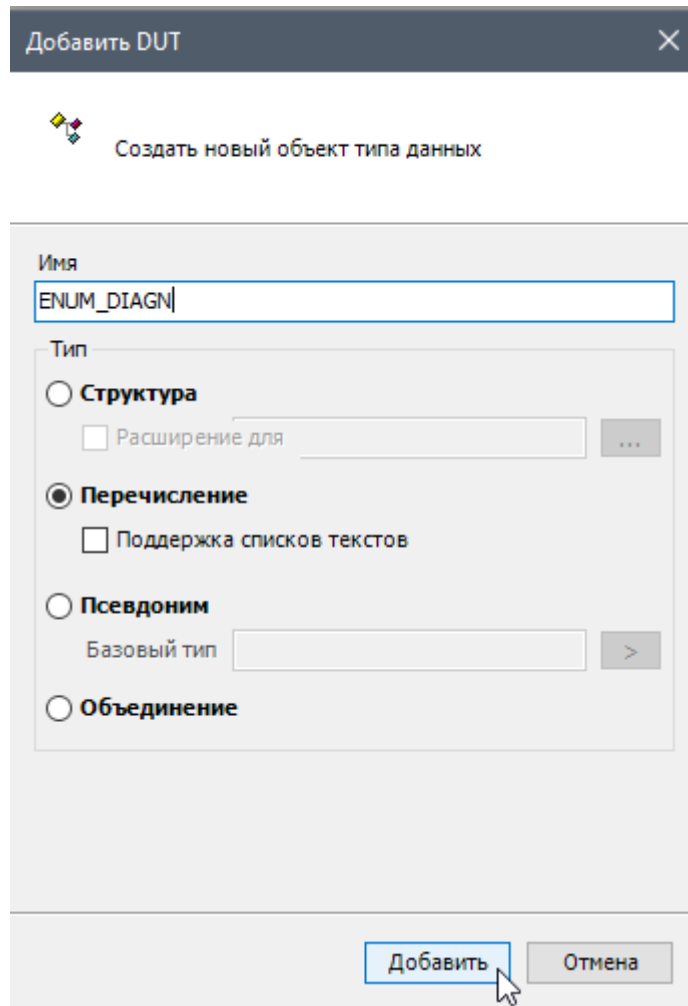
1. Создайте в папке "01. Типы данных" вложенную папку "1.2. Перечисления".



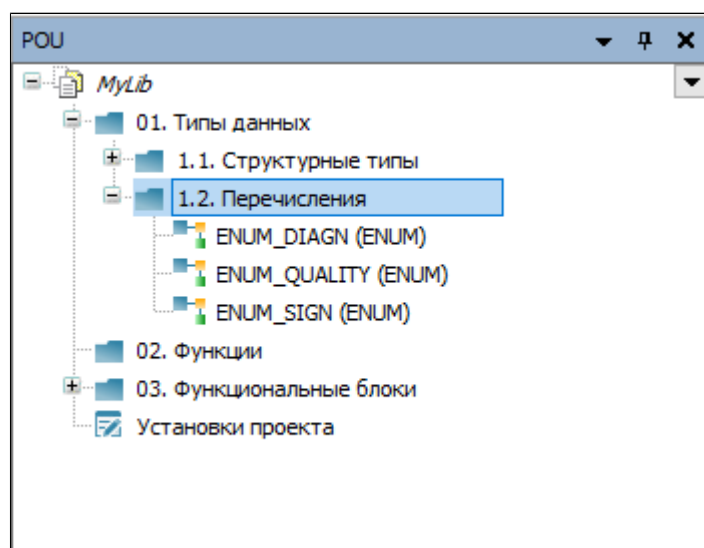
2. Выделите папку "1.2. Перечисления" и в контекстном меню выберите команду "Добавление объекта" -> "DUT...".



3. Выберите тип "Перечисление", укажите название "ENUM\_DIAGN" и нажмите кнопку "Добавить".



4. Аналогичным образом добавьте перечисление "ENUM\_SIGN" и "ENUM\_QUALITY". Должна получиться следующая структура объектов:



5. В редакторе типа "ENUM\_DIAGN" удалите атрибуты `{attribute 'qualified_only'}`, `{attribute 'strict'}` и укажите возможные значения :

```
ENUM_DIAGN x
1 {attribute 'ps.sync'}
2 //Диагноз канала
3 TYPE ENUM_DIAGN :
4 (
5     FAULT      :=0,    //Недостоверный
6     NOCOMPARE :=1,    //Несравнение
7     OK         :=2     //OK
8 );
9 END_TYPE
```

Отметьте тип "ENUM\_DIAGN" атрибутом **{attribute 'ps.sync'}**, чтобы определить экземпляры данного типа как безопасные для резервирования.



Не забывайте указывать комментарии.

6. В редакторе типа "ENUM\_SIGN" аналогично удалите атрибуты **{attribute 'qualified\_only'}**, **{attribute 'strict'}** и укажите возможные значения:

```
ENUM_SIGN x
1 {attribute 'ps.sync'}
2 //Сигнализация порогов
3 TYPE ENUM_SIGN :
4 (
5     UNKNOWN    :=0,    //Ошибка задания порогов
6     AH         :=1,    //Достигнут верхний аварийный порог
7     AL         :=2,    //Достигнут нижний аварийный порог
8     WH         :=3,    //Достигнут верхний предупредительный порог
9     WL         :=4,    //Достигнут нижний предупредительный порог
10    OK          :=5,    //Параметр находится в пределах порога
11    OFF         :=6,    //Сигнализация отключена
12    AH2        :=7,    //Достигнут верхний аварийный порог №2
13    AL2        :=8,    //Достигнут нижний аварийный порог №2
14 );
15 END_TYPE
```

Отметьте тип "ENUM\_SIGN" атрибутом **{attribute 'ps.sync'}**, чтобы определить экземпляры данного типа как безопасные для резервирования.



Не забывайте указывать комментарии.

7. В редакторе типа "ENUM\_QUALITY" аналогично удалите атрибуты {attribute 'qualified\_only'}, {attribute 'strict'} и укажите возможные значения:

```
ENUM_QUALITY x
1  {attribute 'ps.sync'}
2  TYPE ENUM_QUALITY :
3  (
4      UNKNOWN      :=0,          //Неопределенное
5      MODULFAULT   :=1,          //Неисправность модуля
6      BREAK        :=2,          //Обрыв
7      OVERLOAD     :=3,          //Перегрузка
8      SENSFAULT    :=4,          //Отказ датчика. Выход сигнала за пороги достоверности
9      OFF          :=5,          //Отключен пользователем или программой
10     FROZEN       :=6,          //Заморожено
11     REPLACE      :=7,          //Подмена значения
12     NOCOMPARE    :=8,          //Несравнение
13     OK           :=9,          //Хорошее
14 );
15 END_TYPE
```

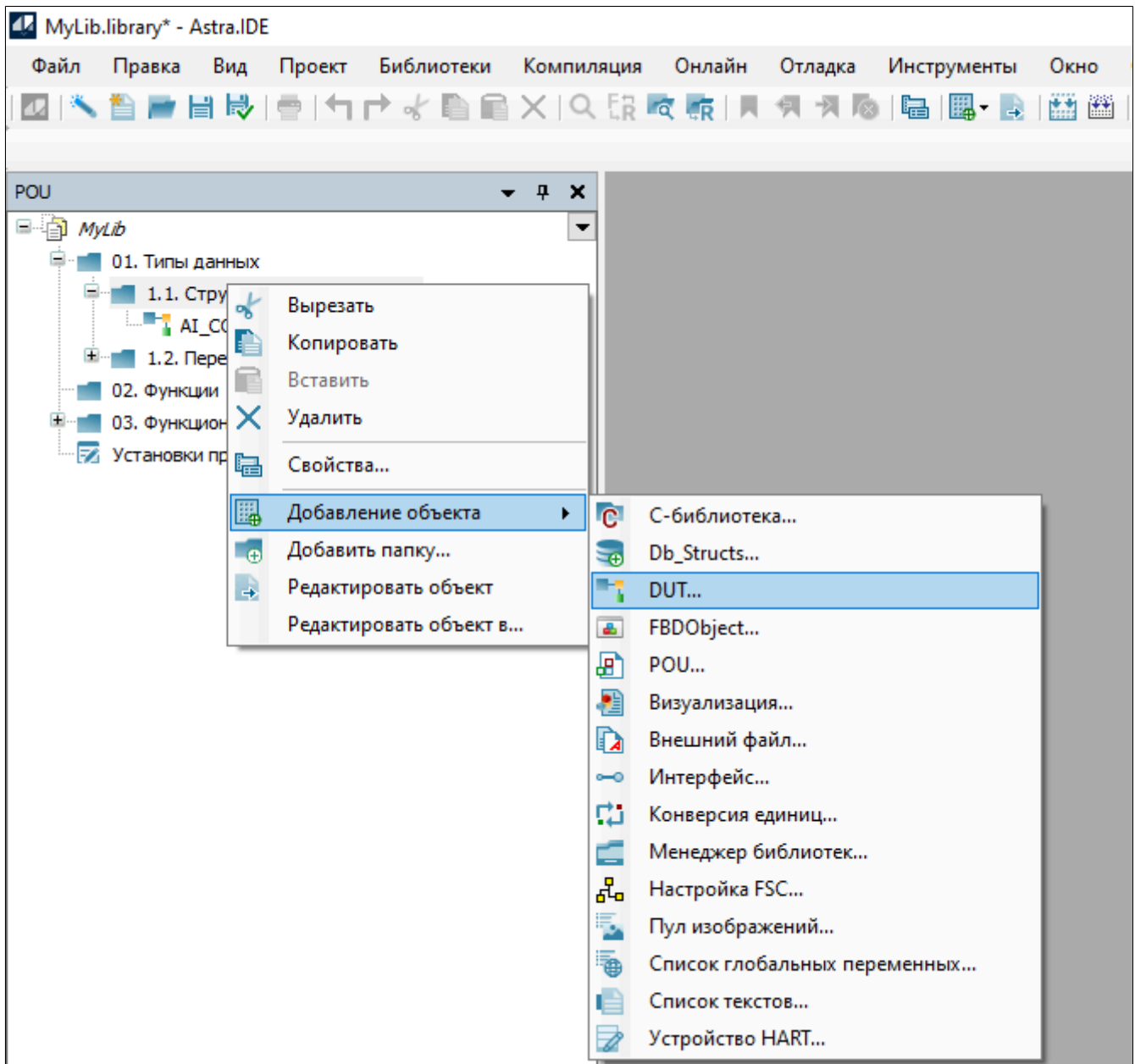
Отметьте тип "ENUM\_QUALITY" атрибутом {attribute 'ps.sync'}, чтобы определить экземпляры данного типа как безопасные для резервирования.



Не забывайте указывать комментарии.

Выходная переменная "OUT" будет представлена в виде переменной структурного типа "AI", который будет включать в себя значение (VALUE) и качество (QUALITY) аналогового сигнала.


8. Выделите папку "1.1. Структурные типы" и в контекстном меню выберите команду "Добавление объекта" -> "DUT...".



9. Укажите название структурного типа "AI" и нажмите кнопку "Добавить".



Добавить DUT ✕

 Создать новый объект типа данных

---

Имя

Тип

Структура  
 Расширение для  ...

Перечисление  
 Поддержка списков текстов

Псевдоним  
 Базовый тип  >

Объединение

---

10. В редакторе структурного типа "AI" добавьте переменные, отвечающие за значение типа REAL и качество аналогового сигнала типа ENUM\_QUALITY.

```

AI x
1  {attribute 'ps.sync'}
2  //Аналоговый выходной сигнал
3  TYPE AI :
4  STRUCT
5      VALUE      : REAL;           //Значение
6      QUALITY    : ENUM_QUALITY;  //Качество сигнала
7  END_STRUCT
8  END_TYPE

```

Отметьте структурный тип "AI" атрибутом **{attribute 'ps.sync'}**, чтобы определить экземпляры данного типа как безопасные для резервирования.

11. Объявите переменные в области "VAR\_OUTPUT". Обратите внимание, что переменная "OUT" является переменной структурного типа "AI", а переменные "SIGN" и "DIAGN" - имеют тип "Перечисление".



Над переменными, которые необходимо передавать на верхний уровень, выставите атрибут **{attribute 'symbol' := 'readwrite'}**.



Над переменными, которые необходимо резервировать, выставите атрибут **{attribute 'ps.add\_redundancy'}**.

```
AI1 x
1 //Аналоговый датчик
2 {attribute 'symbol' := 'none'}
3 FUNCTION_BLOCK AI1
4 VAR_INPUT
5     IN          : REAL;           //Значение сигнала с канала
6     CH_FAULT    : BOOL;           //Ошибка на канале
7     MODULFAULT  : BOOL;           //Ошибка модуля
8     {attribute 'symbol' := 'readwrite'}
9     {attribute 'ps.add_redundancy'}
10    CONFIG      : AI_CONFIG;      //Настройки
11    SIGN_OFF    : BOOL;           //Отключение сигнализации: FALSE - ВКЛ, TRUE - ОТКЛ (из алгоритма)
12    SENSFAULT   : BOOL;           //Флаг недостоверности датчика (из алгоритма)
13    {attribute 'symbol' := 'readwrite'}
14    {attribute 'ps.add_redundancy'}
15    MAN_ON      : BOOL;           //Задание режима: TRUE - режим подмены, FALSE - штатный режим
16    {attribute 'symbol' := 'readwrite'}
17    {attribute 'ps.add_redundancy'}
18    {attribute 'ps.setpoint':='0.0'}
19    MAN_VALUE   : REAL;           //Значение в режиме подмены
20 END_VAR
21 VAR_OUTPUT
22     {attribute 'symbol' := 'readwrite'}
23     {attribute 'ps.add_redundancy'}
24     OUT         : AI;             //Выходной сигнал
25     {attribute 'symbol' := 'readwrite'}
26     FACT_VALUE  : REAL;           //Фактическое значение сигнала
27     {attribute 'symbol' := 'readwrite'}
28     SIGN        : ENUM_SIGN;      //Сигнализация
29     {attribute 'symbol' := 'readwrite'}
30     DIAGN       : ENUM_DIAGN;     //Диагноз
31 END_VAR
32 VAR
33 END_VAR
```

Таким образом, мы объявили входные/выходные переменные функционального блока. Внутренние и временные переменные объявляются аналогичным образом.



Не используйте в функциональных блоках вход-выходные переменные.

## 1.2.2.2.4. Основные правила формирования библиотечного ФБ

1. Над объявлением ФБ ставится атрибут `{attribute 'symbol' := 'none'}`.
2. Над переменными, которые необходимо передавать на верхний уровень (ВУ) необходимо поставить атрибут `{attribute 'symbol' := 'readwrite'}`.
3. Переменные, которые необходимо резервировать (счетчики, уставки, значения с предыдущих тактов и др.) помечаются атрибутом `{attribute 'ps.add_redundancy'}`.
4. Переменные, которые необходимо хранить в энергонезависимой памяти (уставки) помечаются атрибутом `{attribute 'ps.setpoint':='xxxxx'}`, где xxxxx - начальное значение параметра (значения по умолчанию).

### ОБЯЗАТЕЛЬНЫЕ ОГРАНИЧЕНИЯ



Не допускается использование в ФБ вход-выходных переменных (IN/OUT).



Не допускается использование наследования в ФБ.



Не допускается использование атрибута передачи данных на ВУ `{attribute 'symbol' := 'readwrite'}` для переменных типа STRING и WSTRING.

## 1.2.2.2.5. Редактор кода

В редактор кода необходимо прописать алгоритм, реализующий логику функционального блока.

```
AI1 x
1 //Аналоговый датчик
2 {attribute 'symbol' := 'none'}
3 FUNCTION_BLOCK AI1
4 VAR_INPUT
5     IN           : REAL;           //Значение сигнала с канала
6     CH_FAULT    : BOOL;           //Ошибка на канале
7     MODULEFAULT : BOOL;           //Ошибка модуля
8     {attribute 'symbol' := 'readwrite'}
9     {attribute 'ps.add_redundancy'}
10    CONFIG       : AI_CONFIG;      //Настройки
11    SIGN_OFF     : BOOL;           //Отключение сигнализации: FALSE - ВКЛ, TRUE - ОТКЛ (из алгоритма)
12    SENSFAULT    : BOOL;           //Флаг недостоверности датчика (из алгоритма)
13    {attribute 'symbol' := 'readwrite'}
14    {attribute 'ps.add_redundancy'}
15    MAN_ON       : BOOL;           //Задание режима: TRUE - режим подмены, FALSE - штатный режим
16    {attribute 'symbol' := 'readwrite'}
17    {attribute 'ps.add_redundancy'}
18    {attribute 'ps.setpoint':='0.0'}
19    MAN_VALUE    : REAL;           //Значение в режиме подмены
20 END_VAR
21 VAR_OUTPUT
22 VALUE:=OUT.VALUE;
23
24 // Анализ порогов сигнализации
25 SIGN1( IN           := OUT.VALUE,
26        OFF          := SIGN_OFF,
27        AH2          := CONFIG.AH2,
28        AH           := CONFIG.AH,
29        AL           := CONFIG.AL,
30        AL2          := CONFIG.AL2,
31        WH           := CONFIG.WH,
32        WL           := CONFIG.WL,
33        HYST        := CONFIG.HYST,
34        OUT         => SIGN);
```



Не забывайте оставлять комментарии в коде.

**Комментарии** — поясняющие строки в программном коде, которые позволяют понять смысл написанного. Они пишутся для людей, но игнорируются компиляторами и интерпретаторами.

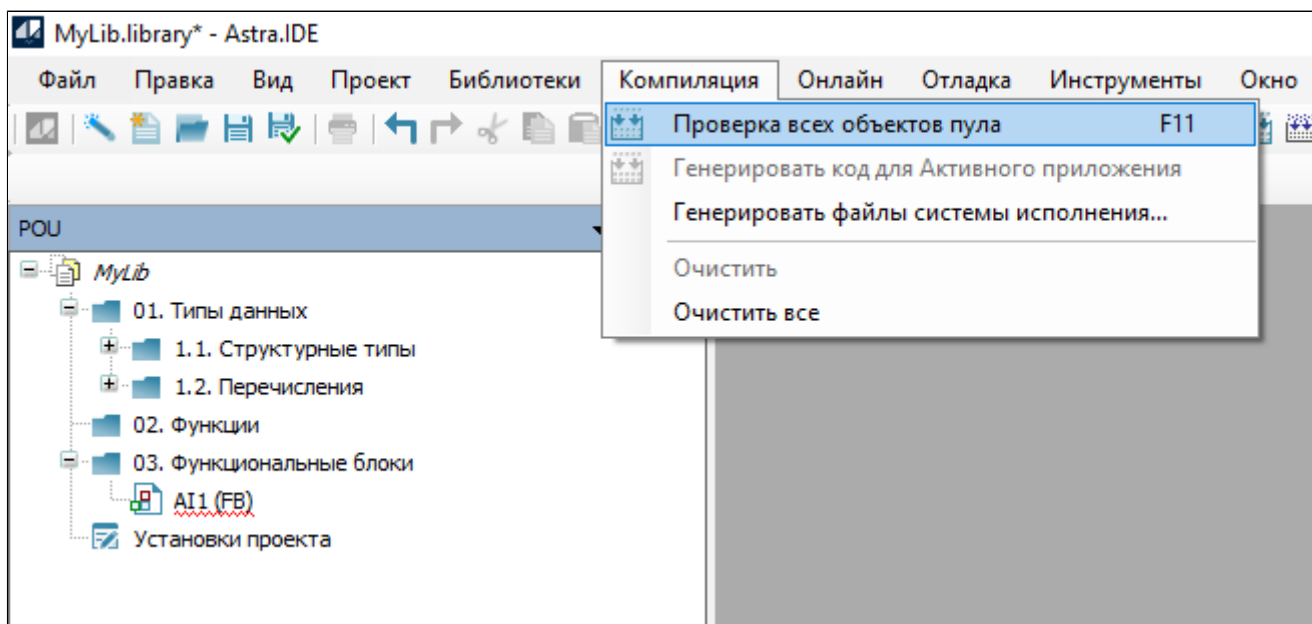
Комментарии помещаются прямо над кодом, к которому они относятся. Так проще понять, о чём речь, не вникая в содержание каждой строчки. Совсем короткие пояснения можно писать справа.

Комментарии полезны тем, что:

- › Помогают быстрее разобраться в коде — если появилась ошибка или нужно что-то изменить в программе. Это важно и разработчику, и тем, кто будет заниматься кодом после него.
- › Не дают запутаться в логике — при создании новых библиотек, процедур, функций и системных переменных.
- › Объясняют результаты работы — при отладке или проверке программы.
- › Описывают сложные алгоритмы и формулы — в математических, физических или экономических расчётах и других сложных вычислениях. Это позволяет разобраться в готовом коде тем, у кого нет глубоких знаний в какой-то предметной области.

После создания алгоритма скомпилируйте проект.

Чтобы скомпилировать проект, выберите команду в меню "Компиляция" -> "Проверка всех объектов пула (F11)".



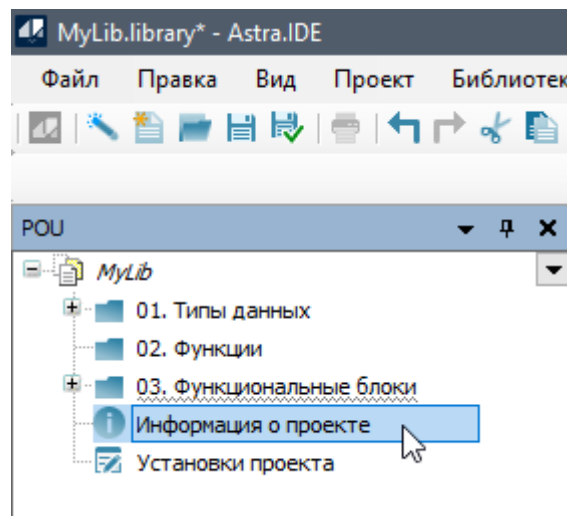
Компилятор проверит Ваш код на возможные ошибки. При наличии ошибок в объекте он будет подчеркнут красной волнистой линией.

## 1.2.2.3. Информация о проекте

Перед выпуском библиотеки необходимо задать информацию о проекте.

Для этого выполните следующие действия:

1. Откройте редактор "Информация о проекте" двойным кликом мыши в дереве объектов.



2. В открывшемся окне заполните поля, при необходимости установите флаг "Выпущен" и нажмите кнопку "ОК".

Информация проекта

Файл | **Общее** | Свойства | Статистика | Лицензия

**Компания** REGLAB

**Заголовок** MyLib

**Версия** 1.0.0.0  Выпущен

Категории библиотек  ...

Доп. имя MyLib

Плейшолдер

Автор REGLAB

Описание

Поля с подписями жирным используются для идентификации библиотек.

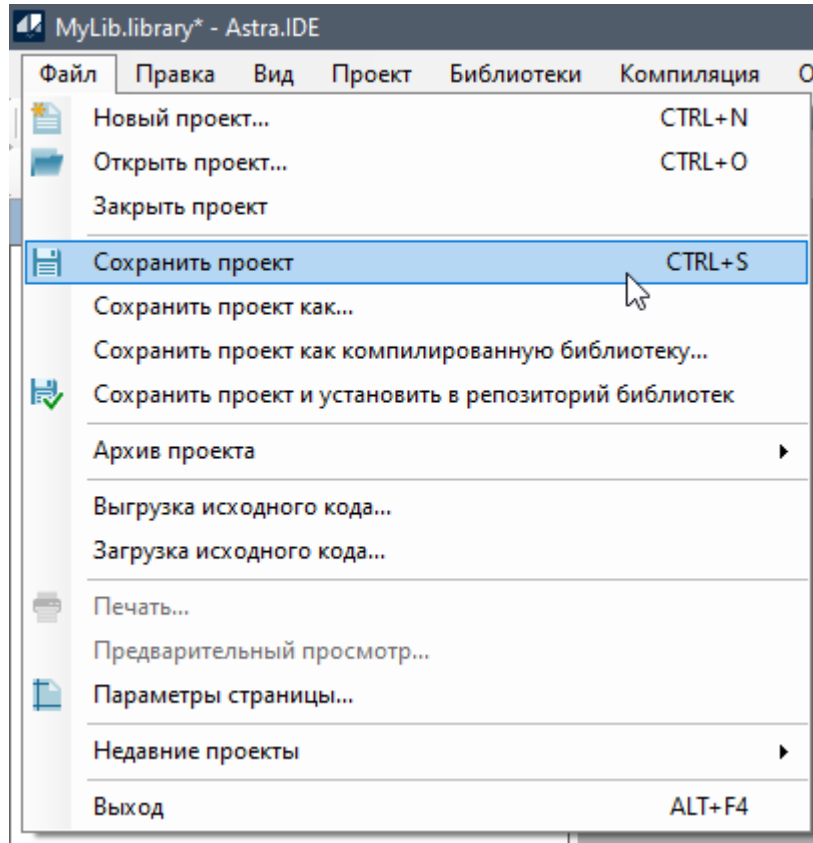
Автоматически генерировать POU 'Информация библиотеки'

Автоматически генерировать POU 'Информация проекта'

OK Отмена

## 1.2.2.4. Сохранение библиотеки

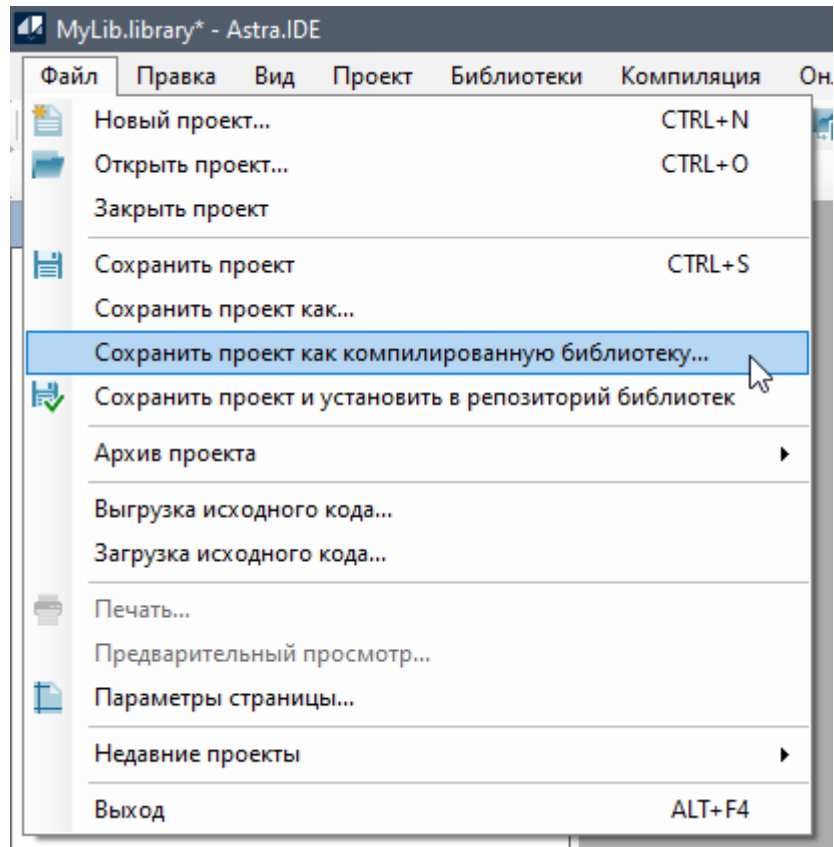
Чтобы сохранить проект, выберите команду в меню "Файл" -> "Сохранить проект".



Библиотека будет сохранена с расширением \*.library.

Чтобы сохранить библиотеку в скомпилированном (закрытом) виде, выберите команду в главном меню "Файл" -> "Сохранить проект как скомпилированную библиотеку...".





Библиотека будет сохранена с расширением \*.compiled-library.

## 1.2.3. Разработка библиотеки типов в Astra.AStudio

[Создание проекта библиотеки](#)

[Создание аспектов](#)

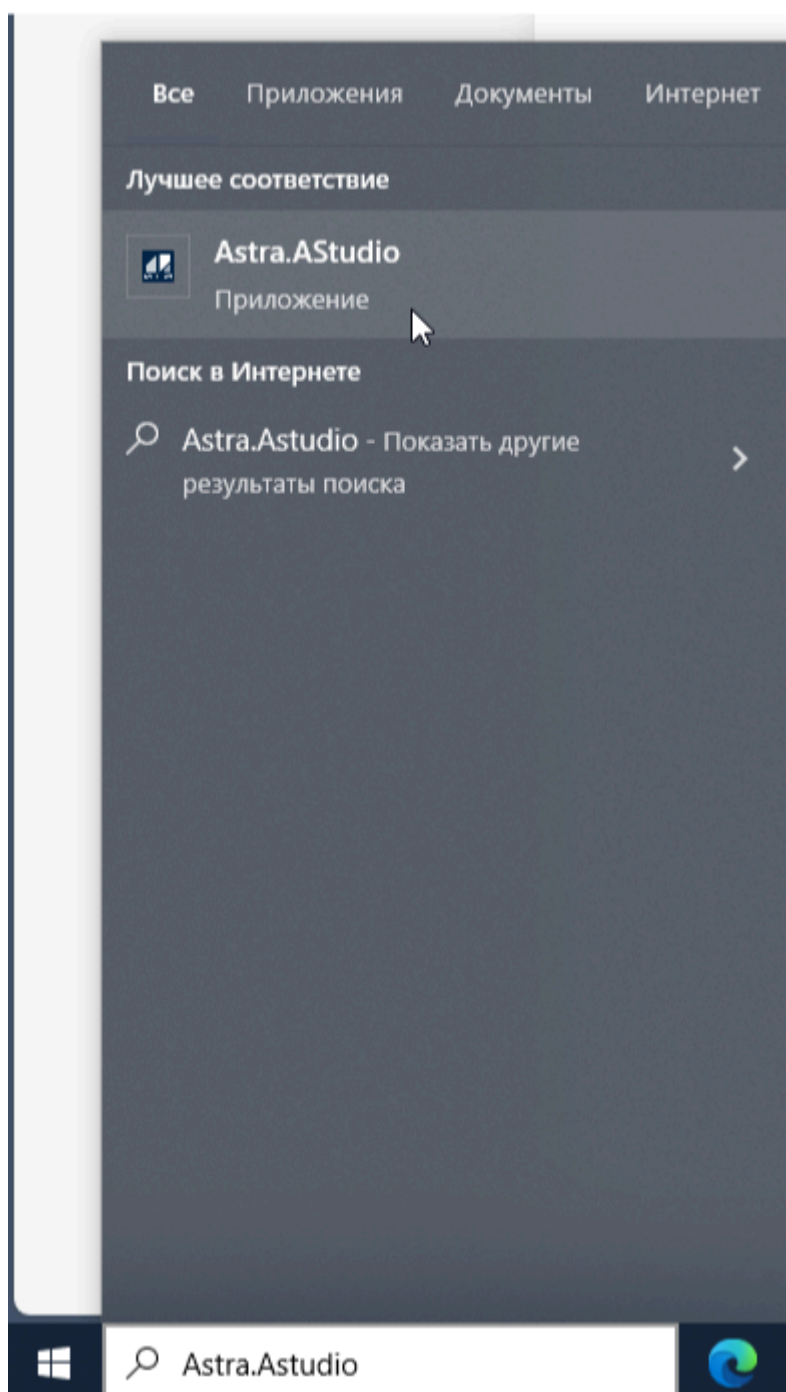
[Создание агрегаторов](#)

[Создание типа](#)

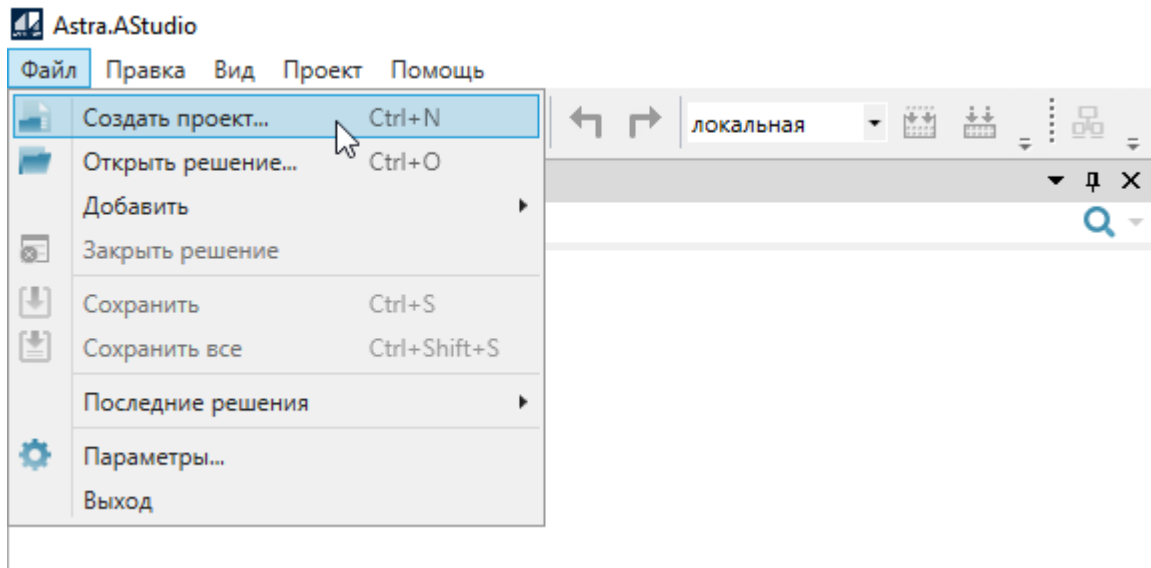
## 1.2.3.1. Создание проекта библиотеки

Чтобы создать новый проект библиотеки типов, выполните следующие действия:

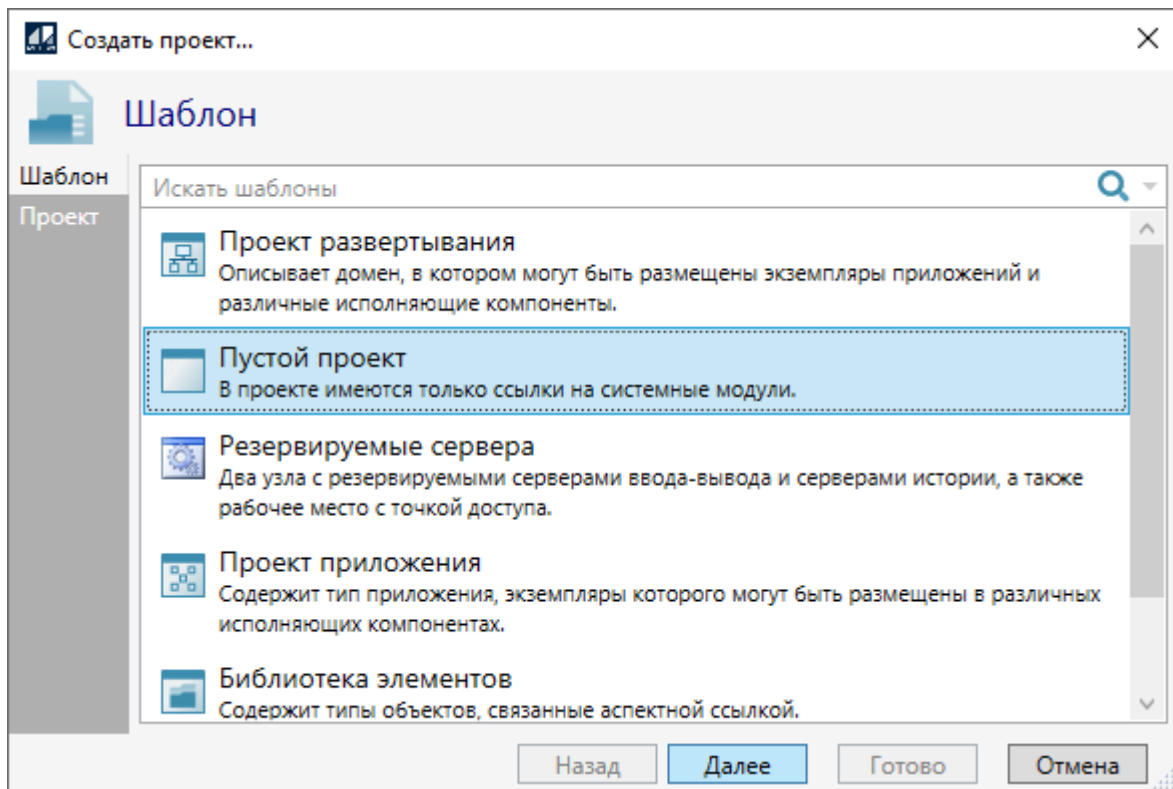
1. Запустите configurator Astra.AStudio.



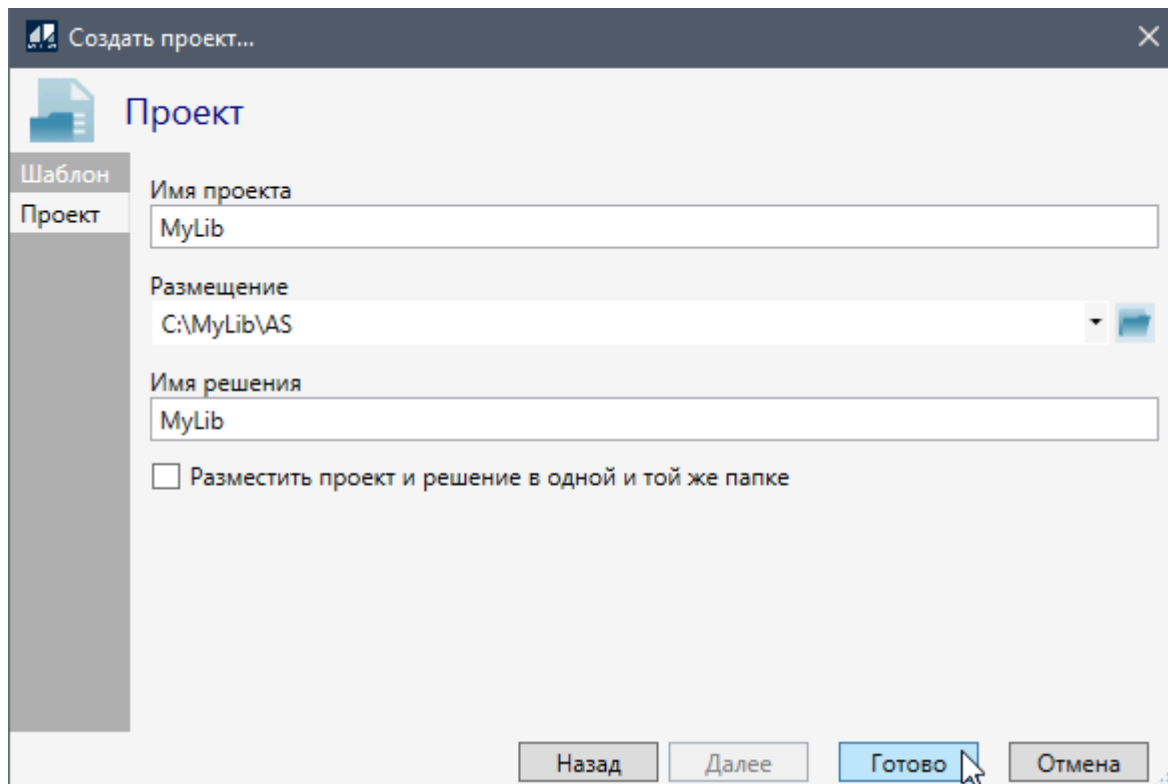
2. Во вкладке "Файл" главного меню выберите команду "Создать проект".



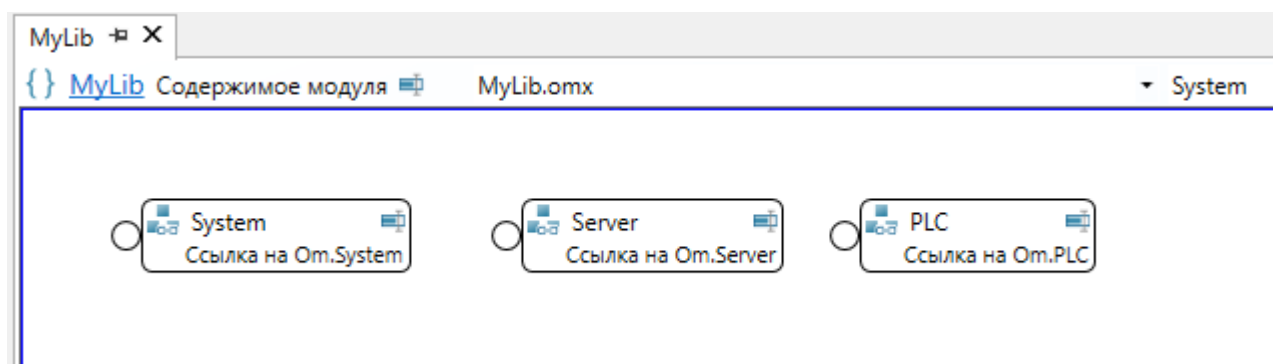
3. В открывшемся окне выберите шаблон "Пустой проект" и нажмите кнопку "Далее".



4. Заполните поля "Имя проекта", "Имя решения" и выберите каталог библиотеки. Нажмите кнопку "Готово".



В созданном решении появятся ссылки на подключенные системные библиотеки.



К одному решению может быть подключено несколько проектов.

## 1.2.3.2. Создание аспектов

**Аспект** - метка или флаг принадлежности к тому или иному представлению.

В нашей библиотеке мы создадим два представления:

- PLC\_Aспект - для представления данных с ПЛК.
- IOS\_Aспект - для представления данных на сервере ввода/вывода.

Уровень представления ПЛК представляет те переменные, которые передаются с ПЛК.

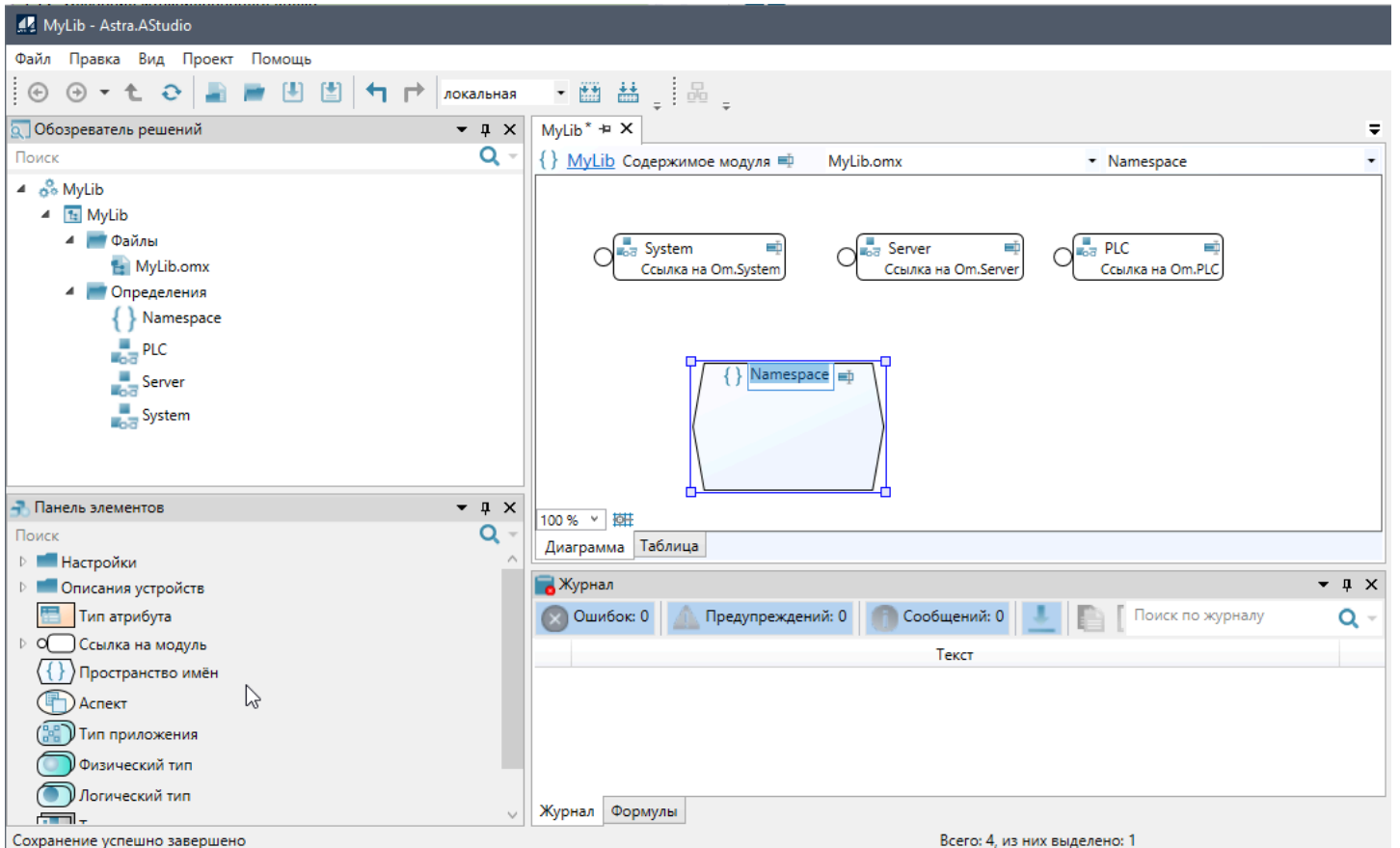
Уровень представления сервера ввода/вывода представляет те переменные, которые передаются на HMI или сторонние системы (MES, ERP и т.д).



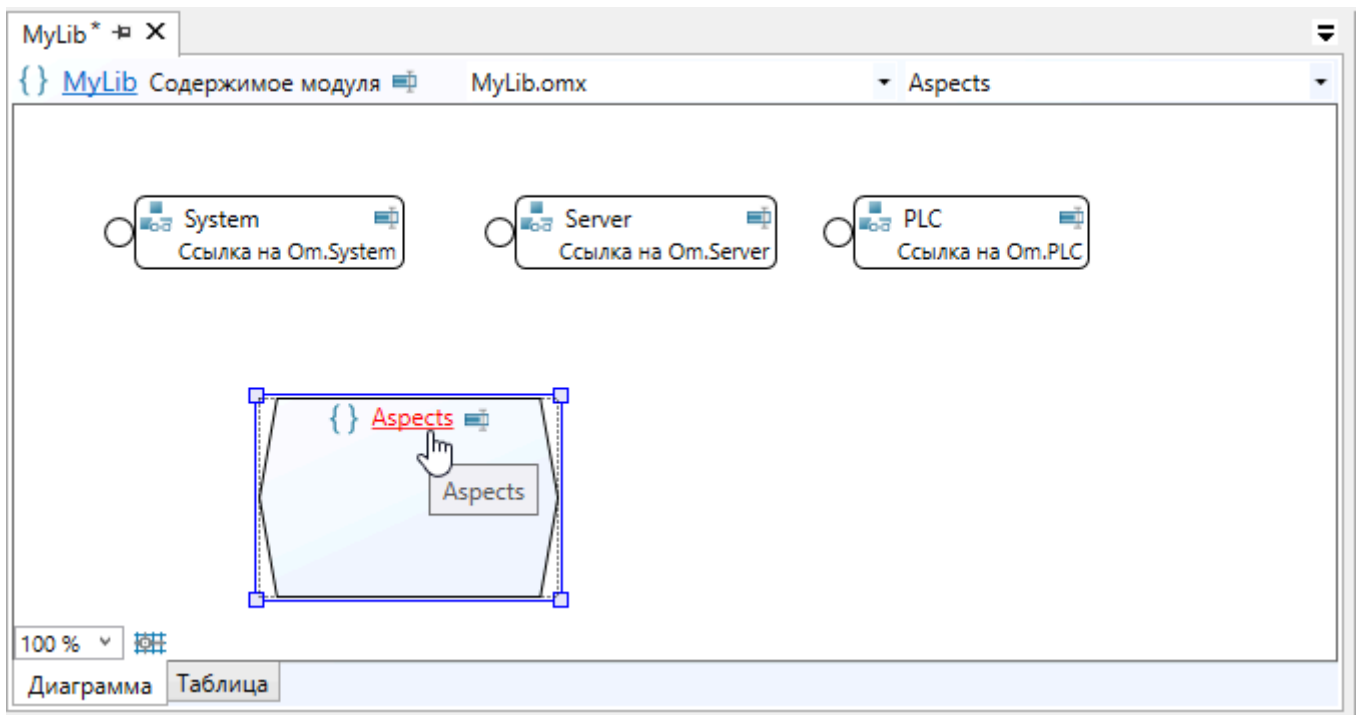
Аспекты используются для быстрого размещения объектов, для которых описаны типы.

Чтобы создать аспекты, выполните следующие действия:

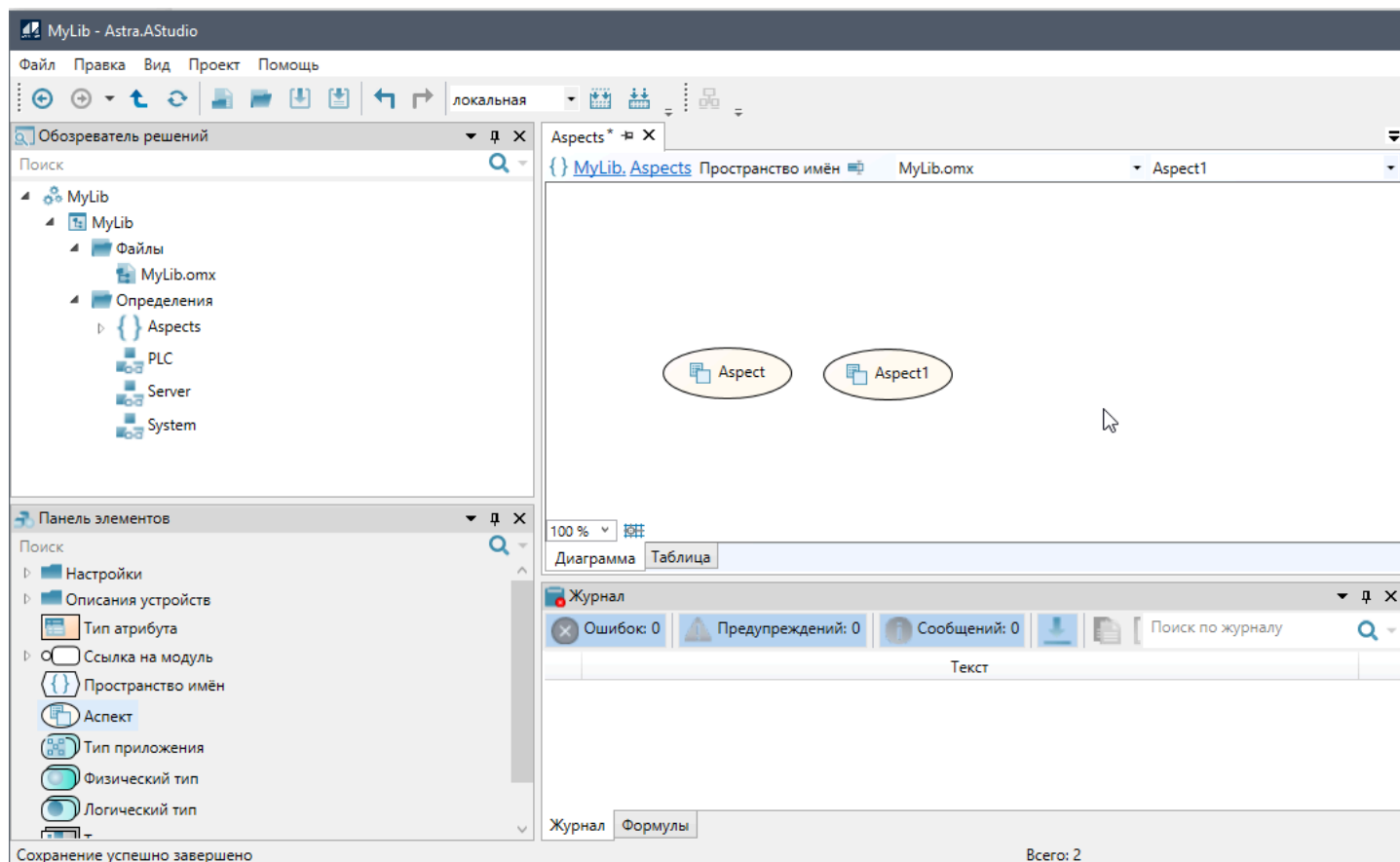
1. Добавьте на рабочую область "Пространство имен" путем перетаскивания из панели элементов:



2. Задайте в свойствах добавленному элементу имя "Aspects" и перейдите в него, кликнув по названию:

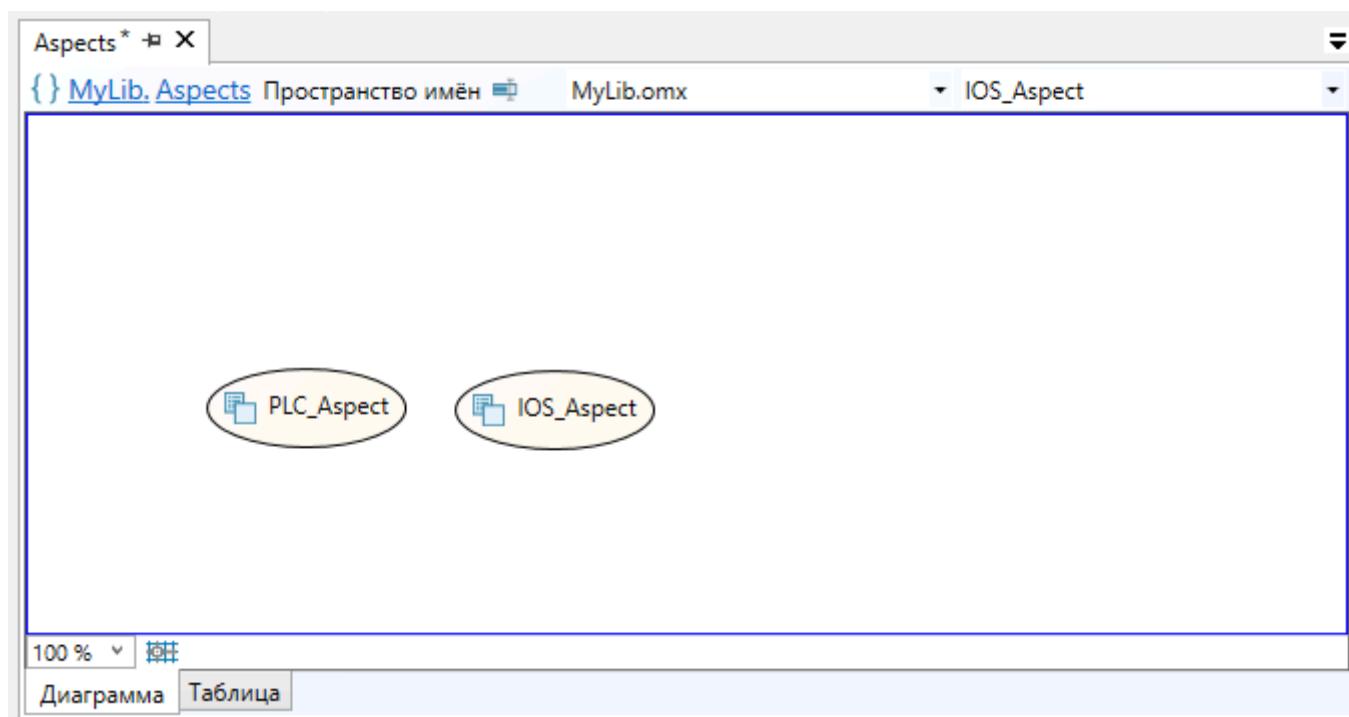


3. В пространство имен "Aspects" добавьте из панели элементов два аспекта:



#### 4. Задайте добавленным аспектам следующие названия:

- PLC\_Aspect - для представления данных с ПЛК.
- IOS\_Aspect - для представления данных на сервере ввода/вывода.





## 1.2.3.3. Создание агрегаторов

**Агрегация событий** – это объединение событий объекта в единую структуру.

Использование агрегации позволяет:

- › определять наличие активных и/или неквитированных событий в объекте;
- › определять максимальную важность среди событий в объекте;
- › квитировать сразу все активные события в объекте.

Для агрегации событий объекта, нужно добавить в него агрегатор.



Агрегатор агрегирует события только того объекта, в который он вложен, не включая вложенные в него объекты.

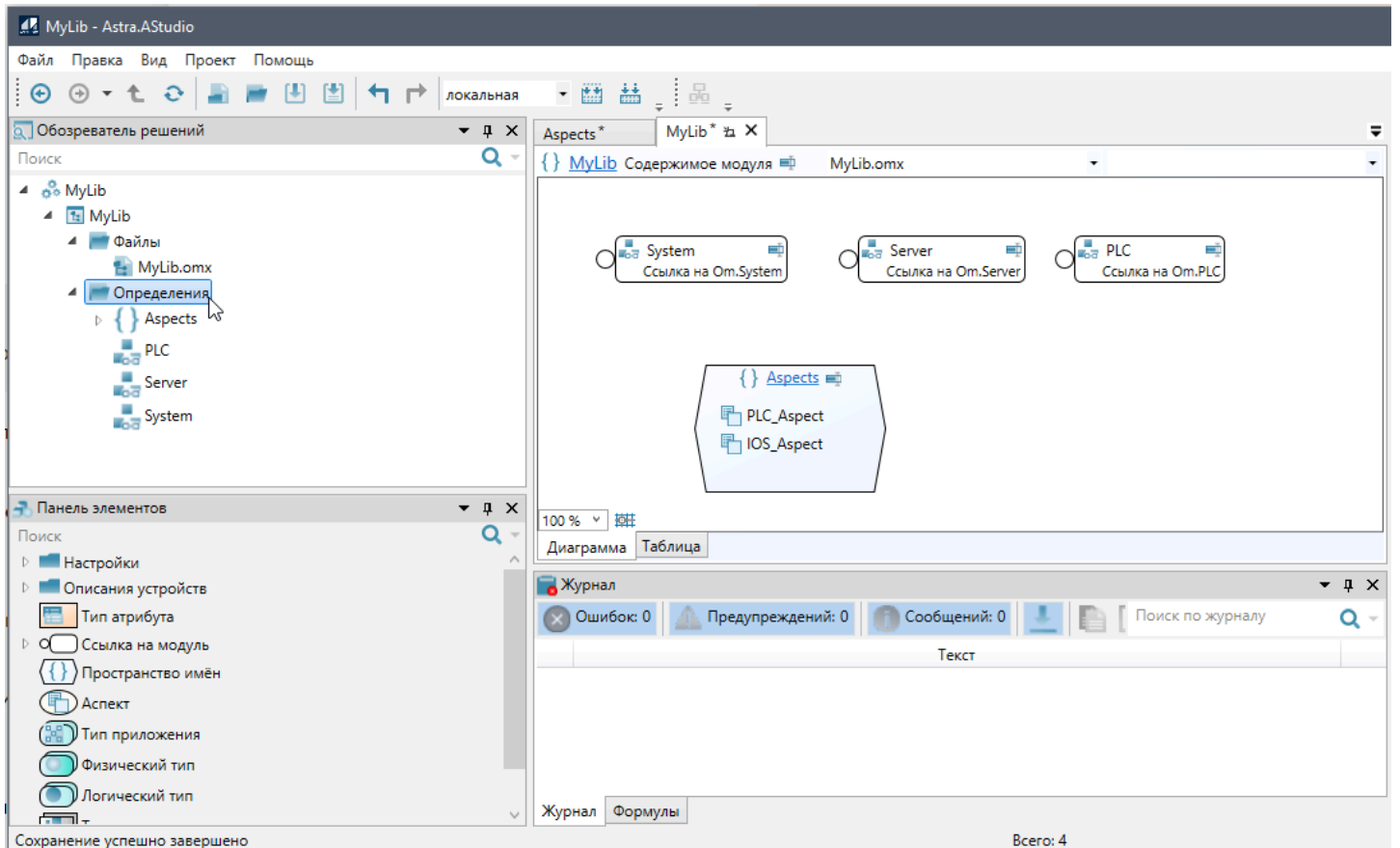
Чтобы агрегировать все события в ветке (в объекте и во всех вложенных объектах), необходимо добавить агрегатор в каждый вложенный объект. Они будут передавать агрегируемую информацию агрегатору, который находится в объекте выше.



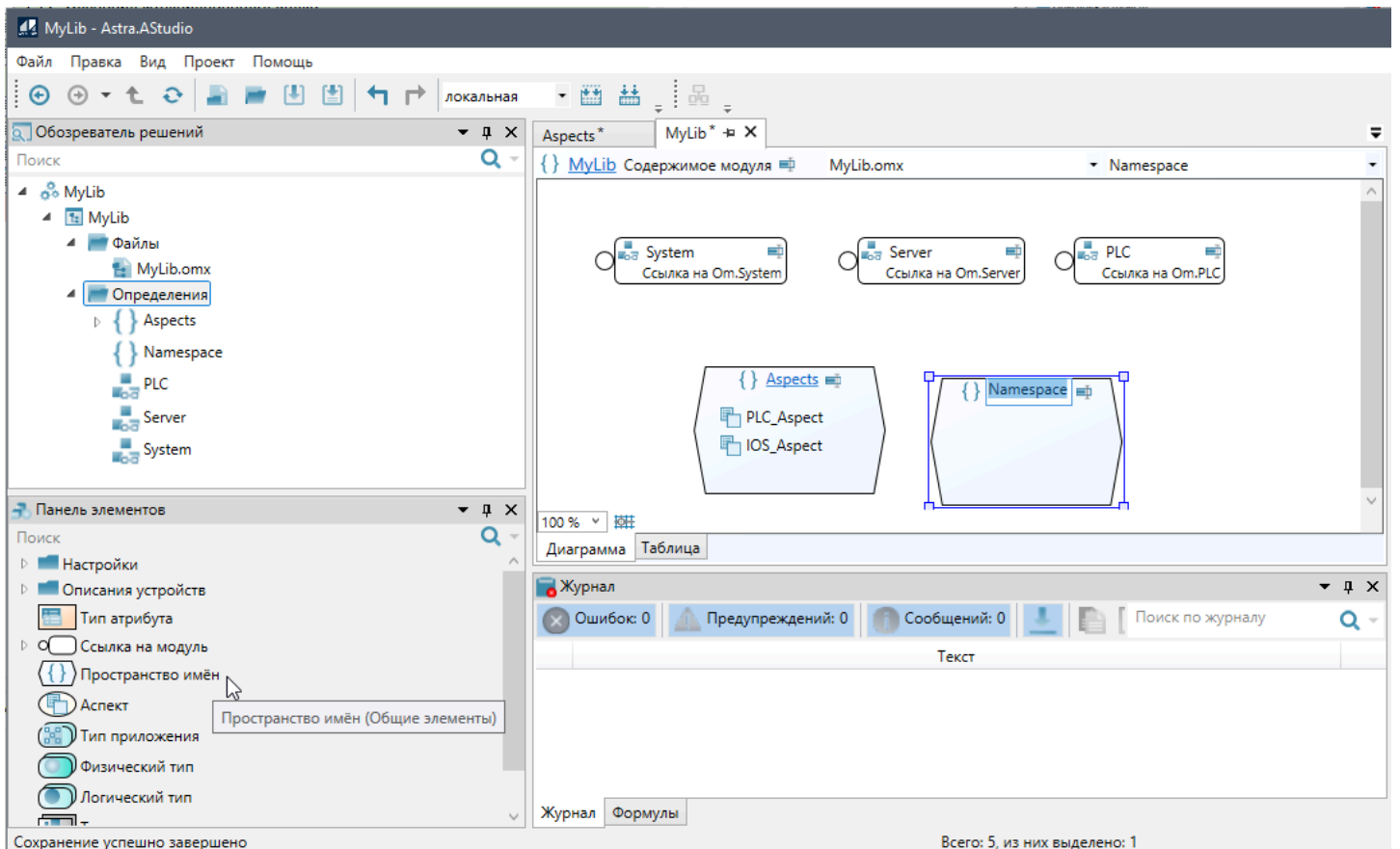
Чтобы информация передавалась между агрегаторами, они должны быть одного типа.

Чтобы создать агрегаторы событий, необходимо выполнить следующие действия:

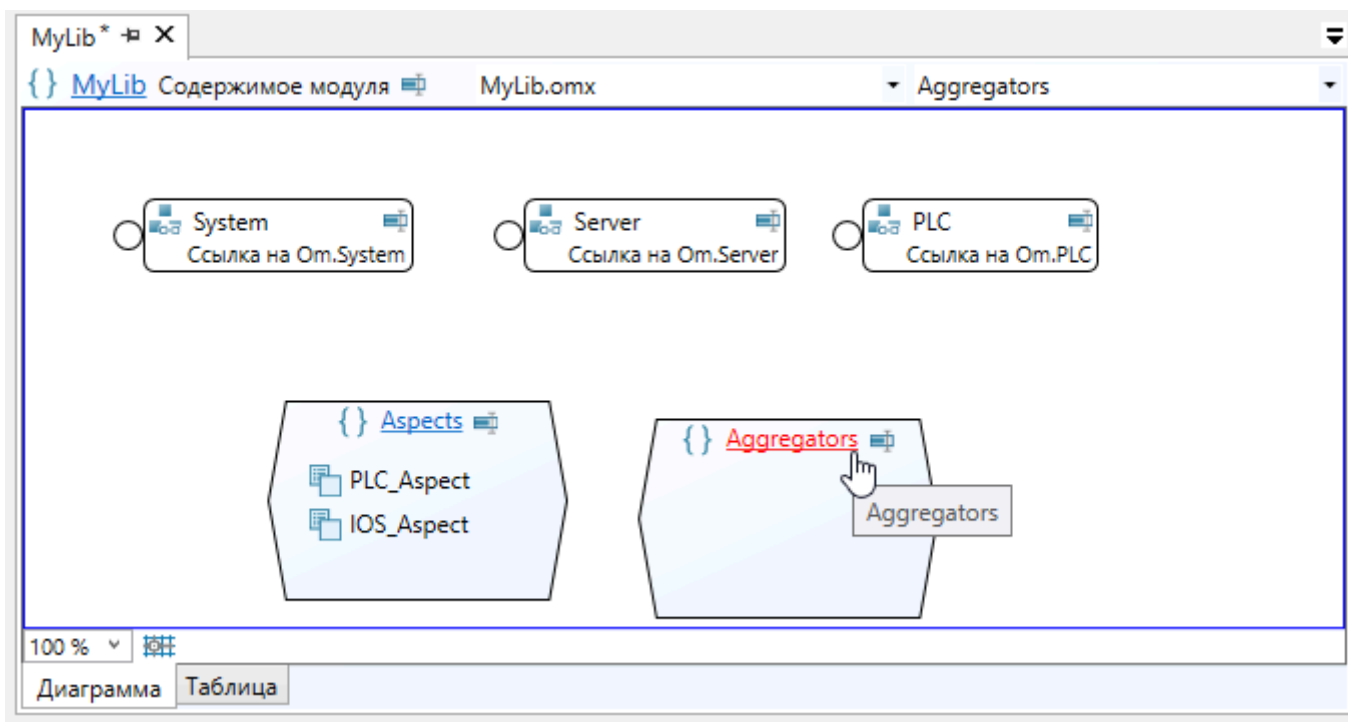
1. Перейдите в определения проекта "MyLib" двойным кликом по объекту "Определения" в обозревателе решений.



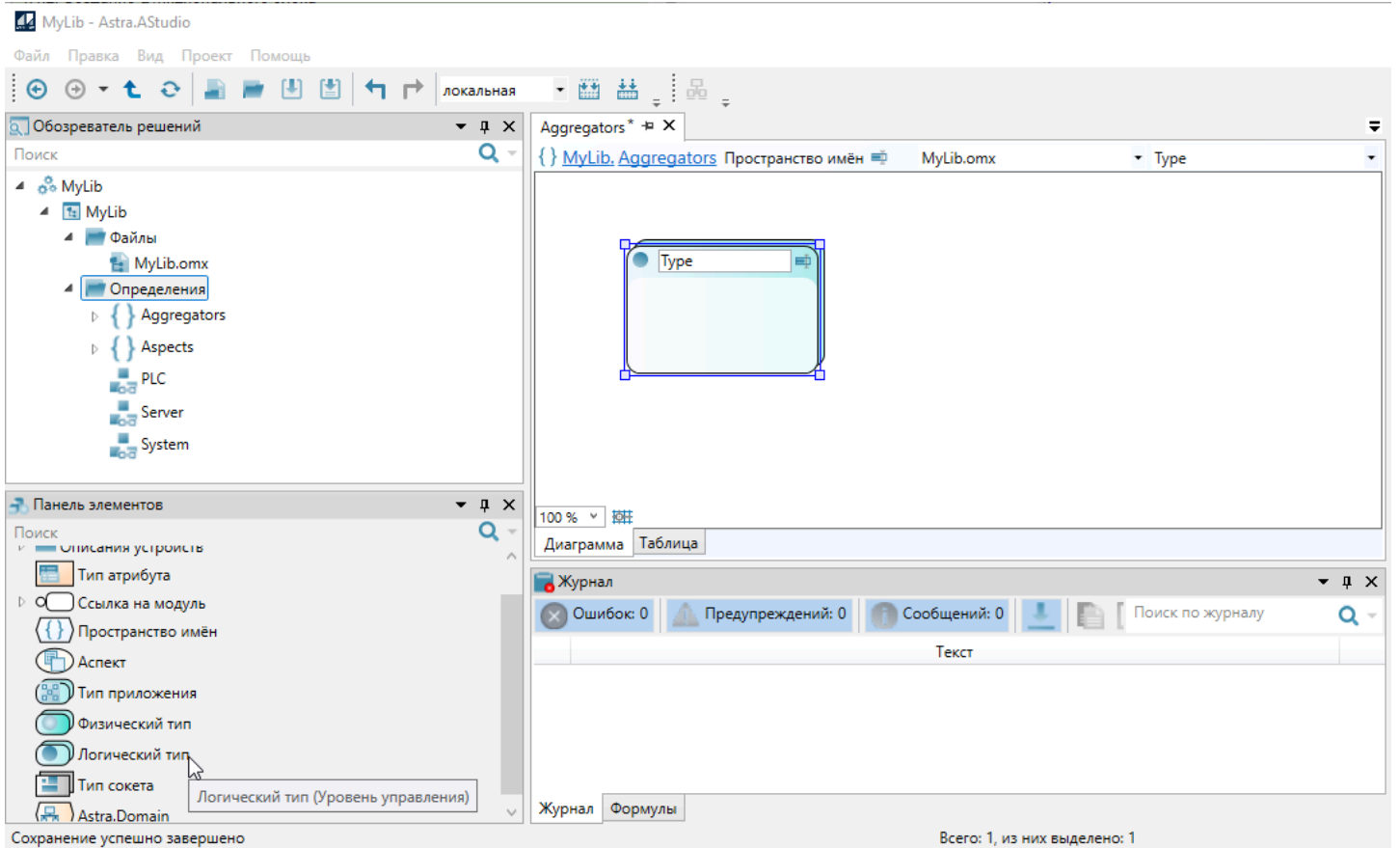
2. Добавьте на рабочую область "Пространство имен" из панели элементов путем перетаскивания.



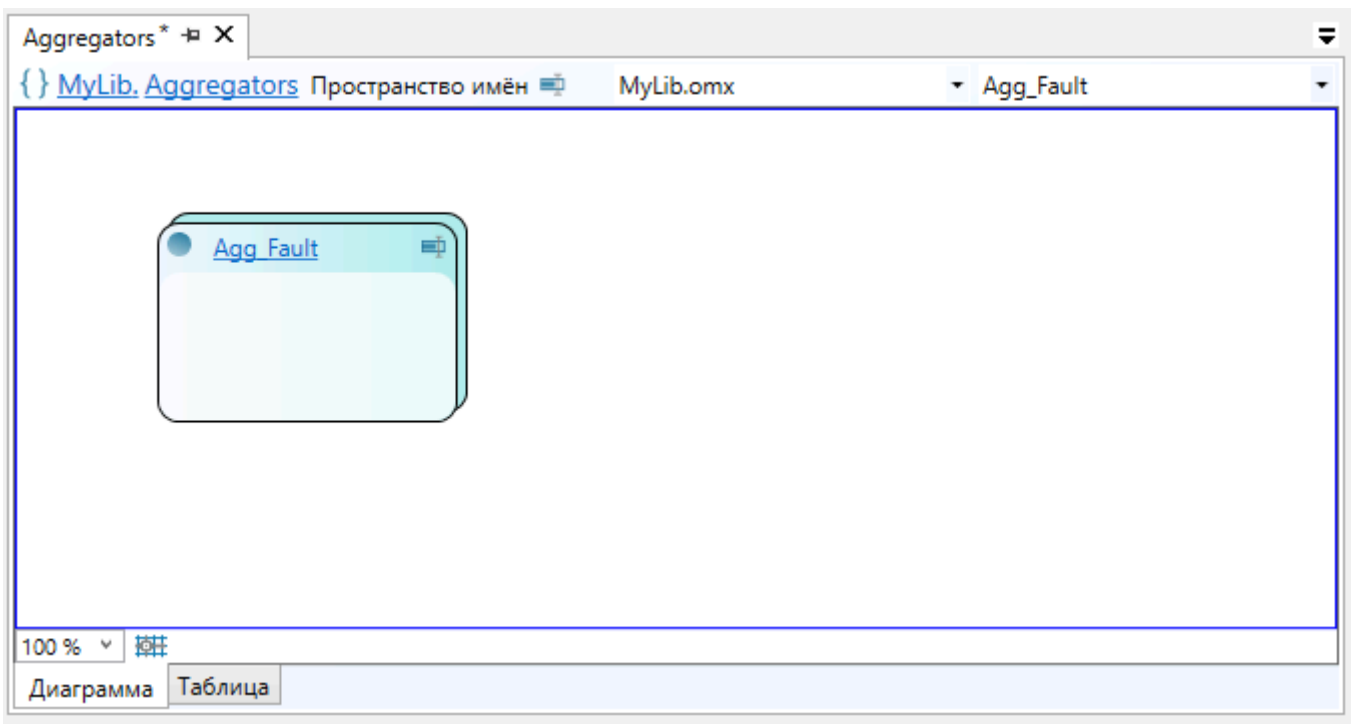
3. Задайте добавленному элементу имя "Aggregators" и перейдите в него, кликнув по названию.



4. Добавьте в пространство имен "Aggregators" элемент "Логический тип" из панели элементов.



5. Задайте добавленному логическому типу имя "Agg\_Fault".



Данный агрегатор будет отвечать за агрегацию отказов.

6. Выделите тип "Agg\_Fault" и в окне "События" установите флаг "Агрегировать события".



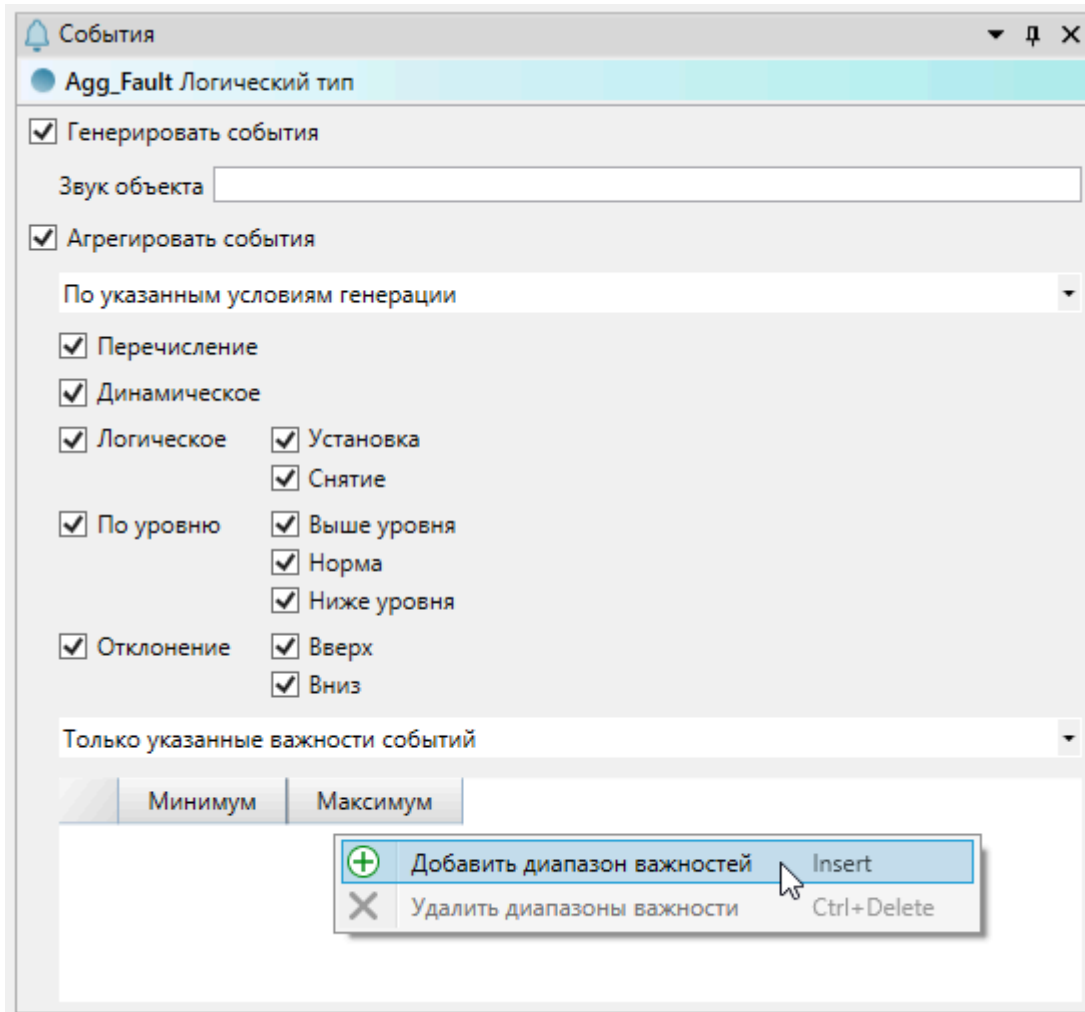
Если окно "События" не отображается, выполните команду в главном меню "Вид" → "События".

The screenshot shows the Aggregators software interface. The main workspace displays a diagram with a single component labeled 'Agg\_Fault'. The 'Properties' window on the right shows the 'Agg\_Fault' type selected. The 'Events' window is open, showing the 'Agg\_Fault' type selected. The 'Generate events' checkbox is checked, and the 'Aggregate events' checkbox is also checked. The 'Aggregate events' dropdown menu is set to 'By specified generation conditions'. The 'Logarithmic' checkbox is checked, and the 'Setup' and 'Removal' checkboxes are also checked. The 'Journal' window at the bottom shows 0 errors, 0 warnings, and 0 messages.

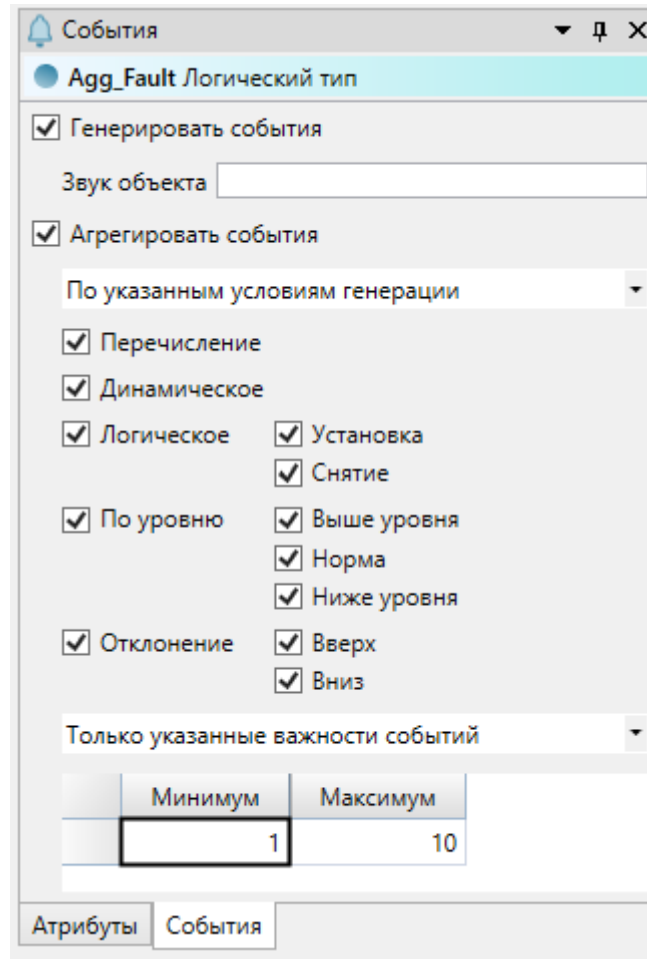
Свойства	
● Agg_Fault Логический тип	
Общие	
Базовый тип	
Аспект	
Представляемый тип	
Имя	Agg_Fault

События	
● Agg_Fault Логический тип	
<input checked="" type="checkbox"/> Генерировать события	
Звук объекта	
<input checked="" type="checkbox"/> Агрегировать события	
По указанным условиям генерации	
<input checked="" type="checkbox"/> Перечисление	
<input checked="" type="checkbox"/> Динамическое	
<input checked="" type="checkbox"/> Логическое	
<input checked="" type="checkbox"/> Установка	
<input checked="" type="checkbox"/> Снятие	

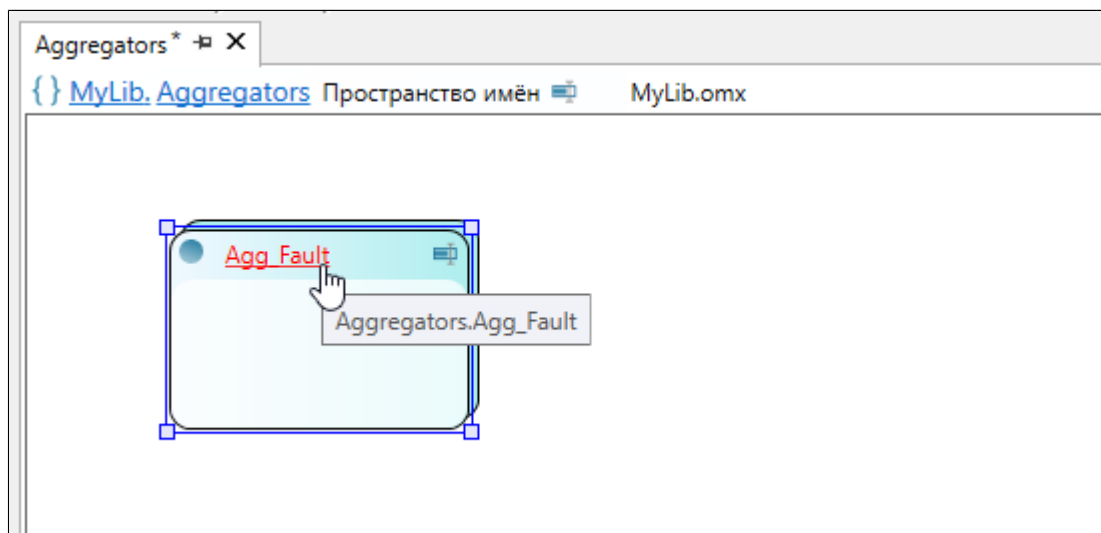
7. Чтобы добавить диапазон важностей для агрегации отказов, в контекстном меню таблицы выберите команду "Добавить диапазон важностей".



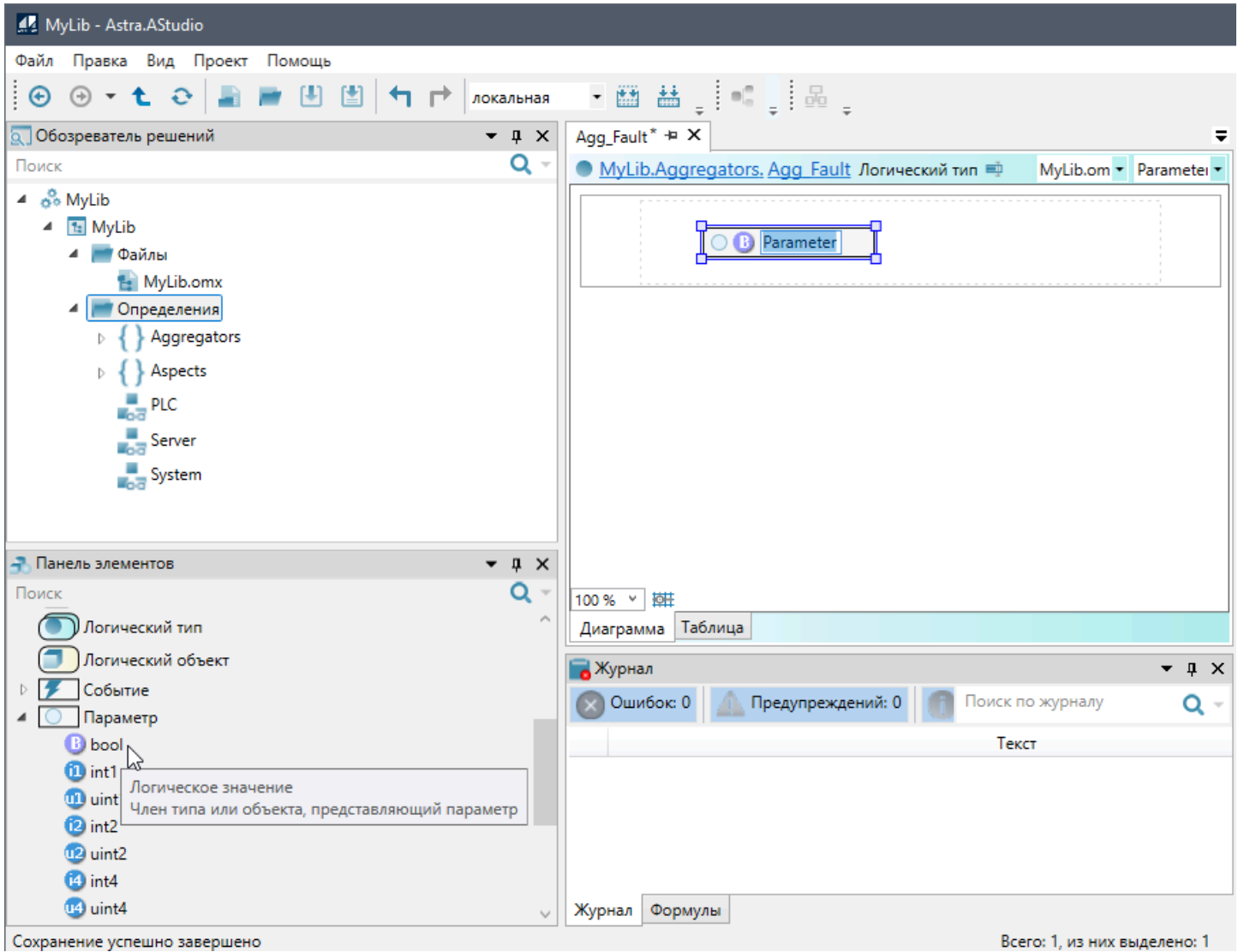
8. Укажите минимум и максимум диапазона важностей в соответствии с цветовой схемой для журнала событий.



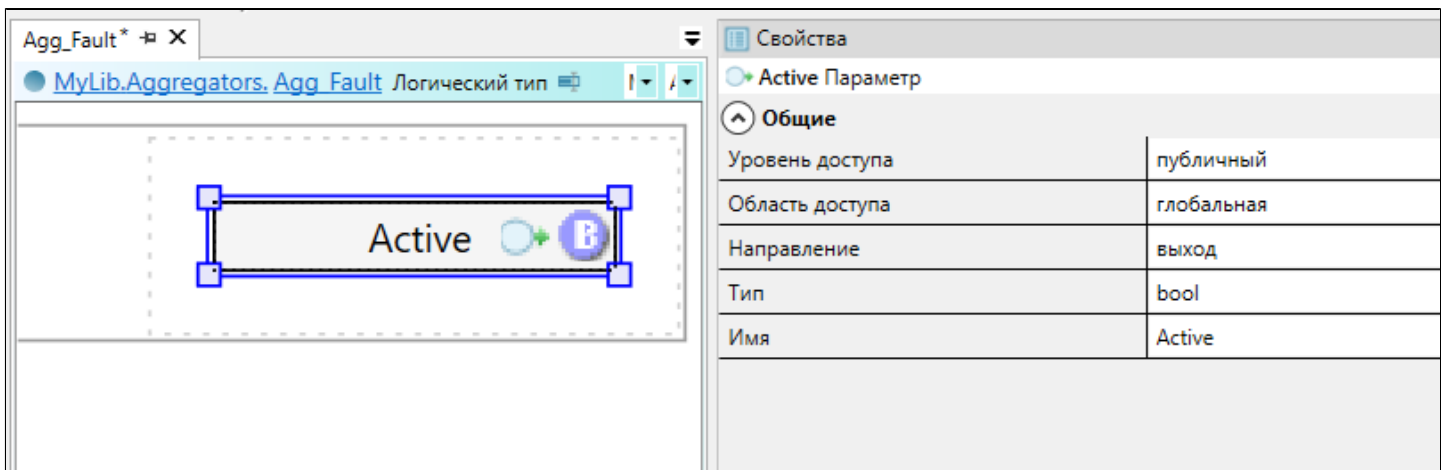
9. Перейдите в объект "Agg\_Fault", кликнув по его названию.



10. Добавьте параметр типа "bool" из раздела "Параметр" панели элементов.



11. Выделите добавленный параметр и в окне "Свойства" укажите имя "Active", а также направление "выход".







Не забывайте нажать клавишу "Enter" при указании свойств, иначе значение не применится.

12. В окне "События" установите флаг "Агрегировать события" и из выпадающего списка выберите тип информации, которую будет агрегировать данный параметр. Для параметра "Active" выберите тип информации "Наличие активных событий".

The screenshot displays the configuration interface for the 'Active' parameter. The 'Properties' window shows the following details:

Property	Value
Уровень доступа	публичный
Область доступа	глобальная
Направление	выход
Тип	bool
Имя	Active

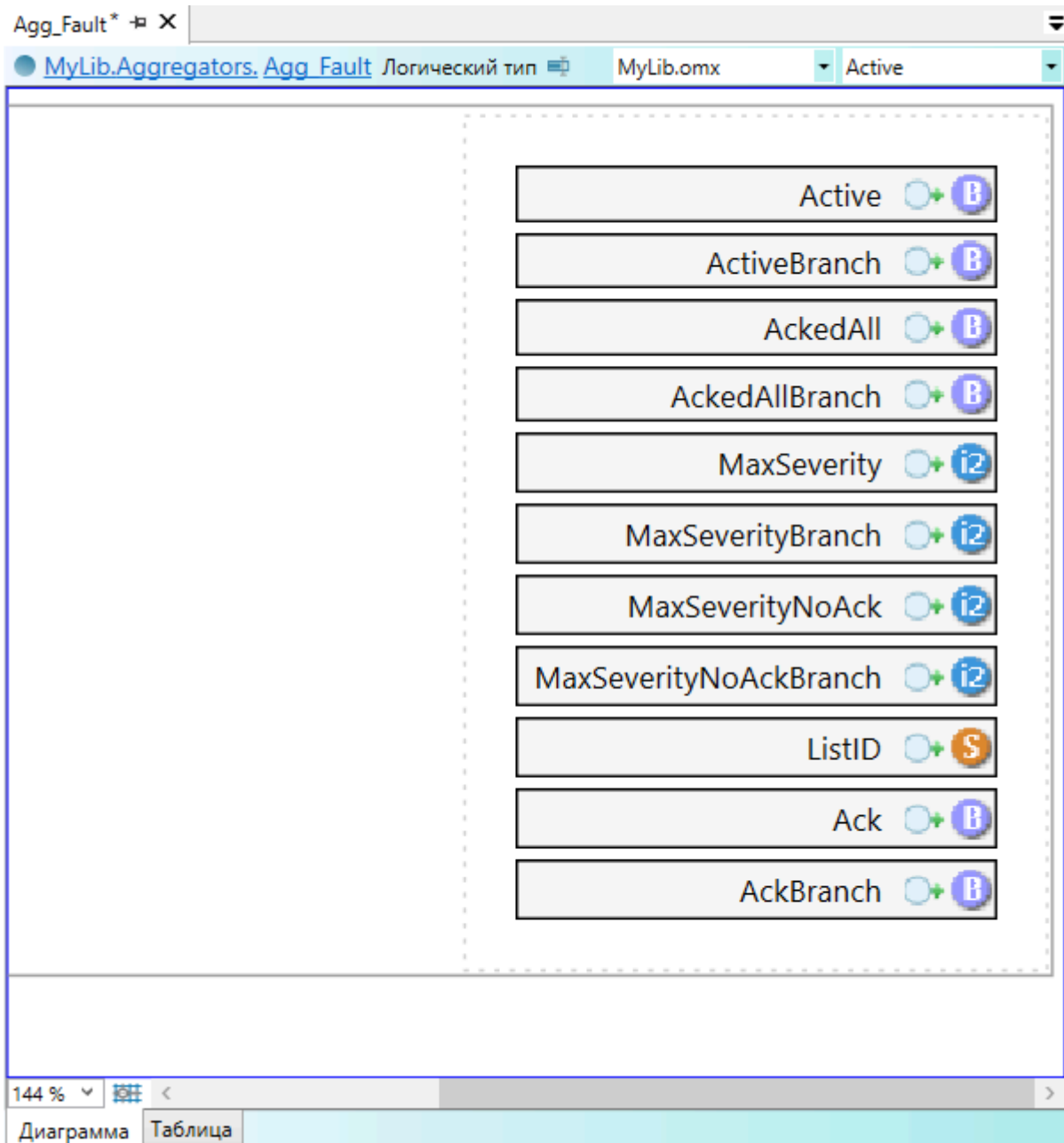
The 'Events' window shows the 'Aggregate events' checkbox checked and the dropdown menu set to 'Наличие активных событий'.

13. Аналогичным образом добавьте другие параметры для агрегации информации в соответствии с описаниями, представленными в таблице:

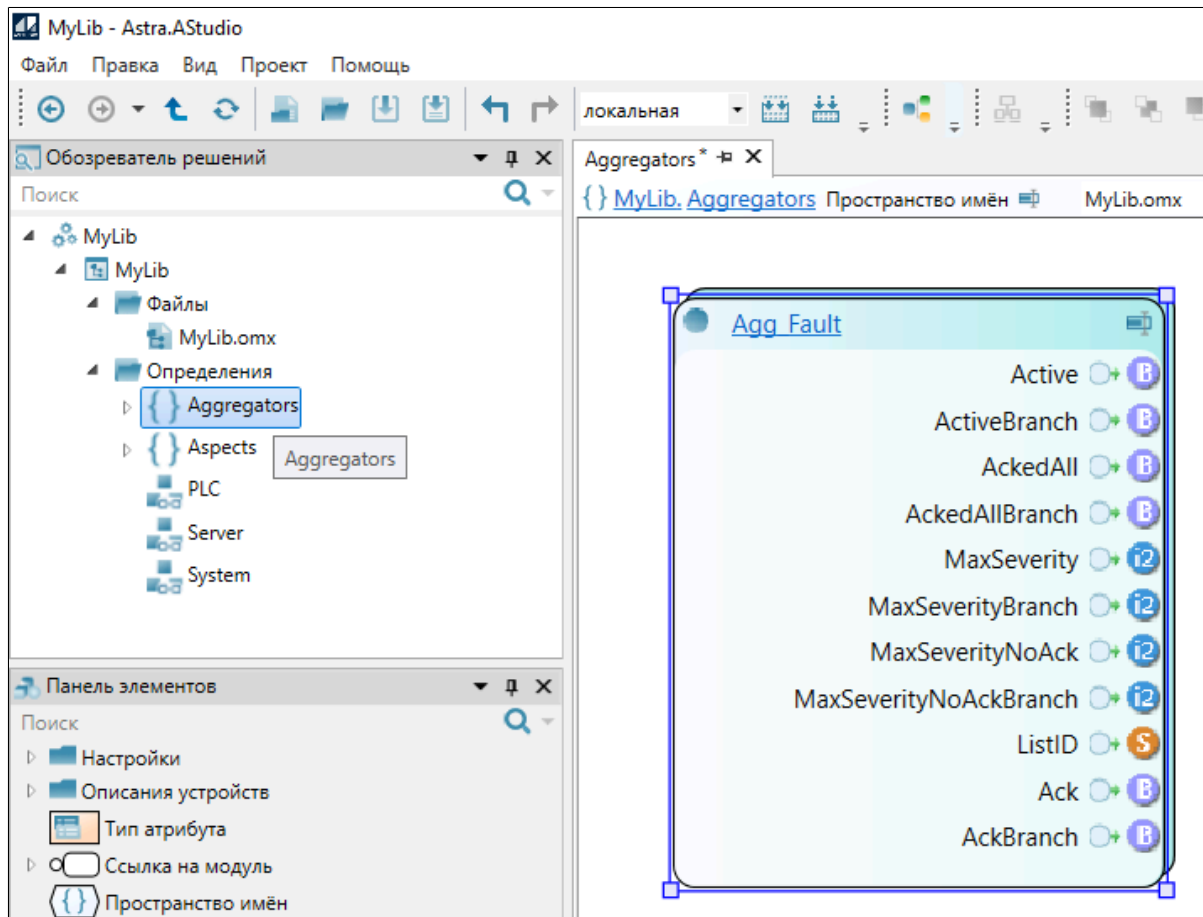
Название параметра	Агрегируемая информация	Тип значений
Active	Наличие активных событий	bool

ActiveBranch	Наличие активных событий в ветке	bool
AckedAll	Все события квитированы	bool
AckedAllBranch	Все события в ветке квитированы	bool
MaxSeverity	Максимальная важность событий	int2, uint2
MaxSeverityBranch	Максимальная важность событий в ветке	int2, uint2
MaxSeverityNoAck	Максимальная важность неквитированных событий	int2, uint2
MaxSeverityNoAckBranch	Максимальная важность неквитированных событий в ветке	int2, uint2
ListID	Список идентификаторов активных объектов	string
Ack	Квитирование событий	bool
AckBranch	Квитирование событий в ветке	bool

В результате должен получиться следующий список параметров:



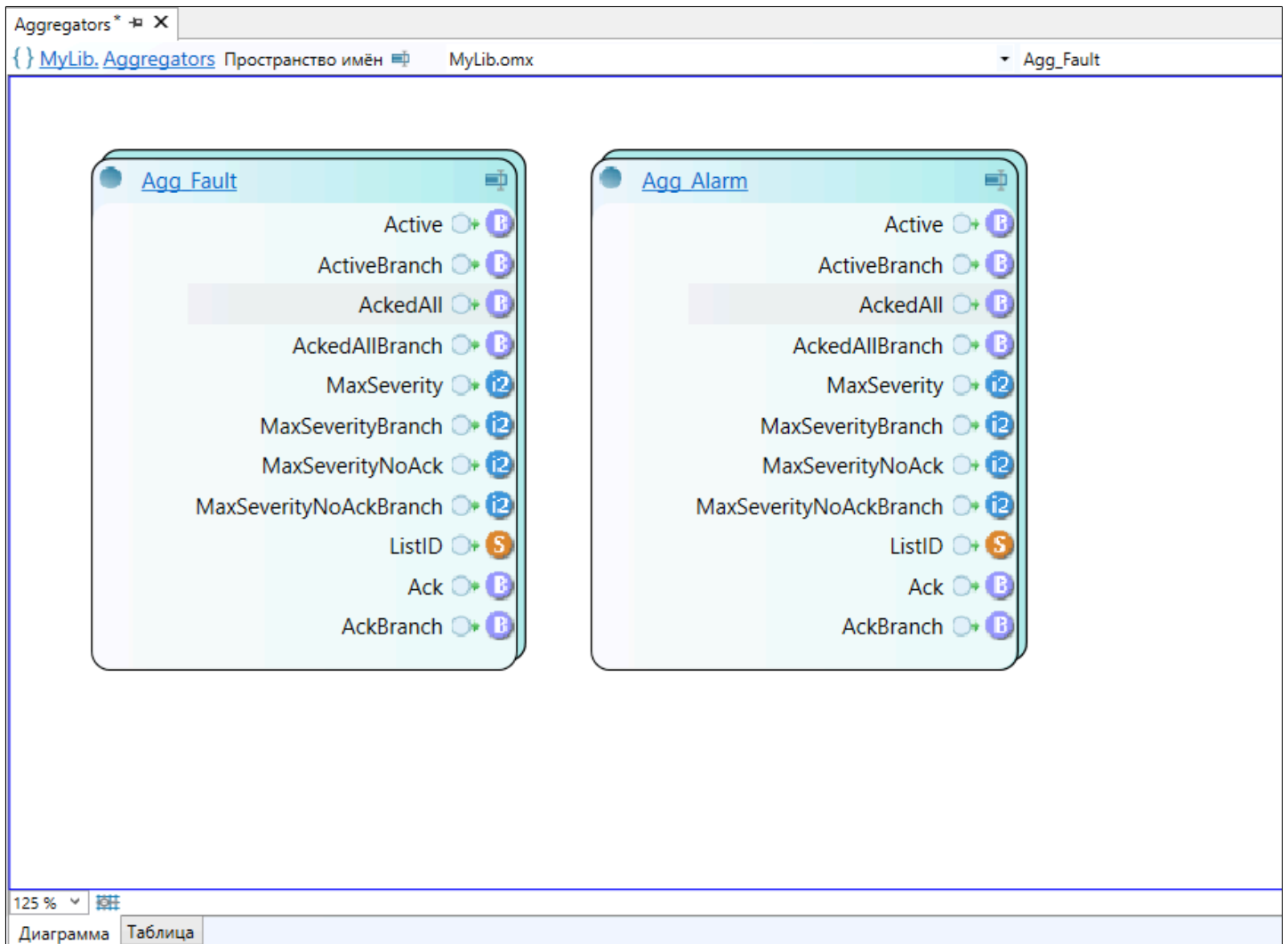
14. Перейдите в пространство имен "Aggregators" двойным кликом по объекту в обозревателе решений.



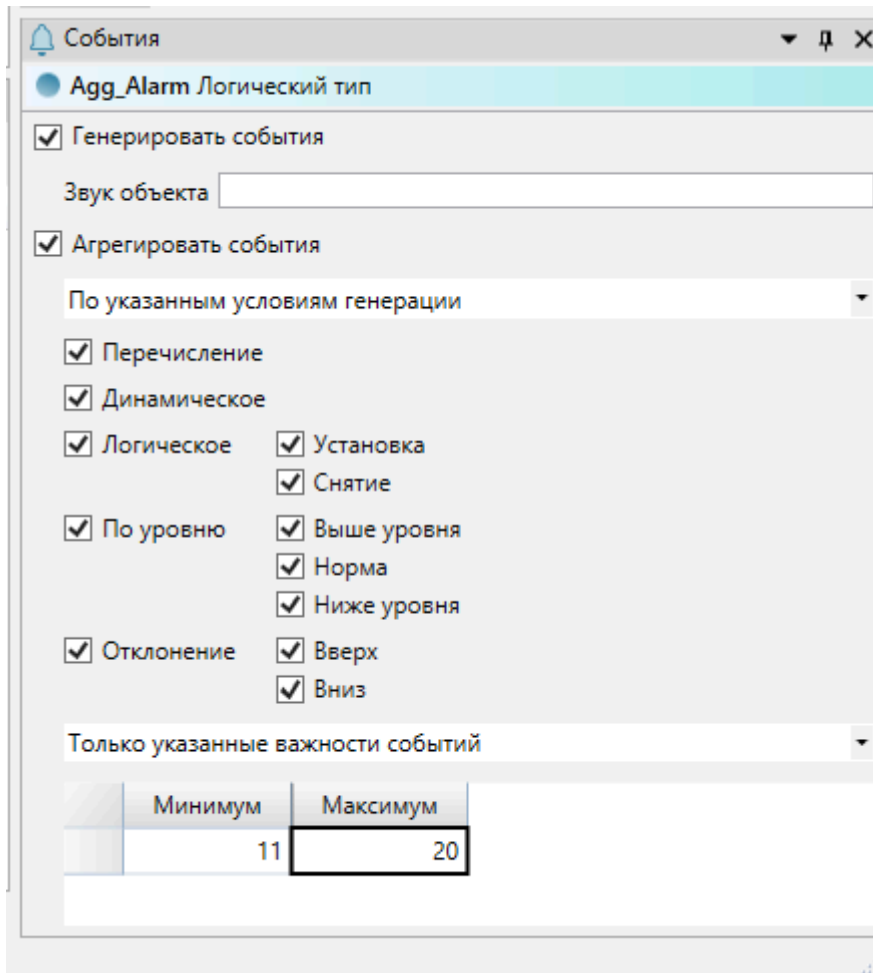
15. Создайте копию логического типа "Agg\_Fault" и переименуйте его в "Agg\_Alarm".



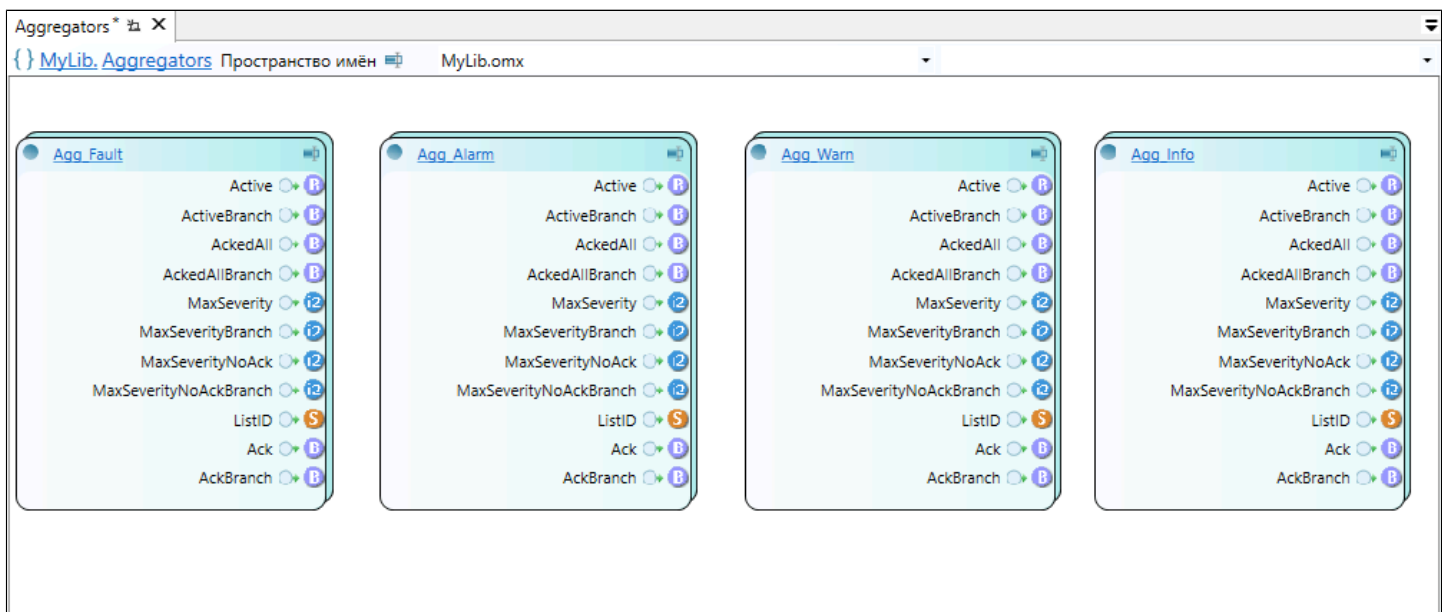
Для создания копии используйте горячие клавиши Ctrl+C, Ctrl+V.



16. Выделите агрегатор "Agg\_Alarm" и в окне "События" укажите диапазон важностей в соответствии с цветовой схемой для журнала событий.



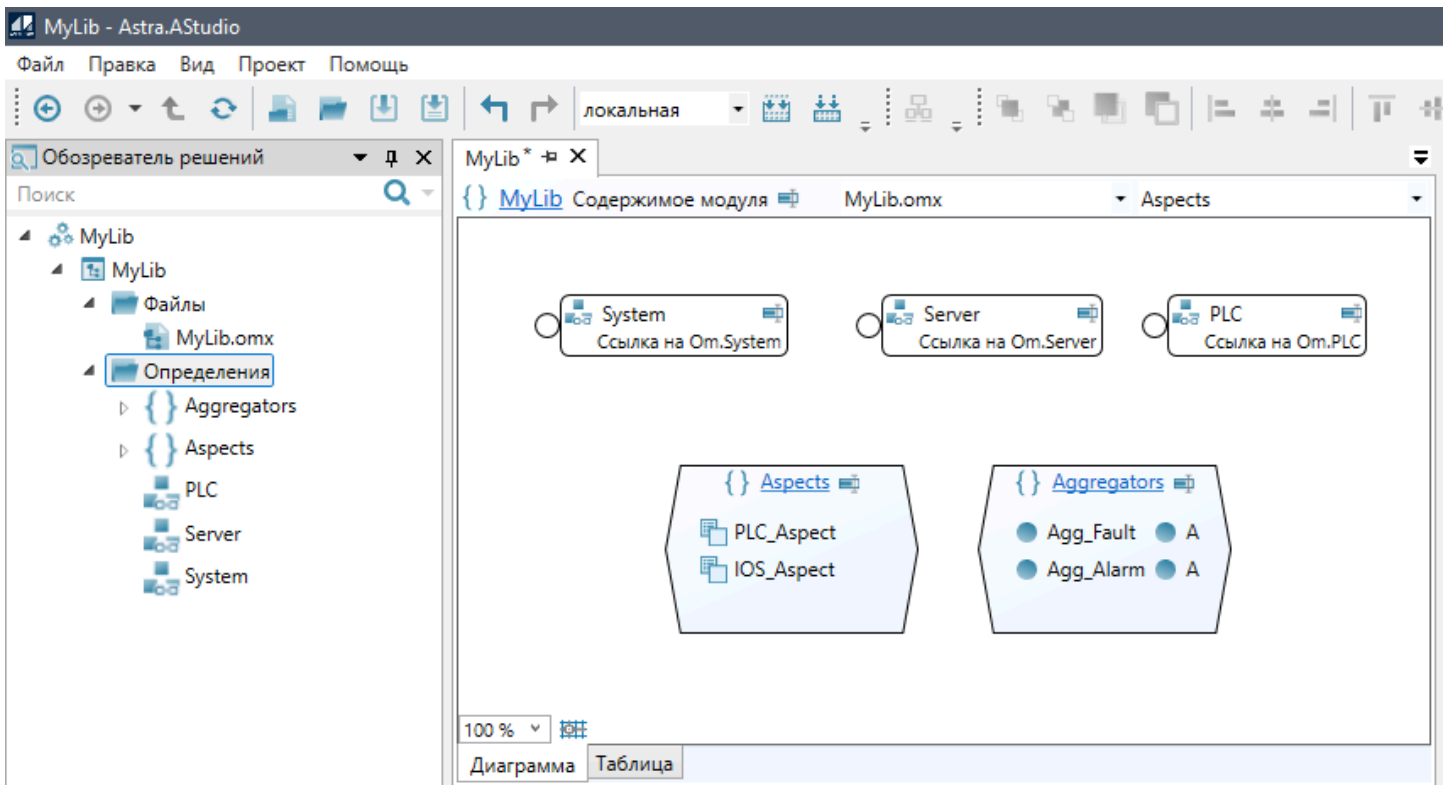
17. Аналогичным образом добавьте агрегаторы событий для предупреждающих и информационных событий.



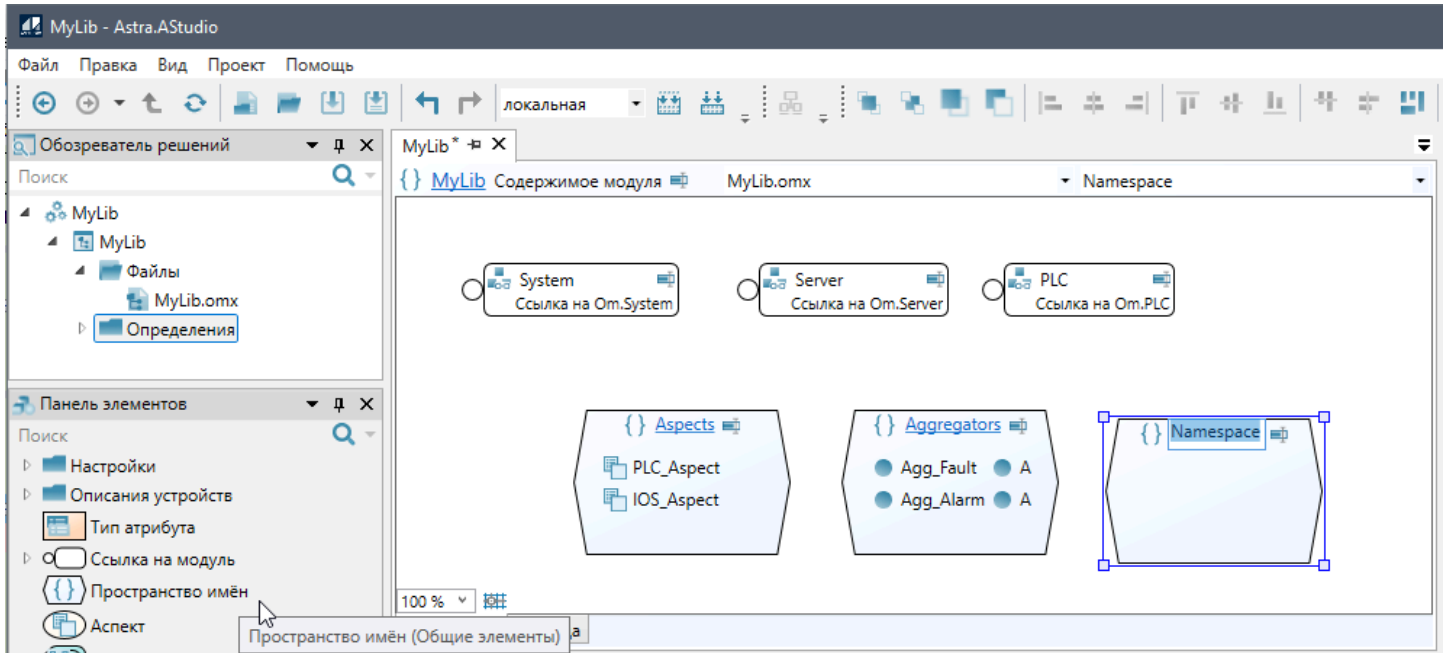
## 1.2.3.4. Создание типа

Чтобы создать тип, необходимо выполнить следующие действия:

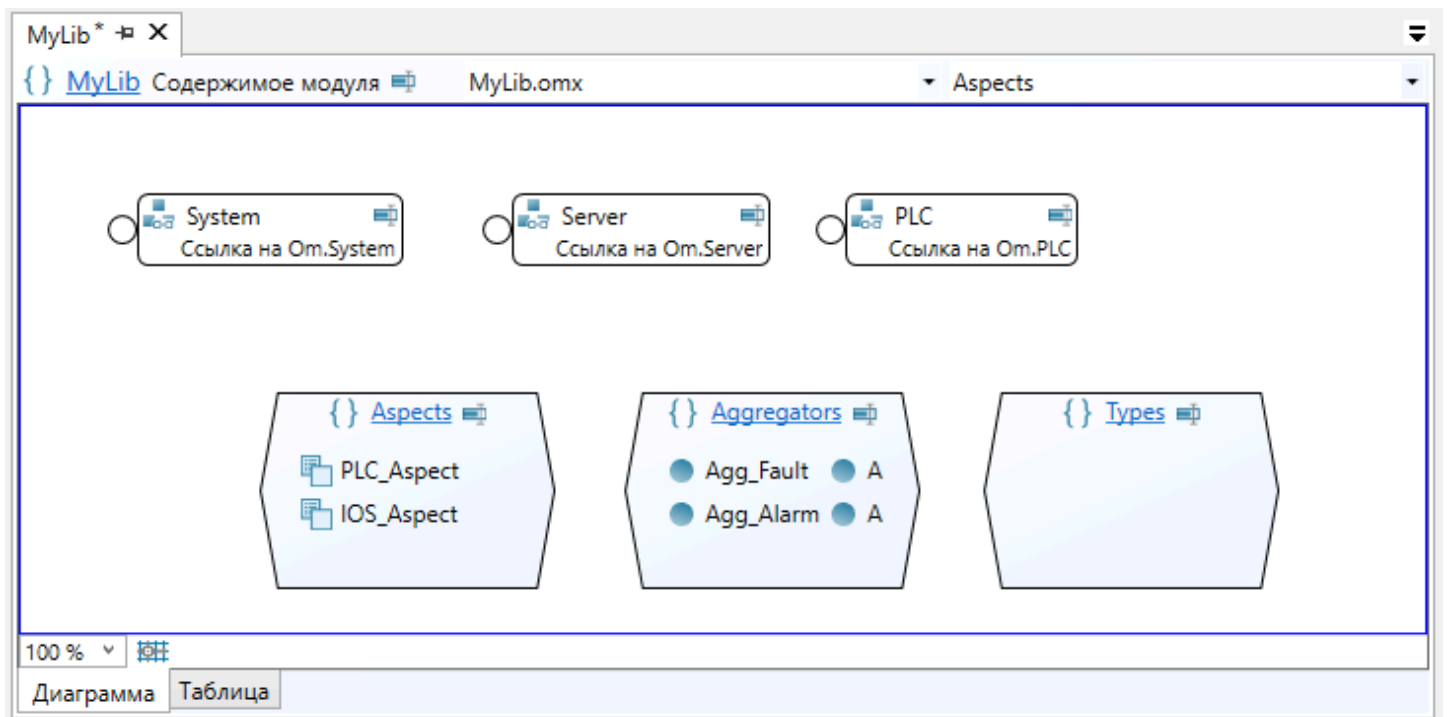
1. Перейдите в определения проекта "MyLib" двойным кликом по объекту в обозревателе решений.



2. Добавьте на рабочую область элемент "Пространство имен" путем перетаскивания из панели элементов.



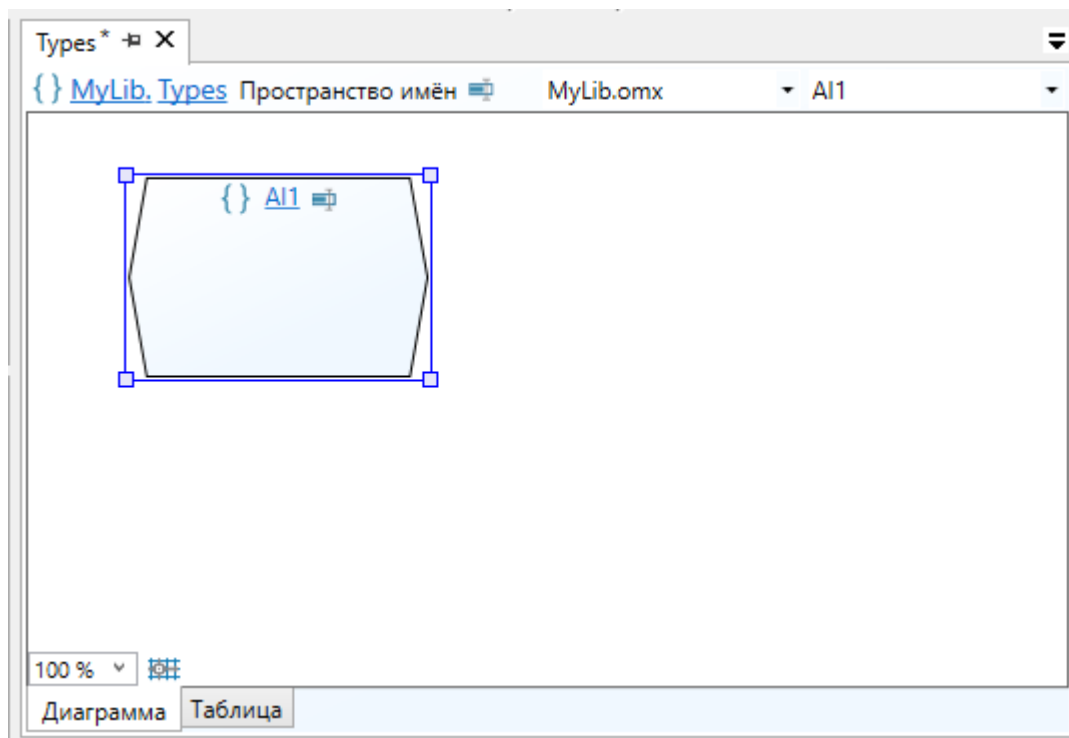
3. Задайте добавленному элементу имя "Types" и перейдите в объект, кликнув по его названию.



В пространстве имен "Types" будут храниться все типы библиотеки.

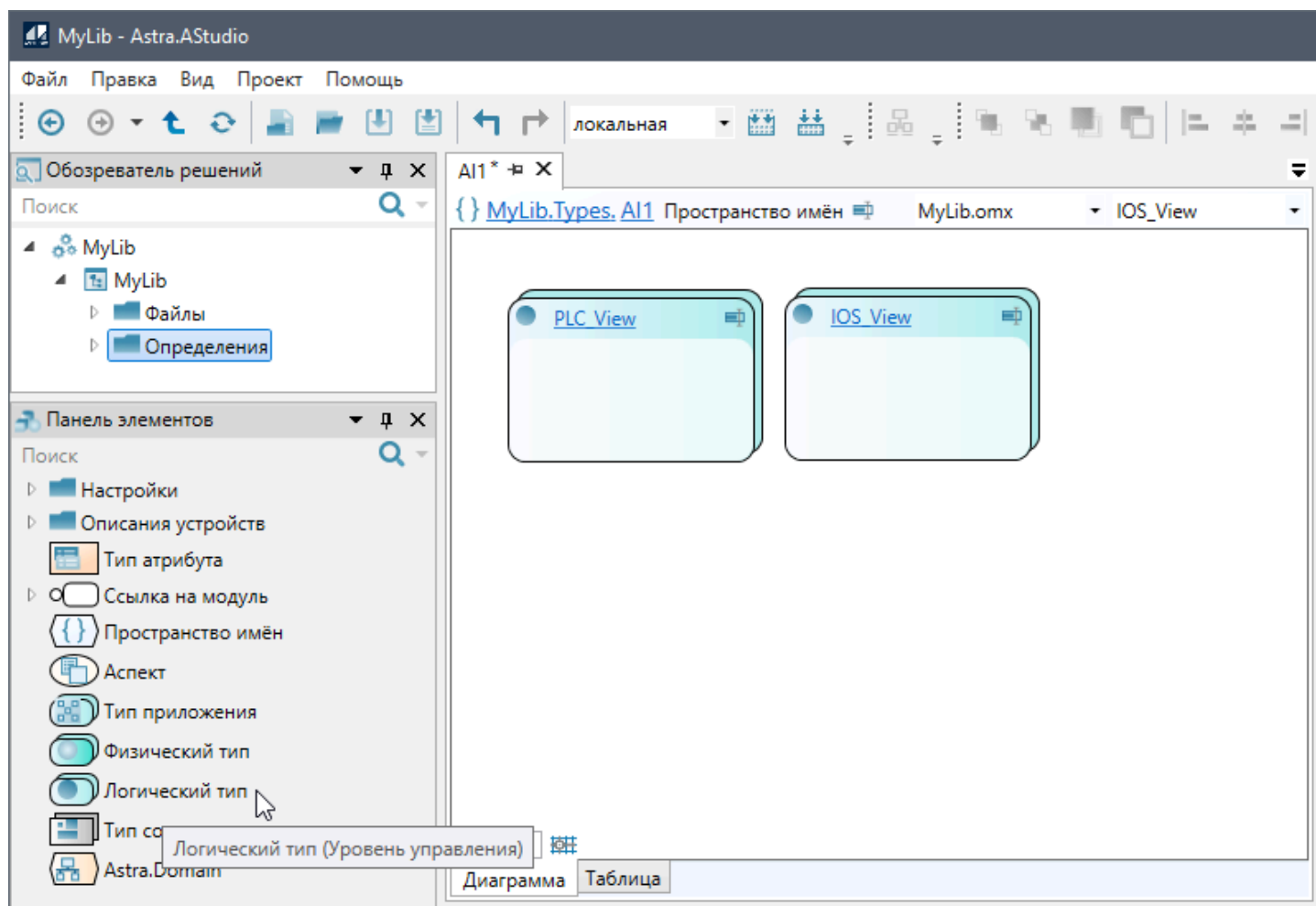
4. Рассмотрим создание типа на примере аналогового датчика. Добавьте из панели элементов пространство имён и в его свойствах укажите имя "AI1".



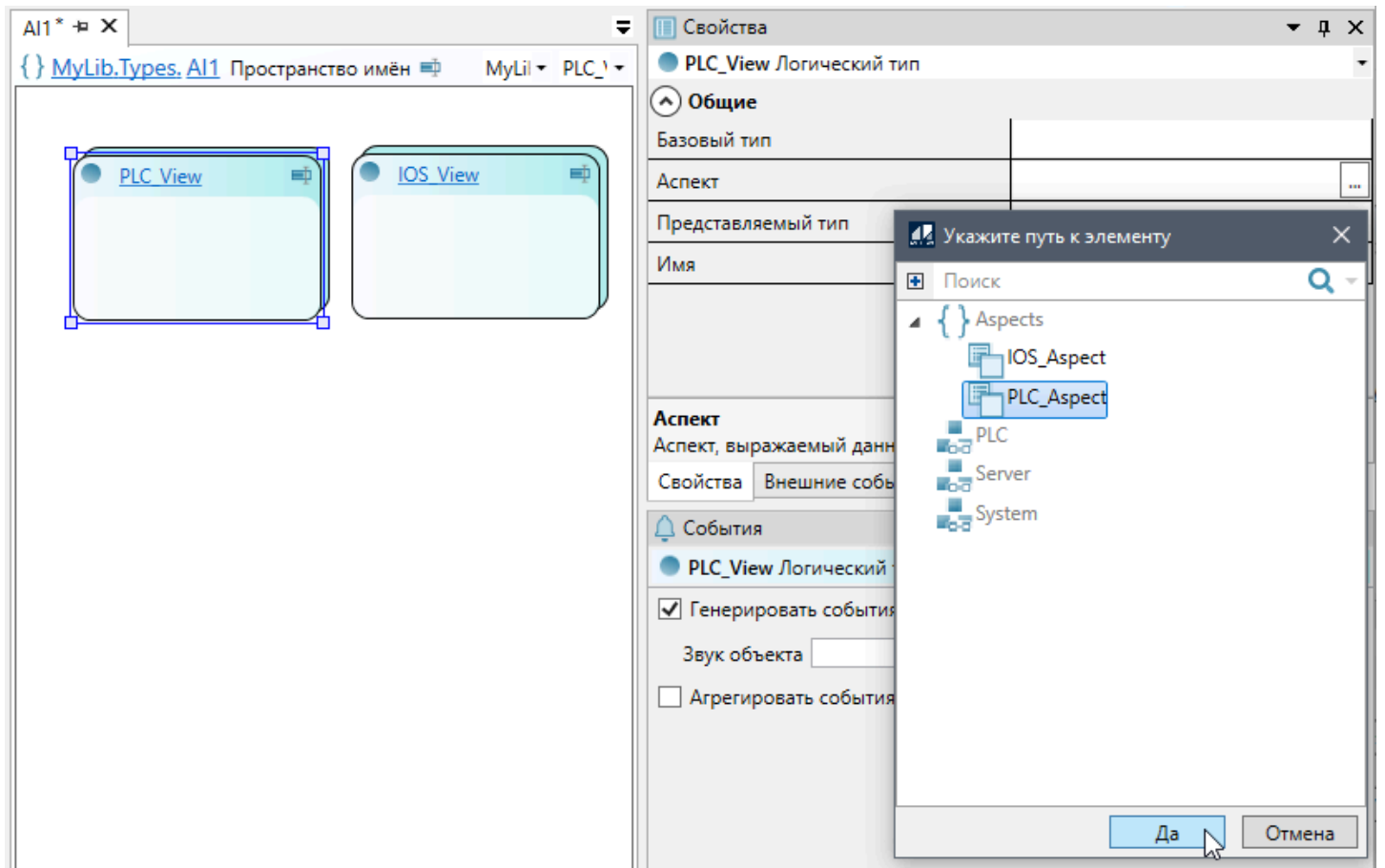


5. Перейдите в пространство имен "AI1", кликнув по названию объекта, и добавьте два логических типа из панели элементов:

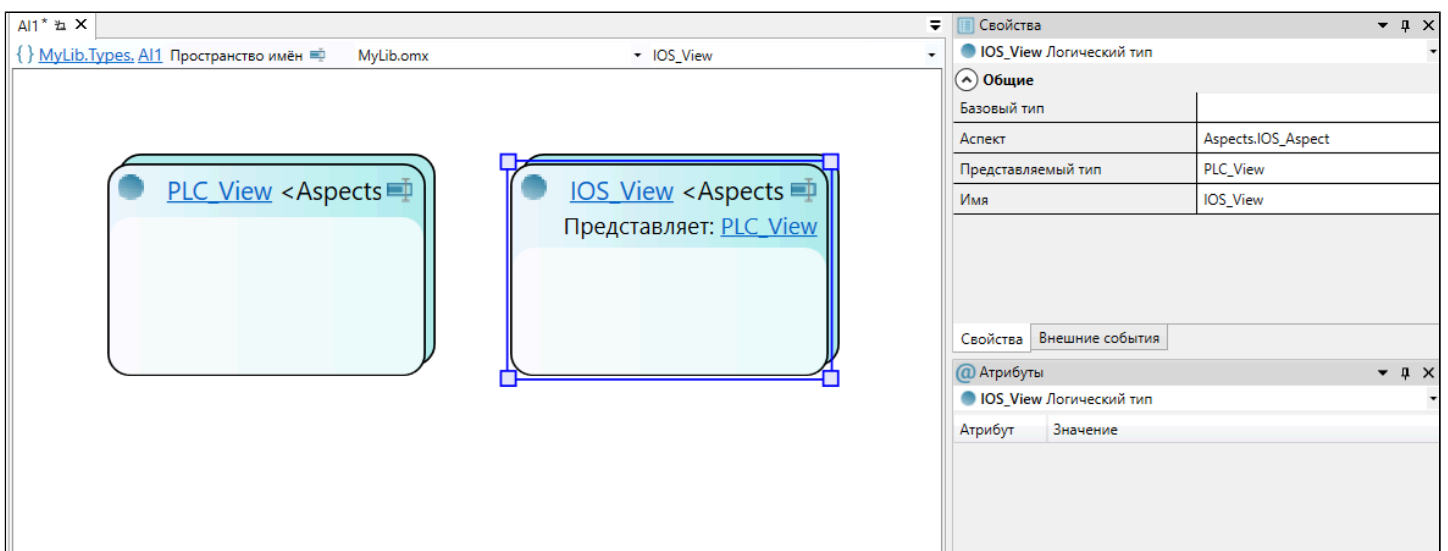
- PLC\_View (представление типа для уровня ПЛК);
- IOS\_View (представление типа для уровня сервера ввода/вывода).



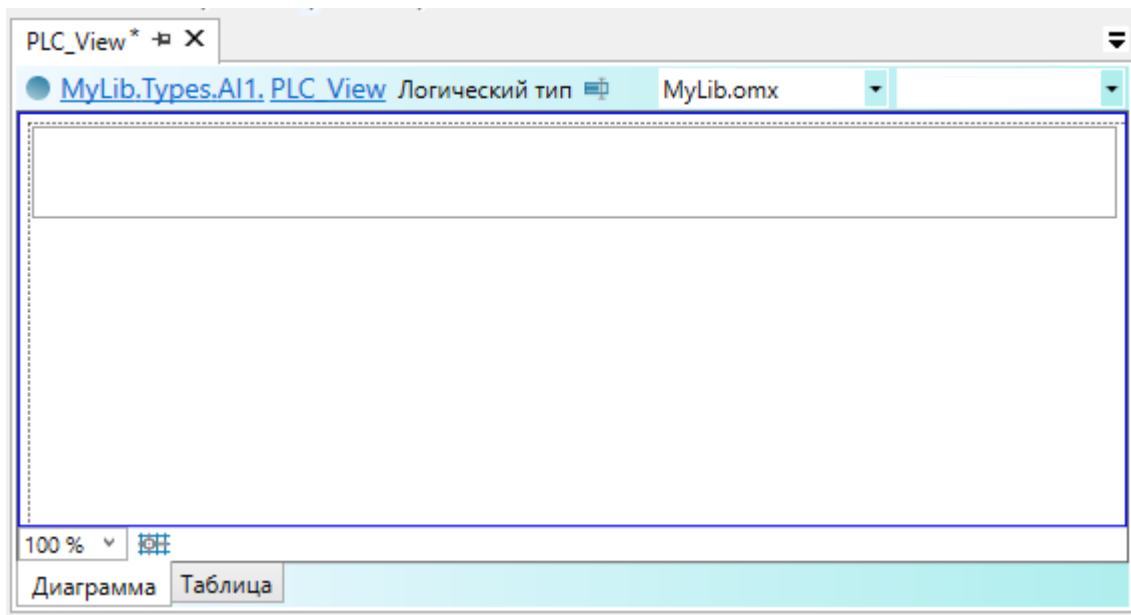
6. Выделите представление "PLC\_View" и в окне свойства в поле "Аспект" укажите аспект "PLC\_Aспект" из пространства имен "Aspects".



7. Аналогичным образом выделите представление "IOS\_View" и в окне свойства в поле "Аспект" укажите аспект "IOS\_View" из пространства имен "Aspects". В поле "Представляемый тип" выберите тип "PLC\_View".



8. Перейдите в представление "PLC\_View" типа "AI1", кликнув по названию объекта.

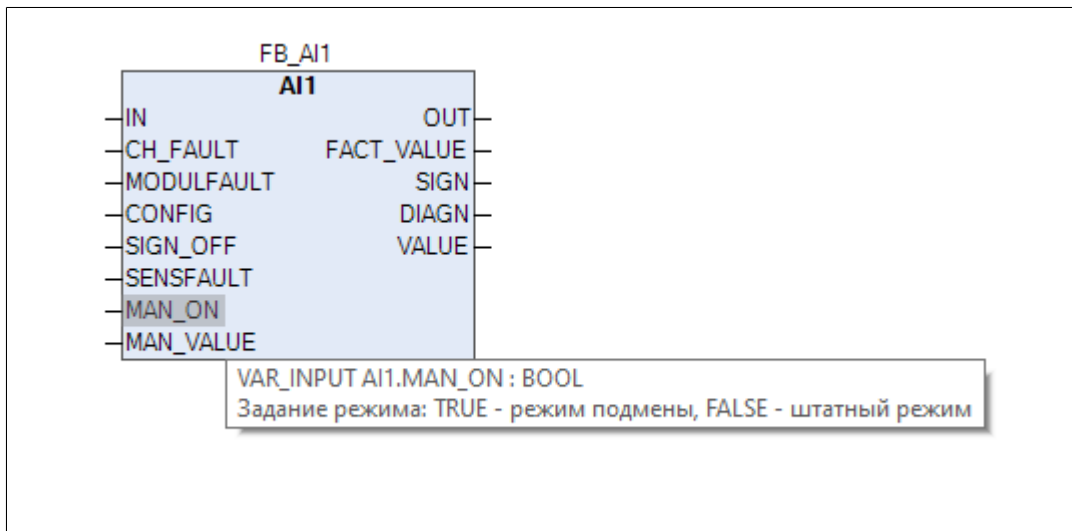


В представление "PLC\_View" необходимо вынести те параметры, которые будут осуществлять обмен данными с ПЛК.

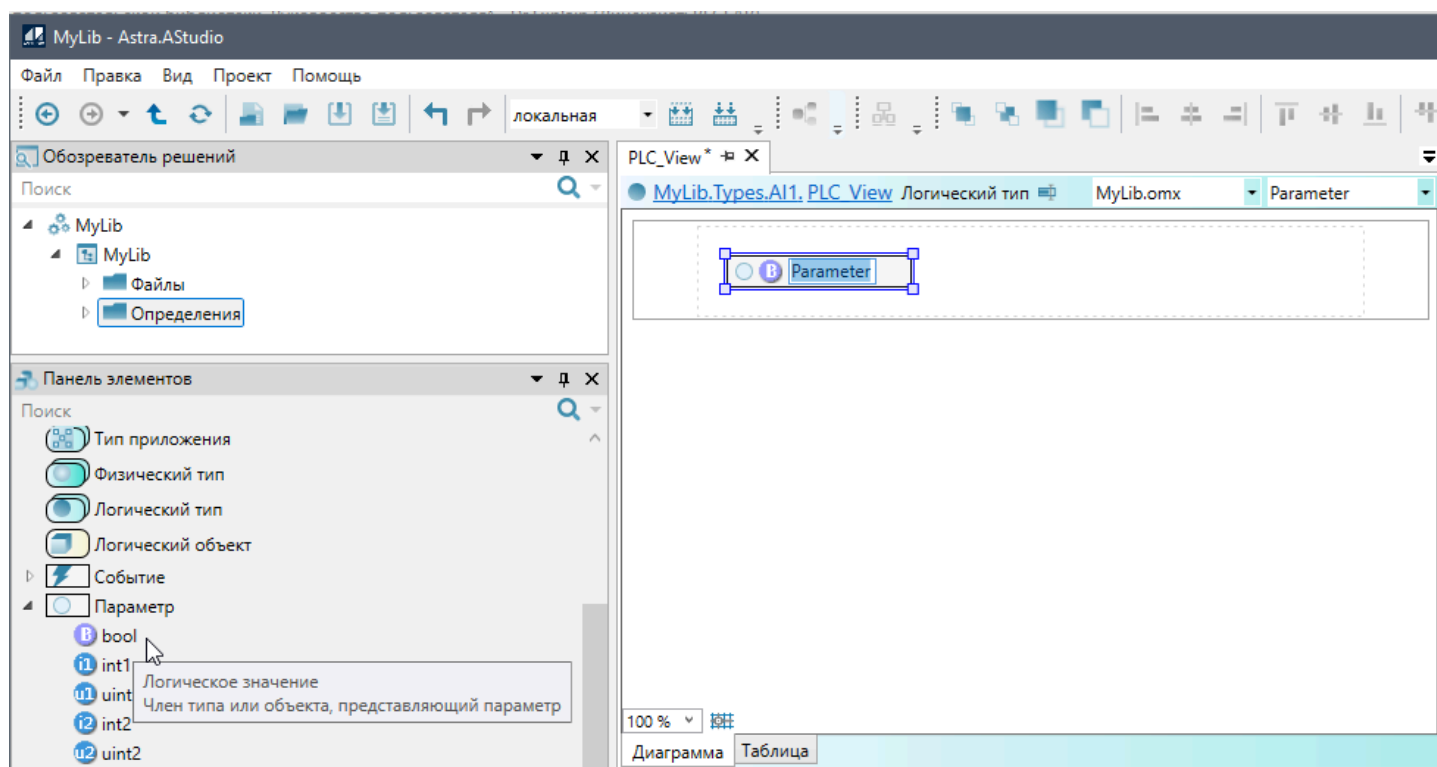
Для этого необходимо посмотреть на реализацию функционального блока аналогового датчика в среде разработки Astra.IDE.

У функционального блока есть входные и выходные переменные для взаимодействия с ВУ. Данные параметры отмечены атрибутом **{attribute 'symbol' := 'readwrite'}**.

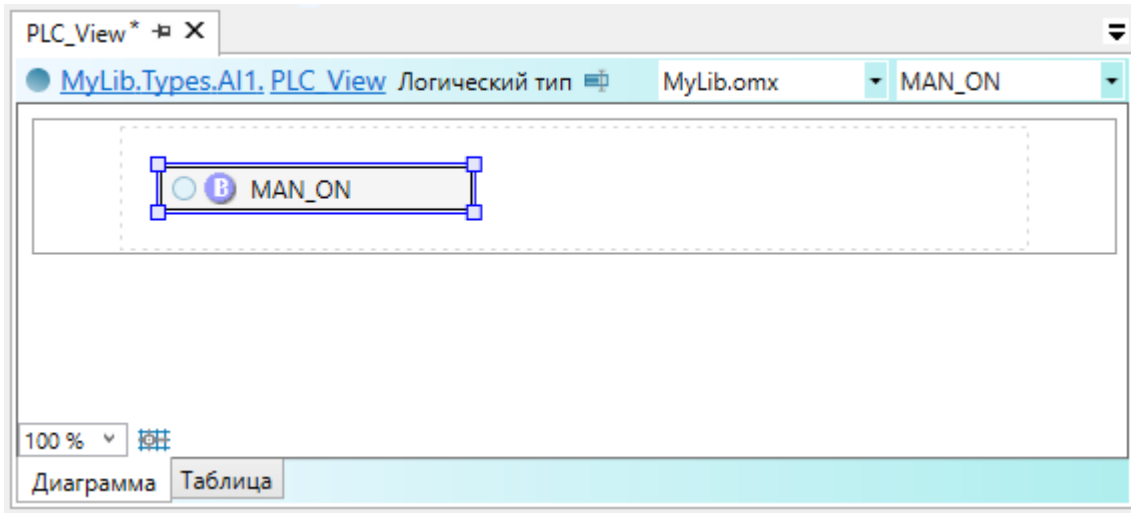
Рассмотрим добавление параметров на примере входной переменной "MAN\_ON" ФБ "AI1". Данный параметр отвечает за команду установки режима аналогового датчика.



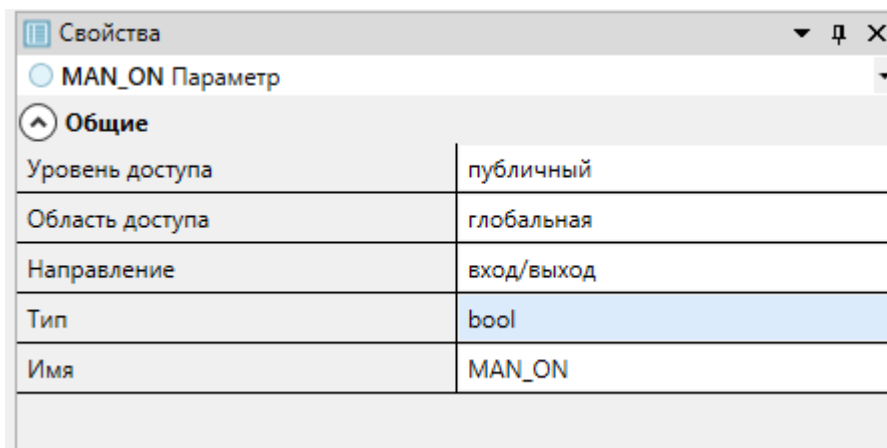
9. В представление "PLC\_View" добавьте параметр типа bool из раздела "Параметр" панели элементов.



10. Выделите добавленный параметр и в окне "Свойства" задайте ему имя "MAN\_ON".



11. Выделите параметр "MAN\_ON" и в окне "Свойства" задайте направление "вход/выход".



Направления параметрам задаются в зависимости от типа взаимодействия с ПЛК:

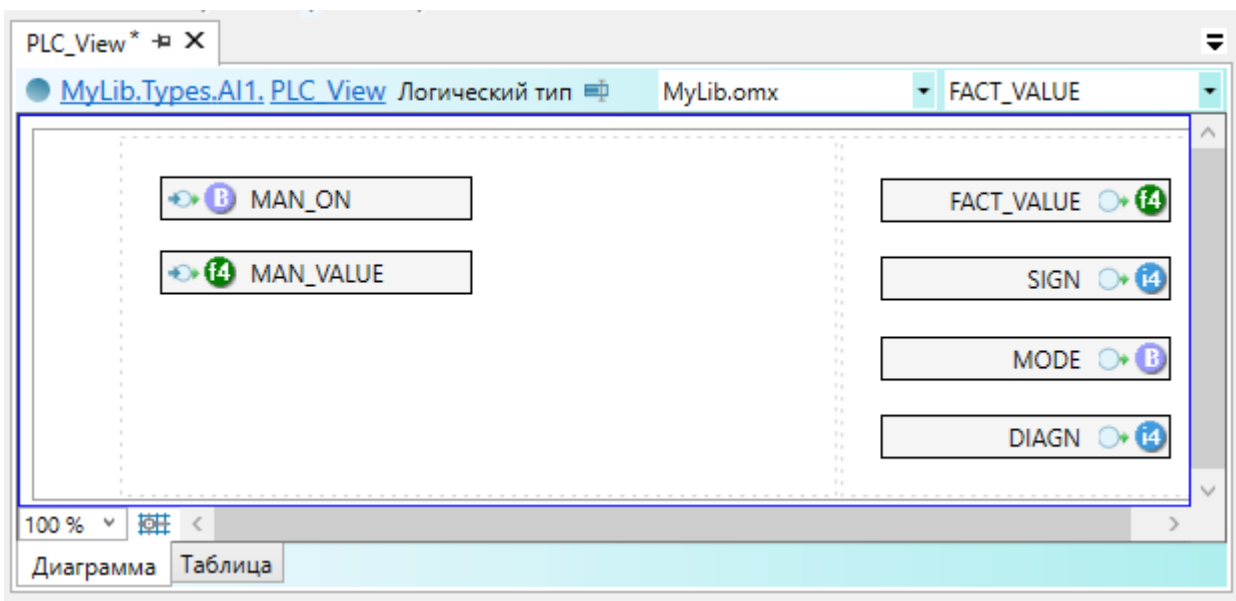
- вход - запись в ПЛК (используется для только входных сигналов);
- выход - чтение из ПЛК (используется для сигналов состояния);
- вход/выход - чтение и запись из ПЛК (используется для команд и уставок).

12. Аналогичным образом добавьте другие параметры ФБ "A11", указанных в таблице:

Параметр	Тип	Направление
----------	-----	-------------

MAN_ON	bool	ВХОД/ВЫХОД
MAN_VALUE	float4	ВХОД/ВЫХОД
FACT_VALUE	float4	ВЫХОД
SIGN	int4	ВЫХОД
MODE	bool	ВЫХОД
DIAGN	int4	ВЫХОД

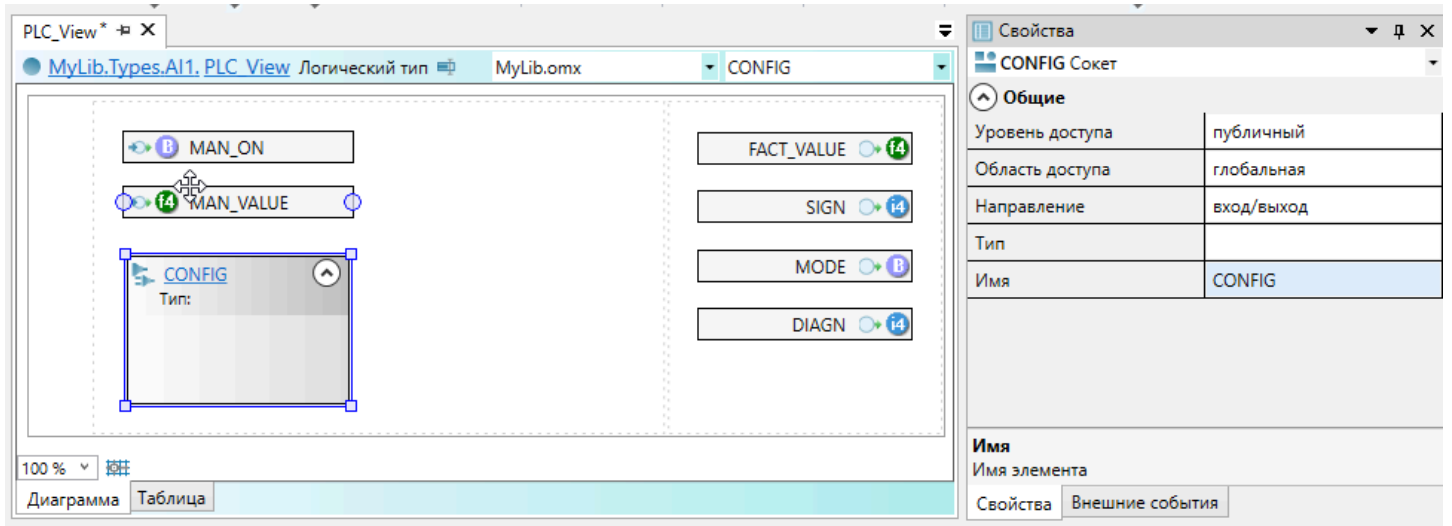
В результате должен получиться следующий вид:



Если в функциональном блоке используются структурные переменные, используйте элемент "Сокет" из панели элементов. Сокет является аналогом структур в Astra.IDE.

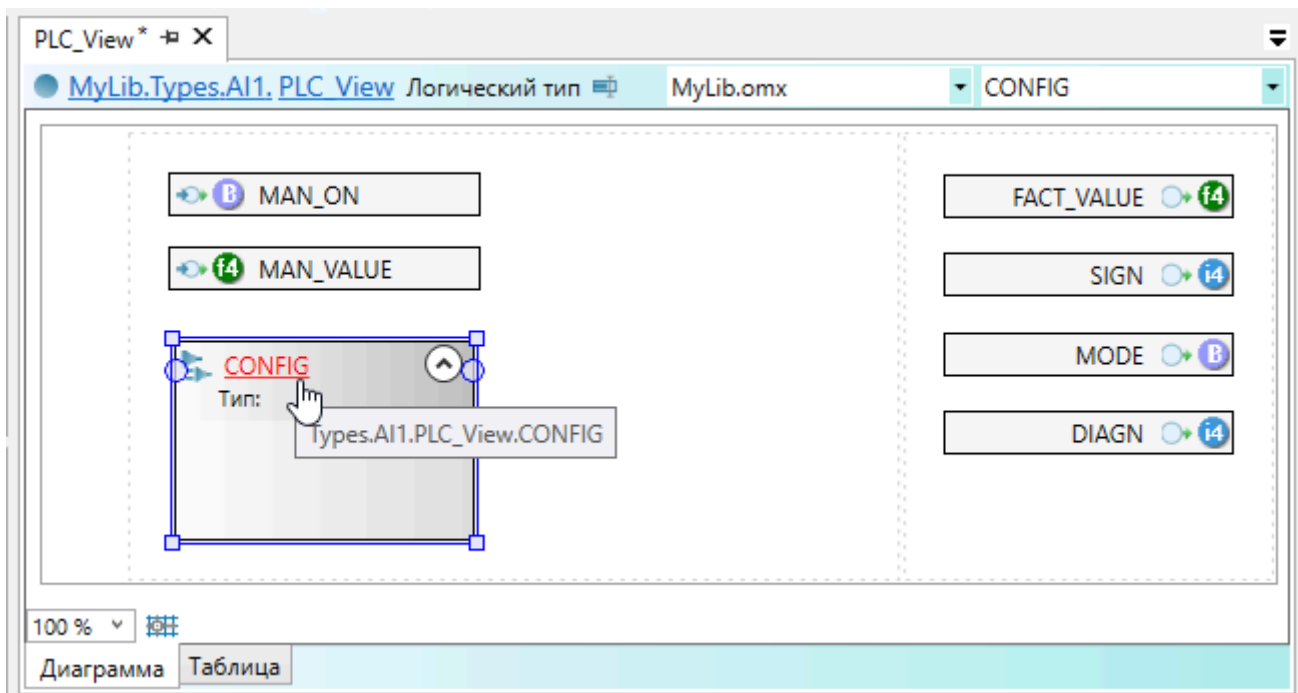
В данном примере будем использовать сокет для выходной структурной переменной "OUT" (значения датчика с качеством) и входной структурной переменной "CONFIG" (настроечные параметры) функционального блока "AI1".

13. Добавьте на рабочую область элемент "Сокет" из панели элементов и задайте ему имя "CONFIG", а также укажите направление "вход/выход".



При указании направления сокета параметры внутри сокета примут то же направление.

14. Перейдите в сокет "CONFIG", кликнув по названию объекта.

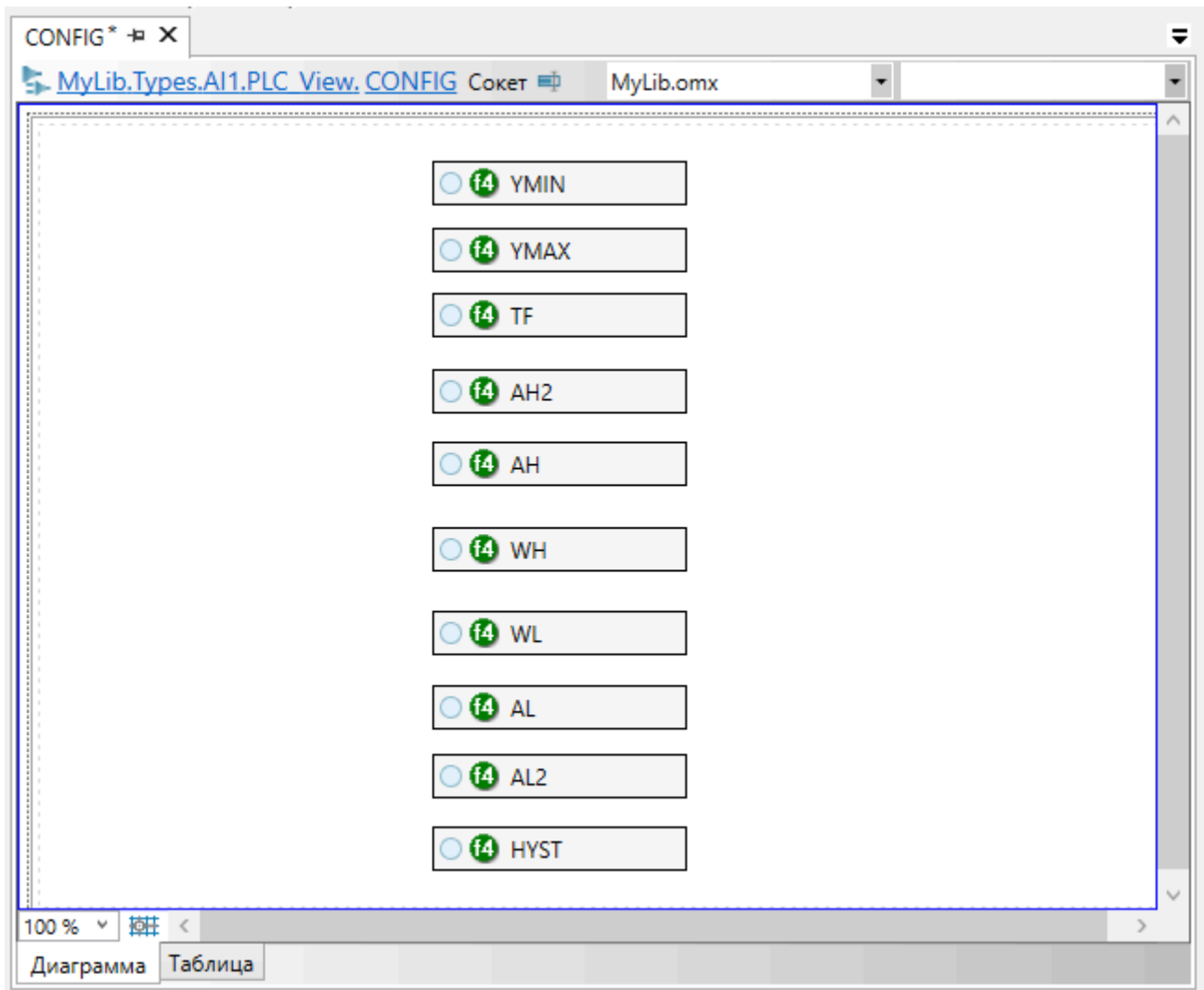


15. Добавьте настроечные параметры функционального блока, воспользовавшись вкладкой "Параметр" панели элементов.



Типы и названия параметров должны соответствовать параметрам в структуре "AI\_CONFIG" в Astra.IDE.





16. Аналогичным образом добавьте сокет "OUT" с направлением "выход".

В результате должно получиться следующее представление аналогового датчика на уровне ПЛК:

The screenshot shows the 'PLC View' application window. The main workspace displays a library of parameters under the 'OUT' category. On the left, there is a 'CONFIG' section with a list of parameters: AN2, AN, WH, WL, AL, AL2, HYST, YMIN, YMAX, and TF. Each parameter has a type icon (e.g., 'f4' for float, 'B' for boolean). On the right, a 'Socket' (OUT) is defined with parameters: FACT\_VALUE (f4), SIGN (i4), MODE (B), and DIAGN (i4). A detailed view of the 'OUT' socket is shown below, with parameters VALUE (f4) and QUALITY (i4). The 'Properties' (Свойства) panel on the right shows the following table:

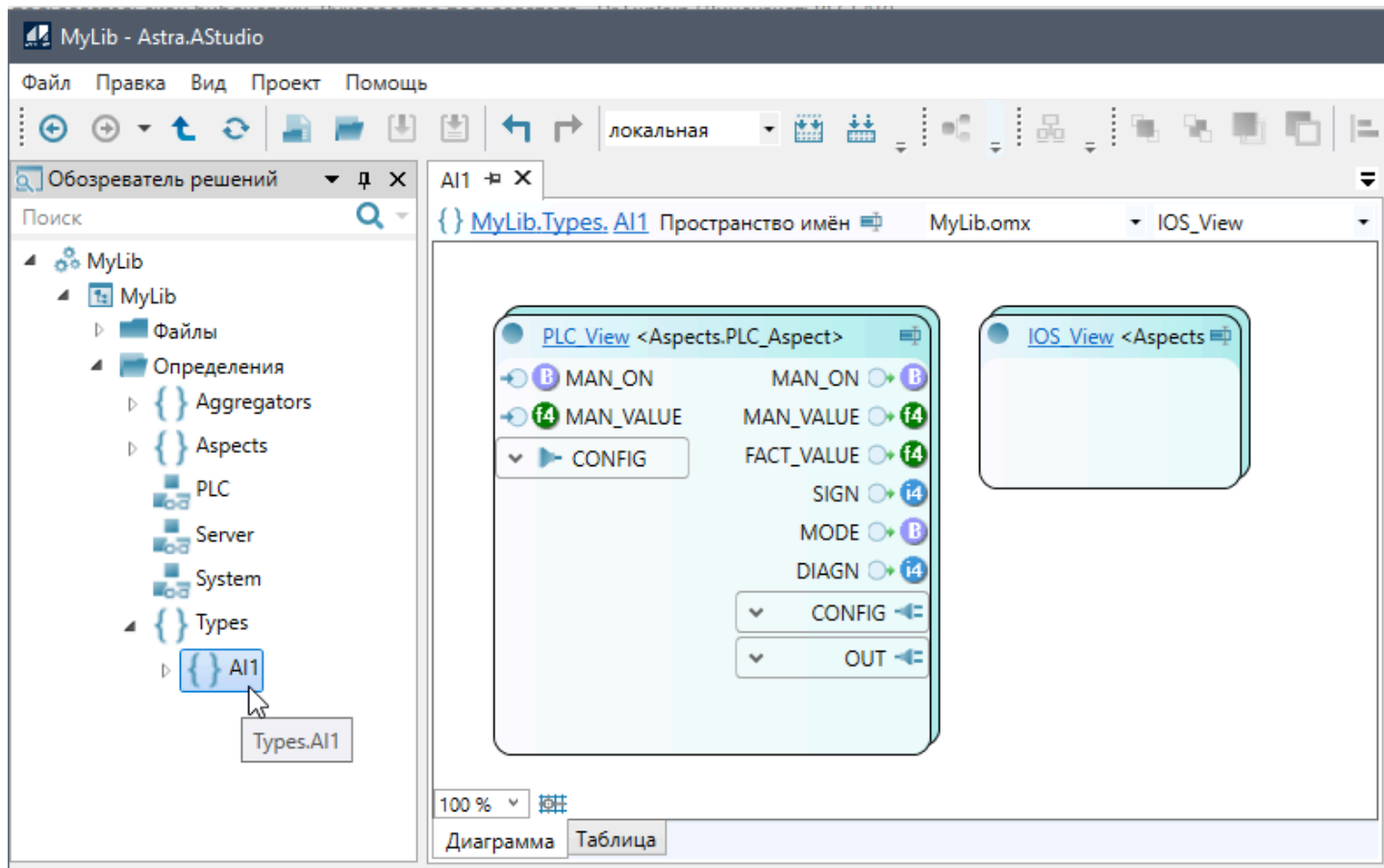
Общие	
Уровень доступа	публичный
Область доступа	глобальная
Направление	выход
Тип	
Имя	OUT

At the bottom of the main workspace, there are tabs for 'Диаграмма' (Diagram) and 'Таблица' (Table), and a zoom level of 100%.

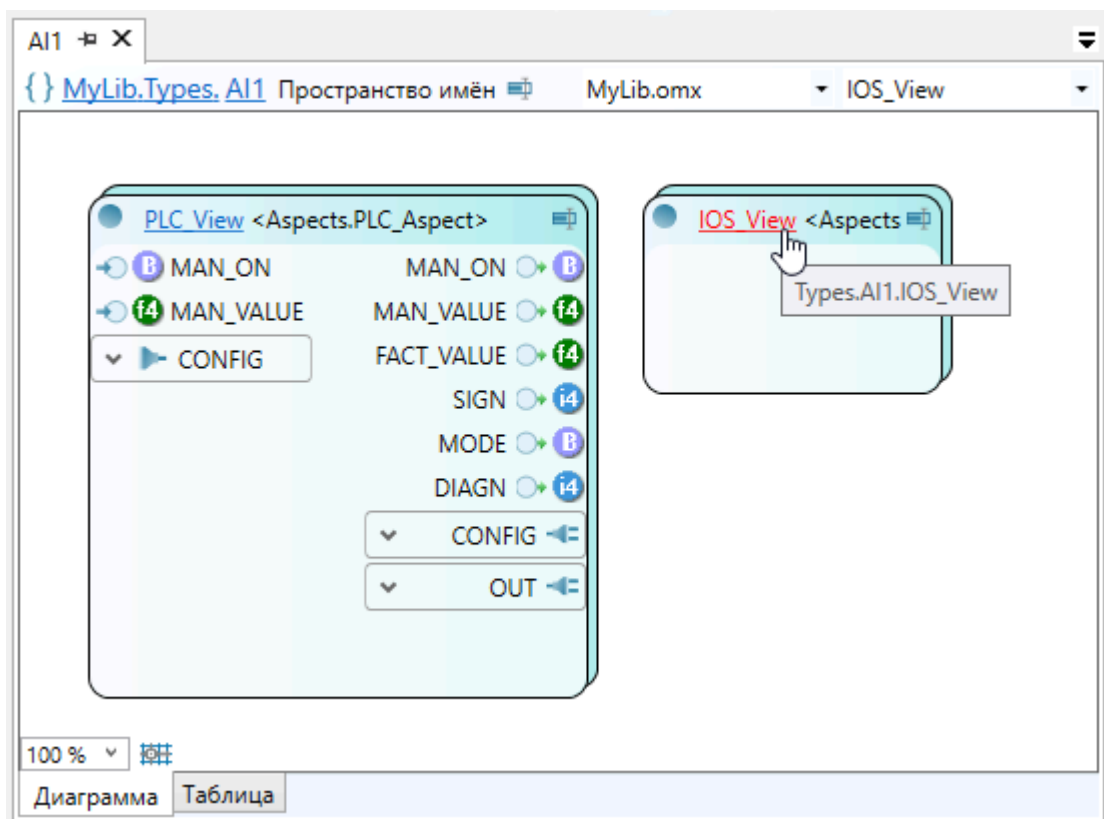


Типы параметров должны совпадать с типами данных в ПЛК.

17. Перейдите в пространство имён "AI1" двойным кликом по названию объекта в обозревателе решений.



18. Перейдите в представление "IOS\_View", кликнув по названию объекта.

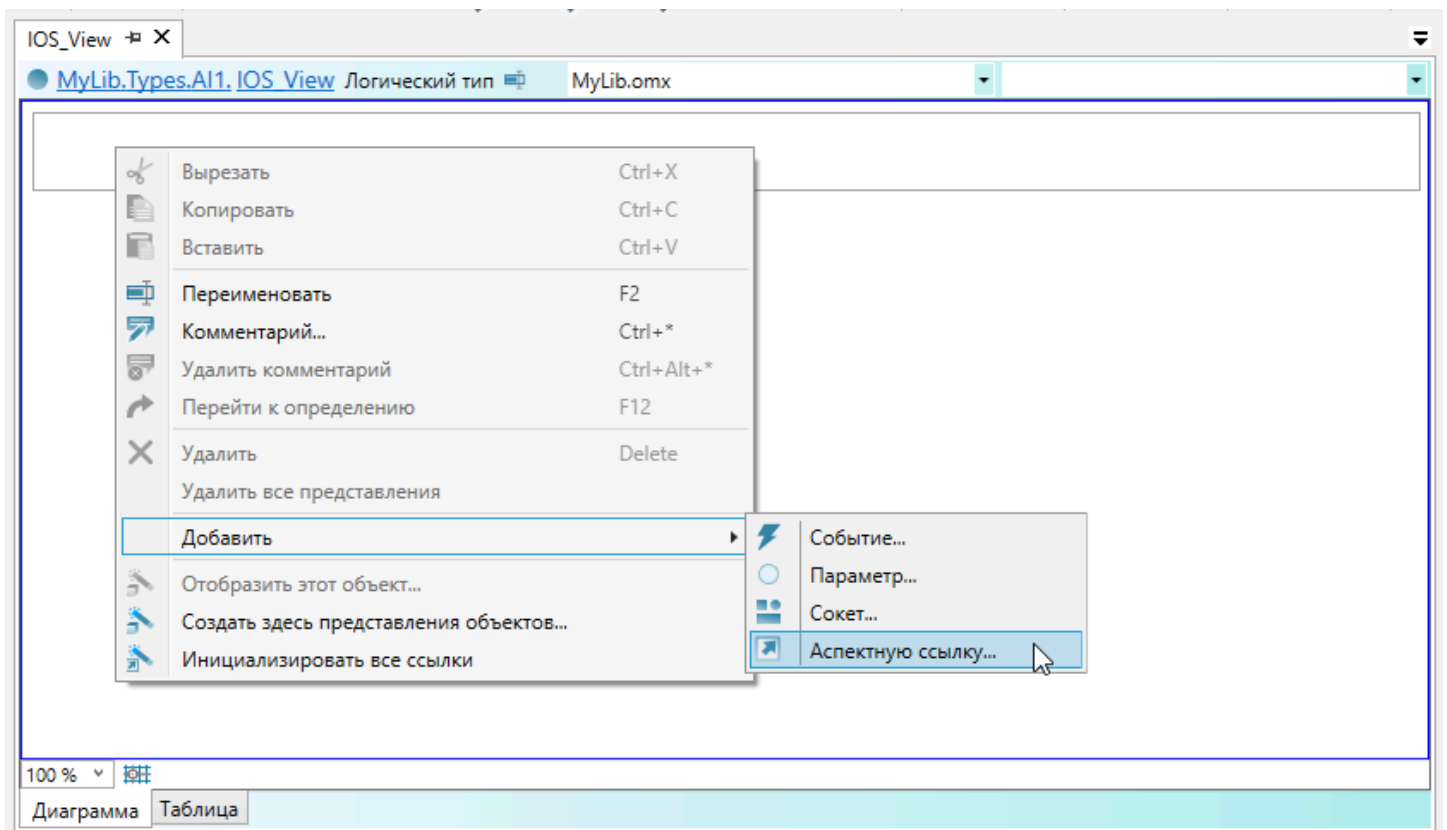


19. Для передачи данных между представлениями используются аспектные ссылки.

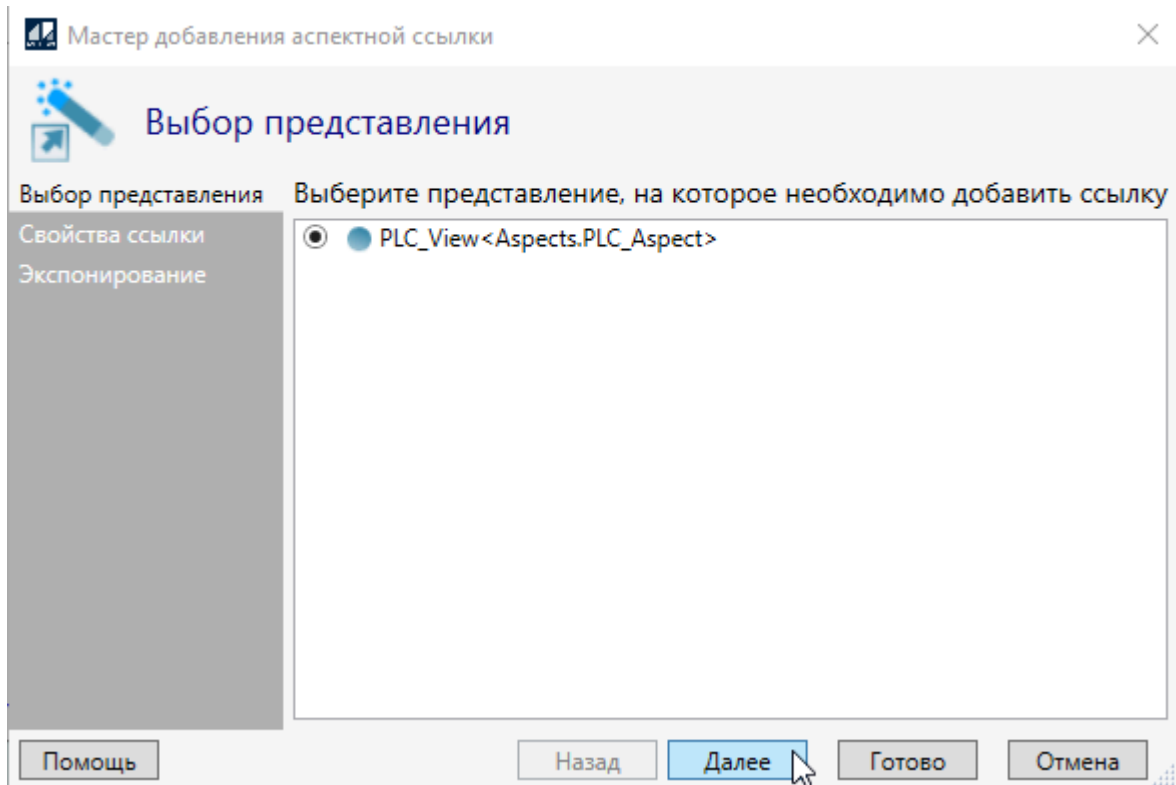


**Аспектная ссылка** – это ссылка, которая добавляется в объект и указывает на другое представление этого объекта.

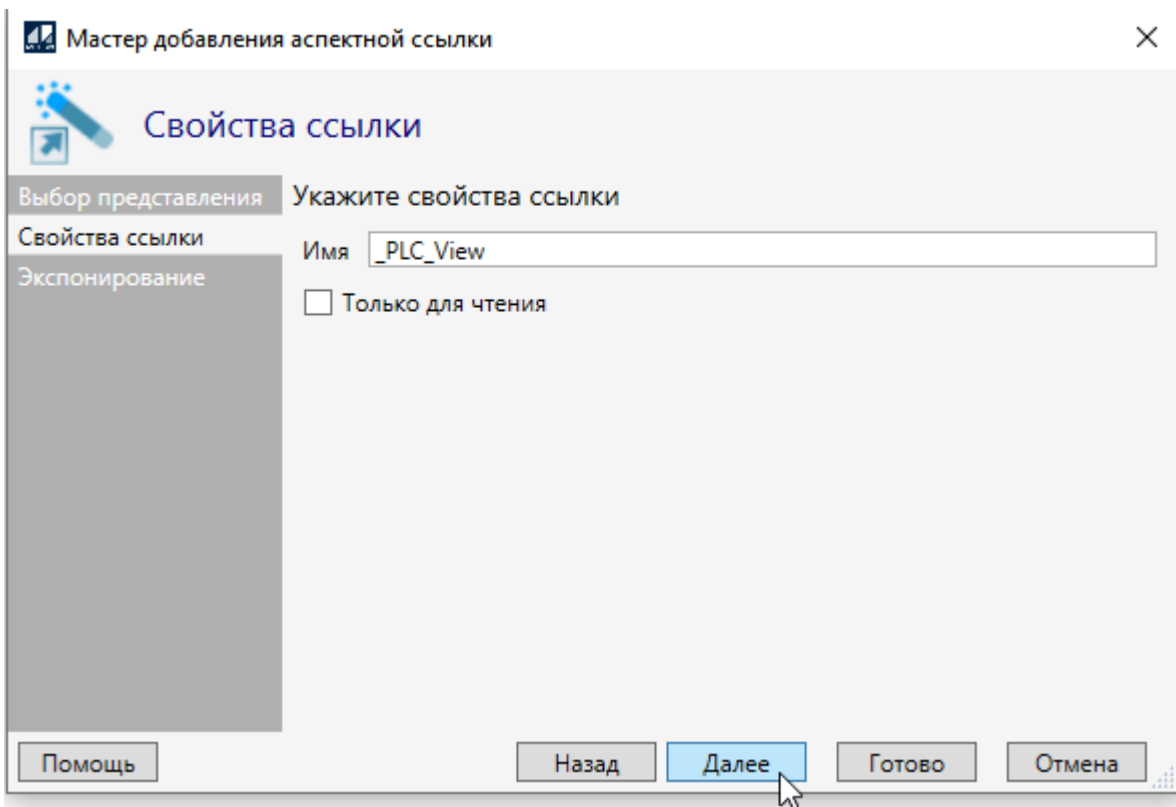
Нажмите ПКМ по свободному полю и выберите из контекстного меню команду "Добавить" -> "Аспектную ссылку".



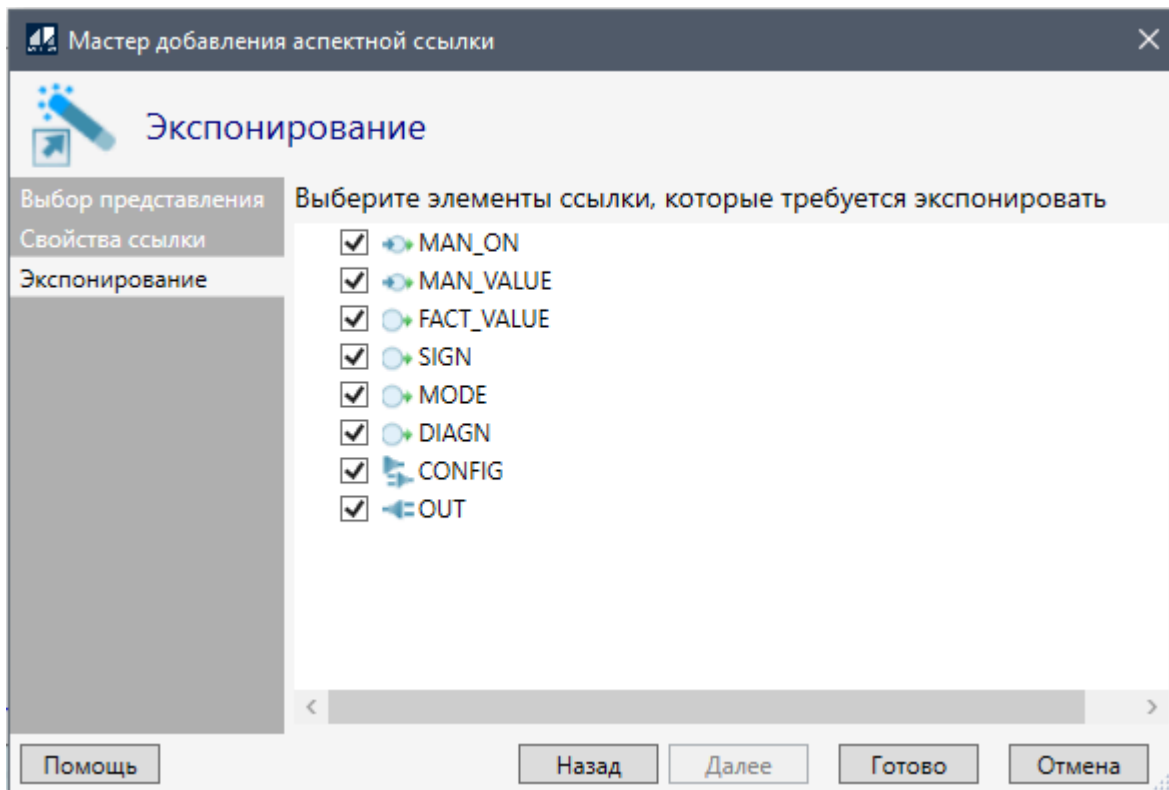
20. В открывшемся окне выберите представление, на которое необходимо добавить ссылку и нажмите кнопку "Далее".



22. Нажмите кнопку "Далее".



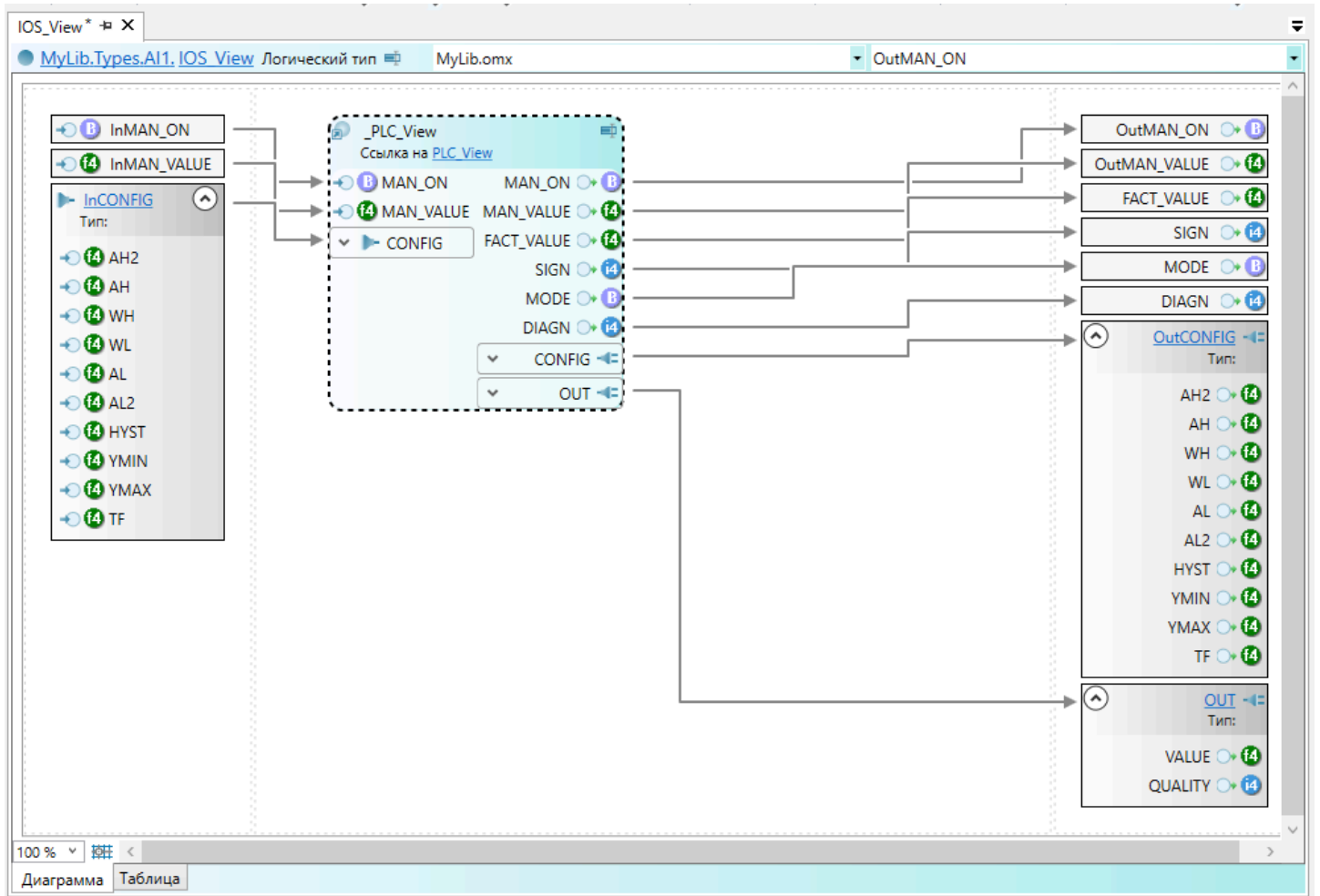
23. Выделите параметры, ссылки на которые требуется экспонировать, и нажмите кнопку "Готово".



В результате образуется связь между представлением сервера ввода/вывода и представлением ПЛК.



Если связи не отобразились, нажмите клавишу "F5".



В самых простых случаях представление на сервере ввода/вывода совпадает с представлением на уровне ПЛК. Однако в большинстве случаев необходимо делать дополнительную обработку параметров на сервере ввода/вывода.

Далее рассмотрим примеры дополнительной обработки на сервере ввода/вывода:

- › [Задание формул](#)
- › [Создание обработчика событий](#)
- › [Добавление атрибутов](#)
- › [Генерация событий](#)
- › [Сохранение в историю](#)

## 1.2.3.4.1. Задание формул

Формулу следует использовать, если значение сигнала является результатом обработки одного или нескольких сигналов и должно пересчитываться при каждом изменении любого из них.



Формулу можно использовать только для следующих типов сигналов:

- › переменная;
- › выходной параметр: параметр, у которого в свойстве Направление указано значение Выход.

Основные правила:

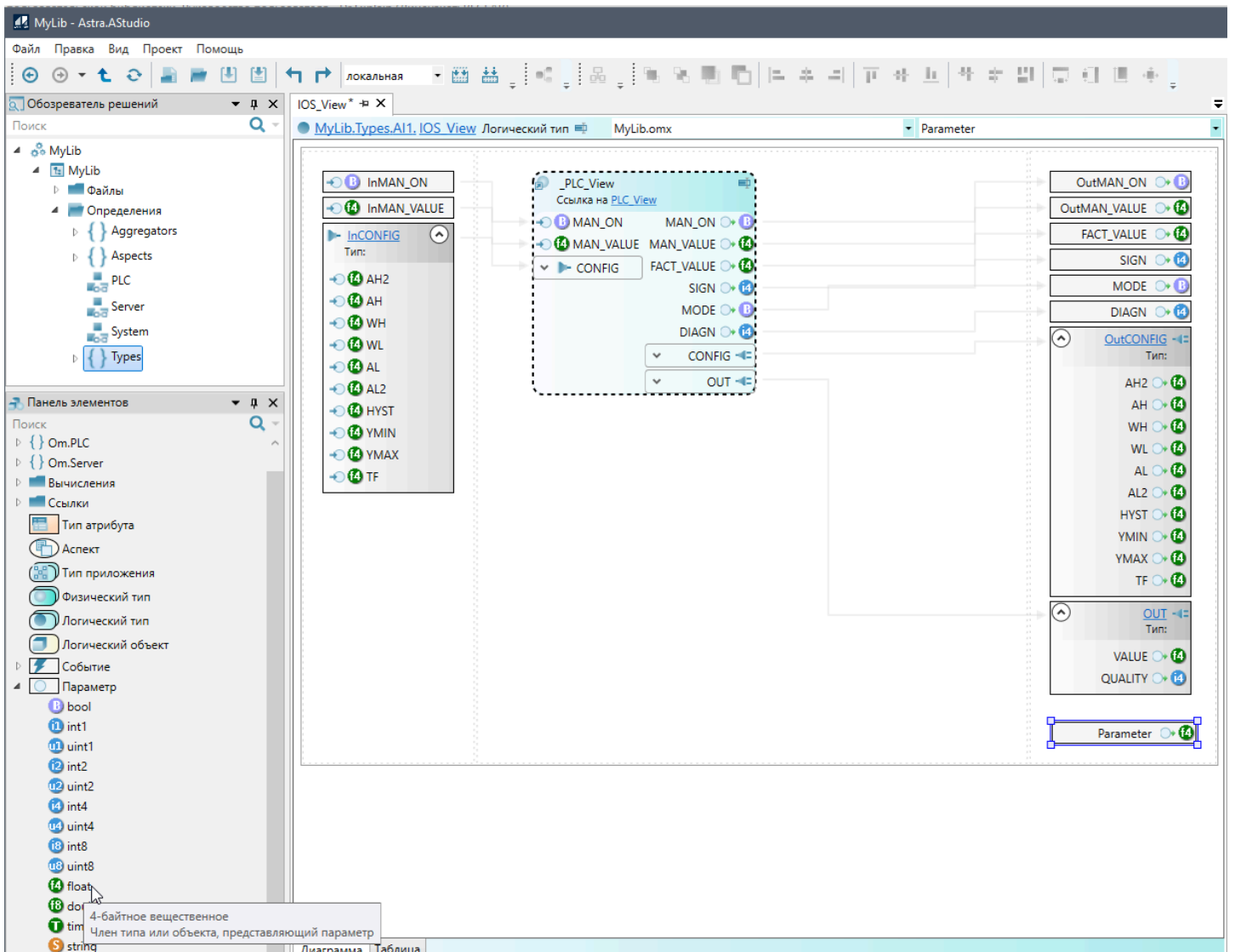
- › формула должна быть написана на языке Astra.Om.
- › формула должна описывать выражение, результат вычисления которого будет присвоен сигналу в качестве значения.
- › результат вычисления должен быть того же типа, что и тип значения сигнала, или неявно к нему приводиться.
- › формула должна содержать хотя бы один сигнал без оператора read. Если в формуле указать read перед именем сигнала, она не будет пересчитываться при изменениях значений этого сигнала. В противном случае формула не будет запущена никогда.
- › в формуле можно использовать значение самого сигнала, значение которого вычисляется: формула не будет пересчитываться при изменении значения этого сигнала.



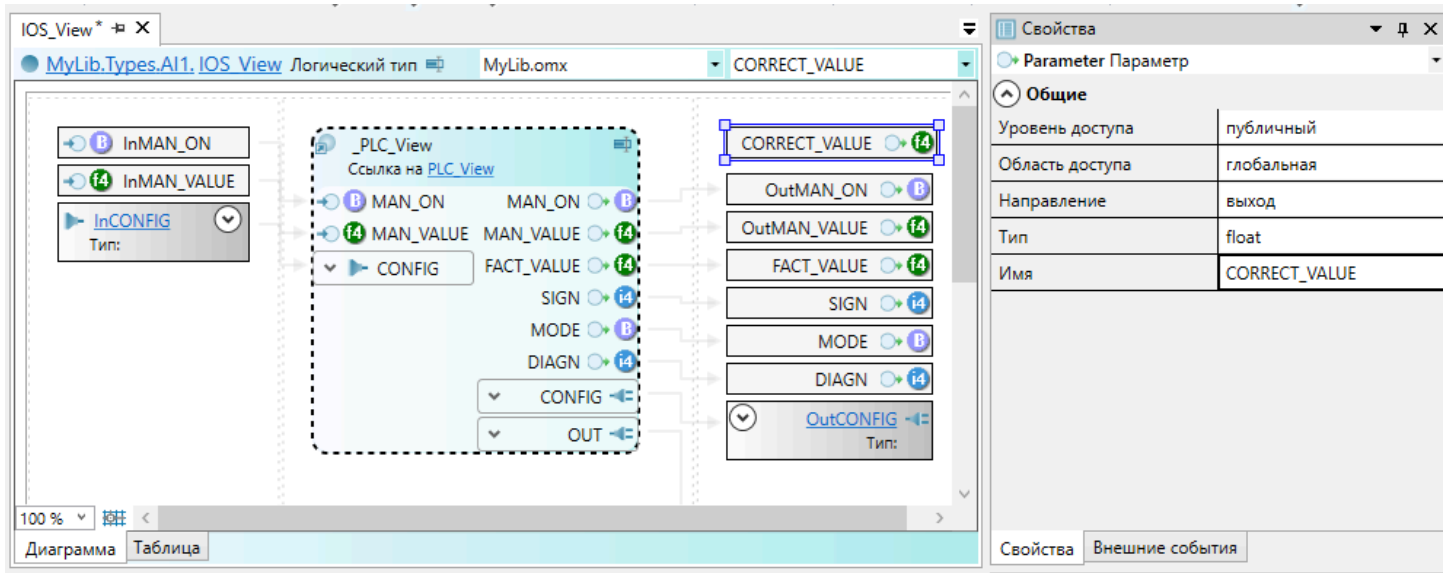
Рассмотрим пример задания формулы для пересчета параметра выходного значения OUT.VALUE на уровне сервера ввода/вывода.

Чтобы добавить параметр для пересчета значения, выполните следующие действия:

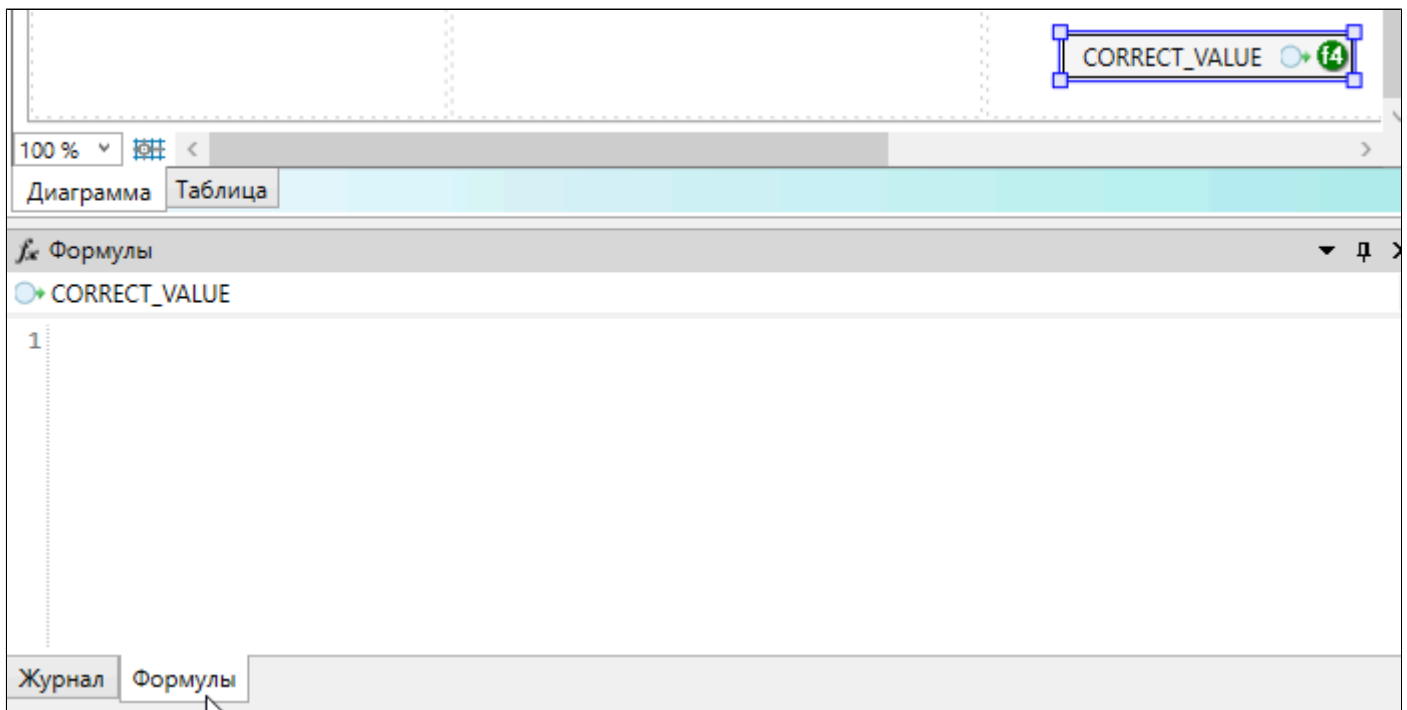
1. Добавьте из панели элементов параметр того же типа, что и параметр, который будет использован для пересчета.



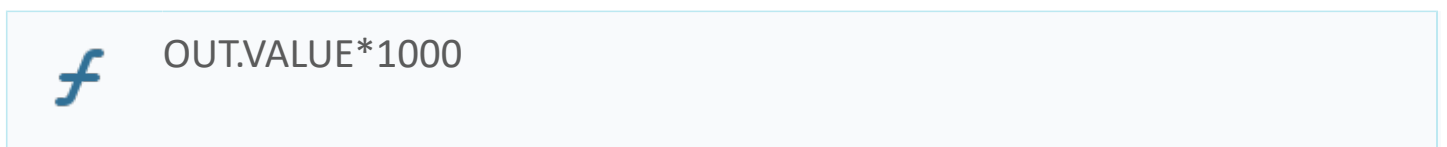
2. Выделите добавленный параметр и в окне "Свойства" укажите имя "CORRECT\_VALUE" и направление "выход".

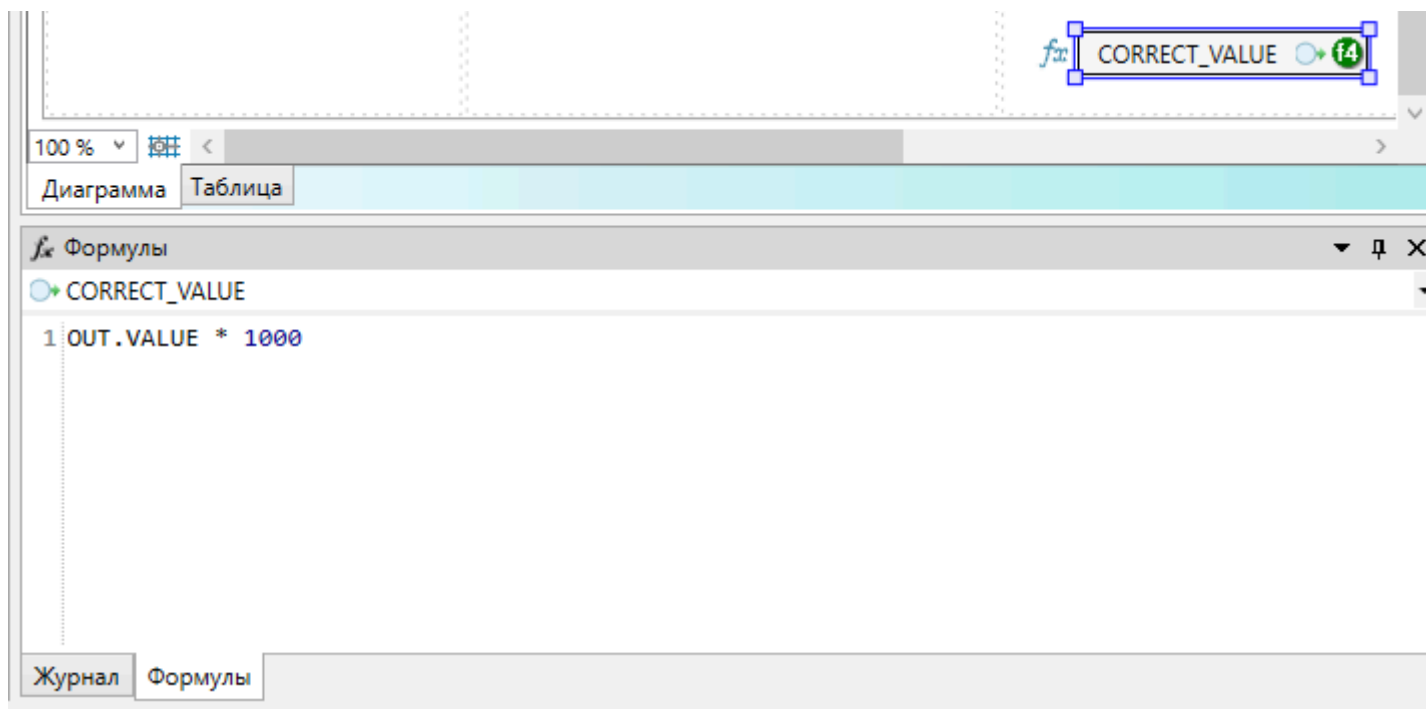


3. Выделите параметр "CORRECT\_VALUE" и в нижней части интерфейса Astra.AStudio перейдите во вкладку "Формулы".



4. В окне редактора формул укажите следующую формулу:





После задания формулы, около параметра появится иконка расчёта значения.

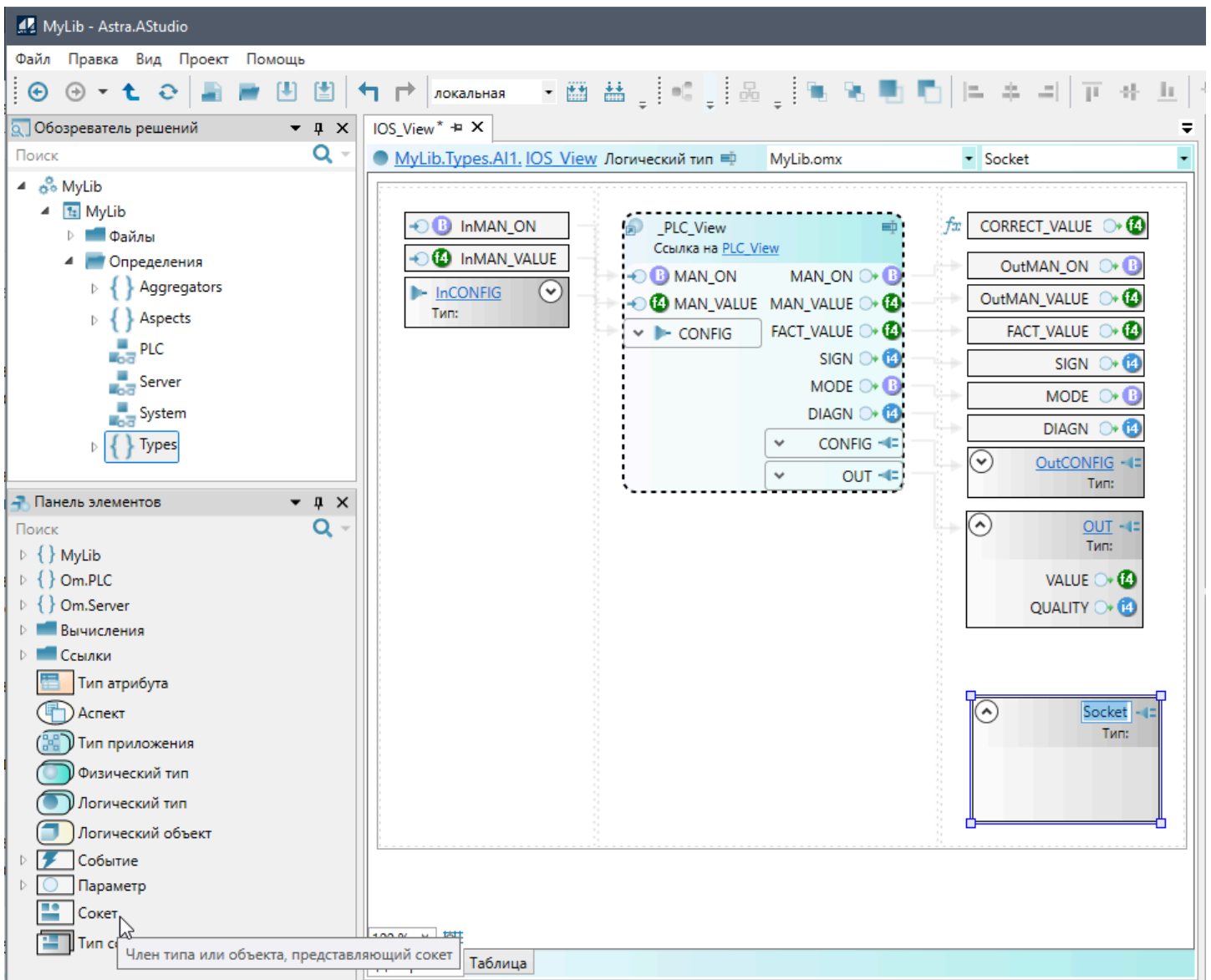
Таким образом можно задавать формулы для параметров представления сервера ввода/вывода. Данное преобразование будет выполняться постоянно. Чтобы пересчет выполнялся по определенному событию или изменению сигнала необходимо воспользоваться [обработчиком событий](#).

## 1.2.3.4.2. Создание обработчика событий

Рассмотрим пример формирования битов в зависимости от значения переменной DIAGN. Для данных целей можно воспользоваться обработчиком событий.

Чтобы создать обработчик события выполните следующие действия:

1. Перенесите элемент "Сокет" на рабочую область из панели элементов.

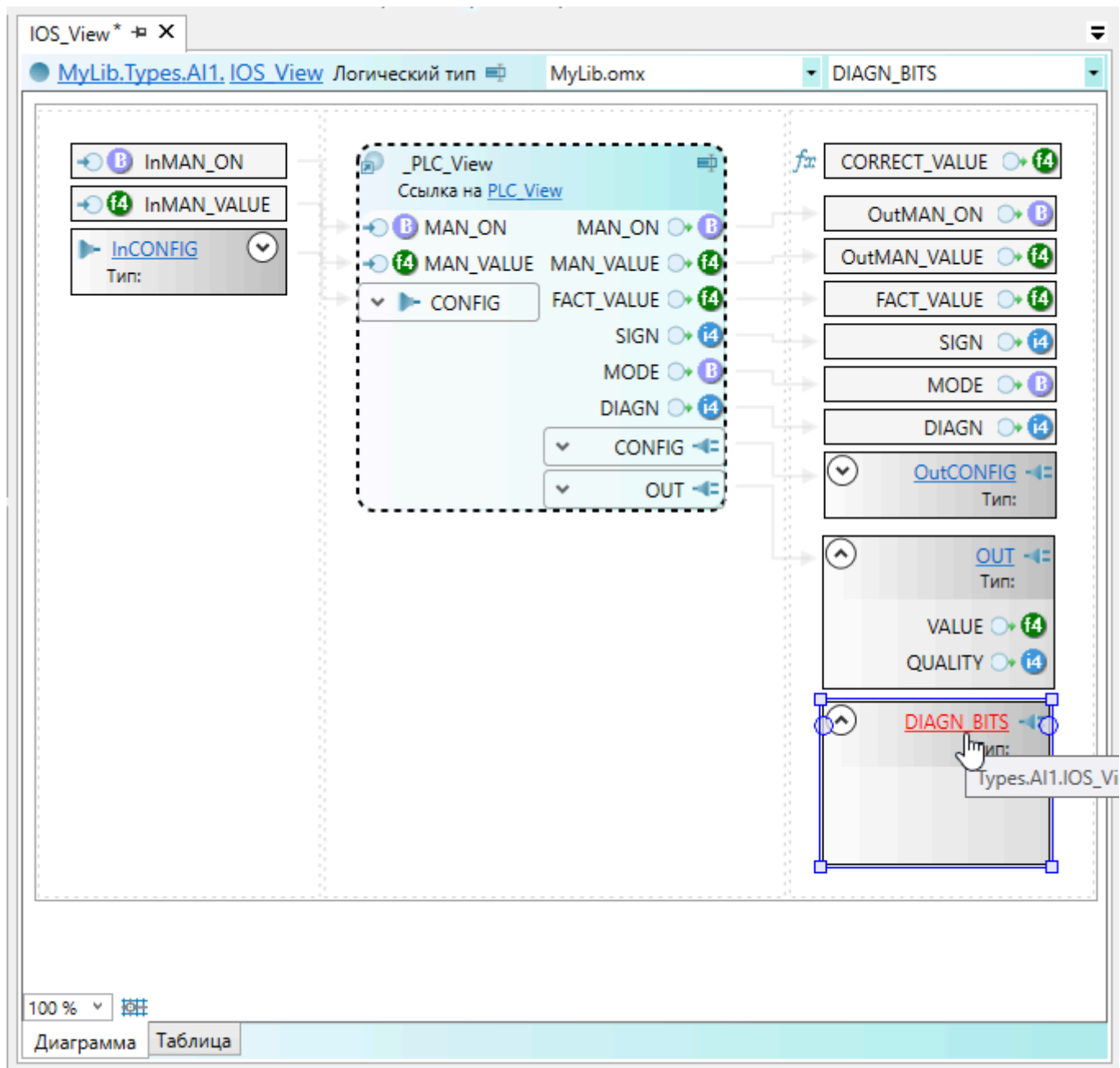


2. Выделите сокет и в окне "Свойства" укажите имя "DIAGN\_BITS" и направление "выход". Данный сокет будет отвечать за отдельные состояния диагноза канала.

The screenshot displays the IOS\_View software interface. The main window shows a diagram of the PLC View components, including InMAN\_ON, InMAN\_VALUE, InCONFIG, MAN\_ON, MAN\_VALUE, CONFIG, SIGN, MODE, DIAGN, CORRECT\_VALUE, OutMAN\_ON, OutMAN\_VALUE, FACT\_VALUE, SIGN, MODE, DIAGN, OutCONFIG, and OUT. The DIAGN\_BITS component is highlighted with a blue border. The properties window on the right shows the QUALITY\_BITS socket properties, including the direction dropdown menu which is currently set to "отсутствует" (absent) and has a mouse cursor hovering over the "выход" (output) option.

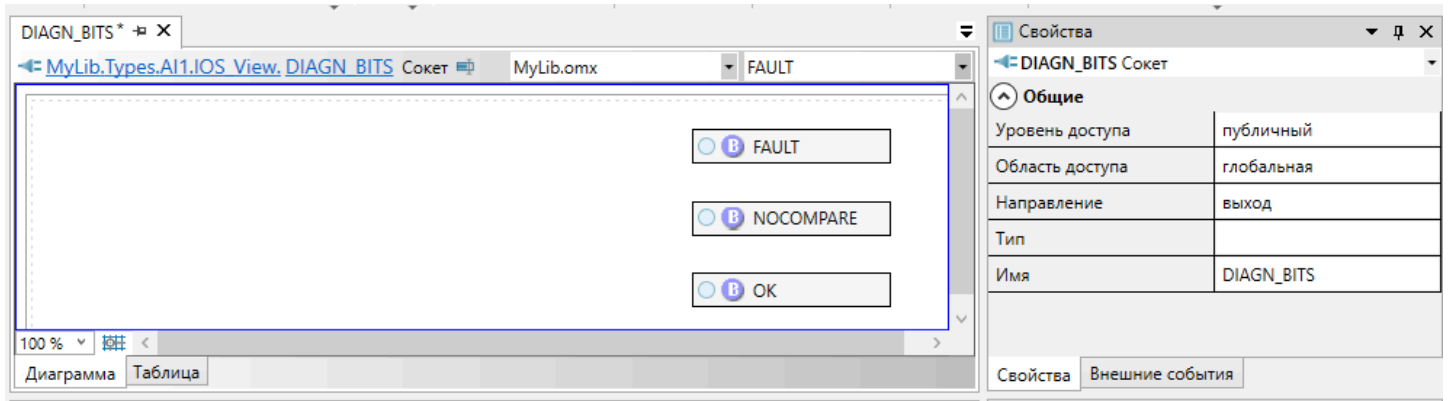
Свойства	
← QUALITY_BITS Сокет	
Общие	
Уровень доступа	публичный
Область доступа	глобальная
Направление	отсутствует
Тип	отсутствует
Имя	вход
	выход
	вход/выход

3. Перейдите в редактор сокета "DIAGN\_BITS", кликнув по названию объекта.



4. В сокет "DIAGN\_BITS" вынесите три параметра типа bool из панели элементов и задайте им следующие названия:

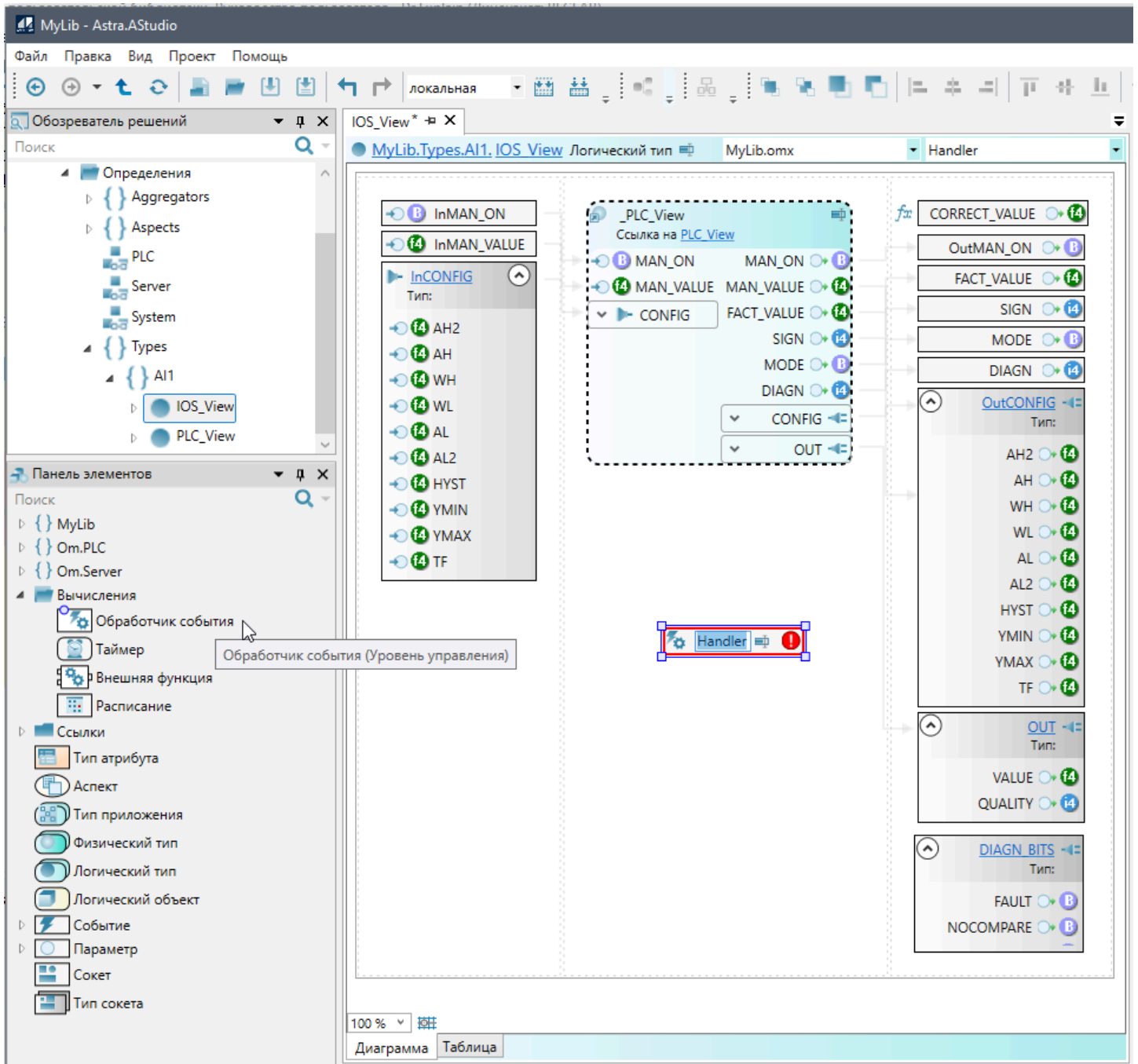
- FAULT
- NOCOMPARE
- OK



5. Вернитесь в представление "IOS\_View" типа и добавьте элемент "Обработчик события" из раздела "Вычисления" панели элементов.

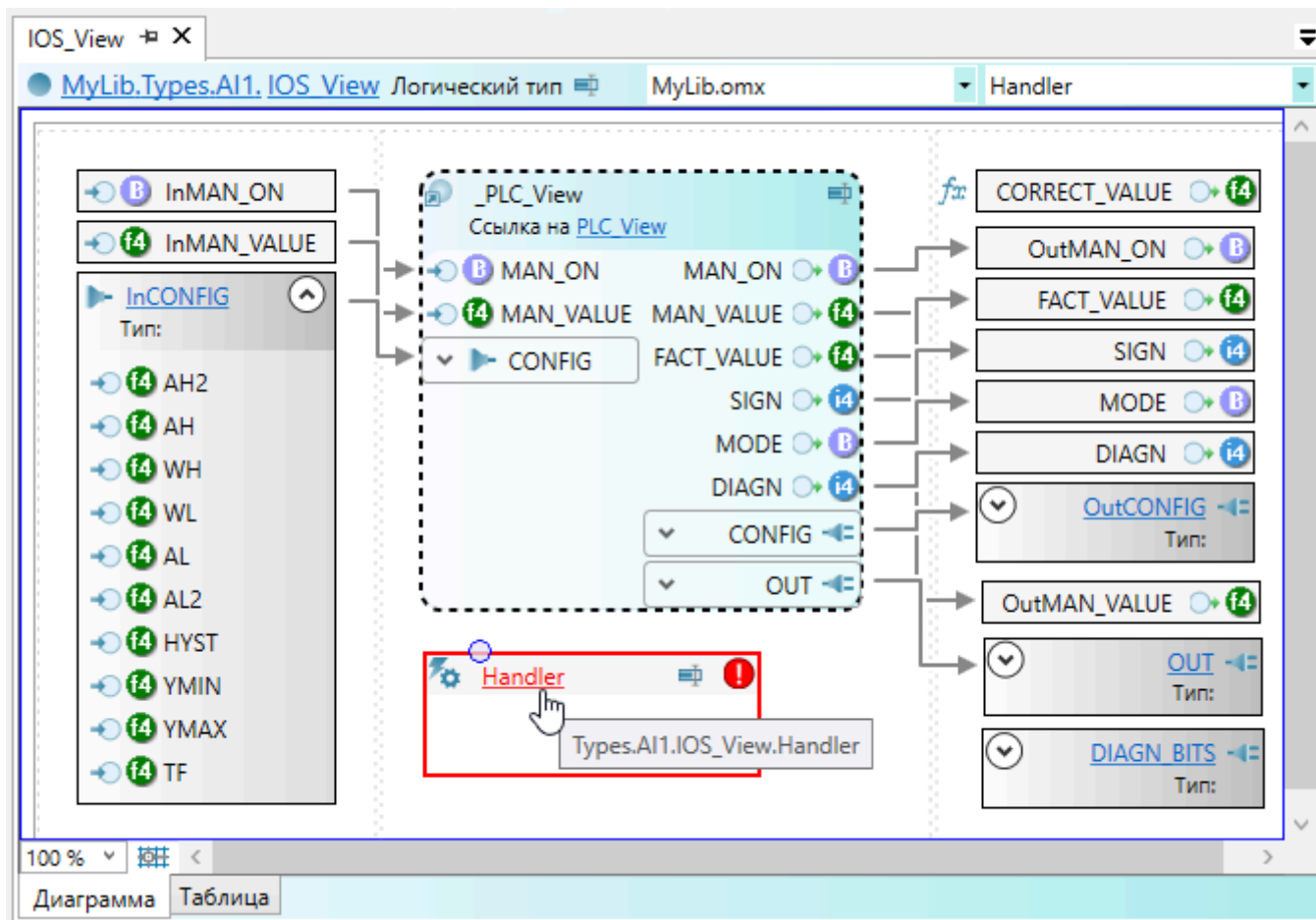


Название можно оставить по умолчанию.



6. Перейдите в редактор кода обработчика, кликнув по названию объекта.





7. В окне редактора задайте следующий код:

```

Handler  IOS_View*
MyLib.Types.AI1.IOS View, Handler  Обработчик события
1 DIAGN_BITS.FAULT      = DIAGN == 0; //Недостоверный
2 DIAGN_BITS.NOCOMPARE = DIAGN == 1; //Несравнение
3 DIAGN_BITS.OK        = DIAGN == 2; //OK

```



Языком программирования для написания кода является Astra.Om. Синтаксис языка приведен в документе "Astra.HMI. Руководство пользователя".


8. Закройте обработчик события воспользовавшись кнопкой "Закреть".

```

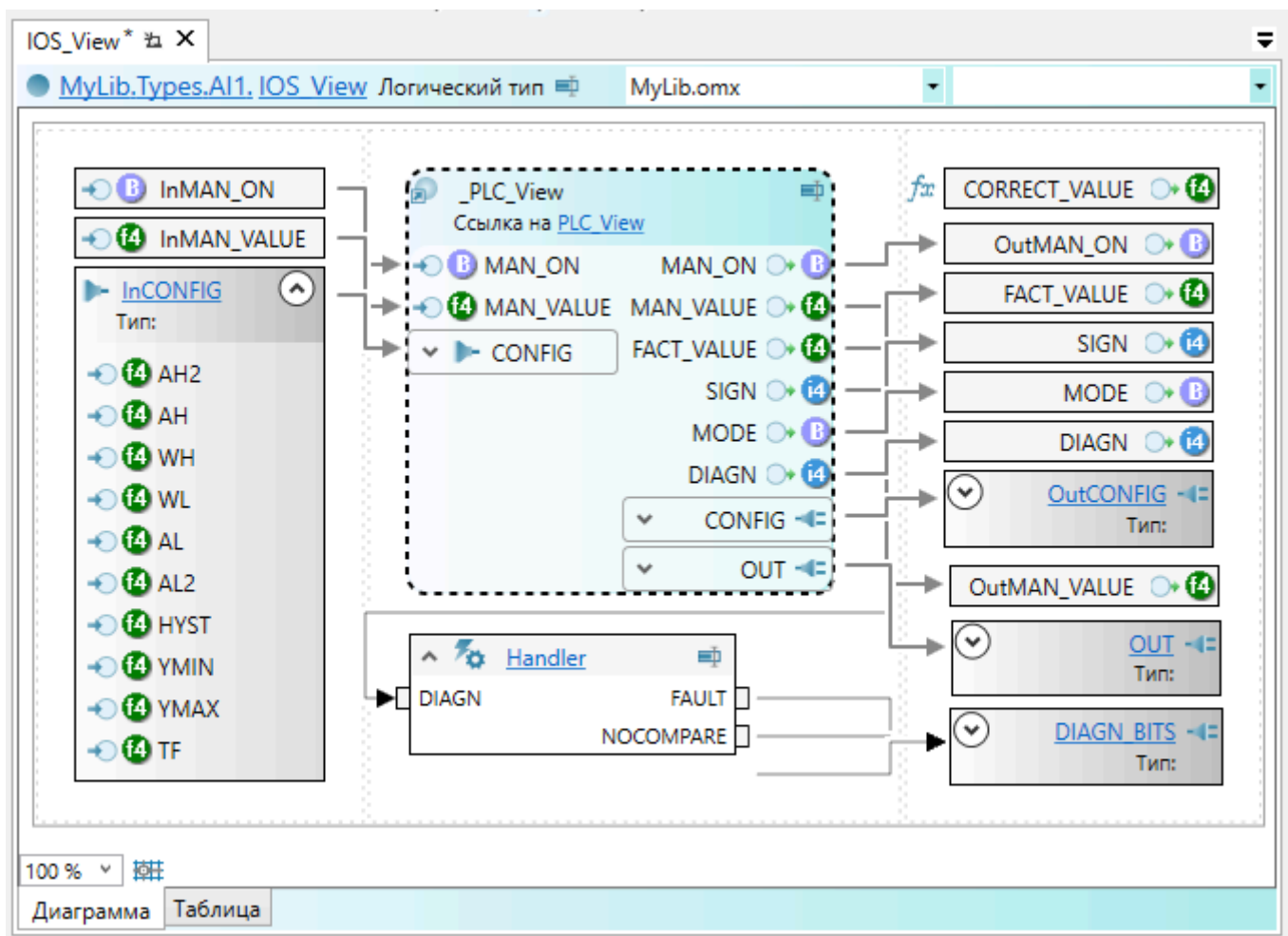
1 DIAGN_BITS.FAULT = DIAGN == 0; //Недостоверный
2 DIAGN_BITS.NOCOMPARE = DIAGN == 1; //Несравнение
3 DIAGN_BITS.OK = DIAGN == 2; //ОК

```

У обработчика появились входные и выходные параметры.

 Чтобы у обработчик появились связи с сокетом, нажмите горячую клавишу F5.

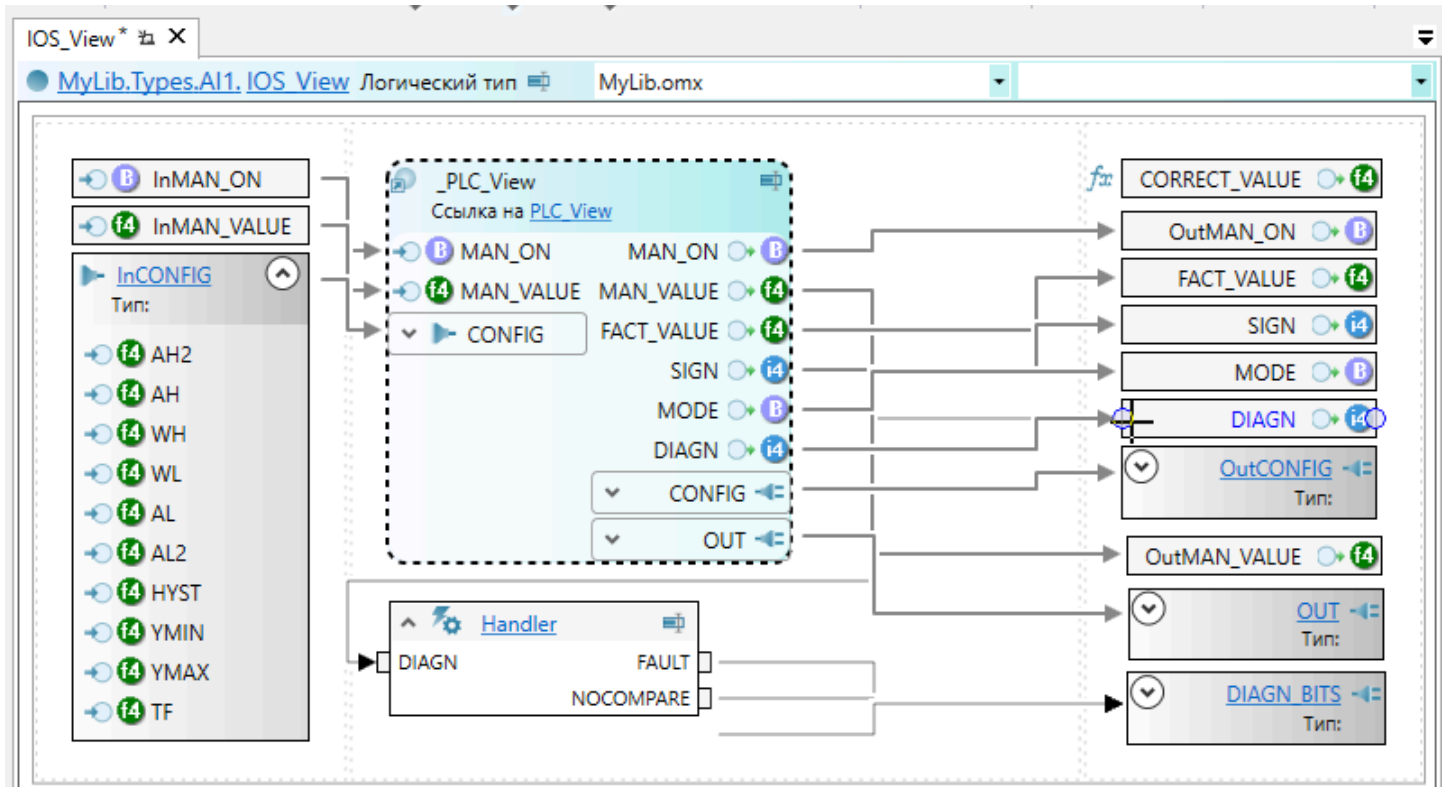
Связи между переменными, задействованными в обработчике событий должны отобразиться на рабочей области.



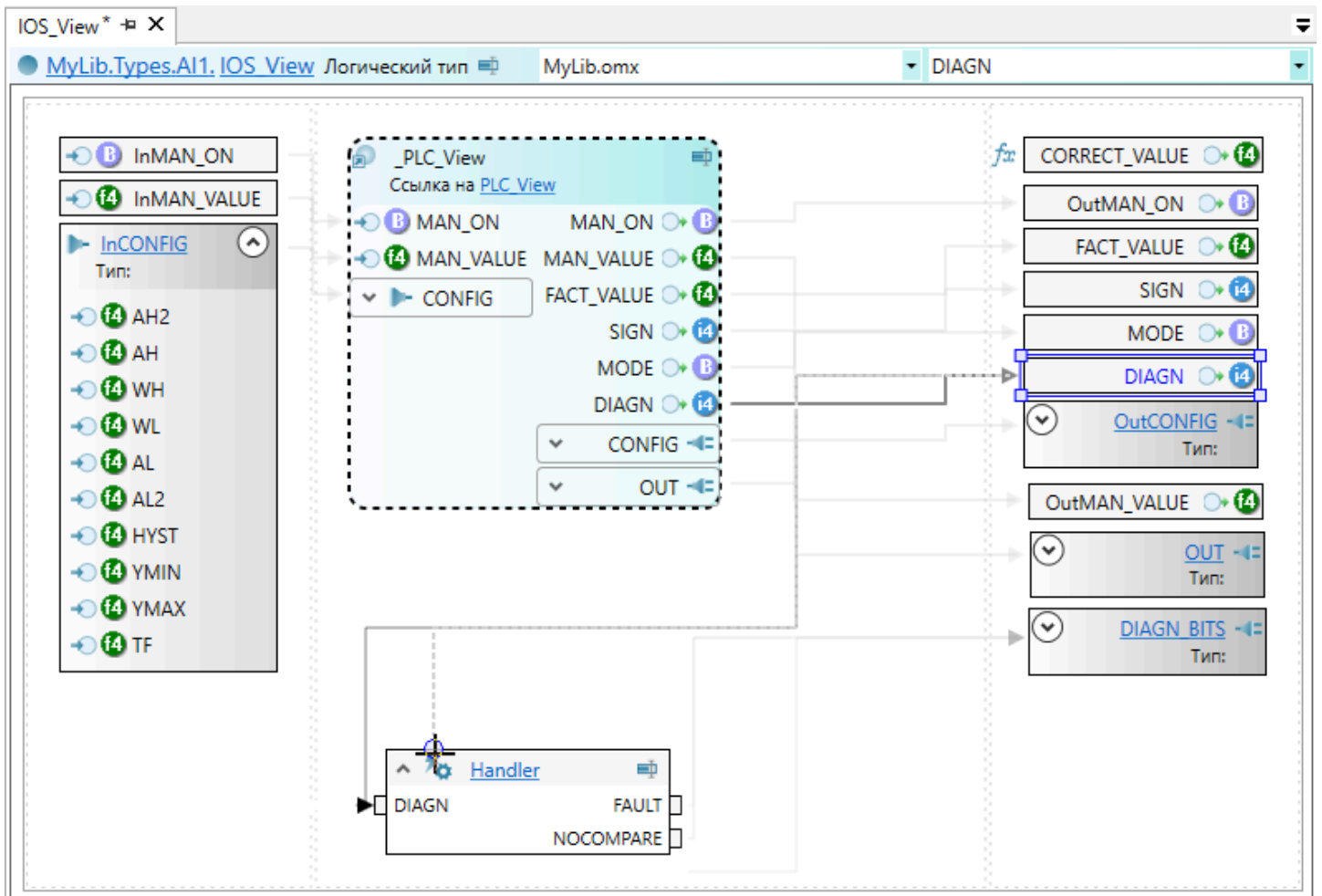


Для того, чтобы обработчик события начал выполнять код, необходимо добавить триггер.

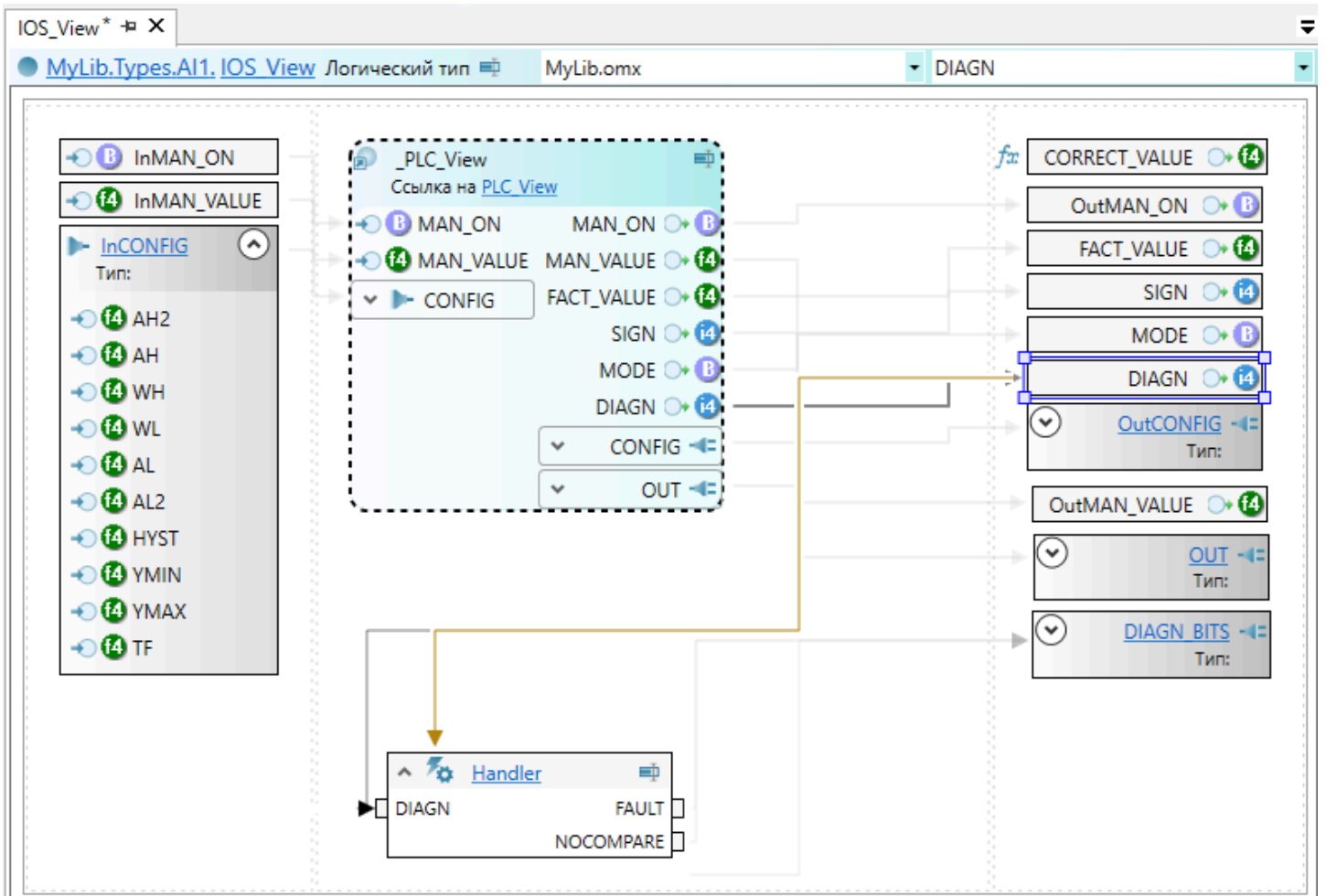
9. Чтобы добавить триггер к обработчику событий, подведите указатель мыши к переменной DIAGN до появления указателя в виде прицела.



Зажмите левой кнопкой мыши область в виде круга и проведите линию к обработчику события.



Линия будет выделена оранжевым цветом. Это и есть триггер, запускающий обработчик событий.



10. Выделите триггер, кликнув по оранжевой стрелке, и в окне "Свойства" в поле "Событие" выберите тип срабатывания.

Свойства	
Триггер	
Общие	
Источник события	DIAGN
Событие	Изменение сигнала
<ul style="list-style-type: none"> <li>Изменение сигнала</li> <li>Обновление</li> <li>Любое изменение</li> <li>Подготовка сообщения</li> </ul>	
Событие	
Событие, являющееся причиной срабатывания триггера	
Свойства	Внешние события

Таким образом можно создавать обработчики события, для выполнения кода по определенному триггеру.

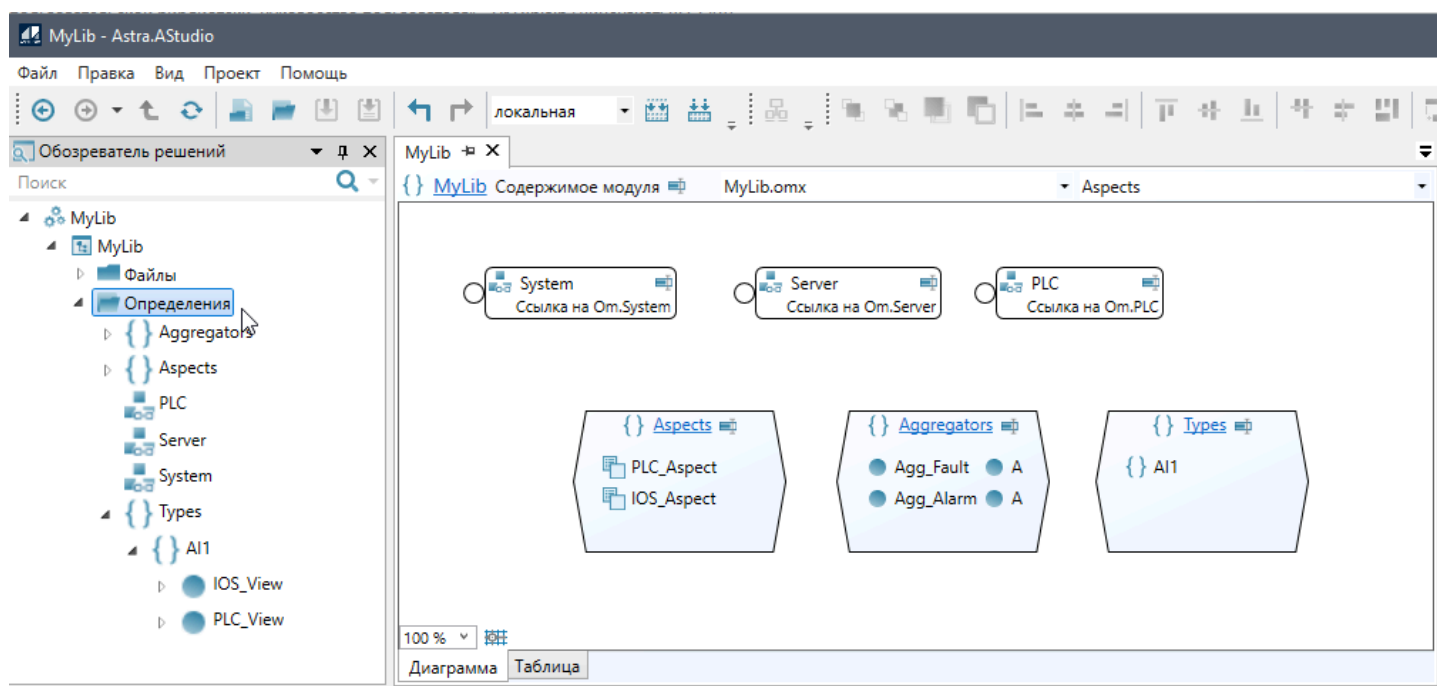
## 1.2.3.4.3. Добавление пользовательских атрибутов

На уровне сервера ввода/вывода можно добавлять дополнительные параметр: название, количество знаков после запятой, обозначение, единицы измерения и т. д.

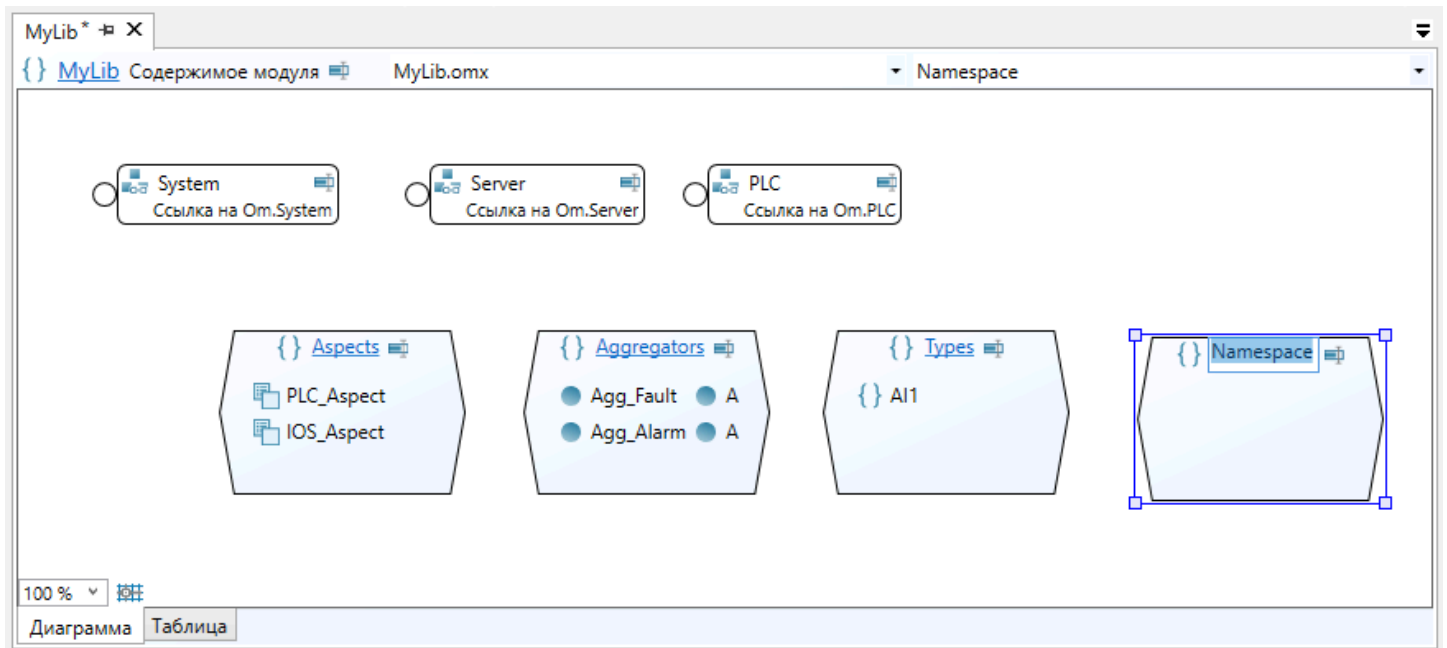
Данные параметры задаются с помощью атрибутов.

Чтобы добавить атрибуты, выполните следующие действия:

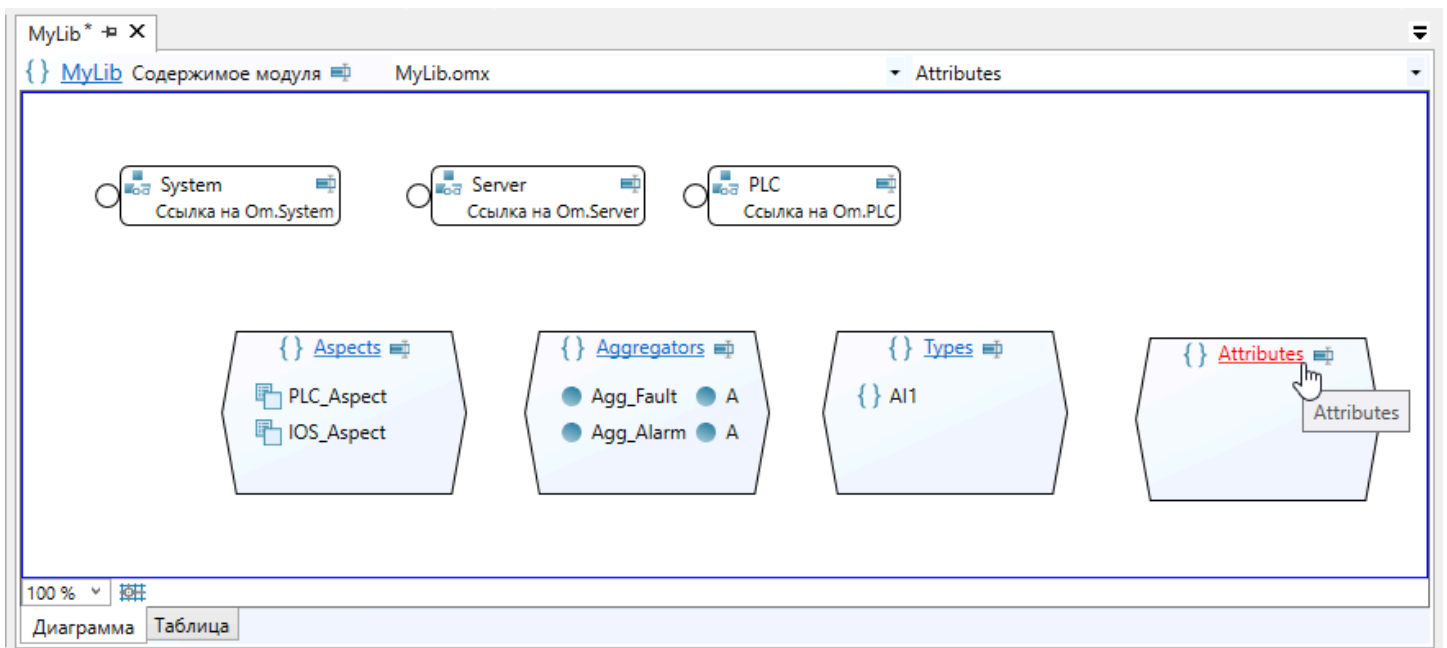
1. Перейдите в определения проекта "MyLib" двойным кликом по объекту "Определения" в обозревателе решений.



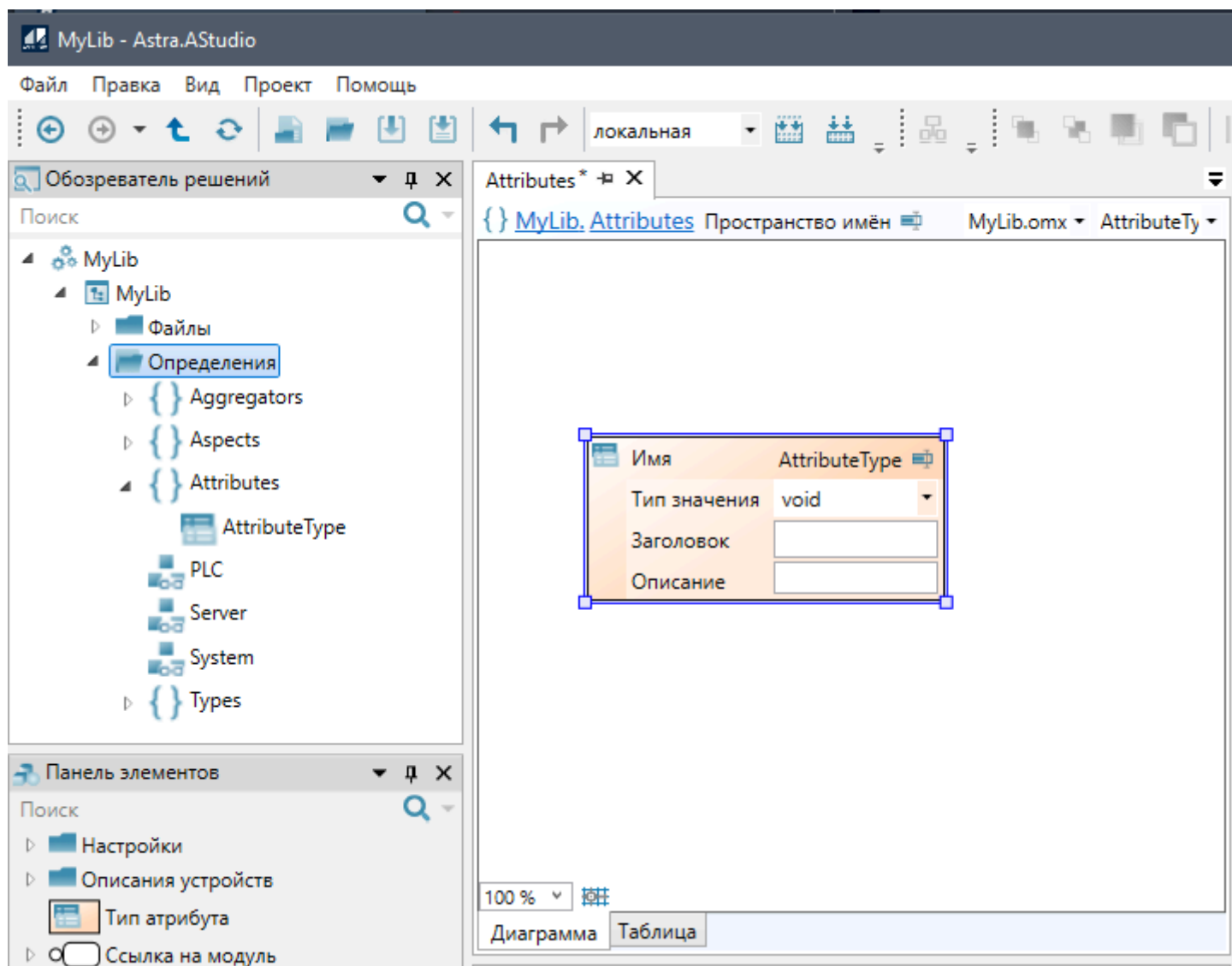
2. Добавьте на рабочую область "Пространство имен" путем перетаскивания из панели элементов.



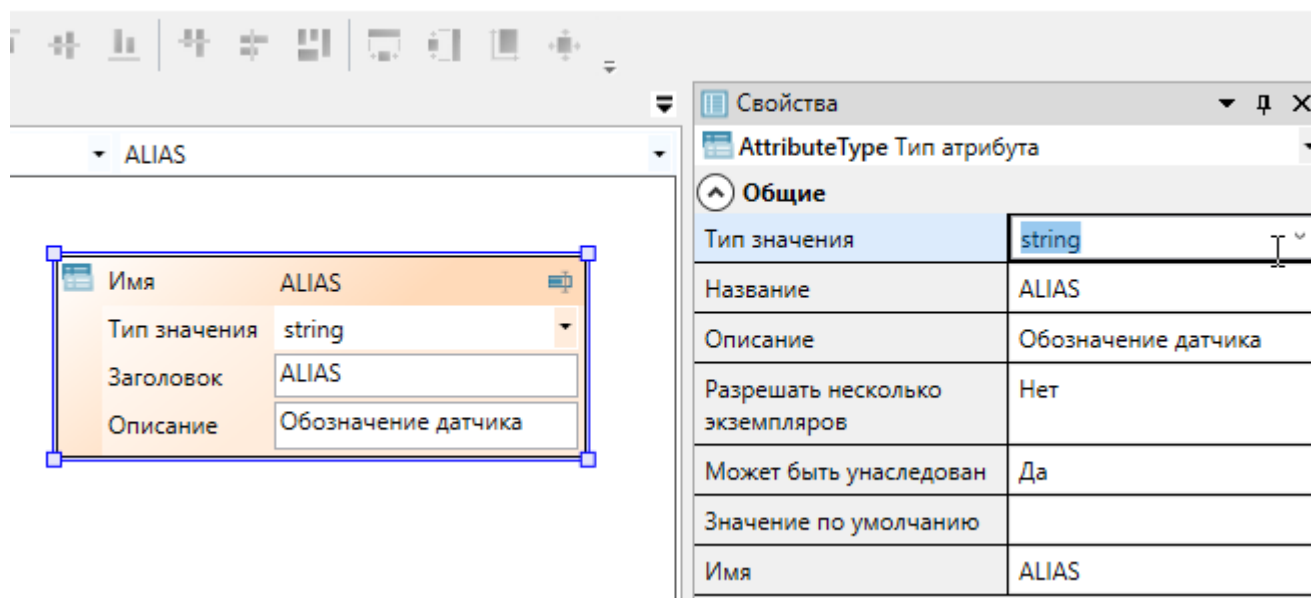
3. Задайте добавленному элементу имя "Attributes" и перейдите в него, кликнув по названию объекта.



4. В пространство имен "Attributes" добавьте элемент "Тип атрибута" из панели элементов.



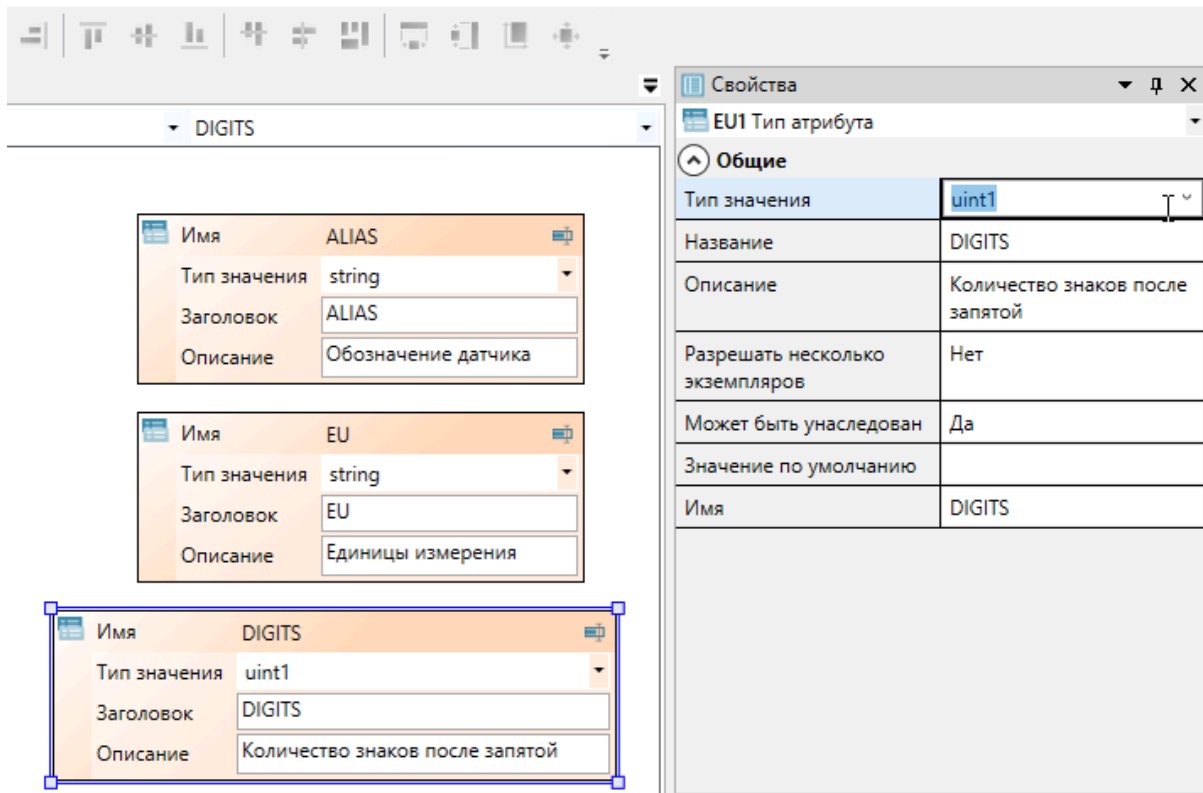
5. Выделите тип атрибута и в окне "Свойства" и задайте следующие значения:





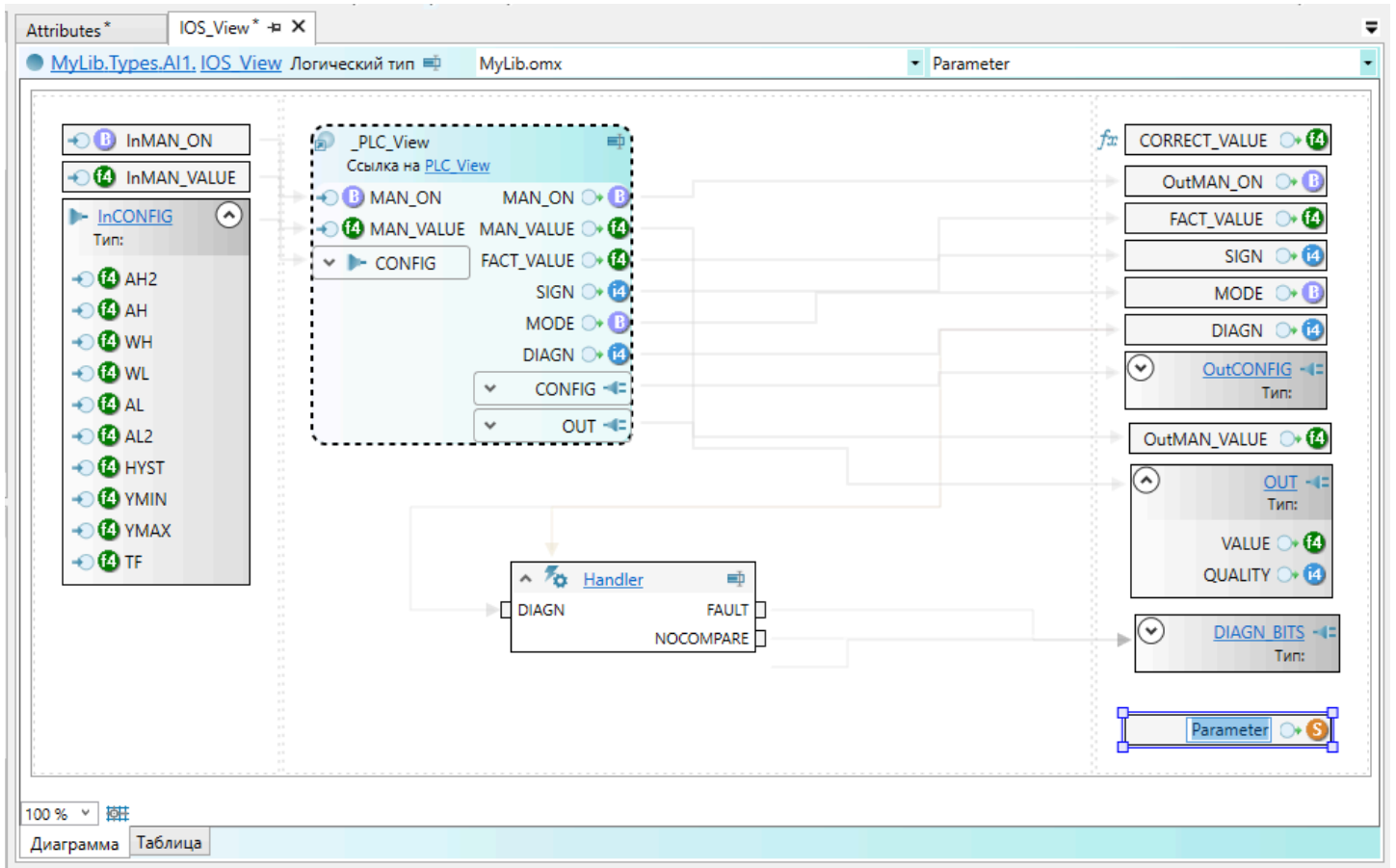
6. Аналогичным образом создайте еще два атрибута:

- › EU (для единиц измерения).
- › DIGITS (для количества знаков после запятой).

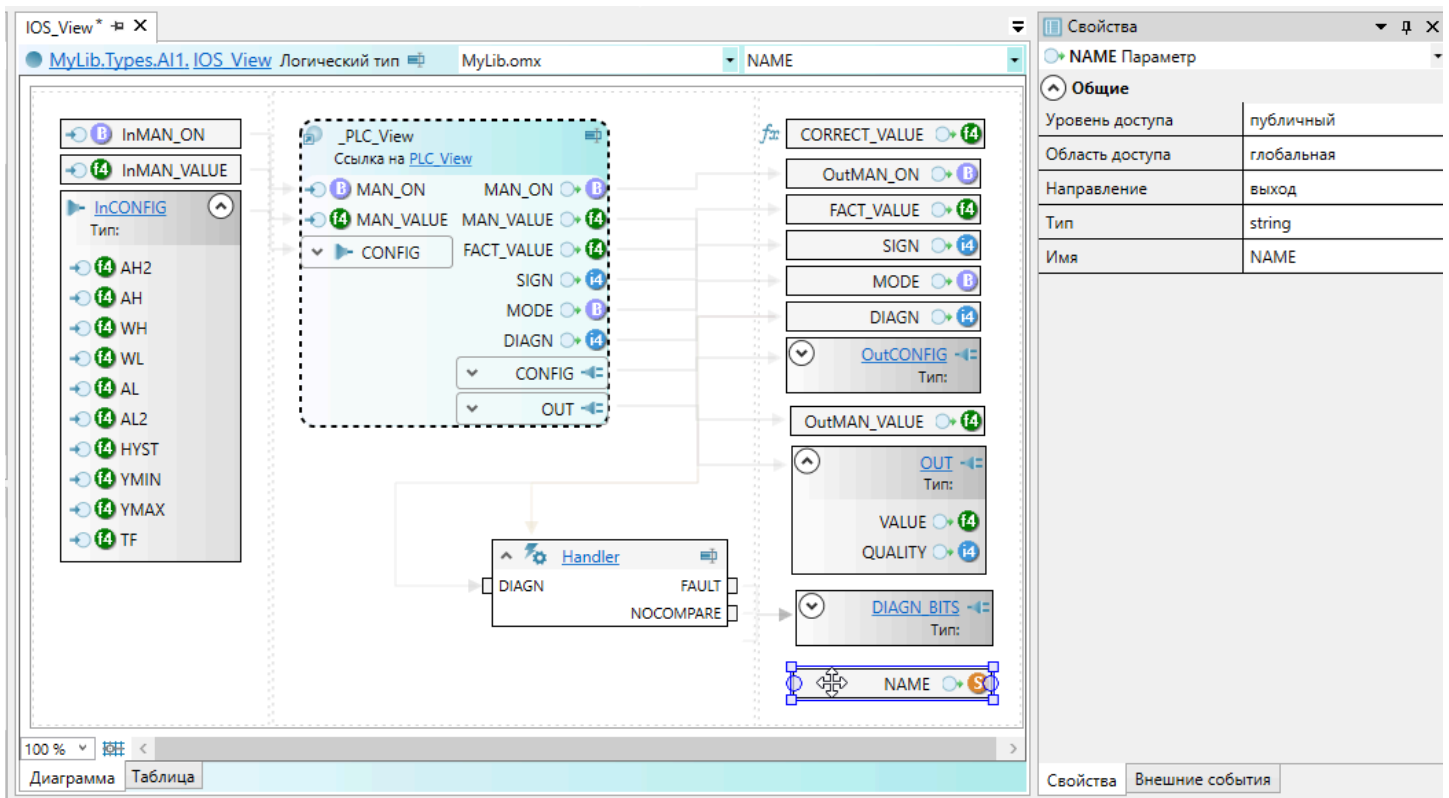


Таким образом мы создали три атрибута, с помощью которых можно задать начальные значения у параметров аналогового датчика в представлении "IOS\_View".

7. Перейдите в представление "IOS\_View" типа "AI1" аналогового датчика и добавьте параметр типа "String" из панели элементов.

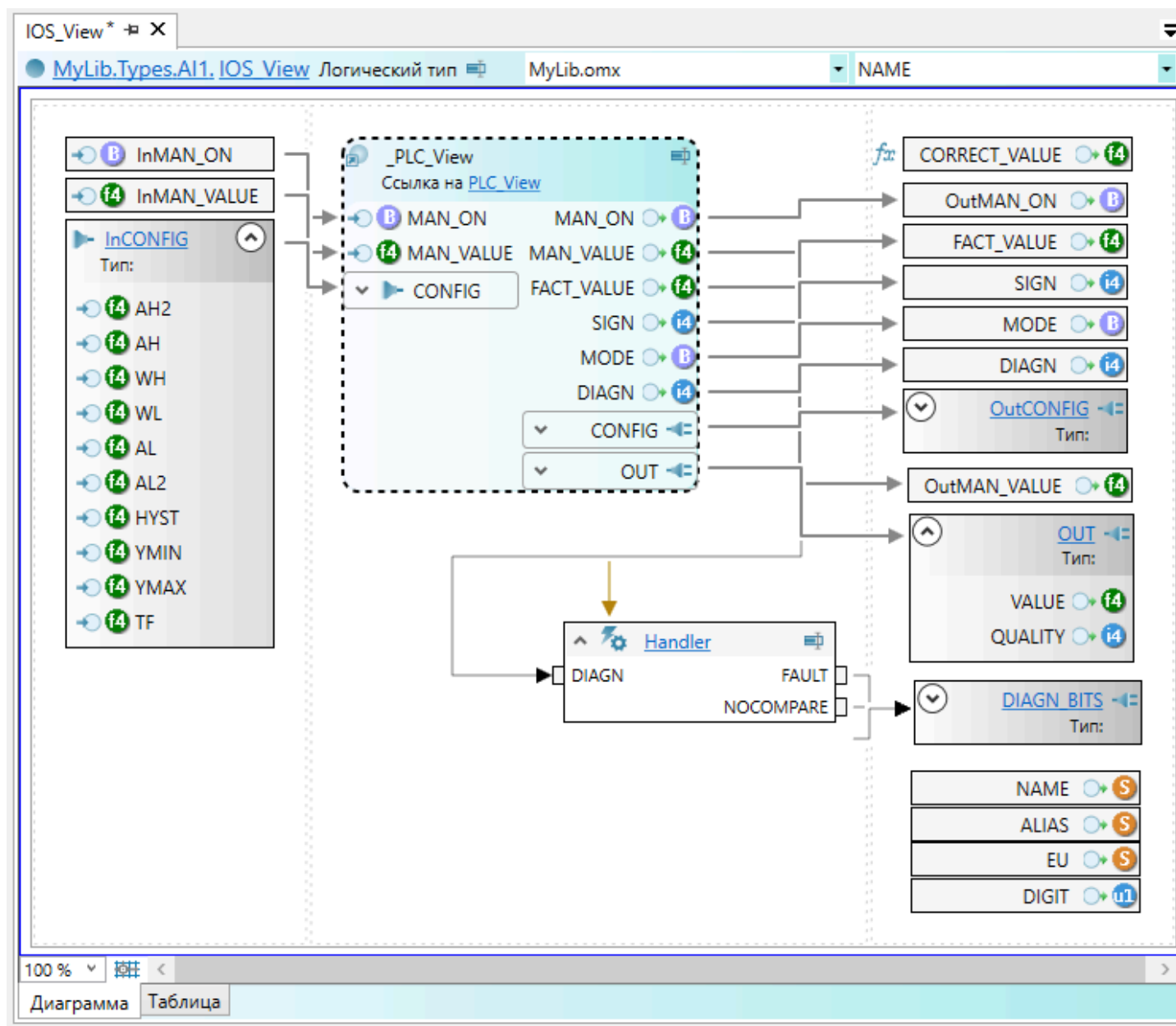


8. Выделите добавленный параметр и в окне "Свойства" укажите имя "NAME" и направление "выход". Данный параметр будет отображать название датчика.



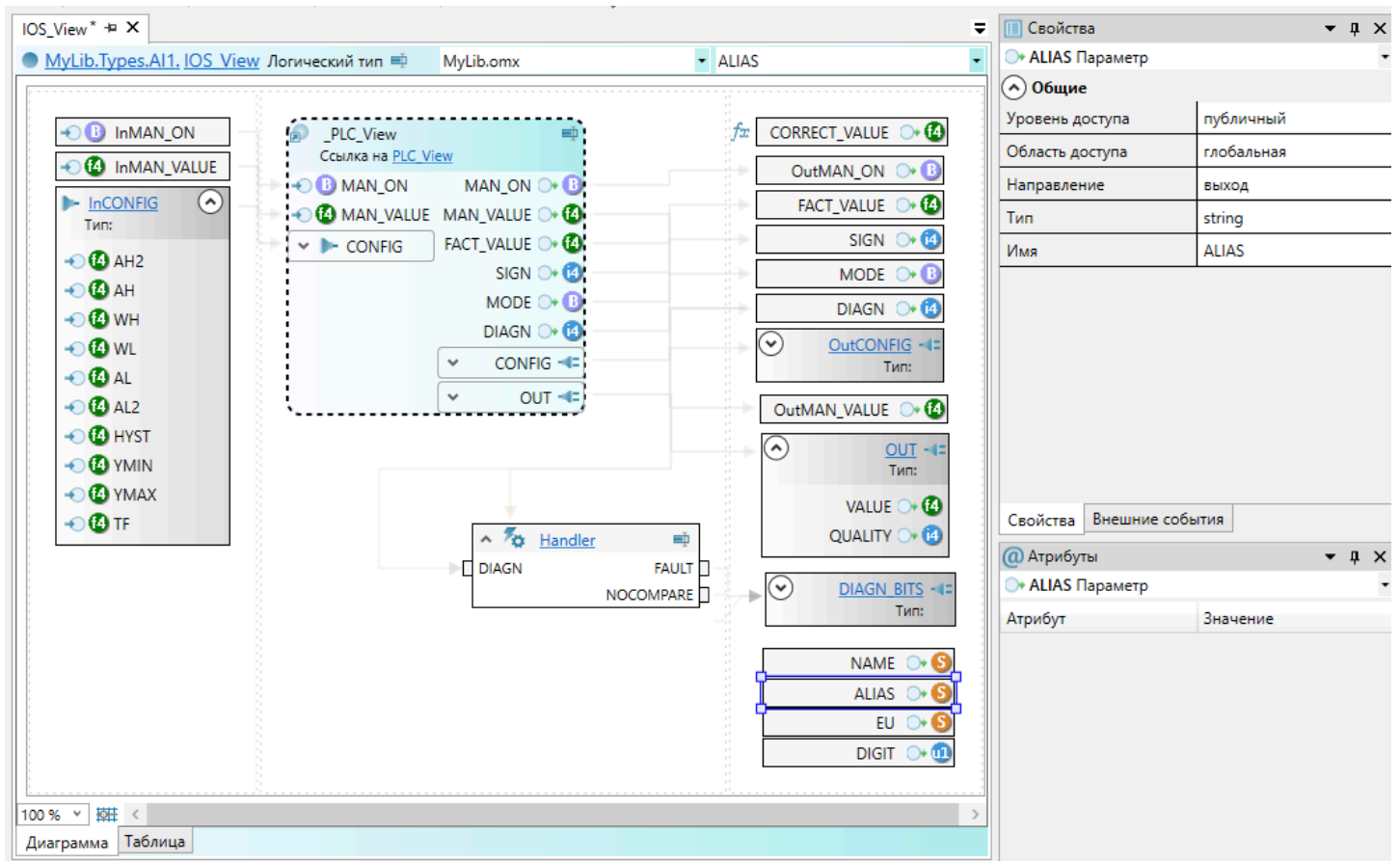
9. Аналогичным образом добавьте следующие параметры:

Название	Тип данных	Описание
ALIAS	string	Обозначение датчика
EU	string	Единицы измерения
DIGITS	uint1	Кол-во знаков после запятой

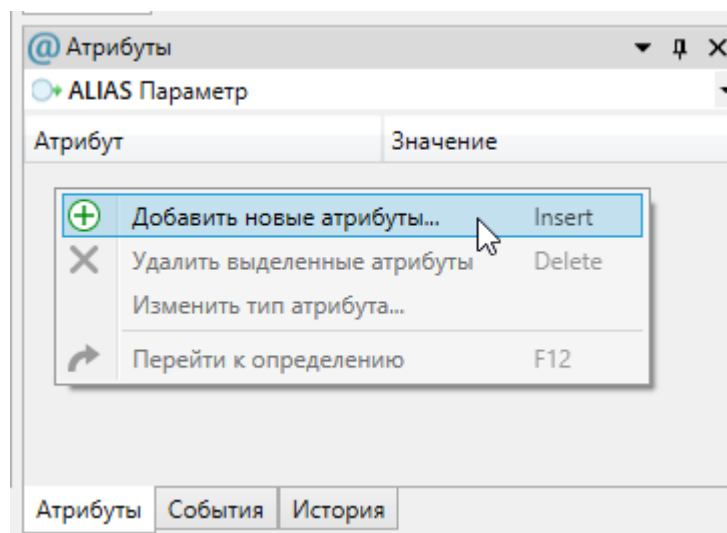


Значения данных параметров будут задаваться у каждого экземпляра датчика

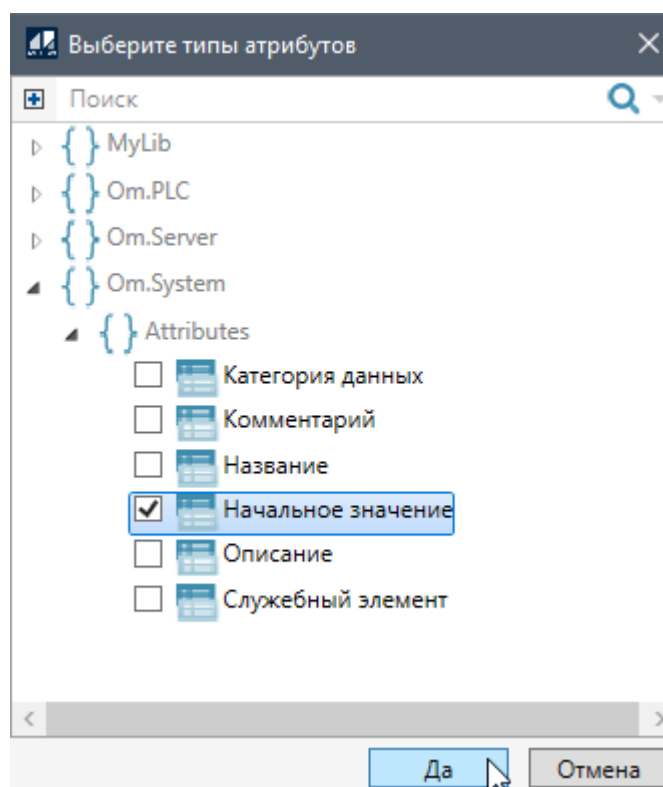
10. Задайте начальное значение для параметра "ALIAS" с помощью атрибута. Выделите параметр и перейдите во вкладку атрибуты.



11. В окне "Атрибуты" нажмите ПКМ по свободному полю и в контекстном меню выберите команду "Добавить новые атрибуты.."



12. В открывшемся окне из раздела "Om.System" выберите атрибут "Начальное значение" и нажмите кнопку "Да".



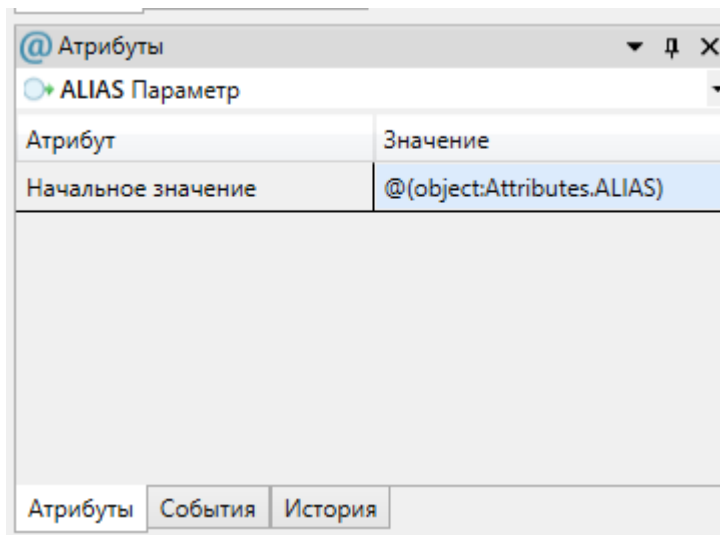
13. В начальном значении необходимо сослаться на ранее созданный атрибут ALIAS.

Чтобы добавить ссылку на атрибут укажите в поле "Начальное значение" следующую конструкцию:

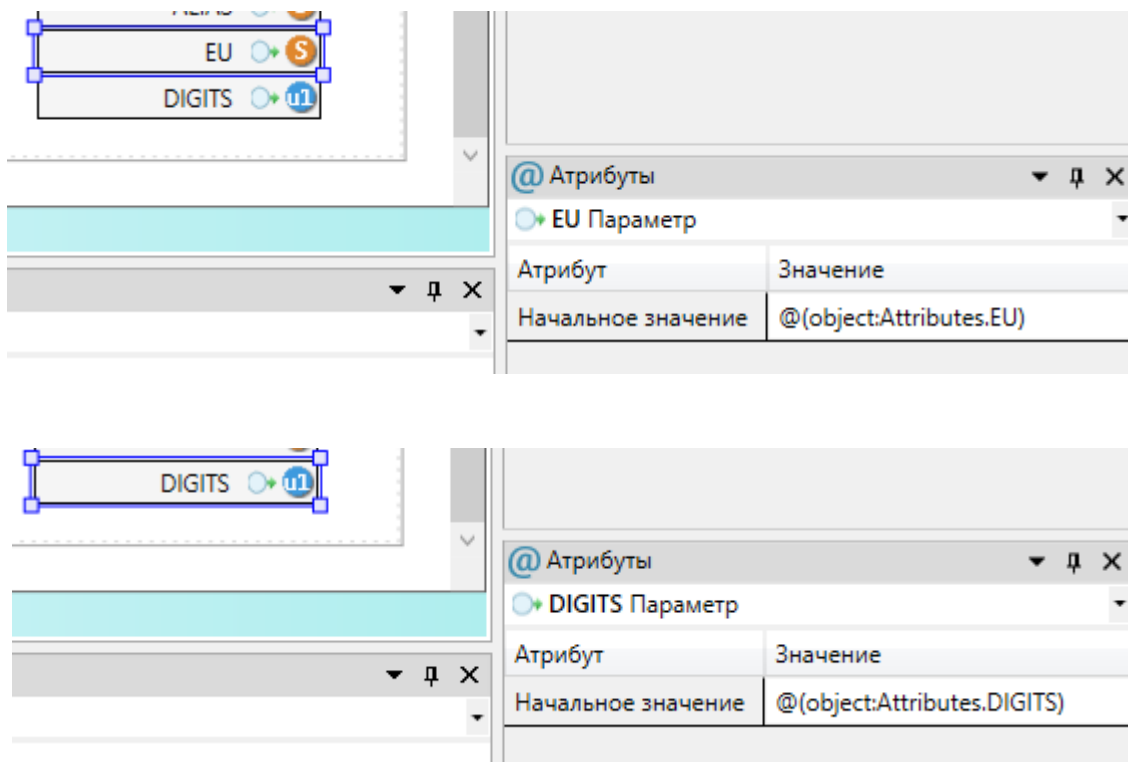


@(object:Attributes.ALIAS)

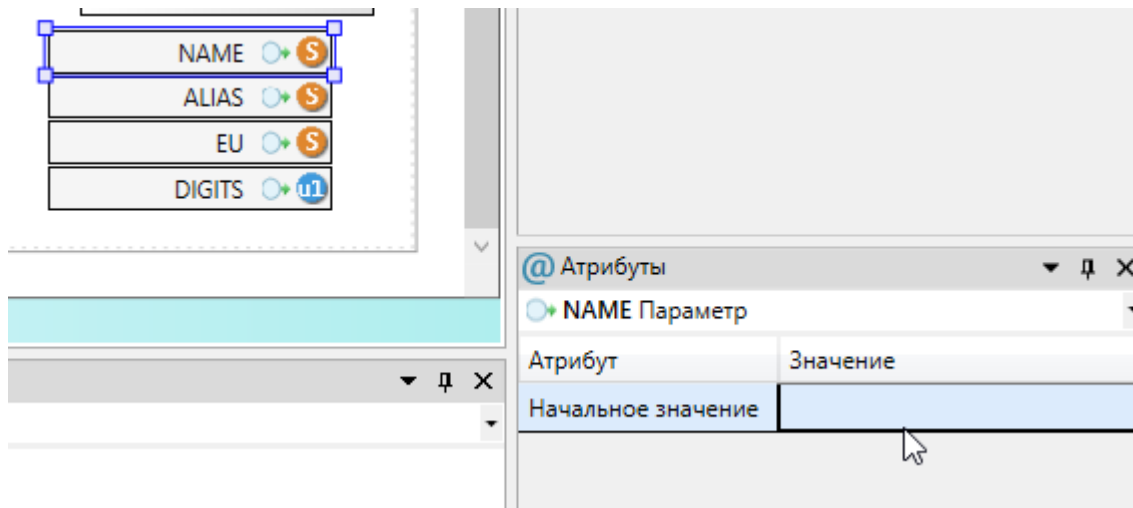
где Attributes - название пространства имен, где содержится атрибут, а ALIAS - название атрибута.



14. Аналогичным образом добавьте ссылку на атрибут для параметров "Единицы измерения" и "Количество знаков после запятой".



15. Выделите параметр "NAME" и в окне "Атрибуты" добавьте атрибут "Начальное значение".

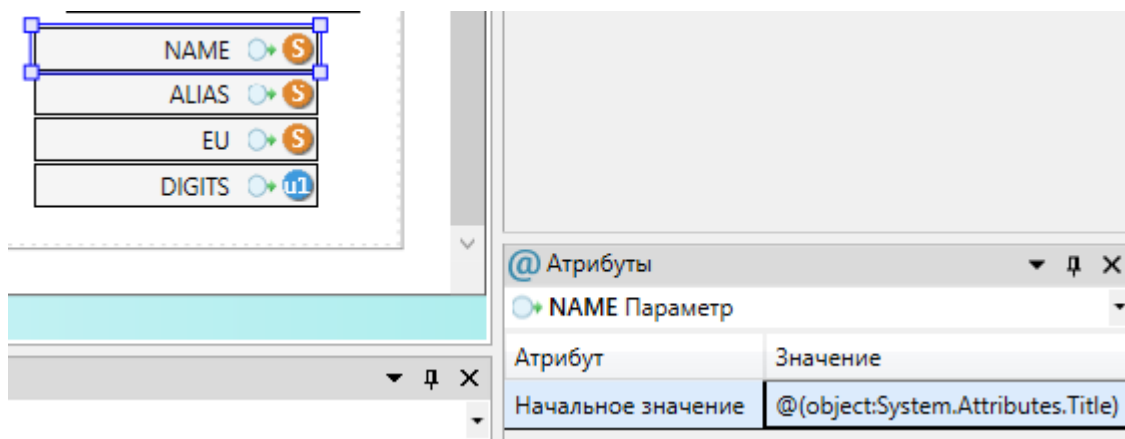


16. Чтобы добавить ссылку на системный атрибут, используйте следующую конструкцию:



`@(object:System.Attributes.Title)`

где System - название библиотеки, Attributes - название пространства имен, где содержится атрибут, а Title - название атрибута.

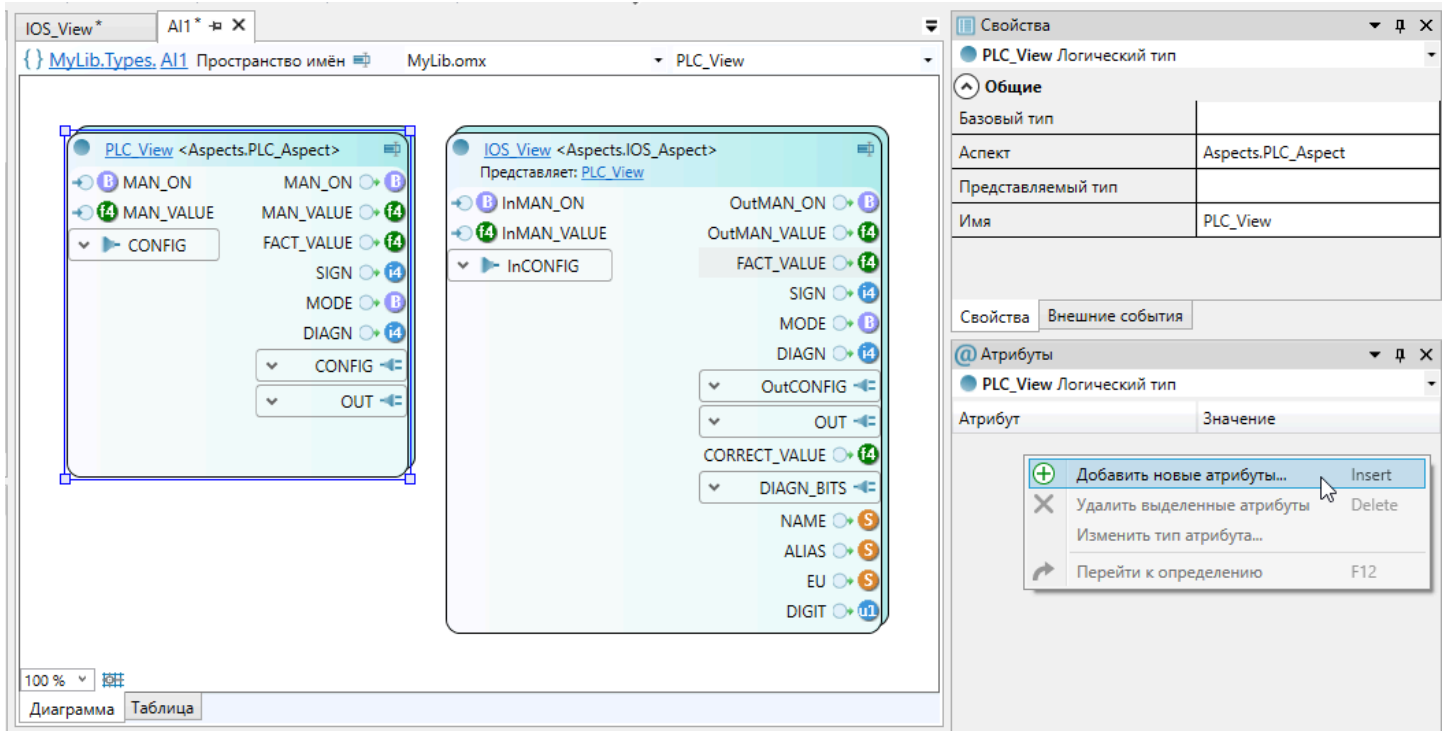


Таким образом, начальные значения добавленных параметров будут ссылаться на значения атрибутов.



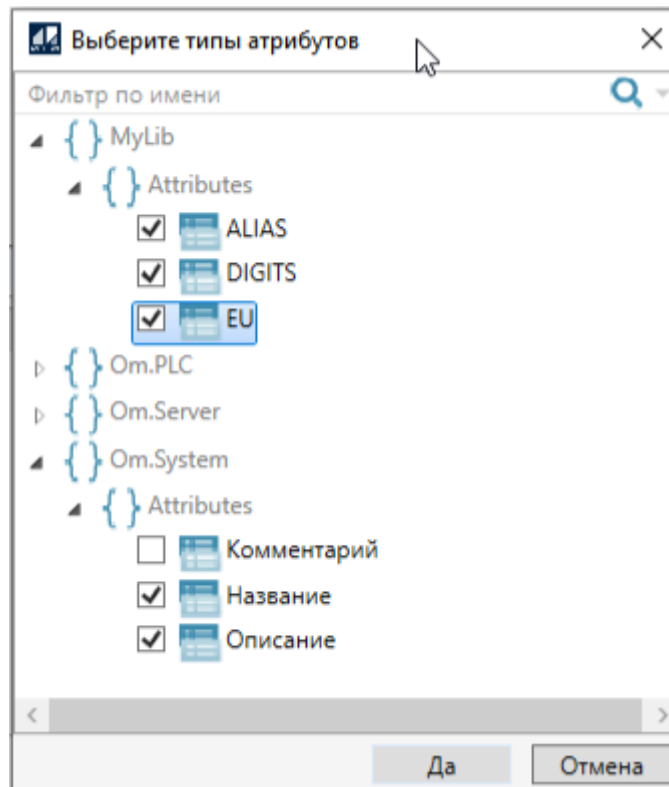
Для того, чтобы значения добавленных параметров не были пустыми, необходимо задать начальные значения у атрибутов.

17. Чтобы задать начальные значения атрибутов, выделите уровень представления "PLC\_View" типа "AI1" в обозревателе решени. В окне атрибуты нажмите ПКМ по свободному полю и из контекстного меню выберите команду "Добавить новые атрибуты...".



18. В открывшемся окне из раздела "MyLib" выберите атрибуты "ALIAS", "DIGITS" и "EU", а из раздела "Om.System" выберите атрибуты "Название" и "Описание".





19. Задайте начальные значения для добавленных атрибутов.

Атрибут	Значение
Название	AI
Описание	KKS
ALIAS	O =
DIGITS	2
EU	eu

Таким образом мы задали начальные значения атрибутов для созданного типа аналогового датчика. Теперь у всех экземпляров типа "AI1" будут заданы данные начальные значения, которые можно будет переопределить у экземпляра.

## 1.2.3.4.4. Генерация событий

Генерация событий настраивается для объектов, размещаемых в Astra.Server.

Чтобы настроить генерацию событий у параметров, выполните следующие действия:

1. Выберите параметр "SIGN", при изменении значений которого будут генерироваться события, и в окне "События" установите флаг "Генерировать события".

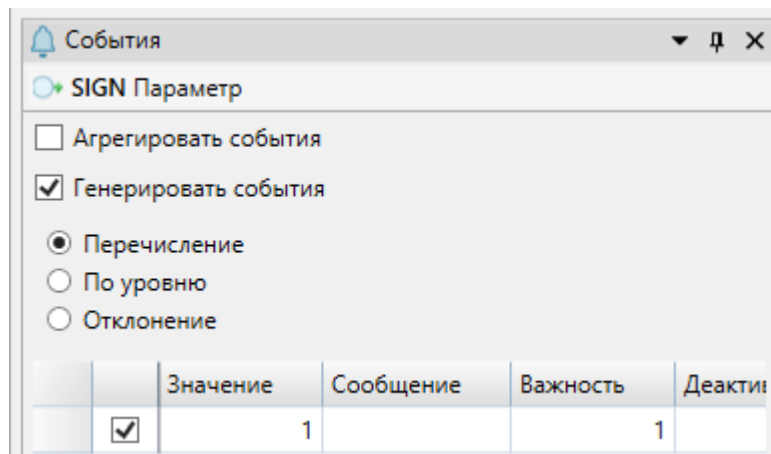


Если вкладка "События" не отображается, выберите команду в главном меню "Вид -> События".

The screenshot shows the software interface for configuring the 'SIGN' parameter. The main window displays a diagram with various input and output parameters. The 'SIGN' parameter is highlighted in blue. The 'Properties' window on the right shows the 'Events' tab, where the 'Generate events' checkbox is checked, and the event type is set to 'Enumeration'.

Свойства	
SIGN Параметр	
Общие	
Уровень доступа	публичный
Область доступа	глобальная
Направление	выход
Тип	int4
Имя	SIGN
Свойства Внешние события	
События	
SIGN Параметр	
<input type="checkbox"/>	Агрегировать события
<input checked="" type="checkbox"/>	Генерировать события
<input type="radio"/>	Перечисление
<input type="radio"/>	По уровню
<input type="radio"/>	Отклонение
	Подусловие
	Значение
	Сообщение
	Важ-

2. В окне "События" выберите тип генерации событий "Перечисление".

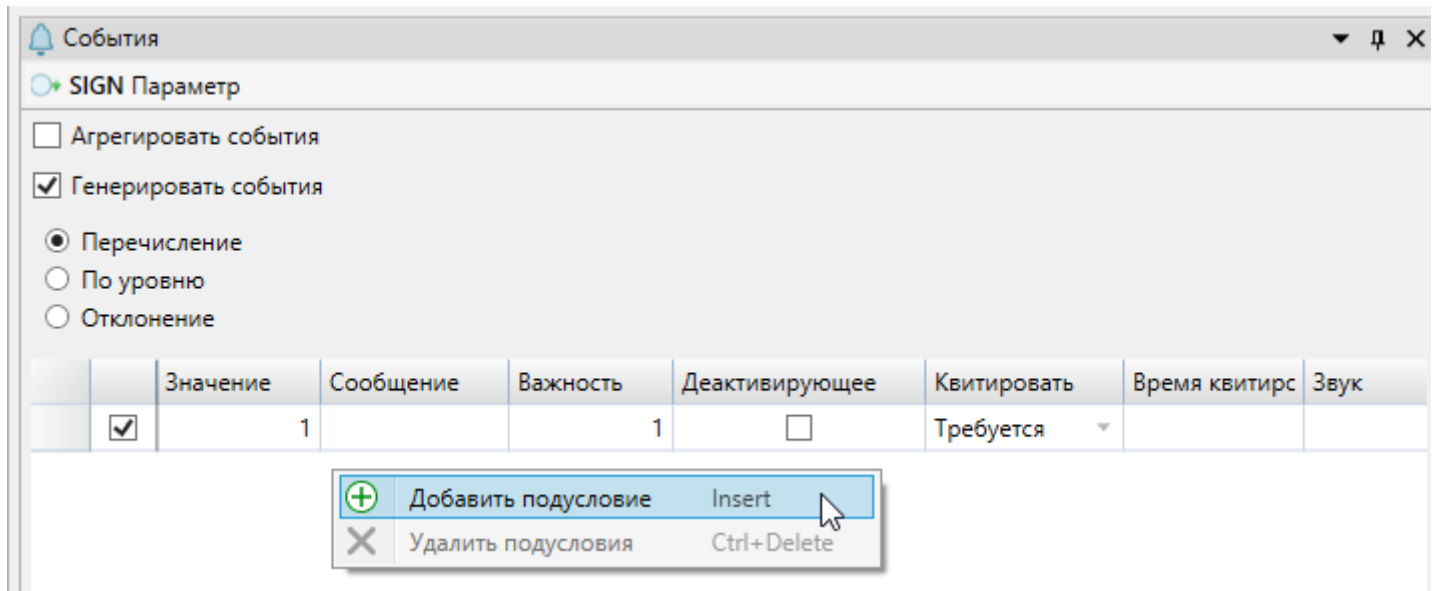


Если флаг не установлен, события в объекте генерироваться не будут.

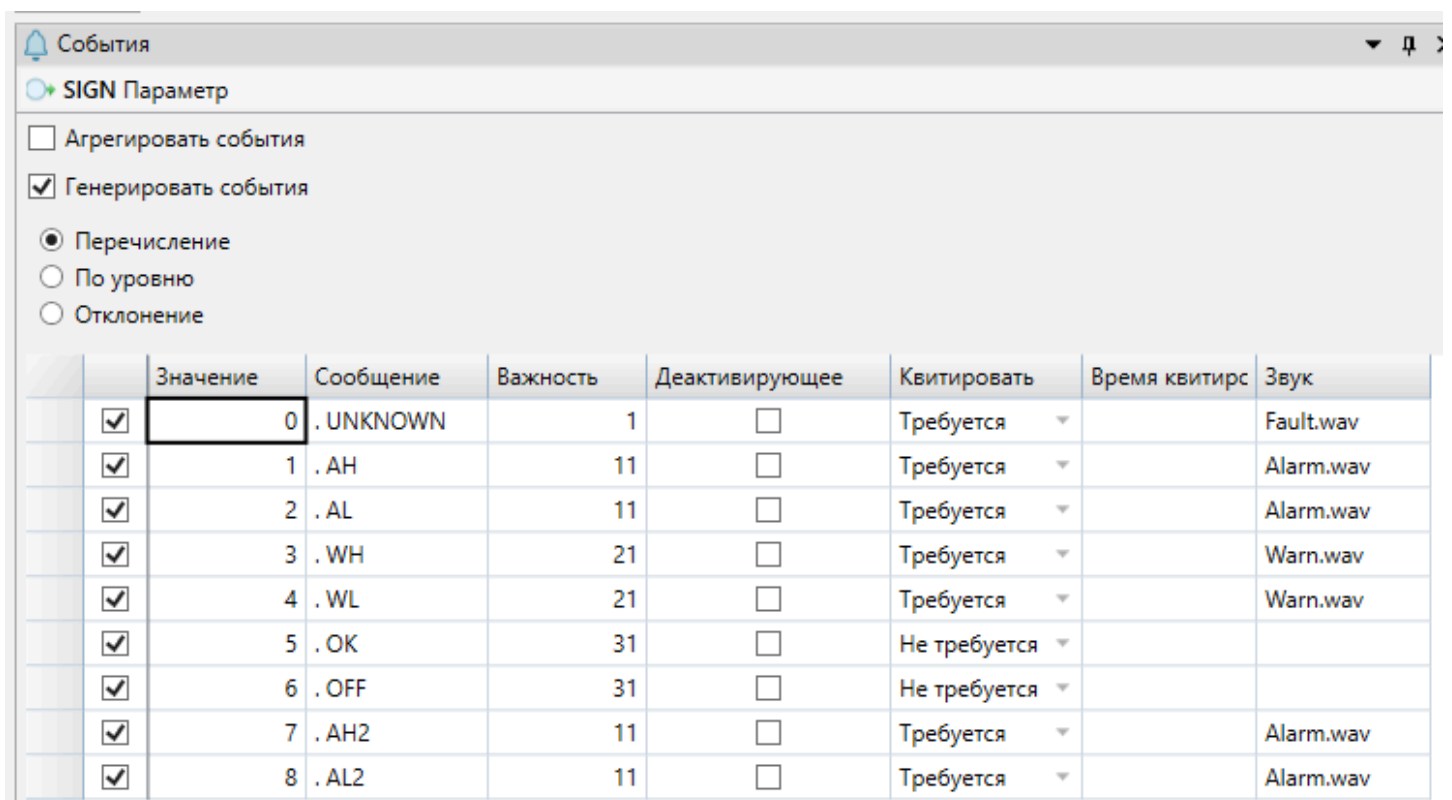
Доступные типы условий зависят от типа сигнала.

Тип сигнала	Типы условий
bool	Дискретный (не требуется выбирать).
int1, uint1, int2, uint2, int4, uint4, int8, uint8	<ul style="list-style-type: none"> <li>&gt; Перечисление</li> <li>&gt; По уровню</li> <li>&gt; Отклонение</li> </ul>
float, double	<ul style="list-style-type: none"> <li>&gt; По уровню</li> <li>&gt; Отклонение</li> </ul>
string	Динамическое (не требуется выбирать).

3. Укажите параметры подусловий события. Чтобы добавить подусловие, в контекстном меню выберите команду "Добавить подусловие" (горячая клавиша «Insert»).



4. Добавьте следующие подусловия и заполните таблицу:



Заголовок столбца	Описание
Значение	Значение, используемое в подусловии. У нечисловых типов (bool, string) отсутствует.

Сообщение	Текст сообщения при выполнении подусловия. При генерации события имеет формат: <b>[имя экземпляра типа][сообщение]</b> .
Важность	Важность события при выполнении подусловия
Деактивирующее	Срабатывание деактивирующего подусловия означает деактивацию условия
Квитировать	Требование квитировать событие: <ul style="list-style-type: none"> <li>› true – требует квитирования</li> <li>› false – не требует квитирования</li> </ul>
Время квитирования	Время, после которого потребуется повторное квитирование
Звук	Имя звукового файла, который будет проигрываться при генерации события. Звуковой файл должен иметь расширение *.wav.

5. Рассмотрим задание генерации события для режима работы датчика. Выделите параметр "MODE" типа bool и перейдите во вкладку "События".

The screenshot shows the 'IOS\_View' application window. On the left, a 'PLC\_View' diagram is visible, containing various parameters like 'MAN\_ON', 'MAN\_VALUE', 'FACT\_VALUE', 'SIGN', 'MODE', 'DIAGN', and 'OUT'. The 'MODE' parameter is highlighted with a blue selection box. On the right, the 'Свойства' (Properties) panel is open, showing the 'MODE' parameter's details. Under the 'События' (Events) tab, the 'MODE' parameter is listed, and the 'Генерировать события' (Generate events) checkbox is checked.

6. Установите флаг "Генерировать события".



У параметра типа bool имеется всего два подусловия: "Установка" и "Снятие".

Подусловие "Установка" срабатывает когда значение параметра true, а подусловие "Снятие" срабатывает когда значение параметра false.

Чтобы активировать подусловие, необходимо установить флаг в соответствующем столбце.

	Подусловие	Сообщение	Важность	Деактивирующее	Квитировать	Время квитирс	Звук
<input checked="" type="checkbox"/>	Установка	. Режим Подмены	31	<input type="checkbox"/>	Требуется		
<input checked="" type="checkbox"/>	Снятие	. Режим Штатный	31	<input type="checkbox"/>	Не требуется		

После настройки генерации событий у сигнала появится атрибут "События", значение которого будет содержать настроенные параметры генерации.

## 1.2.3.4.5. Сохранение в историю



Настройки истории не влияют на сигналы, которые находятся не в Astra.Server.

Чтобы сохранять значения параметров типа в базу данных Astra.Historian, выполните следующие действия:

1. Выделите параметр "OUT.VALUE" и во вкладке "История" установите флаг "Сохранять историю".



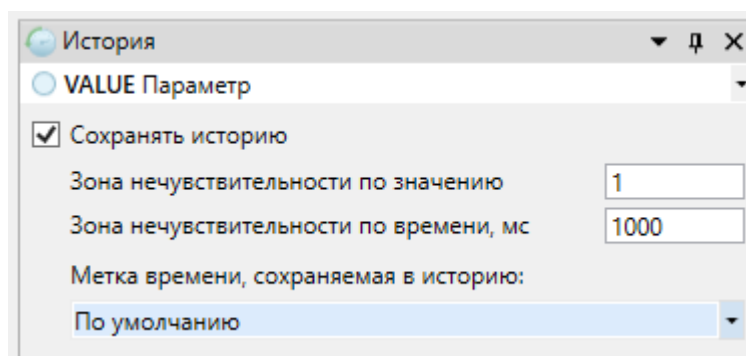
Если данная вкладка не отображается, ее можно добавить, воспользовавшись командой "Вид" -> "История".

The screenshot shows the Astra.Historian software interface. On the left, a ladder logic diagram is displayed with various input and output modules. The central module is labeled "\_PLC\_View" and contains several parameters: MAN\_ON, MAN\_VALUE, FACT\_VALUE, SIGN, MODE, DIAGN, and OUT. The OUT module is highlighted with a blue border. On the right, the 'Свойства' (Properties) panel is open, showing the 'История' (History) tab. The 'VALUE Параметр' (VALUE Parameter) is selected, and the 'Сохранять историю' (Save history) checkbox is checked. The 'Зона нечувствительности по значению' (Value deadband) and 'Зона нечувствительности по времени, мс' (Time deadband, ms) fields are visible, along with a dropdown for 'Метка времени, сохраняемая в историю:' (Time tag saved to history:). The 'Тип' (Type) is set to 'float' and the 'Имя' (Name) is 'VALUE'.



События настраивать не нужно. Все события сохраняются в историю.

2. При необходимости задайте значения зон нечувствительности по значению и времени. Данные настройки являются опциональными.



Зоны нечувствительности необходимо использовать, чтобы уменьшить количество сохраняемых значений в базу данных. Они позволяют не записывать в историю значения сигналов, которые изменяются слишком часто или незначительно.

Фильтр	Описание
Зона нечувствительности по значению	Значение не будет сохранено, если оно отличается от предыдущего сохранённого меньше, чем на значение фильтра.
Зона нечувствительности по времени, мс	Интервал времени после сохранения значения, в течение которого новые значения сохраняться не будут.



Значение не будет сохранено в историю, если оно отсеяно хотя бы одним фильтром.

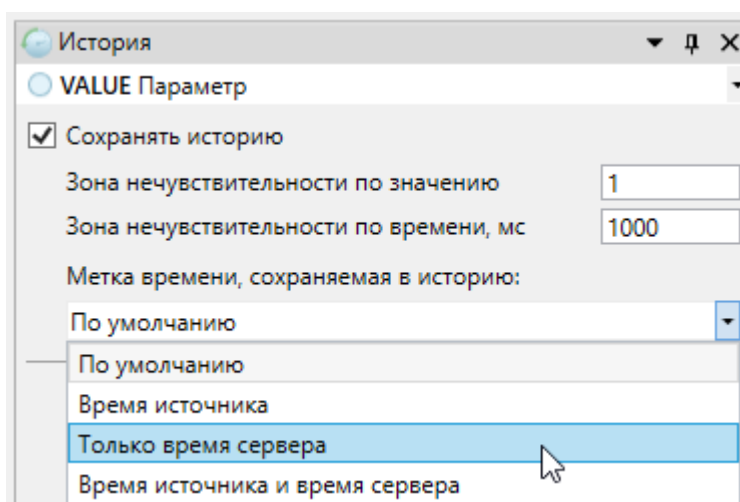
## Пример

Пусть датчик температуры отправляет текущую температуру раз в секунду. Чтобы не сохранять каждое полученное значение, установите в фильтре "Зона нечувствительности по значению" значение 1. Теперь сохраняться будут только те значения, когда температура увеличилась или уменьшилась на 1 градус и более от предыдущего сохранённого значения.



3. Из выпадающего списка "Метка времени, сохраняемая в историю" выберите "Только время сервера" (опционально).

Если выбрана данная опция, вместо метки времени, полученной от устройства, в историю будет сохраняться время Astra.Server в момент, когда значение было записано в сигнал.



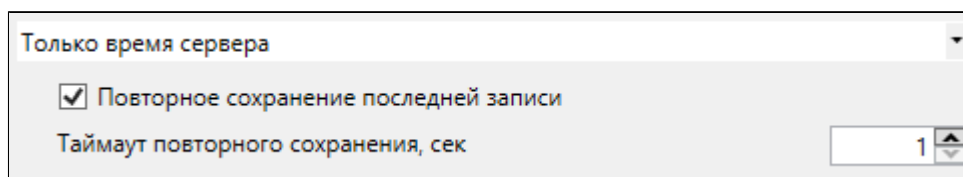
Это необходимо, если значения приходят в Astra.Server с запозданием, например из-за задержек связи. В этом случае нужно установить флаг, чтобы в историю сохранилась не метка времени, когда значение возникло, а метка времени, когда значение было получено и обработано.

Для получения метки времени с устройства выберите опцию "По умолчанию" или "Время источника".

Если необходимо получать значение метки времени с источника и с сервера одновременно, выберите из выпадающего списка "Время источника и время сервера", но при данной опции размер записи данных в Astra.Historian увеличится.

4. При выборе опций "Только время сервера" или "Время источника и время сервера", отобразится флаг "Повторное сохранение последней записи". Установите флаг и выберите время таймаута повторного сохранения, если

необходимо записывать значение сигнала в Astra.Historian, через указанное время, даже если значение параметра не изменилось.



Только время сервера

Повторное сохранение последней записи

Таймаут повторного сохранения, сек

## 1.2.3.4.6. Добавление пользовательского столбца в Astra.HMI.Alarms

Для добавления пользовательского столбца в Astra.HMI.Alarms выполните следующие действия.

1. Откройте проект Astra.AStudio и создайте тип атрибута из панели элементов.

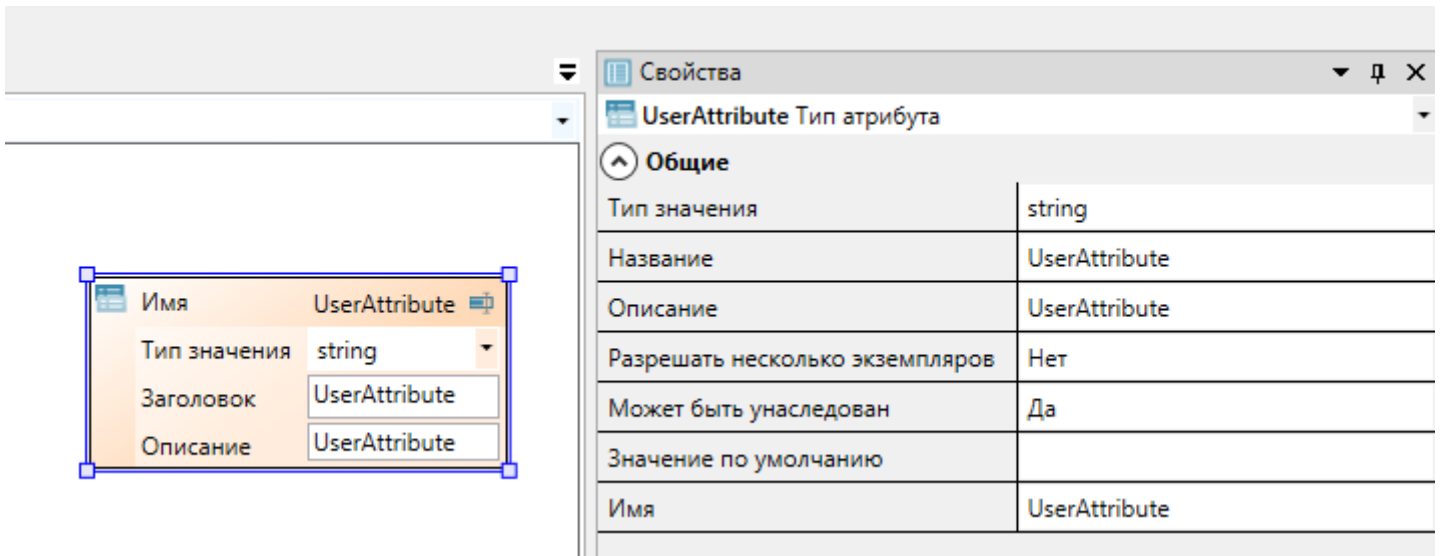
The screenshot displays the Astra.AStudio interface. In the top-left corner, there is a tree view with nodes for 'PLC', 'Server', and 'System'. The main workspace shows a dialog box for creating a new attribute type, titled 'AttributeType'. The dialog has the following fields:

Имя	AttributeType
Тип значения	void
Заголовок	
Описание	

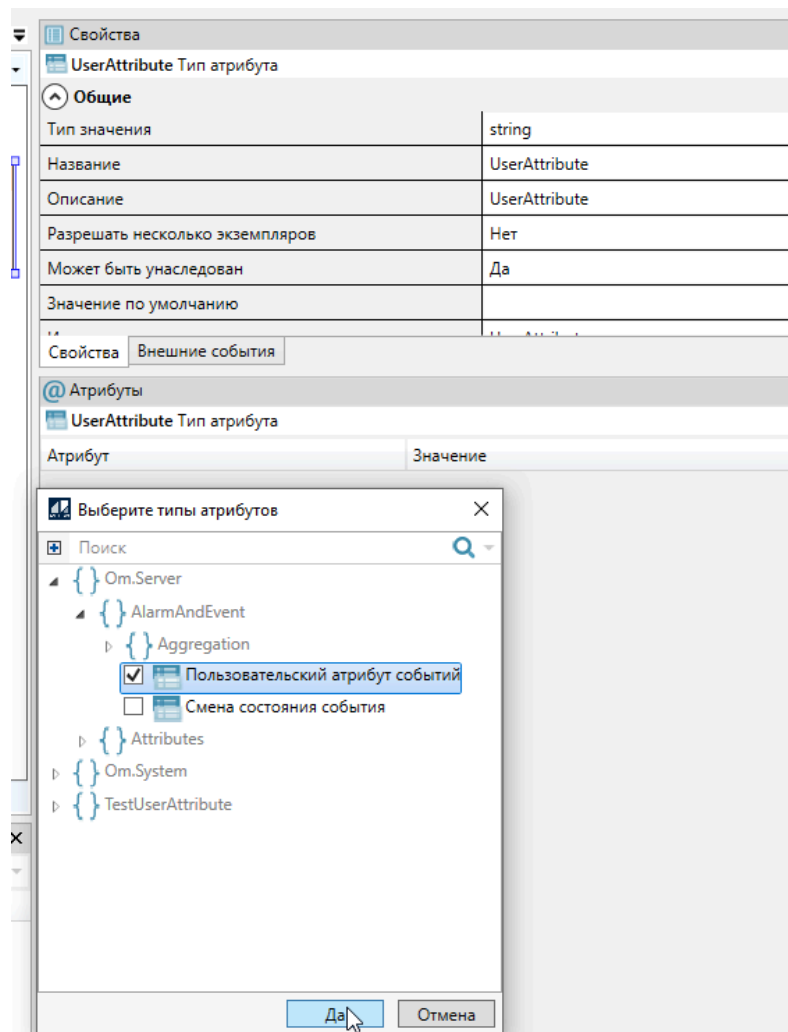
Below the dialog, there are tabs for '100 %', 'Диаграмма', and 'Таблица'. The 'Таблица' tab is active. At the bottom left, the 'Панель элементов' (Element Panel) is visible, with a search bar and a list of elements: 'Настройки', 'Описания устройств', 'Тип атрибута' (highlighted), 'Ссылка на модуль', and 'Пространство имён'. At the bottom right, the 'Журнал' (Log) window shows the following messages:

- Начало компиляции TestUserAttribute -----
- Компиляция TestUserAttribute успешно завершена (длительность: 00:00:00)
- Начало построения TestUserAttribute -----
- Построение выходных файлов для AstraRegul...

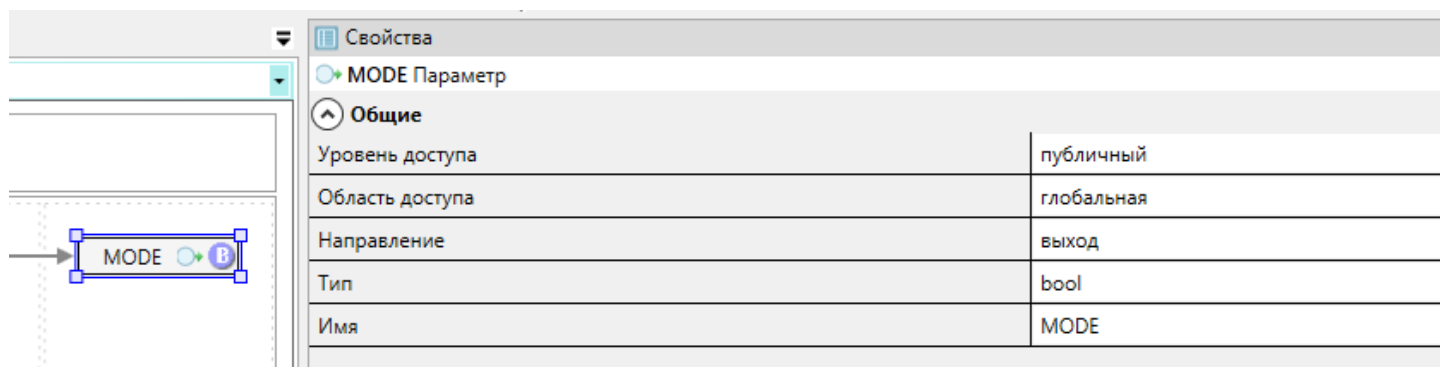
2. Заполните свойства у созданного типа атрибута, задайте имя, например, UserAttribute.



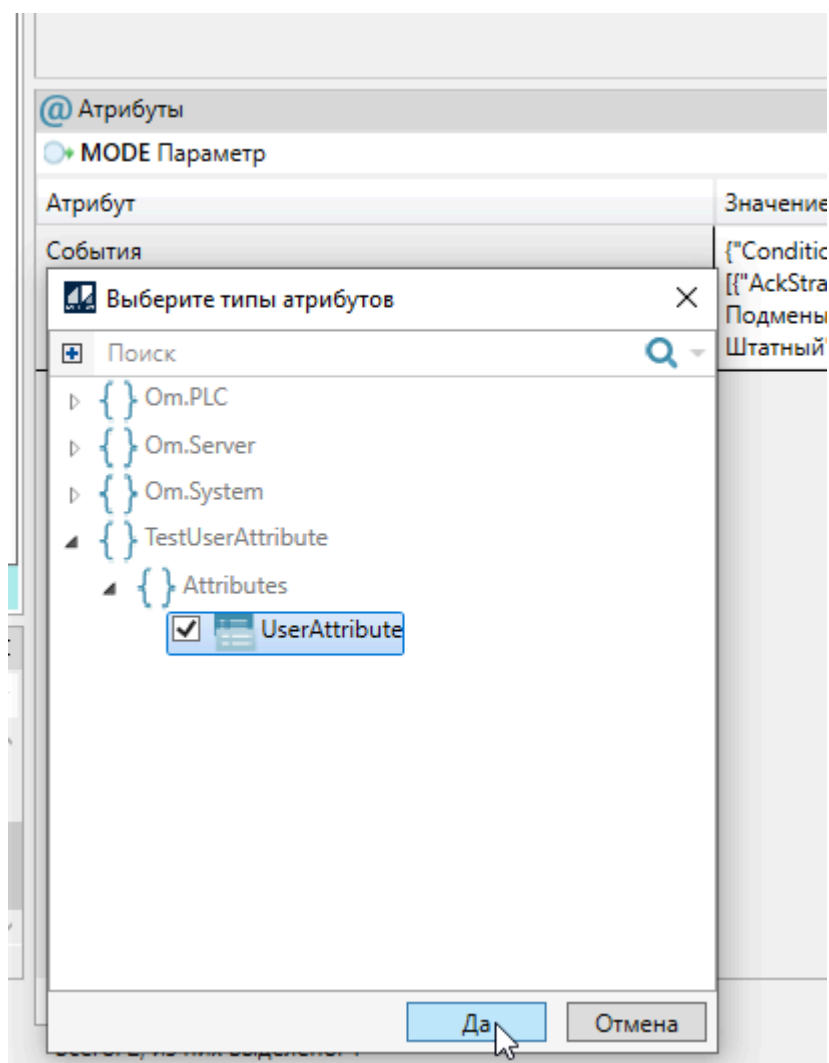
3. Во вкладке "Атрибуты" при помощи контекстного меню добавьте атрибут "Пользовательский атрибут событий".



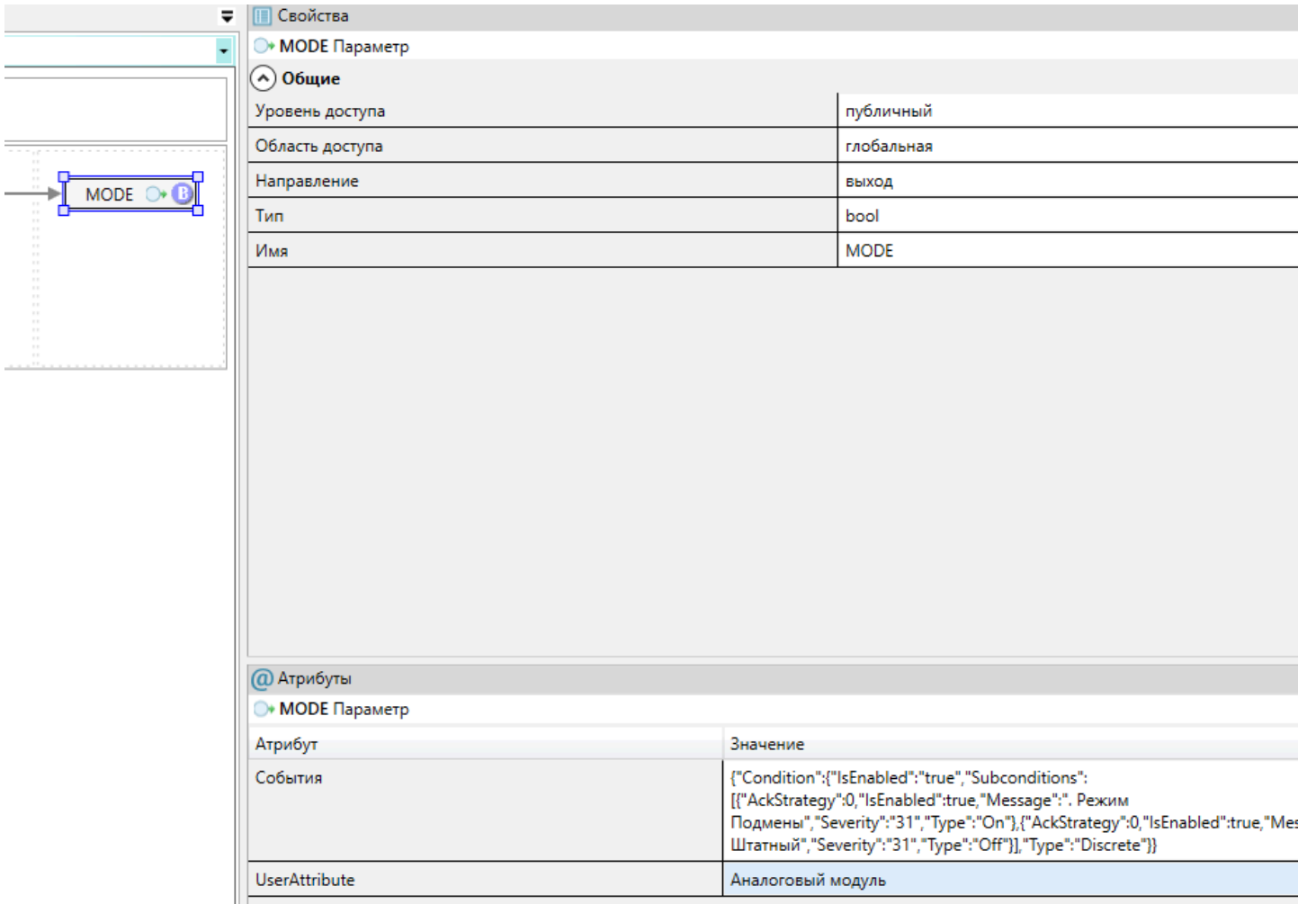
4. Перейдите к библиотечному типу AI1 в представление IOS\_View и выделите параметр MODE.



5. Перейдите на вкладку "Атрибуты" и добавьте пользовательский атрибут с помощью контекстного меню.



## 6. Задайте значение "Аналоговый модуль" созданному атрибуту.



The screenshot displays the configuration interface for a 'MODE' parameter. On the left, a 'MODE' parameter block is shown in a diagram. The main area is divided into two sections: 'Свойства' (Properties) and 'Атрибуты' (Attributes).

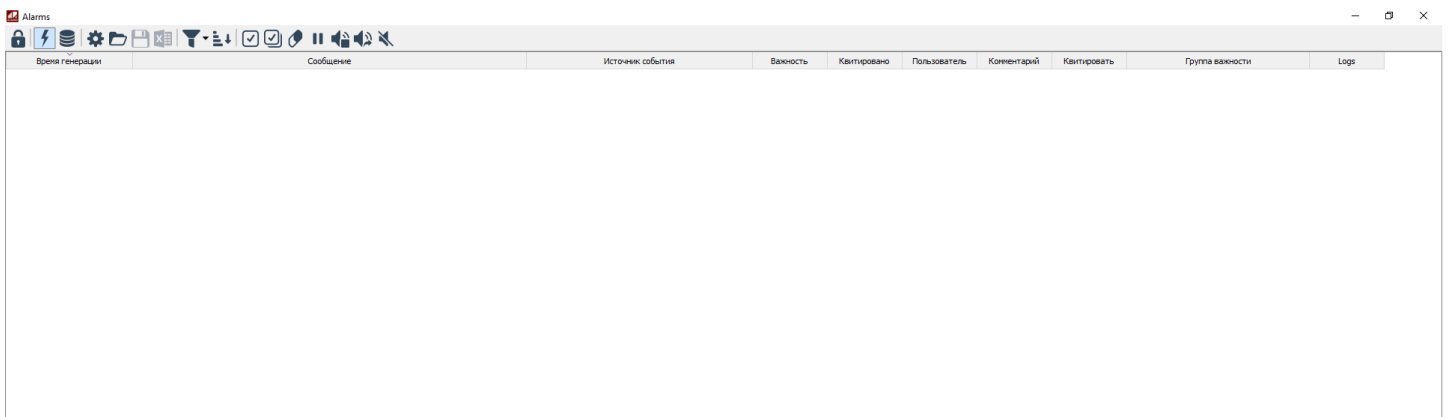
**Свойства (Properties):**

Свойство	Значение
Уровень доступа	публичный
Область доступа	глобальная
Направление	выход
Тип	bool
Имя	MODE

**Атрибуты (Attributes):**

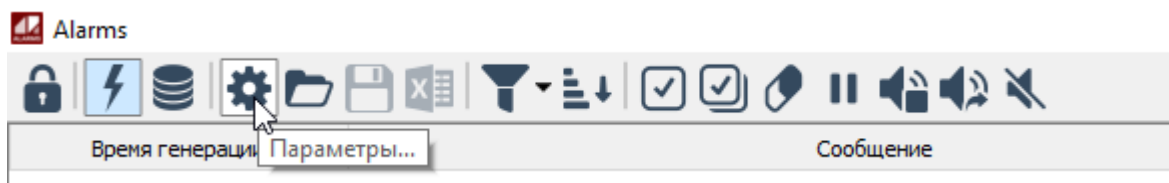
Атрибут	Значение
События	<pre>{"Condition":{"IsEnabled":"true","Subconditions":[{"AckStrategy":0,"IsEnabled":true,"Message":". Режим Подмены","Severity":"31","Type":"On"}, {"AckStrategy":0,"IsEnabled":true,"Message":"Штатный","Severity":"31","Type":"Off"}],"Type":"Discrete"}}</pre>
UserAttribute	Аналоговый модуль

## 7. После применения конфигурации проекта на Astra.Server, в которой используется данный библиотечный тип AI1, откройте Astra.HMI.Alarms.



The screenshot shows the Astra.HMI.Alarms application window. The title bar reads 'Alarms'. The interface includes a toolbar with various icons and a table with columns for alarm management. The table headers are: 'Время генерации', 'Сообщение', 'Источники событий', 'Важность', 'Категоризовано', 'Пользователь', 'Комментарий', 'Категоризовать', 'Группа важности', and 'Log'. The main area of the window is currently empty.

8. Откройте окно "Параметры" нажав на кнопку Параметры на панели инструментов.



9. В открывшемся окне перейдите на вкладку "Вид таблицы".

The image shows the 'Parameters' dialog box with the 'Table View' tab selected. The dialog has a sidebar on the left with various settings categories, and a main area with a table and formatting options.

**Параметры**

Важность событий  
Настройка событий  
**Вид таблицы**  
Панель инструментов  
Комментарии  
Фильтры  
Экспорт  
Механизмы очистки  
Контекстное меню  
Печать

**Вид таблицы**

Описание	Заголовок	Показывать	Ширина	Формат времени
Время генерации	Время генерации	<input checked="" type="checkbox"/>	170	
Время срабатывания	Время срабатывания	<input type="checkbox"/>	145	
Время деактивации	Время деактивации	<input type="checkbox"/>	145	
Сообщение	Сообщение	<input checked="" type="checkbox"/>	526	
Источник события	Источник события	<input checked="" type="checkbox"/>	302	
Важность	Важность	<input checked="" type="checkbox"/>	100	
Квитировано	Квитировано	<input checked="" type="checkbox"/>	100	
Время квитирования	Время квитирования	<input type="checkbox"/>	130	
Пользователь	Пользователь	<input checked="" type="checkbox"/>	100	
Комментарий	Комментарий	<input checked="" type="checkbox"/>	100	
Активность	Активность	<input type="checkbox"/>	100	
Квитировать	Квитировать	<input checked="" type="checkbox"/>	100	
Качество сигнала	Качество сигнала	<input type="checkbox"/>	100	
Значение сигнала	Значение сигнала	<input type="checkbox"/>	100	
Группа важности	Группа важности	<input checked="" type="checkbox"/>	244	
Logs	Logs	<input checked="" type="checkbox"/>	100	

Высота заголовка: 23      Шрифт заголовка: 8      Ж К Ц  
Высота строки: 25      Шрифт строки: 8      Ж К Ц

Вверх  
Вниз  
Добавить  
Удалить  
Изменить

OK      Отмена      Применить

10. В правой части окна выберите кнопку "Добавить".

Параметры

Вид таблицы

Описание	Заголовок	Показывать	Ширина	Формат времени
Время генерации	Время генерации	<input checked="" type="checkbox"/>	170	
Время срабатывания	Время срабатывания	<input type="checkbox"/>	145	
Время деактивации	Время деактивации	<input type="checkbox"/>	145	
Сообщение	Сообщение	<input checked="" type="checkbox"/>	526	
Источник события	Источник события	<input checked="" type="checkbox"/>	302	
Важность	Важность	<input type="checkbox"/>	100	
Квнтировано	Квнтировано	<input type="checkbox"/>	100	
Время квнтирования	Время квнтирования	<input type="checkbox"/>	130	
Пользователь	Пользователь	<input type="checkbox"/>	100	
Комментарий	Комментарий	<input type="checkbox"/>	100	
Активность	Активность	<input type="checkbox"/>	100	
Квнтировать	Квнтировать	<input type="checkbox"/>	100	
Качество сигнала	Качество сигнала	<input type="checkbox"/>	100	
Значение сигнала	Значение сигнала	<input type="checkbox"/>	100	
Группа важности	Группа важности	<input type="checkbox"/>	244	
Logs	Logs	<input checked="" type="checkbox"/>	100	

Важность событий  
Настройка событий  
Вид таблицы  
Панель инструментов  
Комментарии  
Фильтры  
Экспорт  
Механизмы очистки  
Контекстное меню  
Печать

Вверх  
Вниз  
Добавить  
Удалить  
Изменить

Новый столбец

Идентификатор

Описание

Тип  
bool

Формат времени

OK Отмена

Высота заголовка 23 Шрифт заголовка 8 Ж К Ч  
Высота строки 25 Шрифт строки 8 Ж К Ч

OK Отмена Применить



11. В открывшемся диалоговом окне в столбце "Идентификатор" введите имя пользовательского атрибута, например, UserAttribute.

ывания	Время срабатывания	<input type="checkbox"/>	145	
вазии	Время деактивации	<input type="checkbox"/>	145	
	Сообщение	<input checked="" type="checkbox"/>	526	
ятия	Источник события	<input checked="" type="checkbox"/>	302	
	Важ		100	
	Кви		100	
ывания	Вре		130	
	Пол		100	
	Ком		100	
	Акт		100	
	Кви		100	
чала	Кач		100	
чала	Знач		100	
сти	Гру		244	
	Logs	<input checked="" type="checkbox"/>	100	

Новый столбец

Идентификатор  
UserAttribute

Описание

Тип  
bool

Формат времени

OK Отмена

12. В поле "Описание" введите описание атрибута, например, Тип модуля.

Изменение столбца

Идентификатор  
UserAttribute

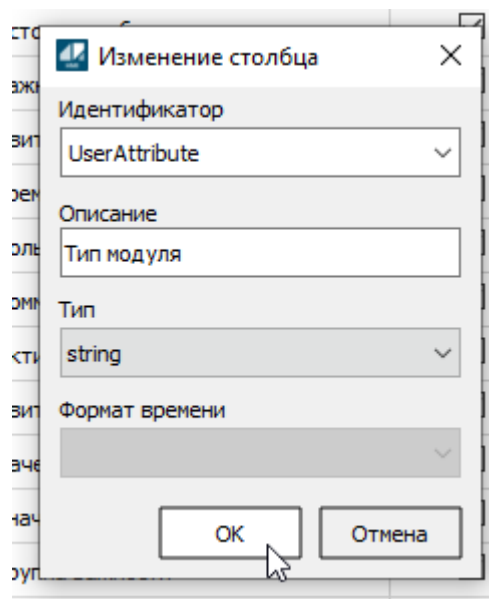
Описание  
Тип модуля

Тип  
bool

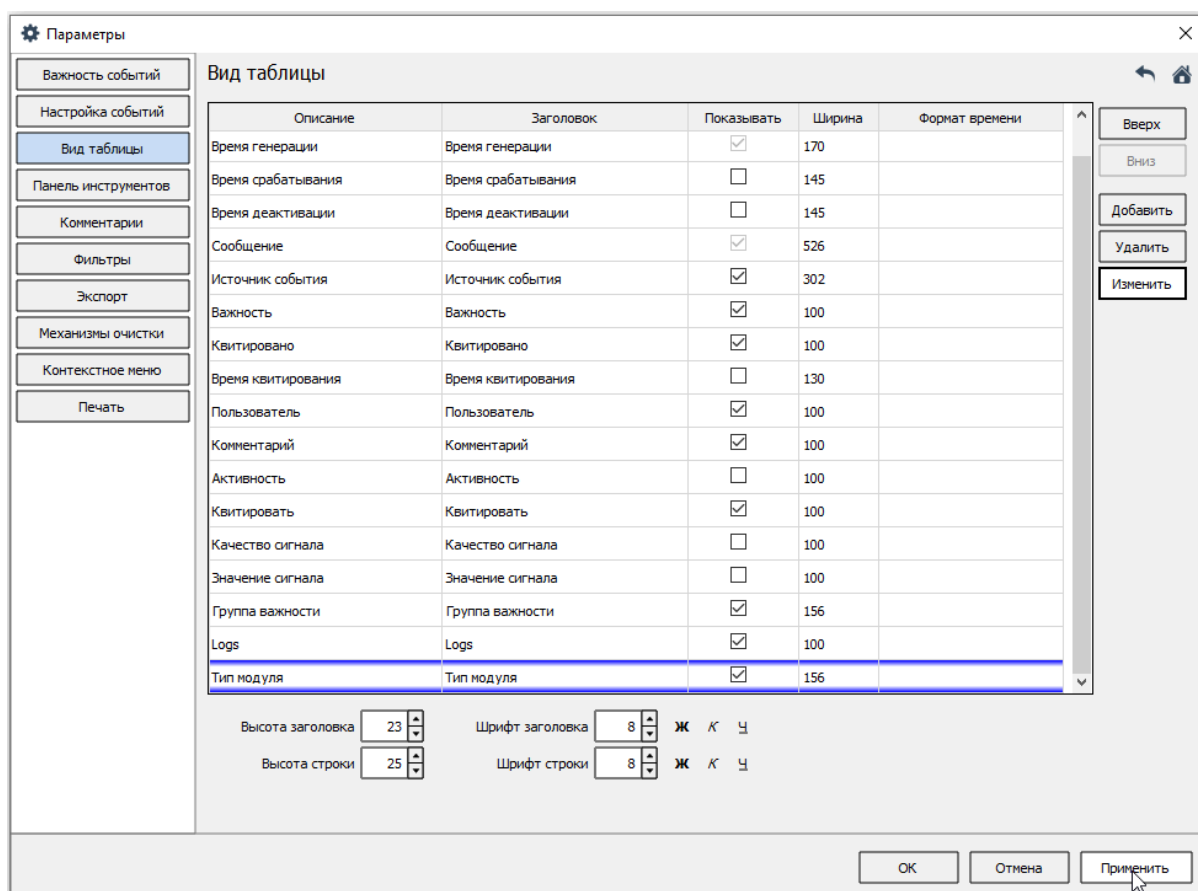
Формат времени

OK Отмена

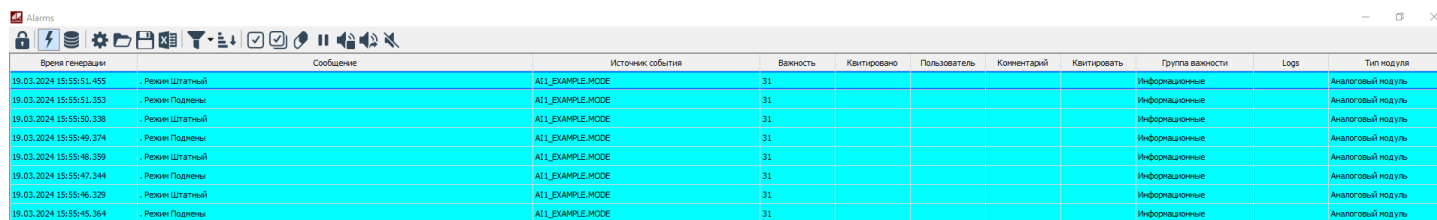
13. Из выпадающего списка выберите тип пользовательского атрибута типа string. И нажмите кнопку ОК.



14. Определите ширину и порядок расположения пользовательского столбца и нажмите кнопку "Применить".



15. Теперь, при генерации события по параметру MODE в столбце "Тип ошибки" будет отображаться значение пользовательского атрибута UserAttribute.



The screenshot shows the 'Alarms' application window. The main area contains a table with the following columns: 'Время генерации', 'Сообщение', 'Источник события', 'Важность', 'Кэшировано', 'Пользователь', 'Комментарий', 'Кэшировать', 'Группа важности', 'Logg', and 'Тип модуля'. The table contains 8 rows of data, all with a 'Важность' of 31 and 'Тип модуля' of 'Аналоговый модуль'.

Время генерации	Сообщение	Источник события	Важность	Кэшировано	Пользователь	Комментарий	Кэшировать	Группа важности	Logg	Тип модуля
19.03.2024 15:55:51.455	Режим Штатный	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:51.353	Режим Подмены	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:50.338	Режим Штатный	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:49.374	Режим Подмены	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:48.359	Режим Штатный	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:47.344	Режим Подмены	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:46.329	Режим Штатный	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль
19.03.2024 15:55:45.364	Режим Подмены	A11_EXAMPLE.MODE	31					Информационные		Аналоговый модуль

Таким образом, можно создавать собственные пользовательские столбцы и добавлять их в журнал событий.

## 1.2.4. Разработка библиотеки мнемосимволов в Astra.HMI

**Мнемосимвол** – это графический образ объекта. Из мнемосимволов строится мнемосхема.

При разработке мнемосимволов применяются объекты графических примитивов (линии, прямоугольники, эллипсы и т.д.), элементы пользовательского интерфейса (кнопки, флаги, прогресс-бары, поля ввода и т.д.), а также встраиваемые компоненты (Astra.HMI.Trends, Astra.HMI.Alarms и другие).



В зависимости от состояния объекта графическое представление мнемосимвола может меняться динамически.

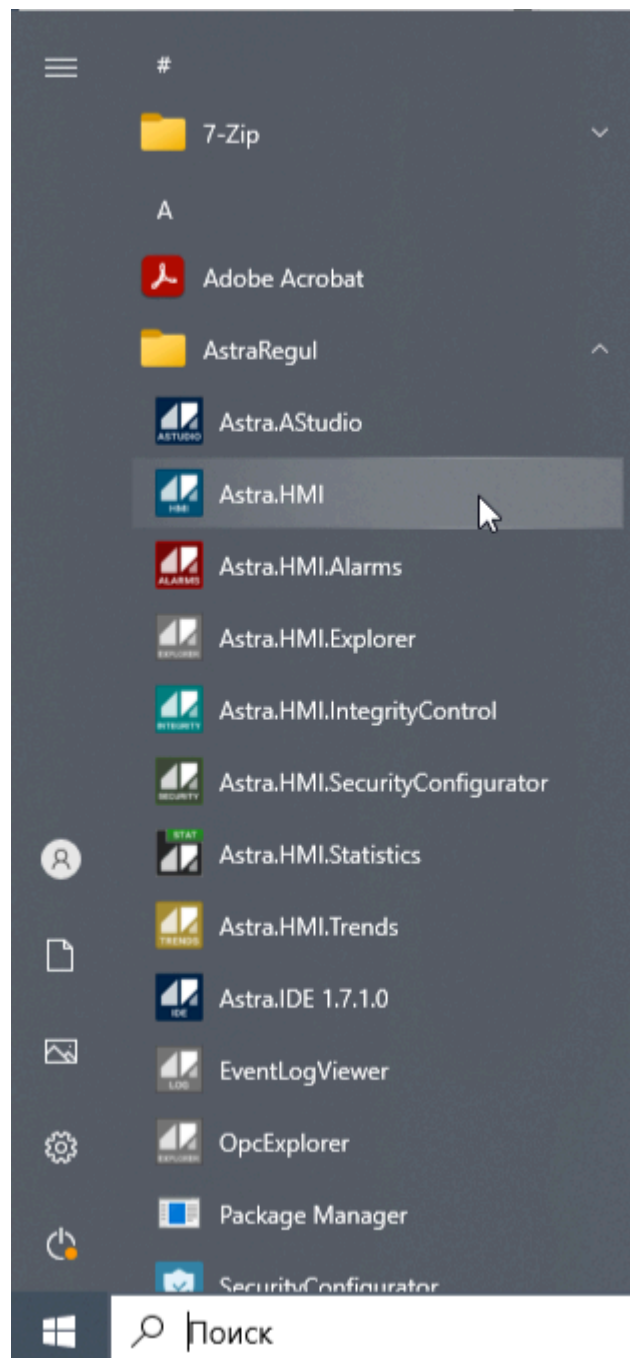
Для создания библиотеки мнемосимволов будем применять принципы объектно-ориентированного программирования.

**Объектно-ориентированное программирование (ООП)** — методология программирования, основанная на представлении программы в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

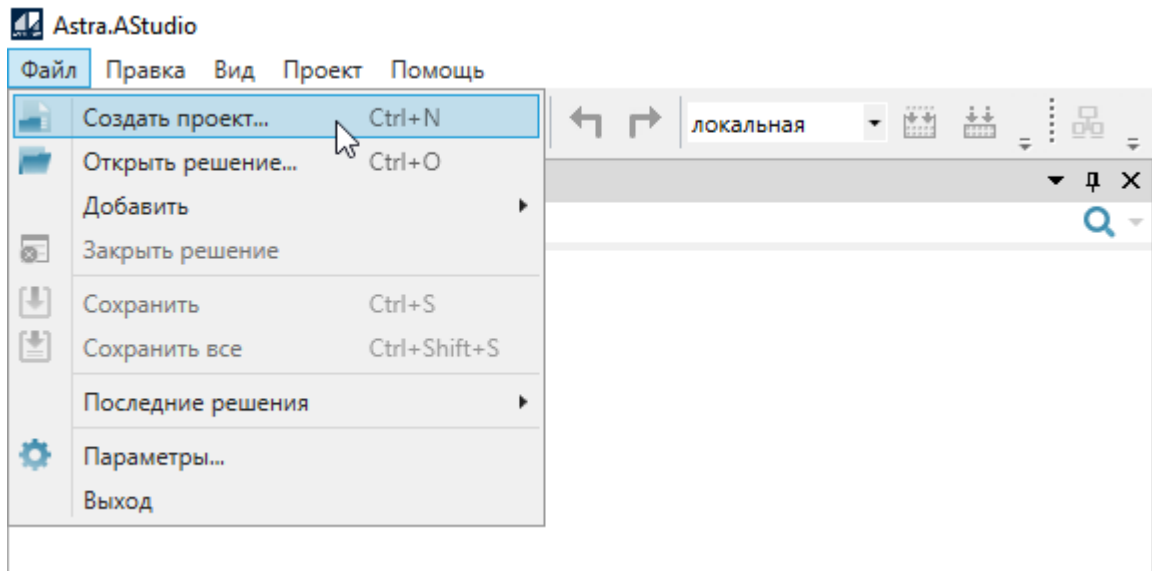
## 1.2.4.1. Создание проекта библиотеки

Чтобы создать новый проект библиотеки мнемосимволов, выполните следующие действия:

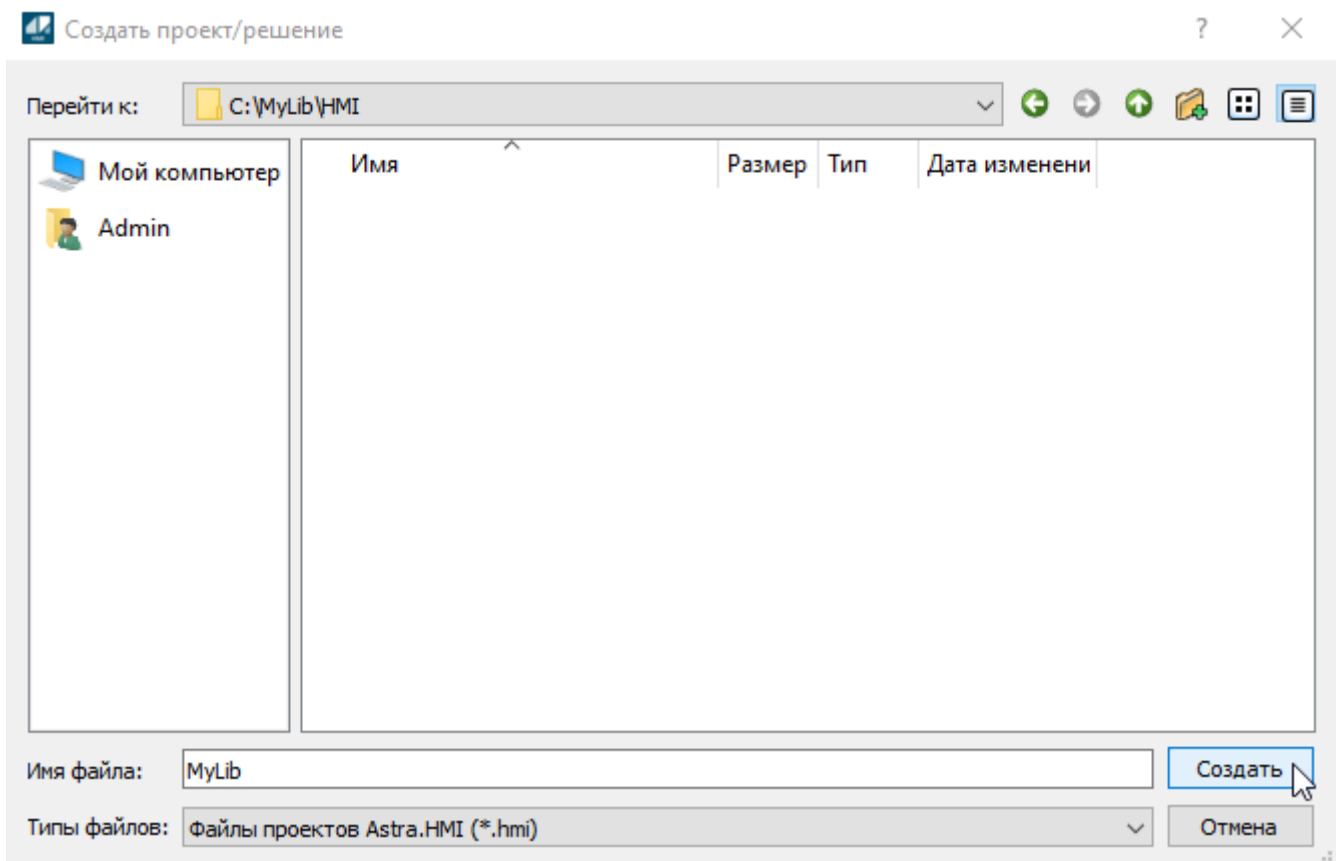
1. Запустите графический редактор Astra.HMI.



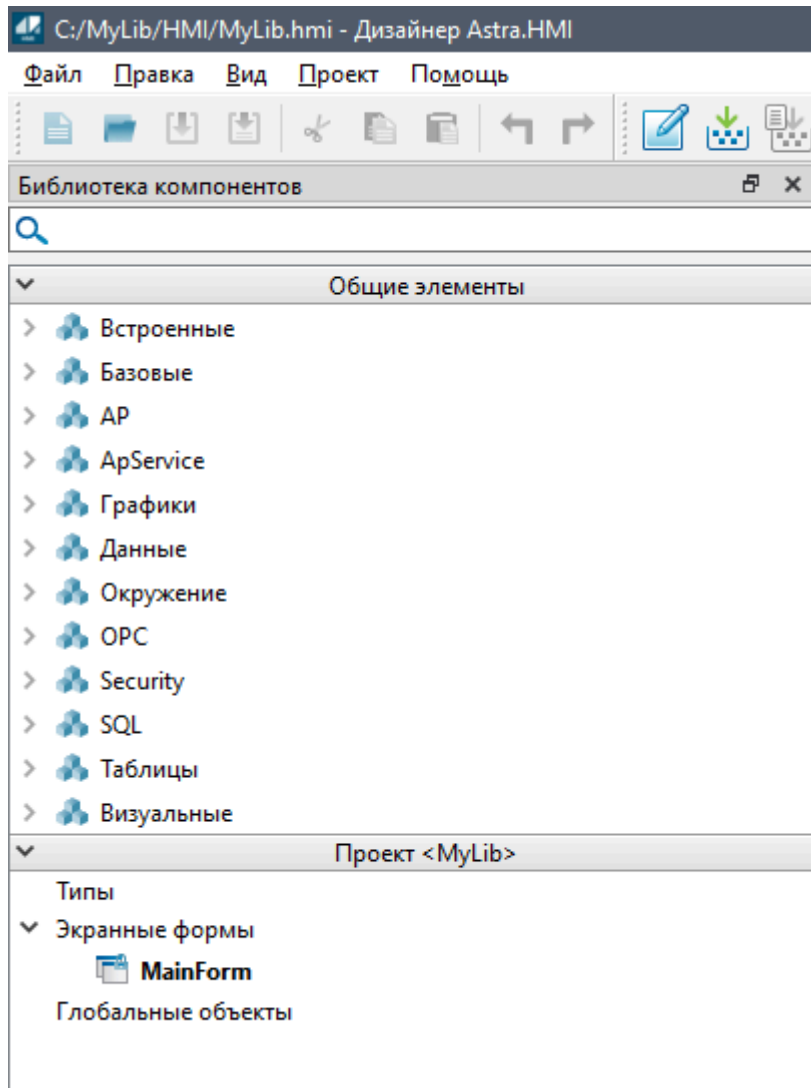
2. Во вкладке "Файл" главного меню выберите команду "Создать проект".



3. В открывшемся диалоговом окне укажите имя и каталог библиотеки и нажмите кнопку "Создать".



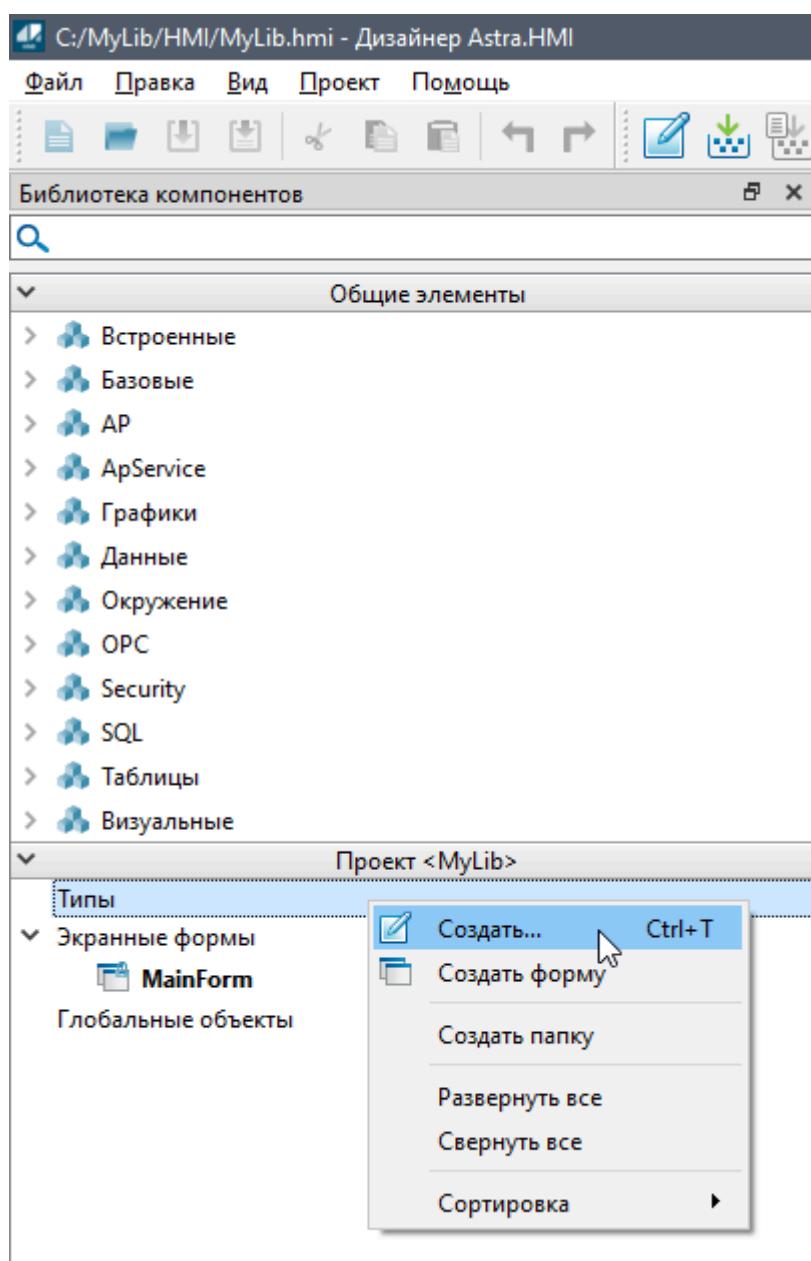
В Astra.HMI.Designer откроется новый проект.



## 1.2.4.2. Создание базового типа агрегаторов событий

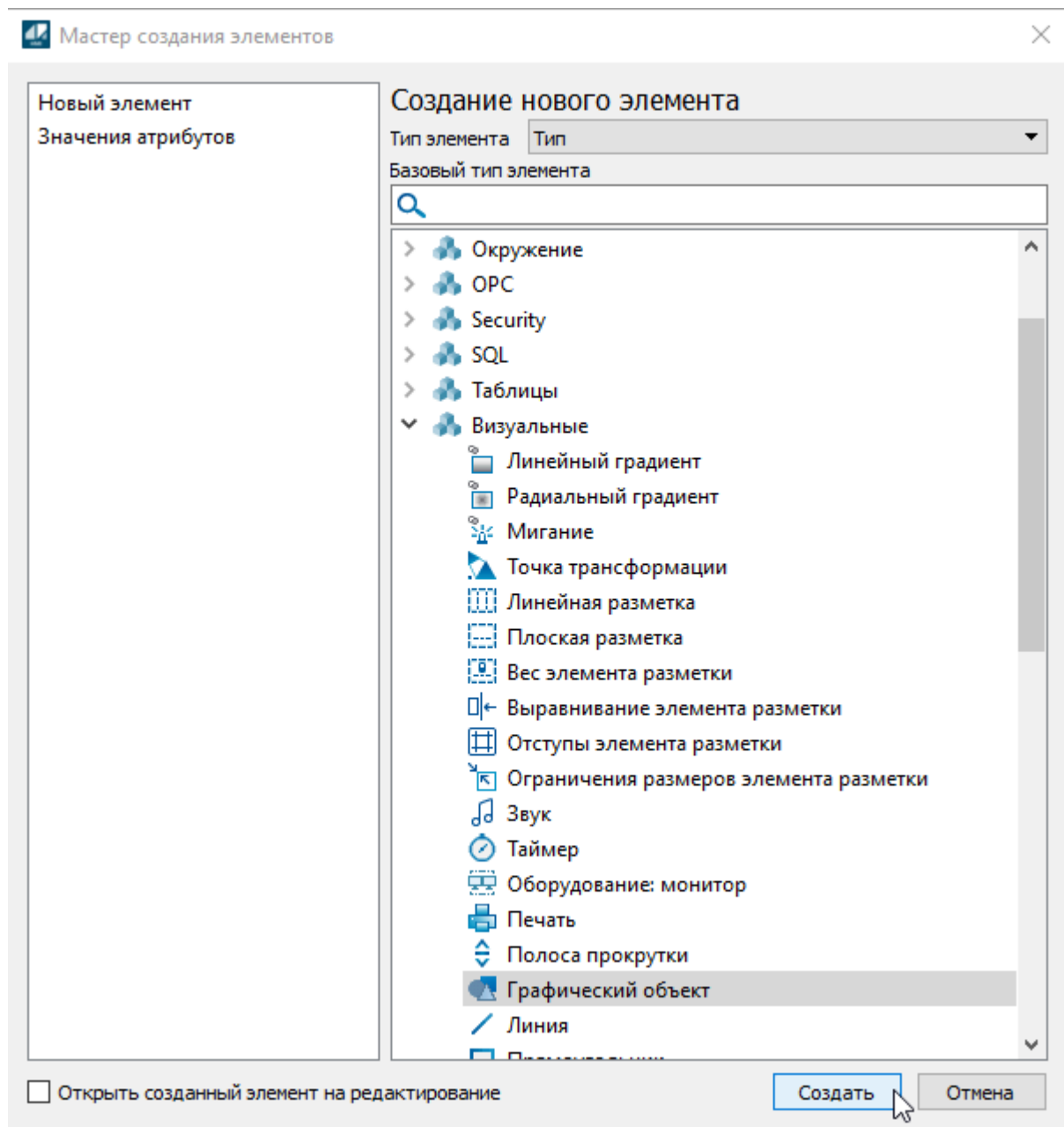
Чтобы использовать агрегаторы событий для отображения состояний в HMI, необходимо выполнить следующие действия:

1. Нажмите правой кнопкой мыши по вкладке "Типы" и в контекстном меню выберите команду "Создать...".

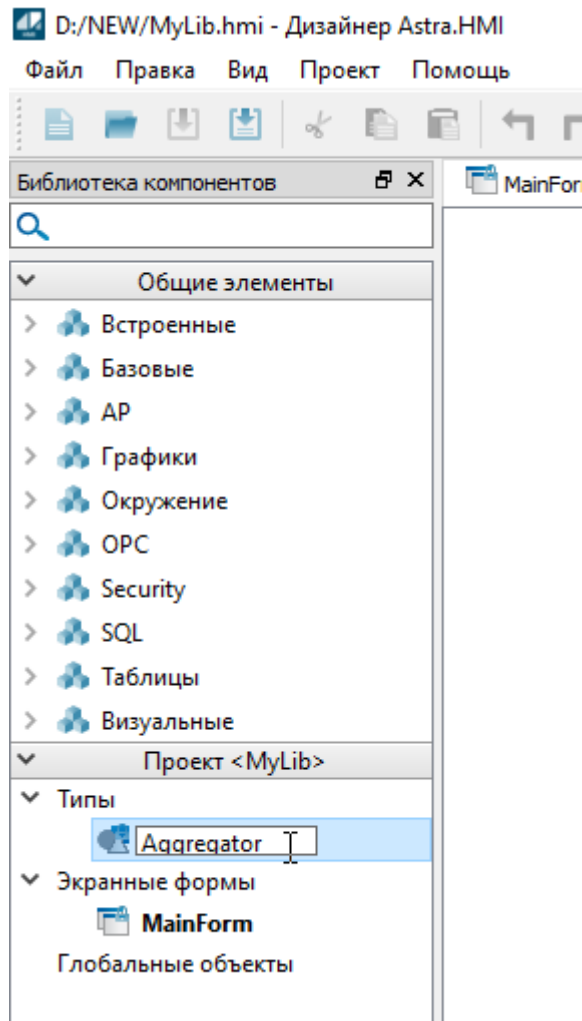


2. В мастере создания элементов раскройте вкладку "Визуальные" и выберите базовый тип элемента "Графический объект". Нажмите кнопку "Создать".

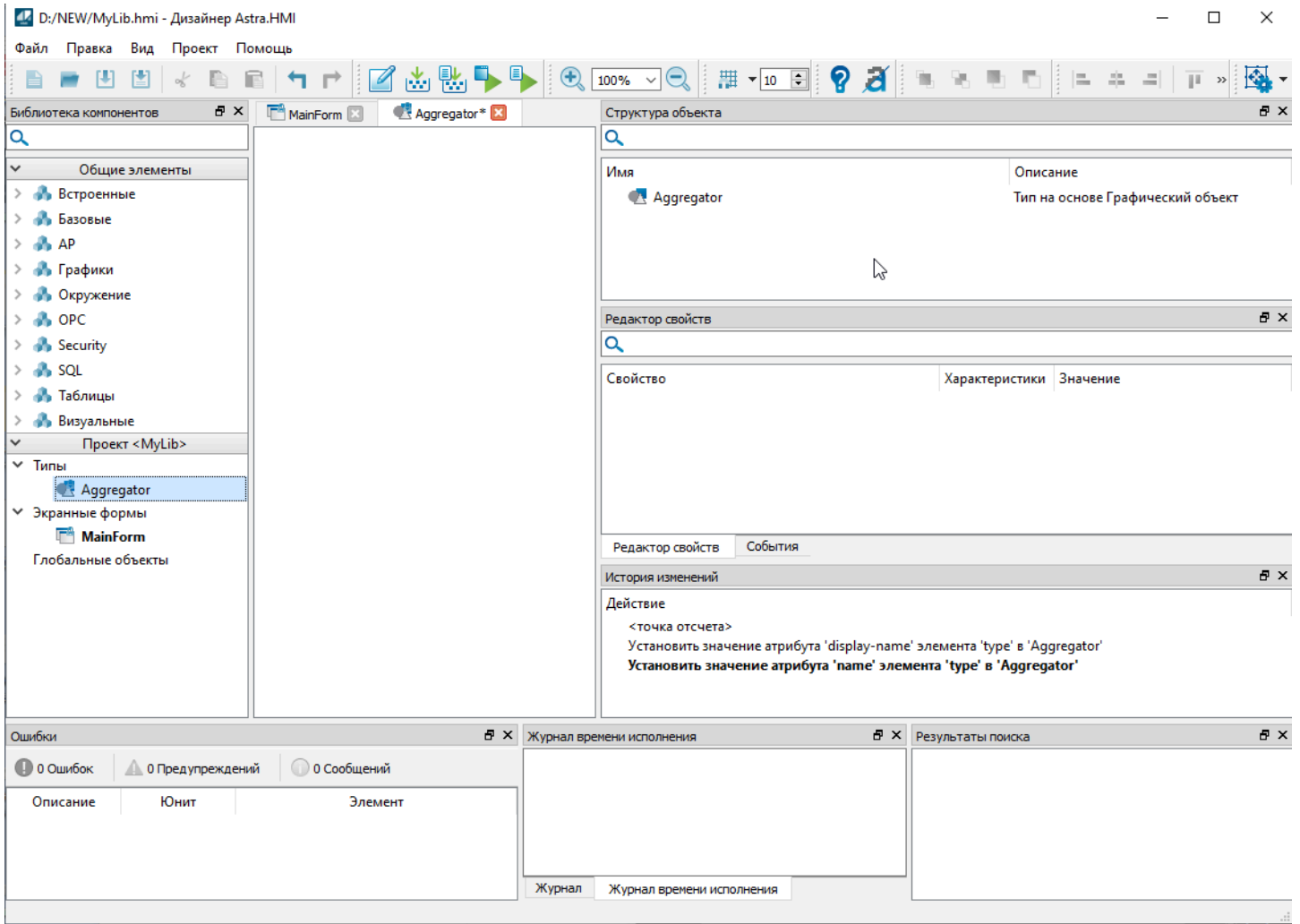


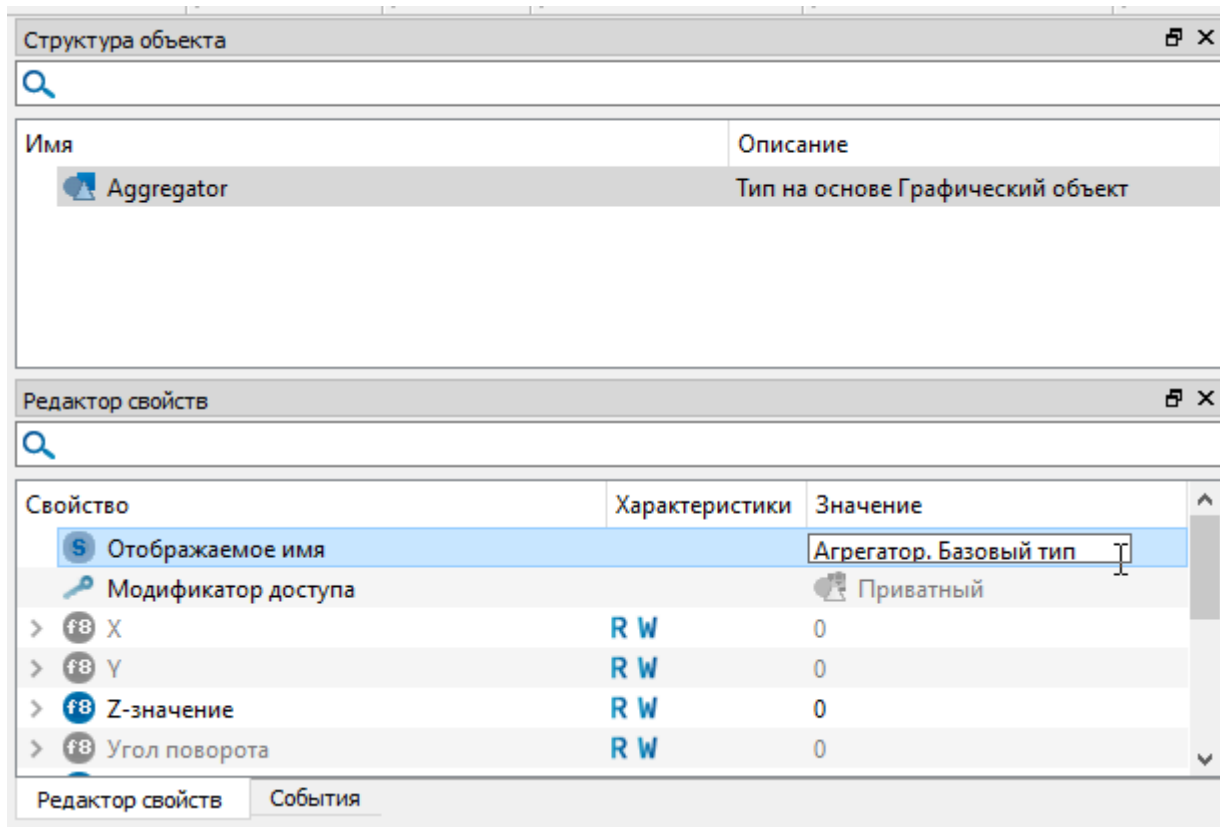


3. Во вкладке "Типы" появится новый графический объект с именем "Dummy\_1". Выделите тип левой кнопкой мыши, нажмите клавишу "F2" и задайте имя "Aggregator".

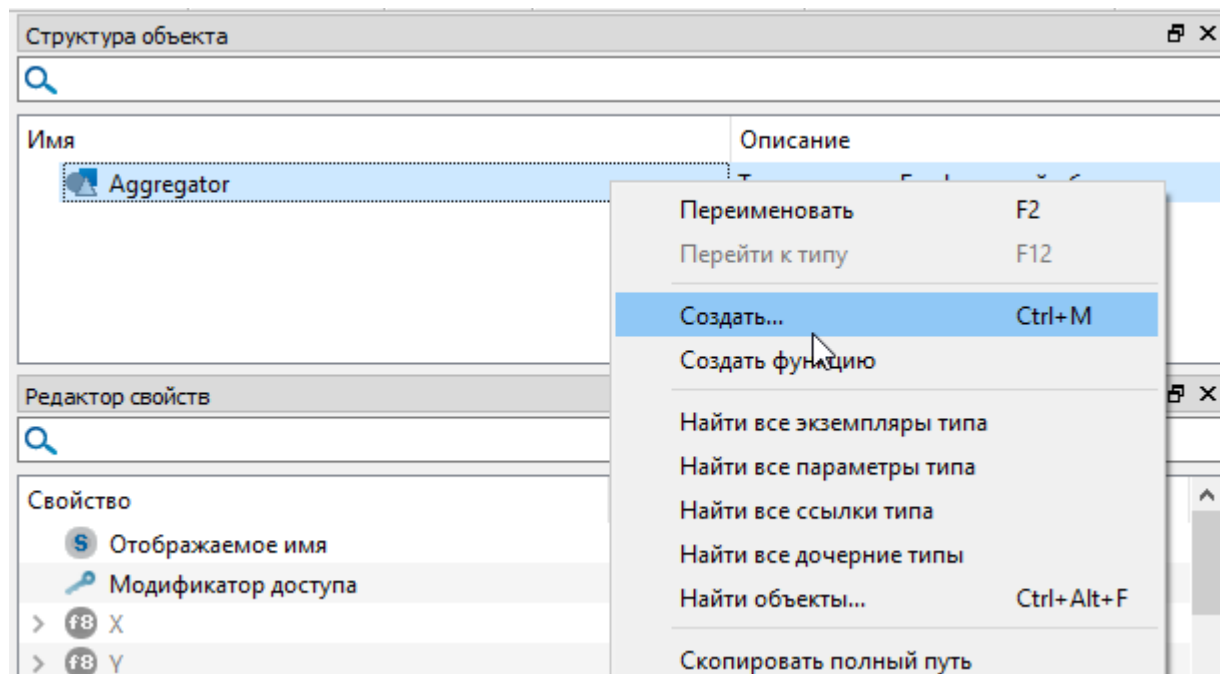


4. Перейдите в тип Aggregator двойным кликом левой кнопки мыши. В окне справа отобразится структура объекта.

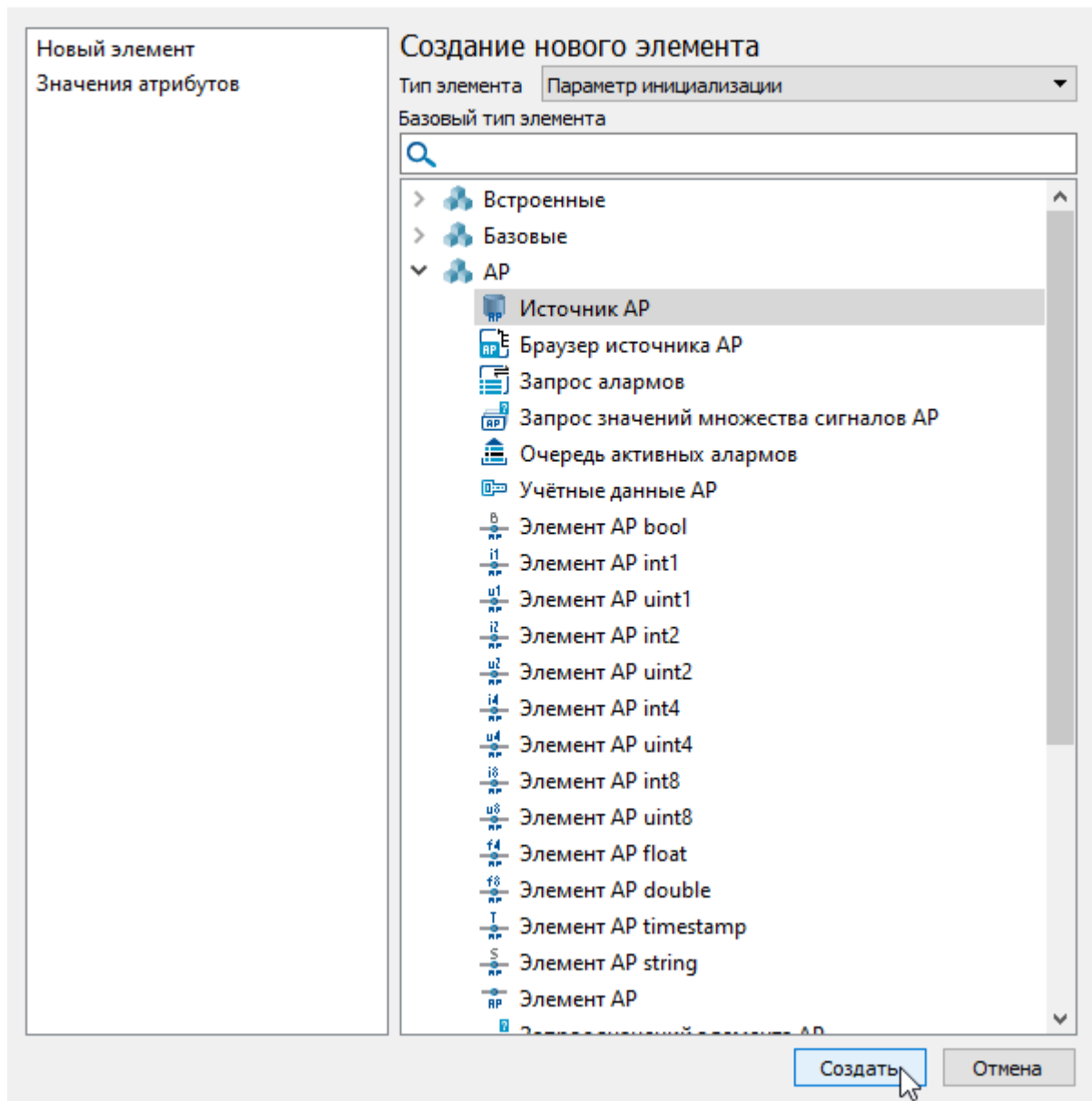




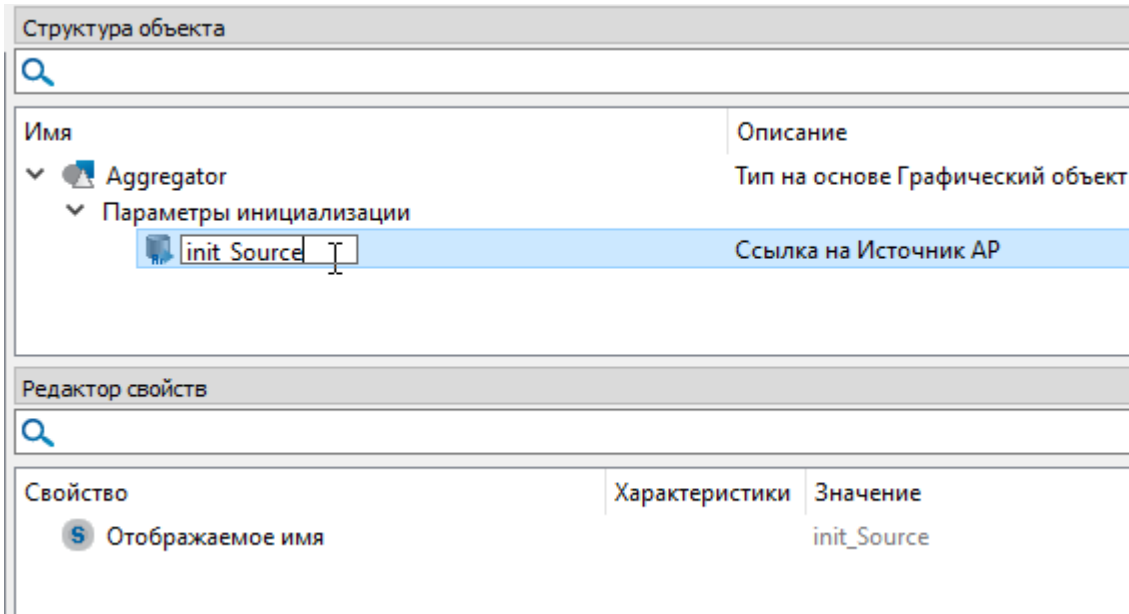
6. Кликните правой кнопкой мыши в структуре объекта по типу "Aggregator" и из контекстного меню выберите команду "Создать...".



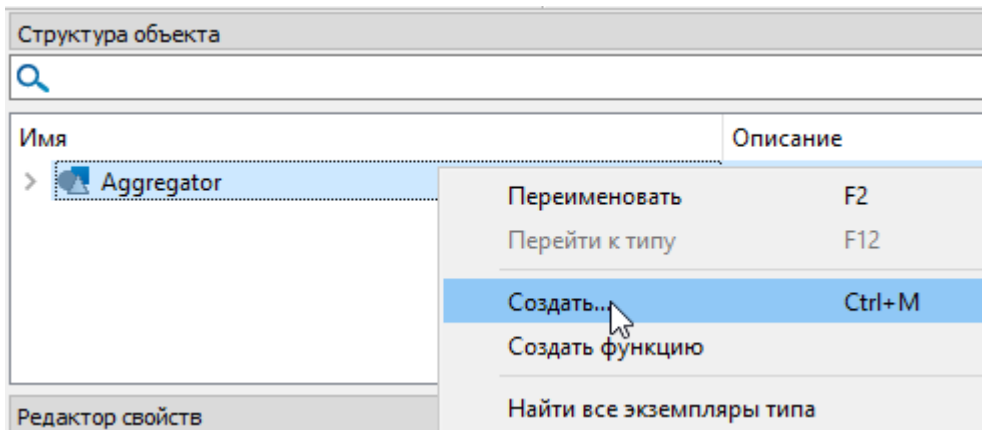
7. В мастере создания элементов в выпадающем списке вверху смените тип элемента на "Параметр инициализации", раскройте папку "AP" и выберите базовый тип "Источник AP". Нажмите кнопку "Создать".



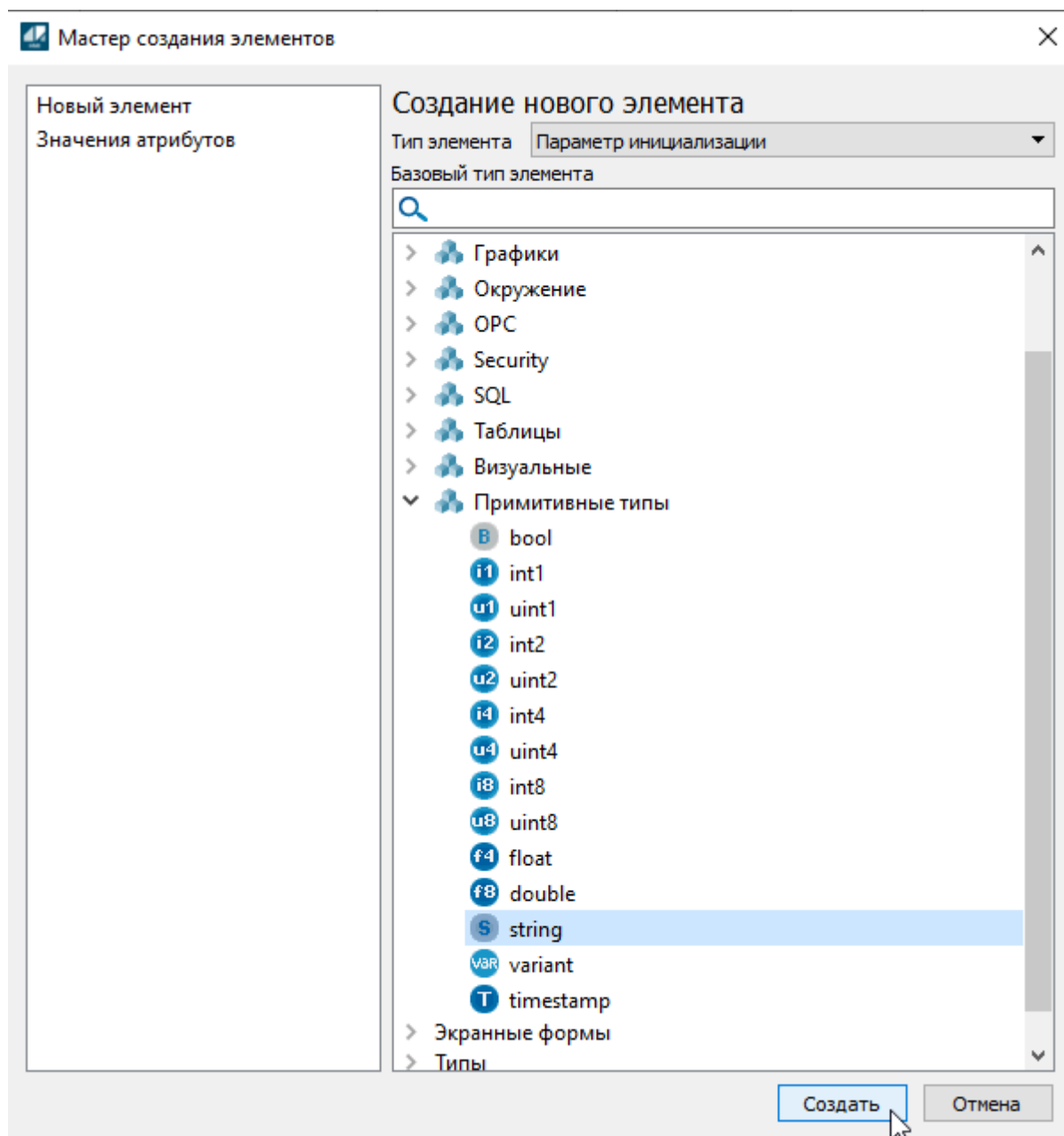
8. В структуре объекта появится вкладка "Параметры инициализации". Переименуйте созданный тип при помощи клавиши "F2" в "init\_Source".



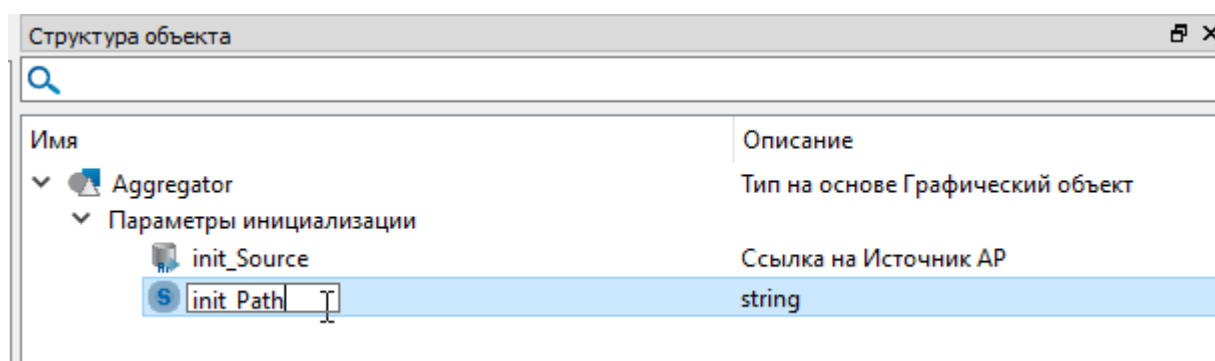
9. Кликните правой кнопкой мыши в структуре объекта по типу "Aggregator" и из контекстного меню выберите команду "Создать...".



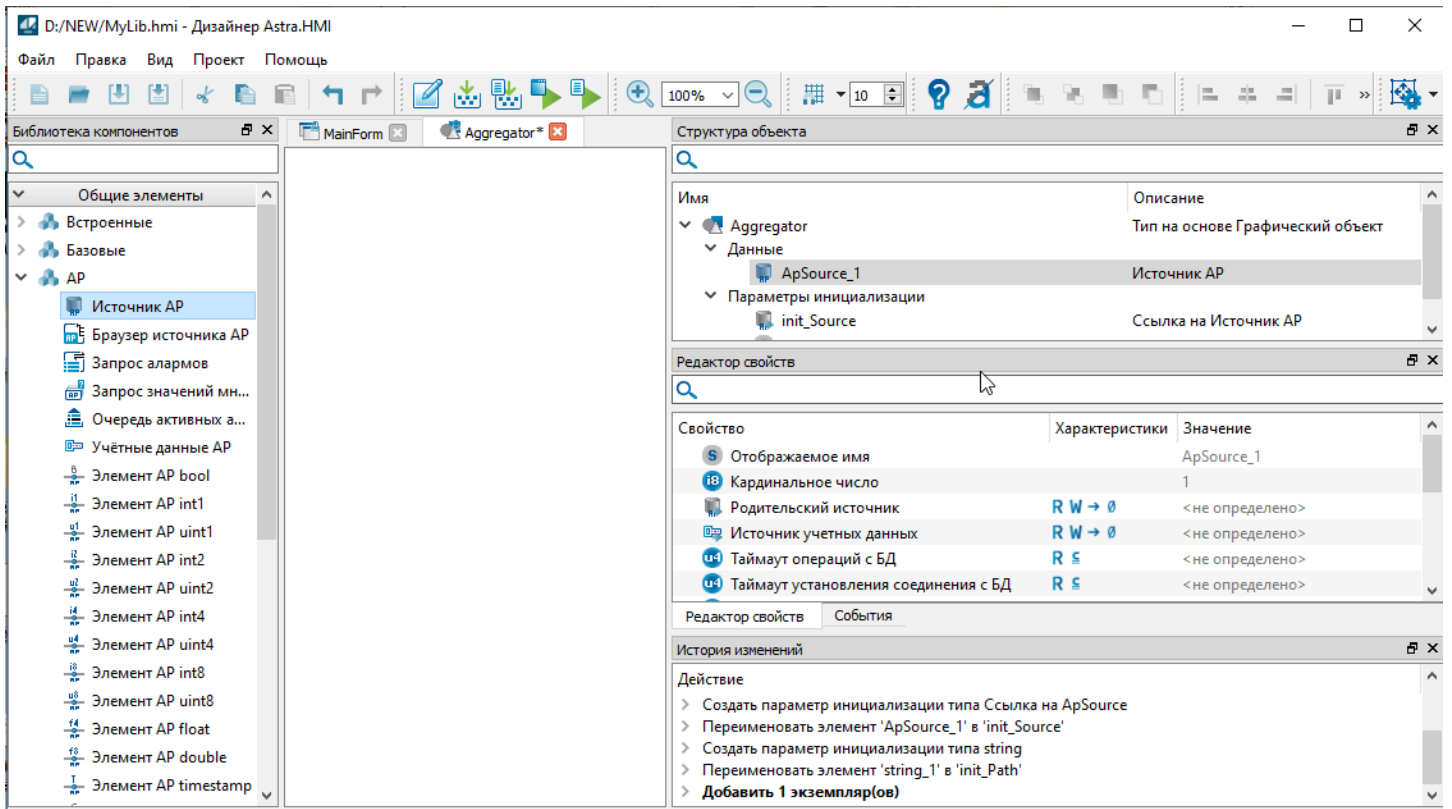
10. В мастере создания элементов в выпадающем списке вверху смените тип элемента на "Параметр инициализации", раскройте папку "Примитивные типы" и выберите базовый тип "string". Нажмите кнопку "Создать".



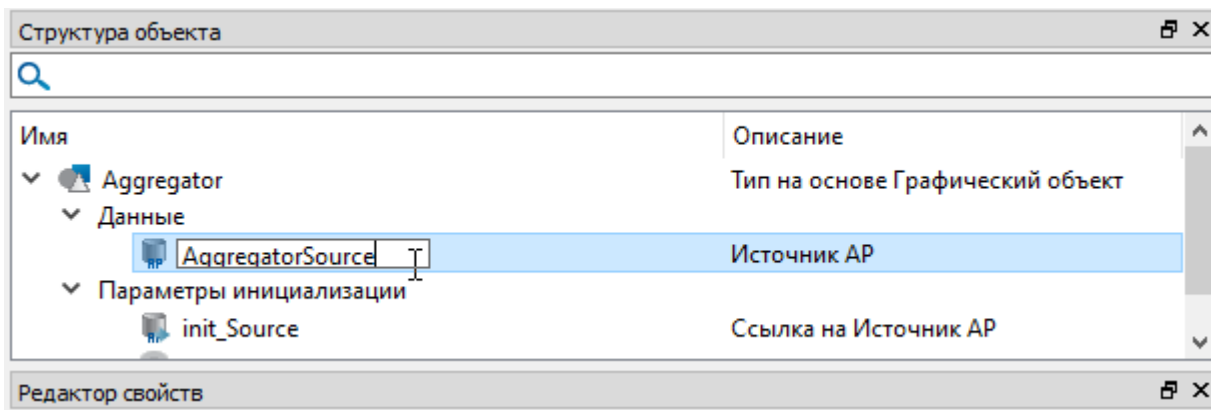
11. Во вкладке "Параметры инициализации" Переименуйте созданный тип "string\_1" при помощи клавиши "F2" в "init\_Path".



12. Во вкладке "Общие элементы" раскройте вкладку "AP" и путем перетаскивания добавьте элемент "Источник AP".

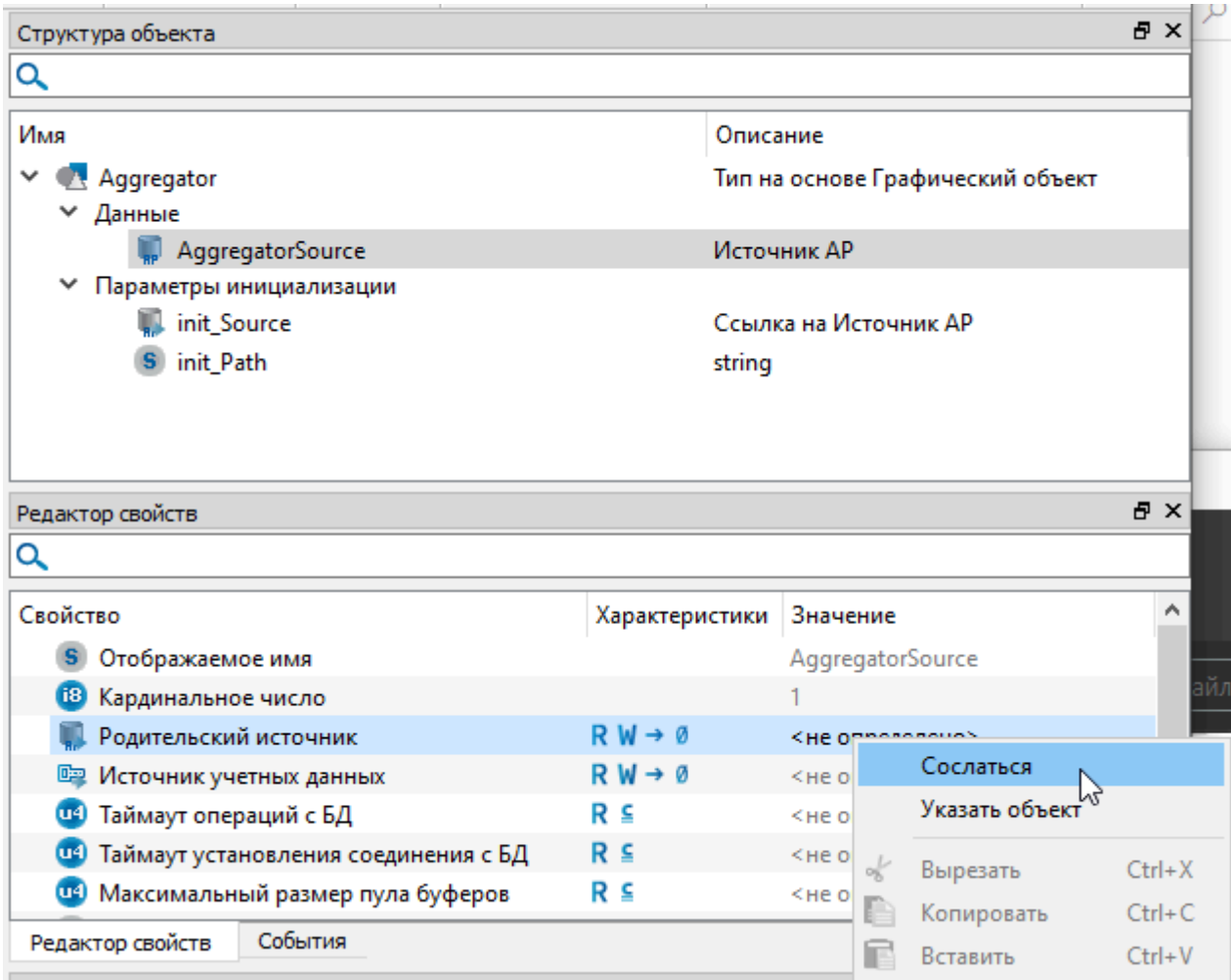


13. Во вкладке "Данные" окна "Структура объекта" переименуйте созданный тип "ApSource\_1" при помощи клавиши "F2" в "AggregatorSource".

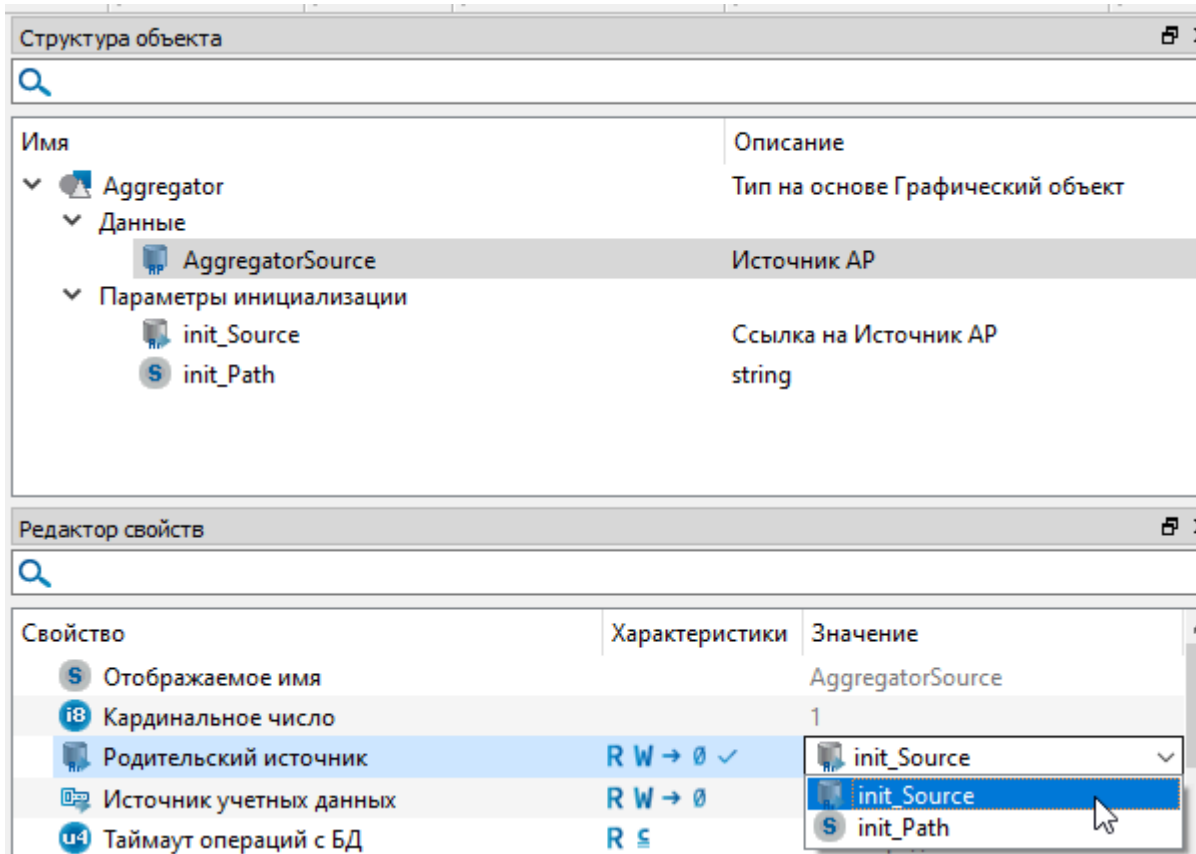


14. Выделите тип "AggregatorSource" нажав левой кнопкой мыши по нему. В окне "Редактор свойств" кликните правой кнопкой мыши по свойству "Родительский источник" в поле "Значение" и из контекстного меню выберите команду "Сослаться".

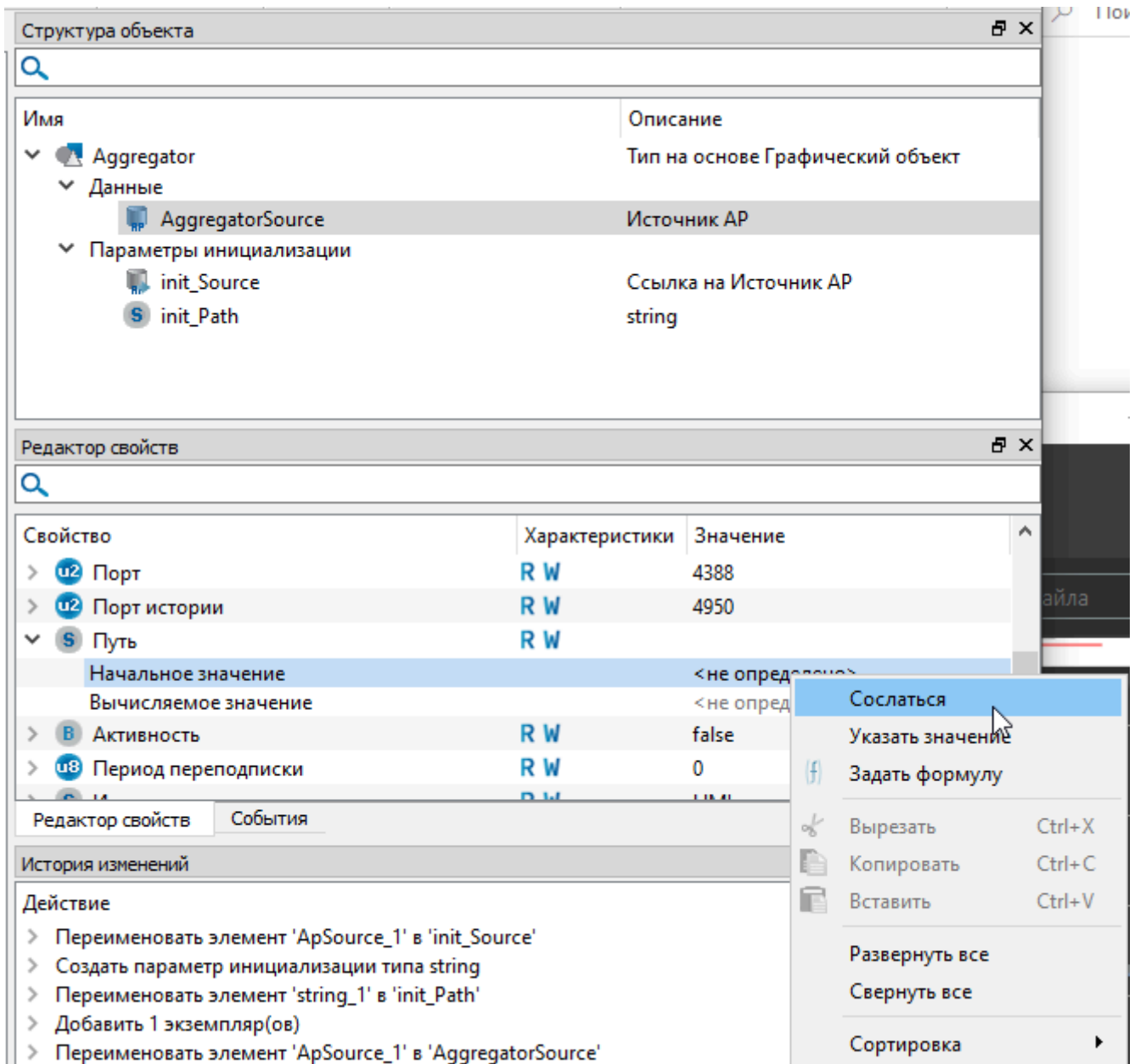




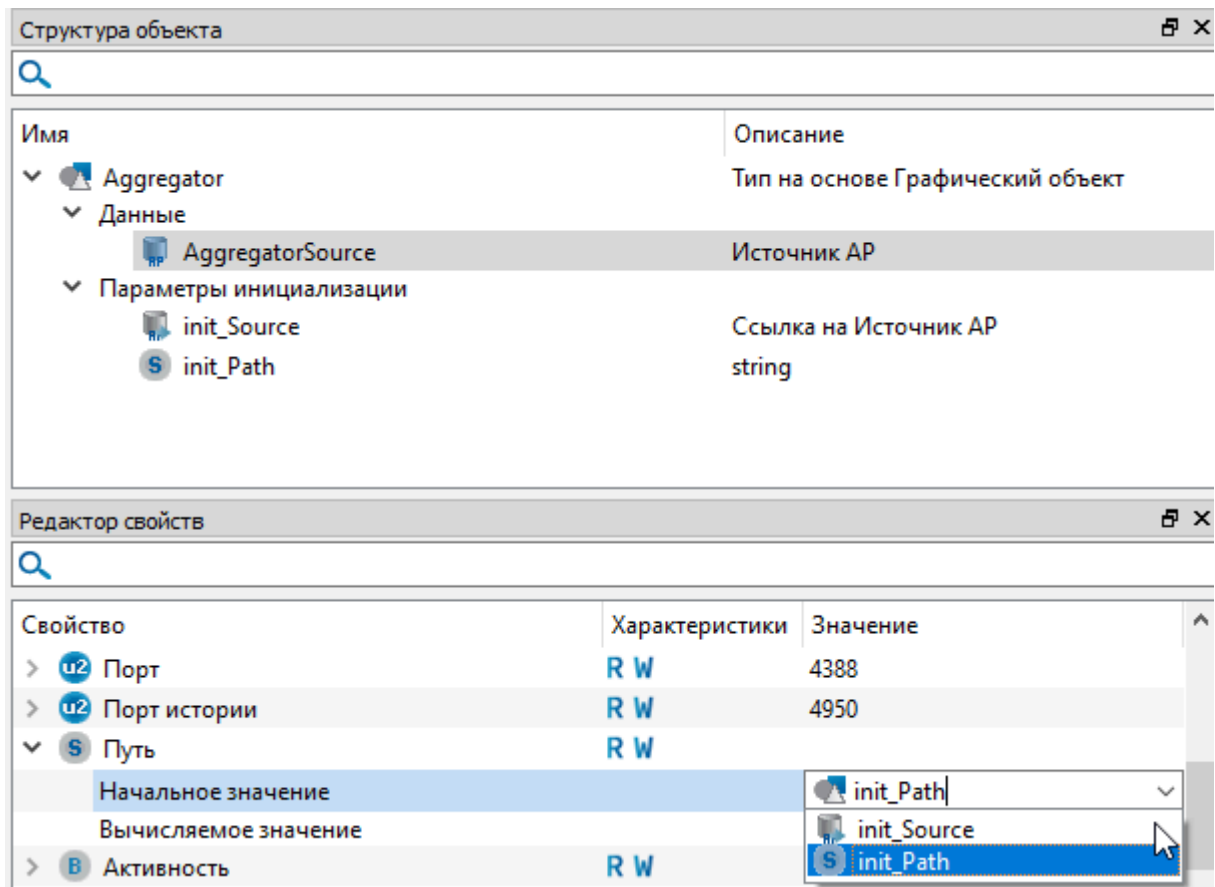
15. В появившемся поле ввода введите название параметра инициализации "init\_Source".



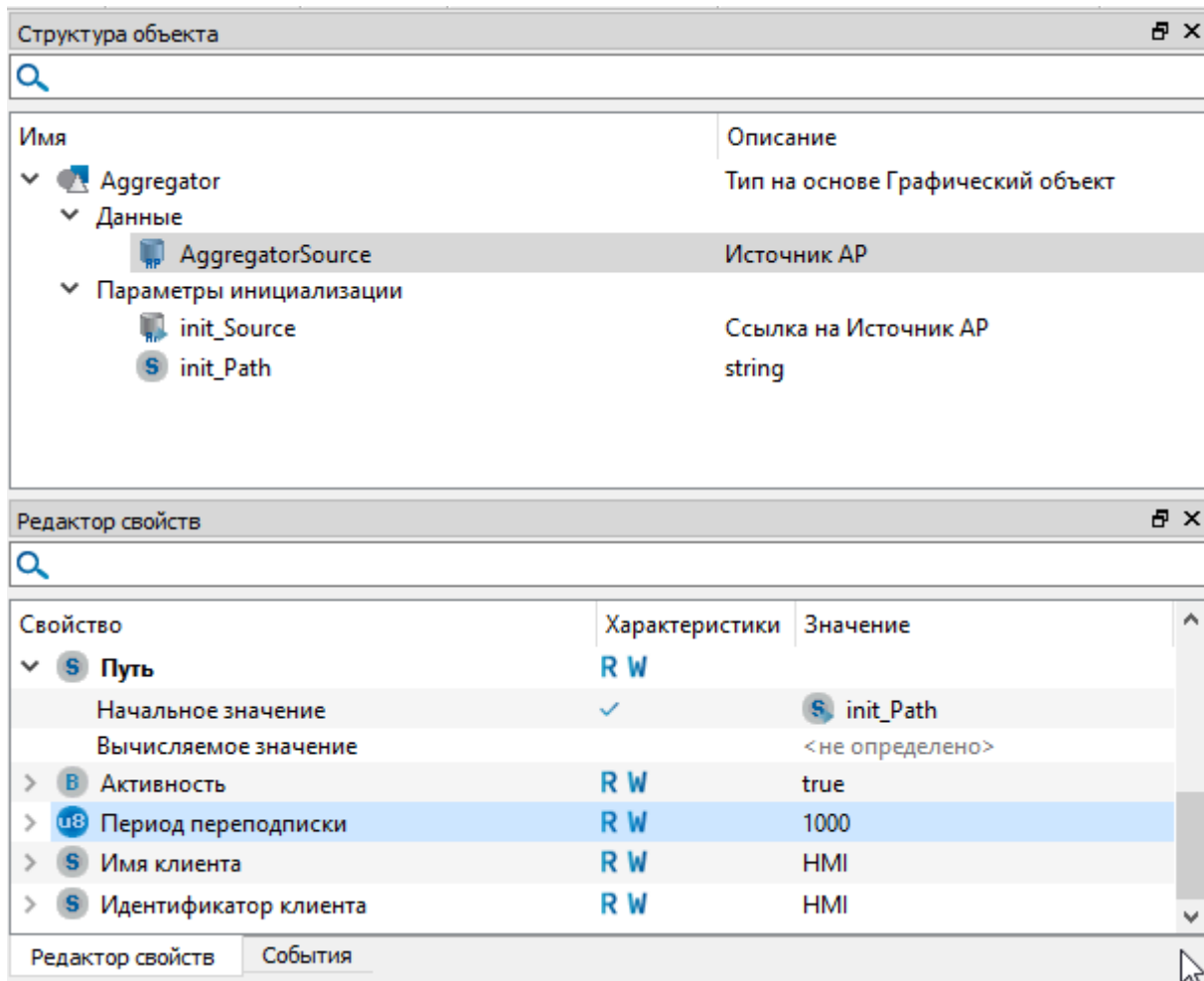
16. Раскройте свойство "Путь" кликнув по крыжику слева. Нажмите правой кнопкой мыши по строке "Начальное значение" и из контекстного меню выберите команду "Сослаться".



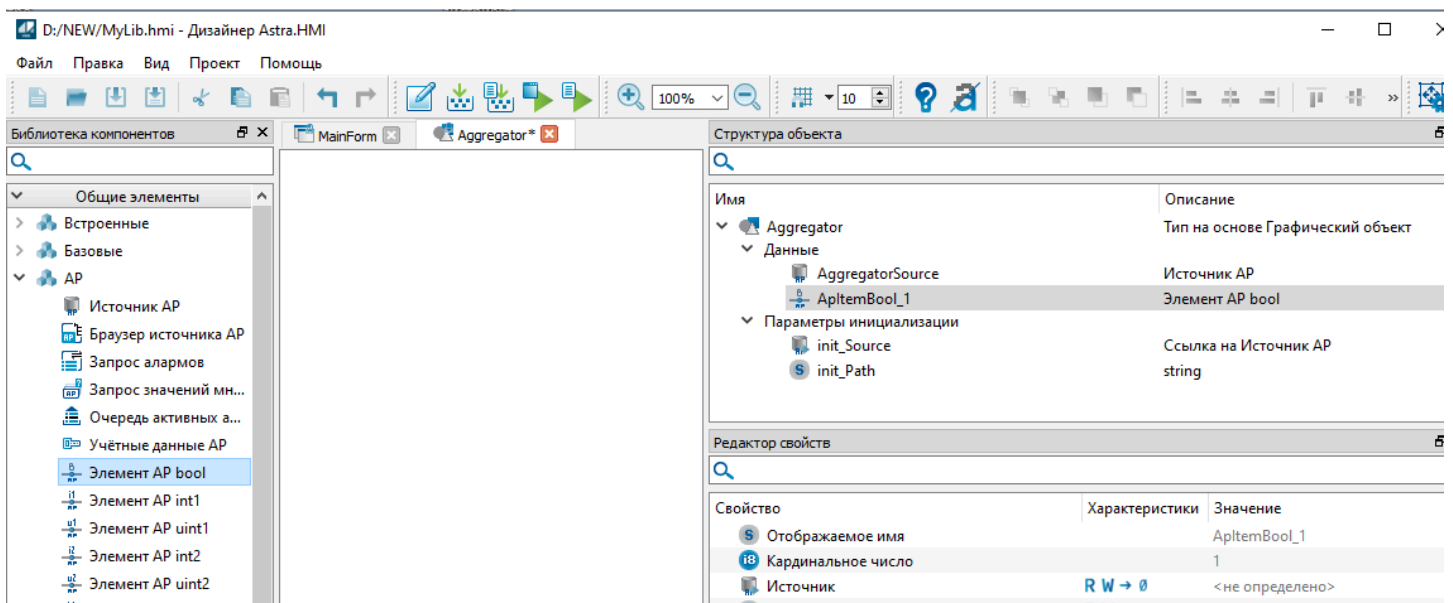
17. В появившемся поле ввода введите название параметра инициализации "init\_Path".



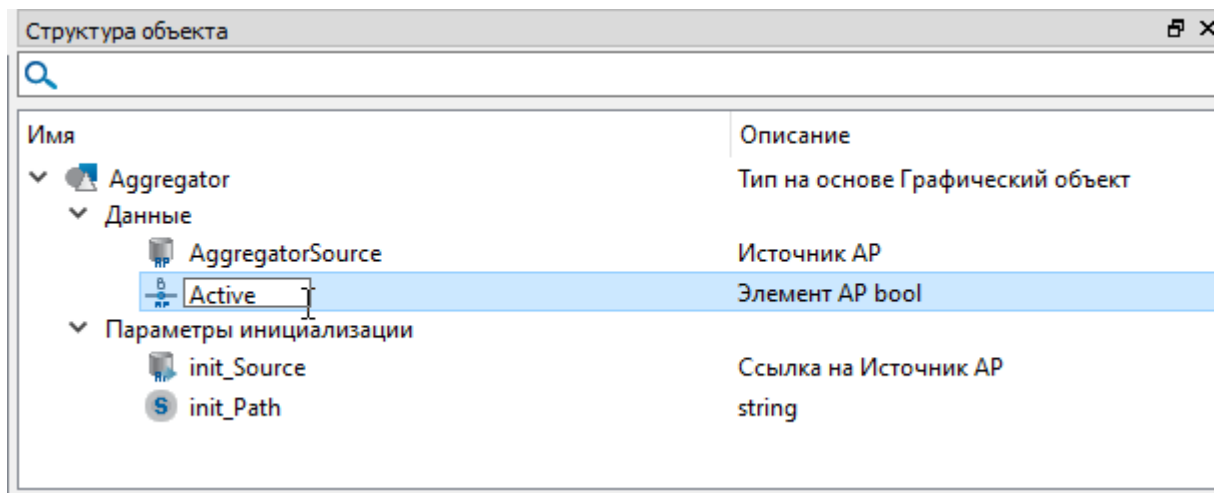
18. Установите свойство "Активность" в значение "true" и свойство "Период переподписки" в значение "1000."



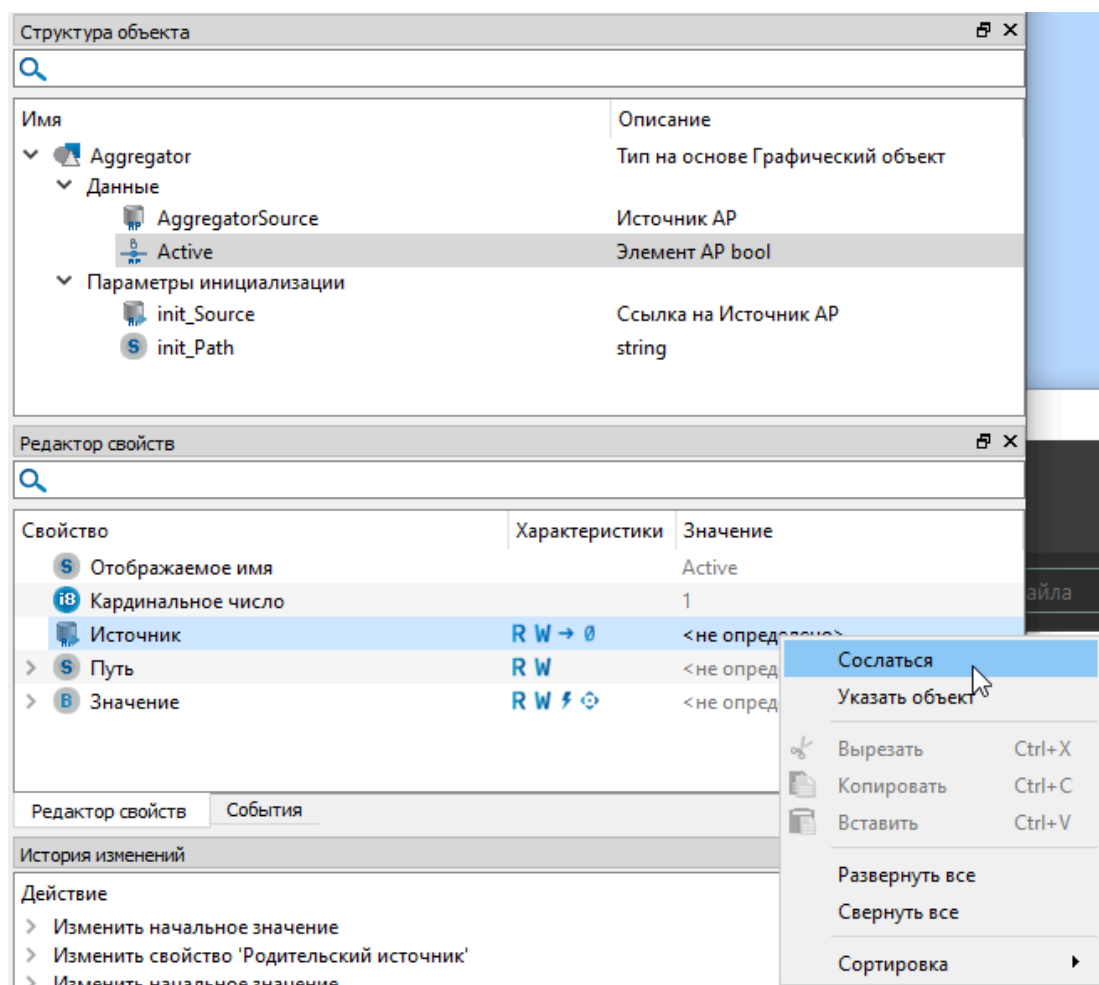
19. Во вкладке "Общие элементы" раскройте вкладку "AP" и путем перетаскивания добавьте "Элемент AP bool".



20. Во вкладке "Данные" окна "Структура объекта" переименуйте созданный тип "ApltemBool\_1" при помощи клавиши "F2" в "Active".



21. Выделите элемент "Active" левой кнопкой мыши. В окне "Редактор свойств" кликните правой кнопкой мыши по свойству "Источник" в поле "Значение" и из контекстного меню выберите команду "Сослаться".



22. В появившемся поле ввода введите название источника AP "AggregatorSource".

Структура объекта

Имя	Описание
Aggregator	Тип на основе Графический объект
Данные	
AggregatorSource	Источник AP
Active	Элемент AP bool
Параметры инициализации	
init_Source	Ссылка на Источник AP
init_Path	string

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		Active
Кардинальное число		1
Источник	R W → 0	AggregatorSource
Путь	R W	AggregatorSource
Значение	R W ⚡ ⚙	<не определено>

23. Кликните двойным кликом по свойству "Путь" в поле значение. В появившемся поле ввода укажите путь до параметра агрегатора "Active", который показывает наличие активных событий в агрегаторе.

Структура объекта

Имя	Описание
Aggregator	Тип на основе Графический объект
Данные	
AggregatorSource	Источник AP
Active	Элемент AP bool
Параметры инициализации	
init_Source	Ссылка на Источник AP
init_Path	string

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		Active
Кардинальное число		1
Источник	R W → 0 ✓	AggregatorSource
Путь	R W	Active
Значение	R W f ↻	<не определено>

24. Аналогичным образом добавьте параметр агрегатора "Ack", который отвечает за квитирование активных событий в агрегаторе.

Структура объекта

Имя	Описание
Aggregator	Тип на основе Графический объект
Данные	
AggregatorSource	Источник AP
Active	Элемент AP bool
Ack	Элемент AP bool
Параметры инициализации	
init_Source	Ссылка на Источник AP
init_Path	string

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		Ack
Кардинальное число		1
Источник	R W → 0 ✓	AggregatorSource
Путь	R W ✓	Ack
Значение	R W f ↻	<не определено>

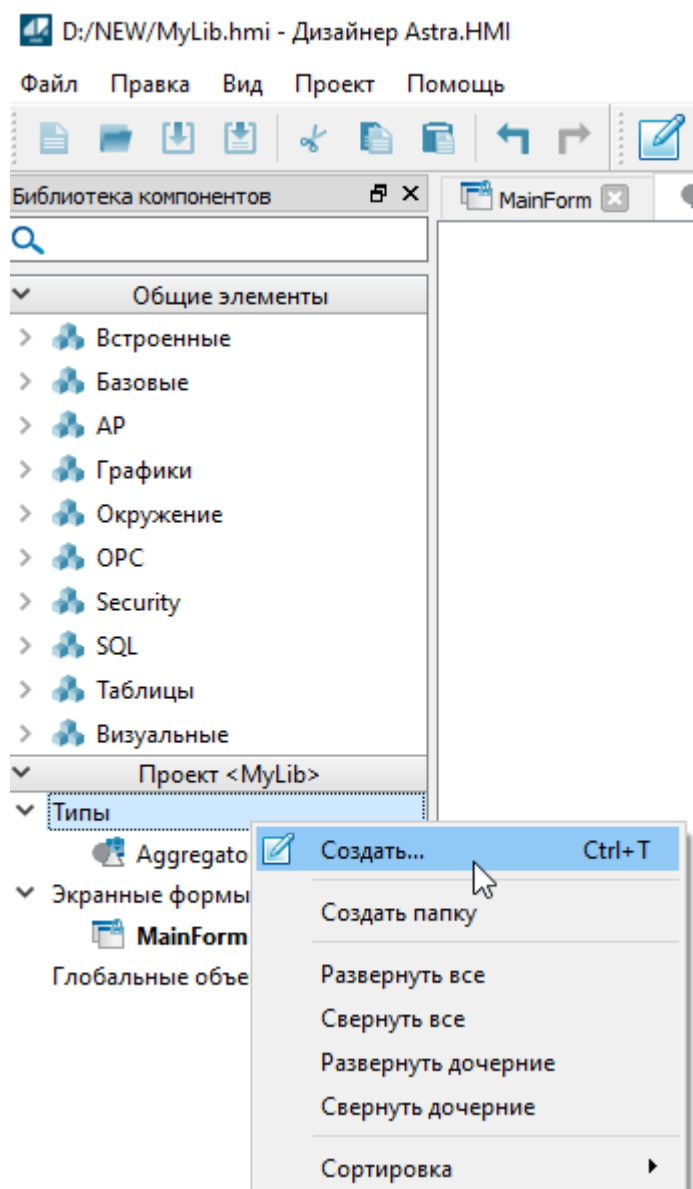


Создание базового типа агрегатора завершено.

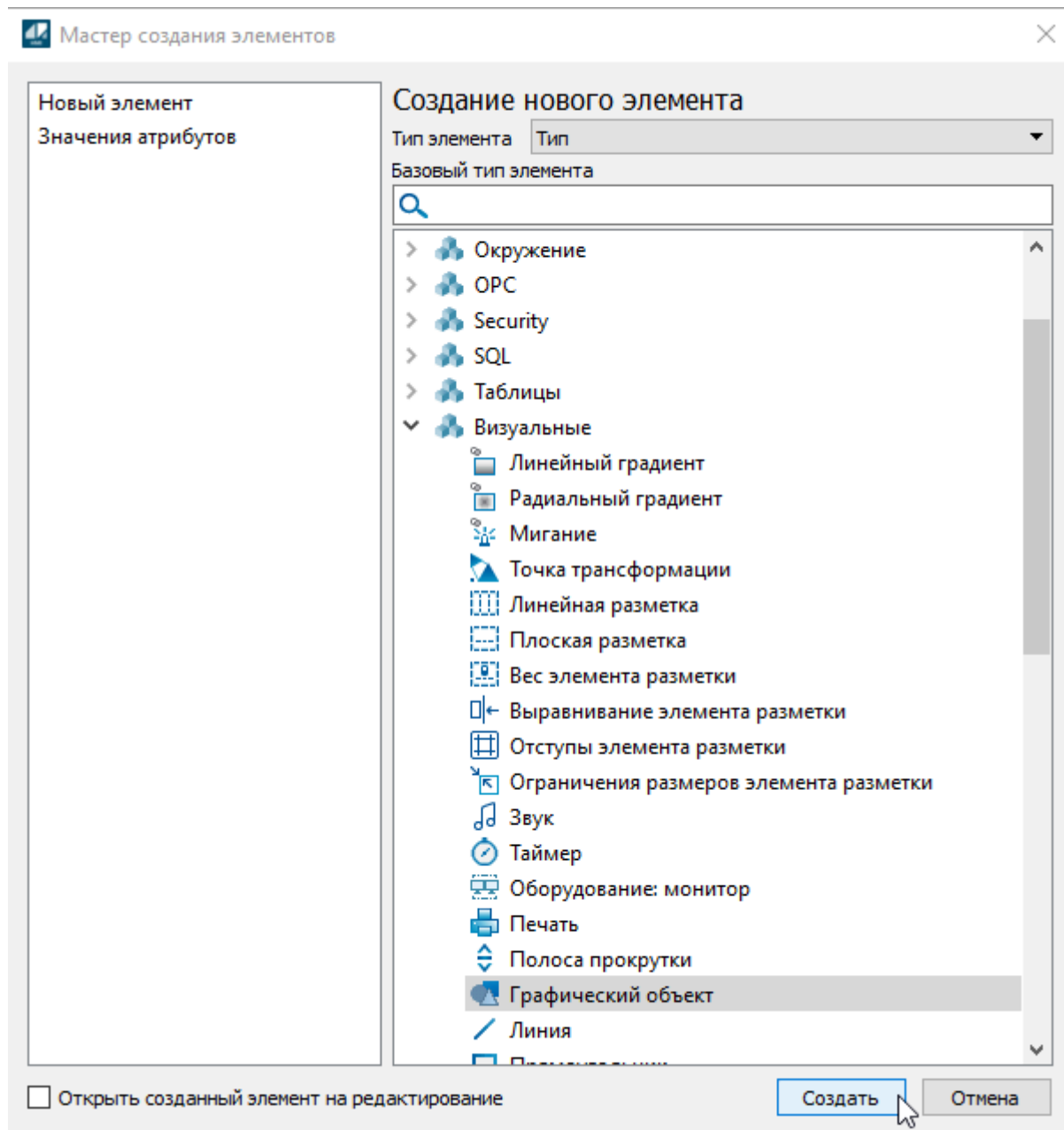
## 1.2.4.3. Создание базового типа мнемосимволов

Чтобы создать базовый тип мнемосимволов, выполните следующие действия:

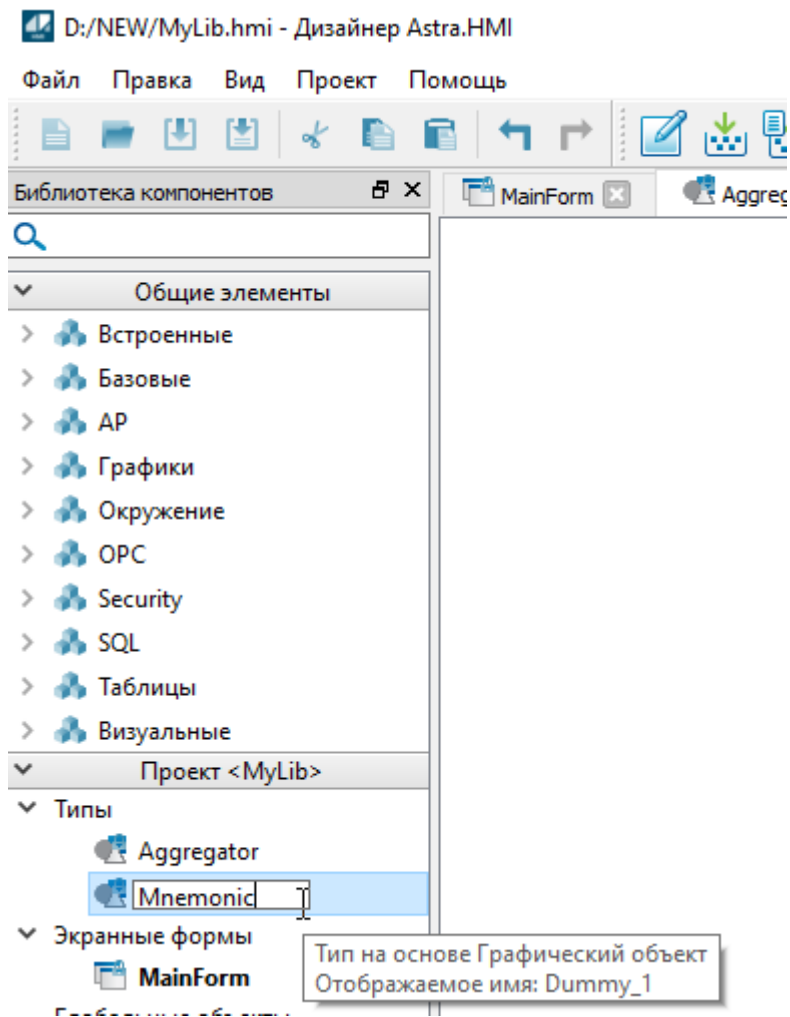
1. Нажмите правой кнопкой мыши по вкладке "Типы" и в контекстном меню выберите команду "Создать...".



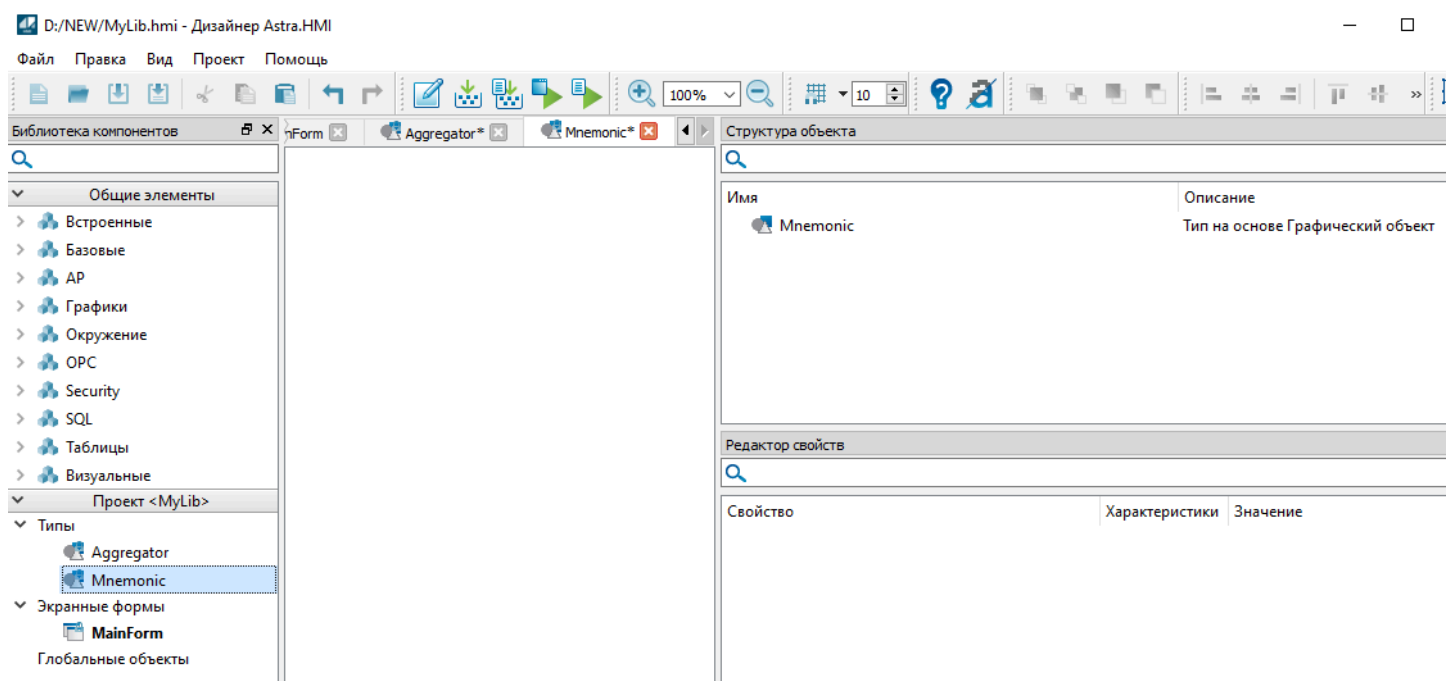
2. В мастере создания элементов раскройте вкладку "Визуальные" и выберите базовый тип элемента "Графический объект". Нажмите кнопку "Создать".



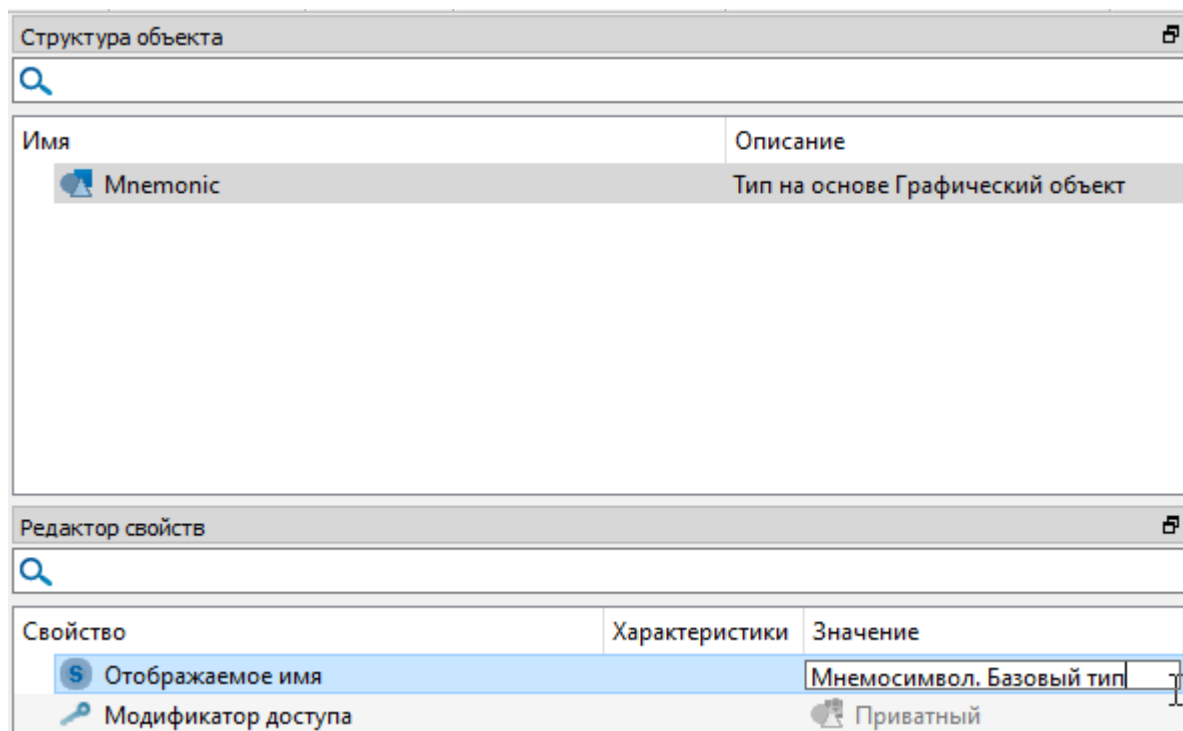
3. Во вкладке "Типы" появится новый графический объект с именем "Dummy\_1". Выделите тип левой кнопкой мыши, нажмите клавишу "F2" и задайте имя "Mnemonic".



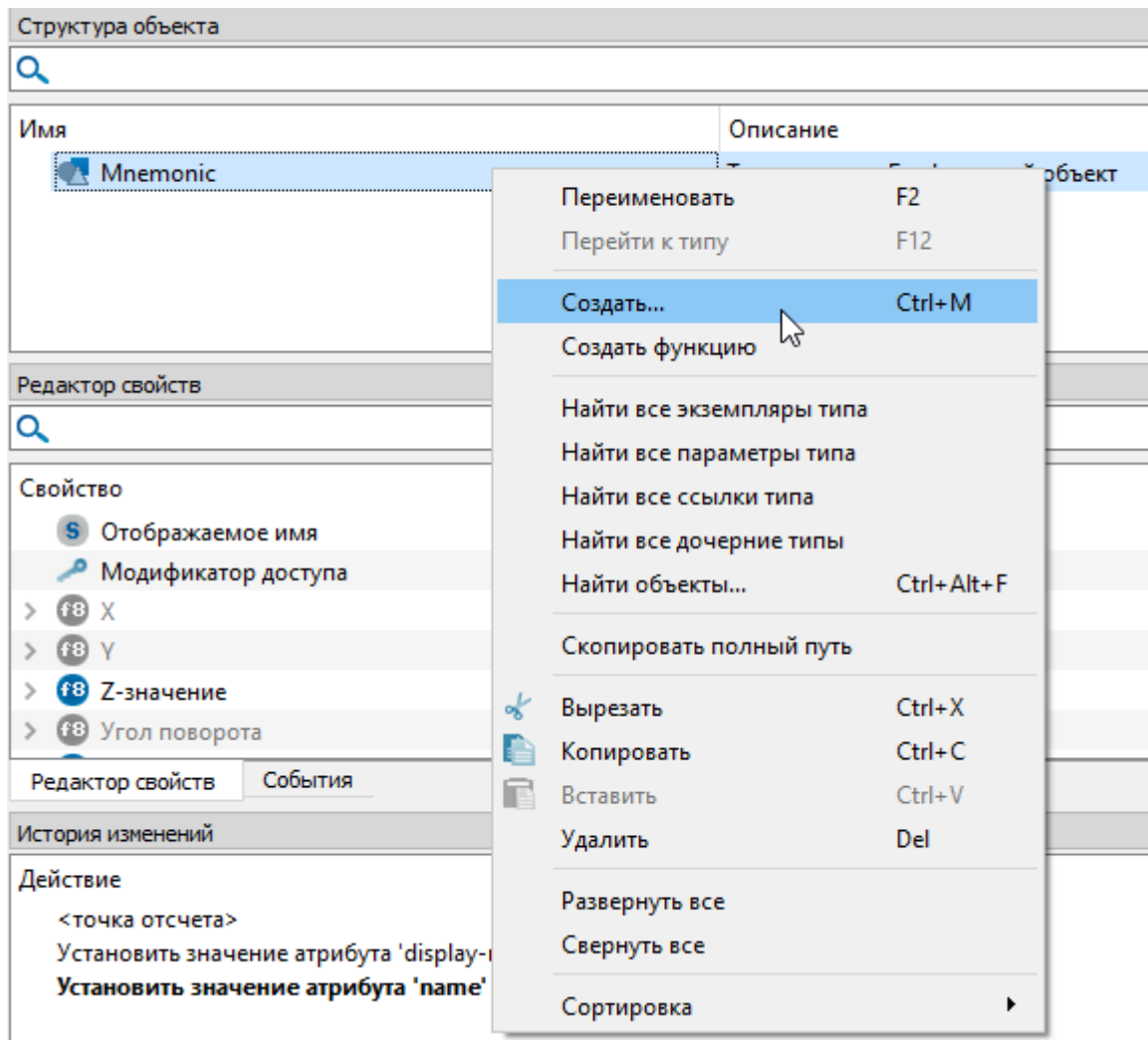
4. Перейдите в тип "Mnemonic" двойным кликом левой кнопки мыши. В окне справа отобразится структура объекта.



5. Выделите тип "Mnemonic" левой кнопкой мыши и в окне "Редактор свойств" задайте отображаемое имя "Мнемосимвол. Базовый тип".



6. Кликните правой кнопкой мыши в структуре объекта по типу "Mnemonic" и из контекстного меню выберите команду "Создать.."



7. В мастере создания элементов в выпадающем списке вверху смените тип элемента на "Параметр инициализации", раскройте папку "AP" и выберите базовый тип "Источник AP". Нажмите кнопку "Создать".

Новый элемент

Значения атрибутов

Создание нового элемента

Тип элемента

Базовый тип элемента

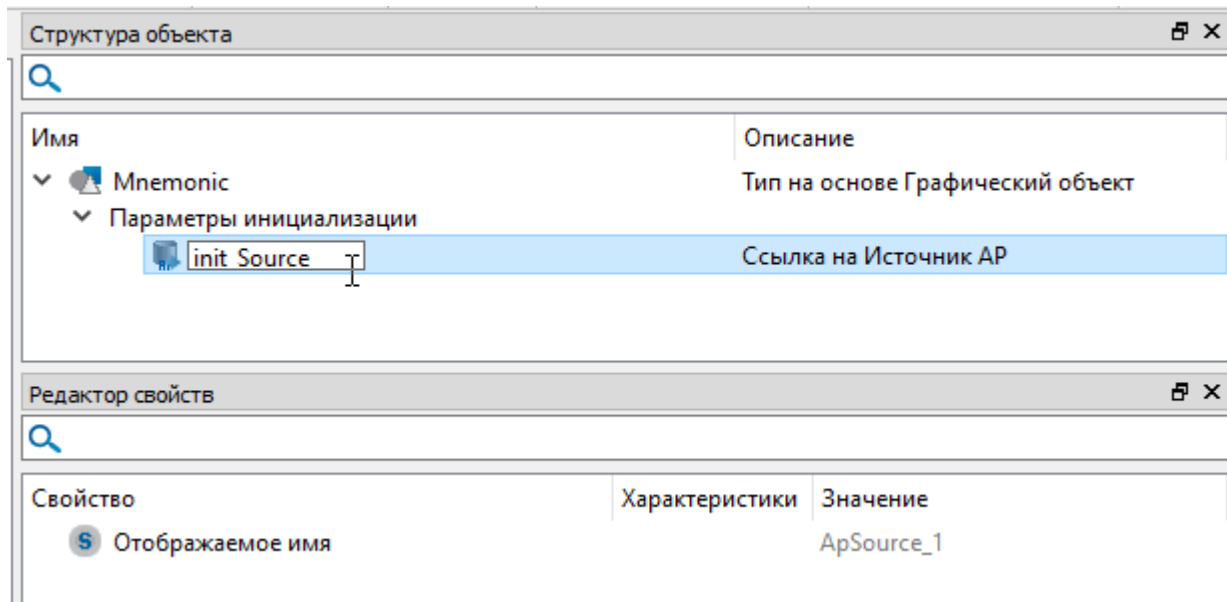


- > Встроенные
- > Базовые
- ▼ AP
  - Источник AP
  - Браузер источника AP
  - Запрос алармов
  - Запрос значений множества сигналов AP
  - Очередь активных алармов
  - Учётные данные AP
  - Элемент AP bool
  - Элемент AP int1
  - Элемент AP uint1
  - Элемент AP int2
  - Элемент AP uint2
  - Элемент AP int4
  - Элемент AP uint4
  - Элемент AP int8
  - Элемент AP uint8
  - Элемент AP float
  - Элемент AP double
  - Элемент AP timestamp
  - Элемент AP string
  - Элемент AP

Создать

Отмена

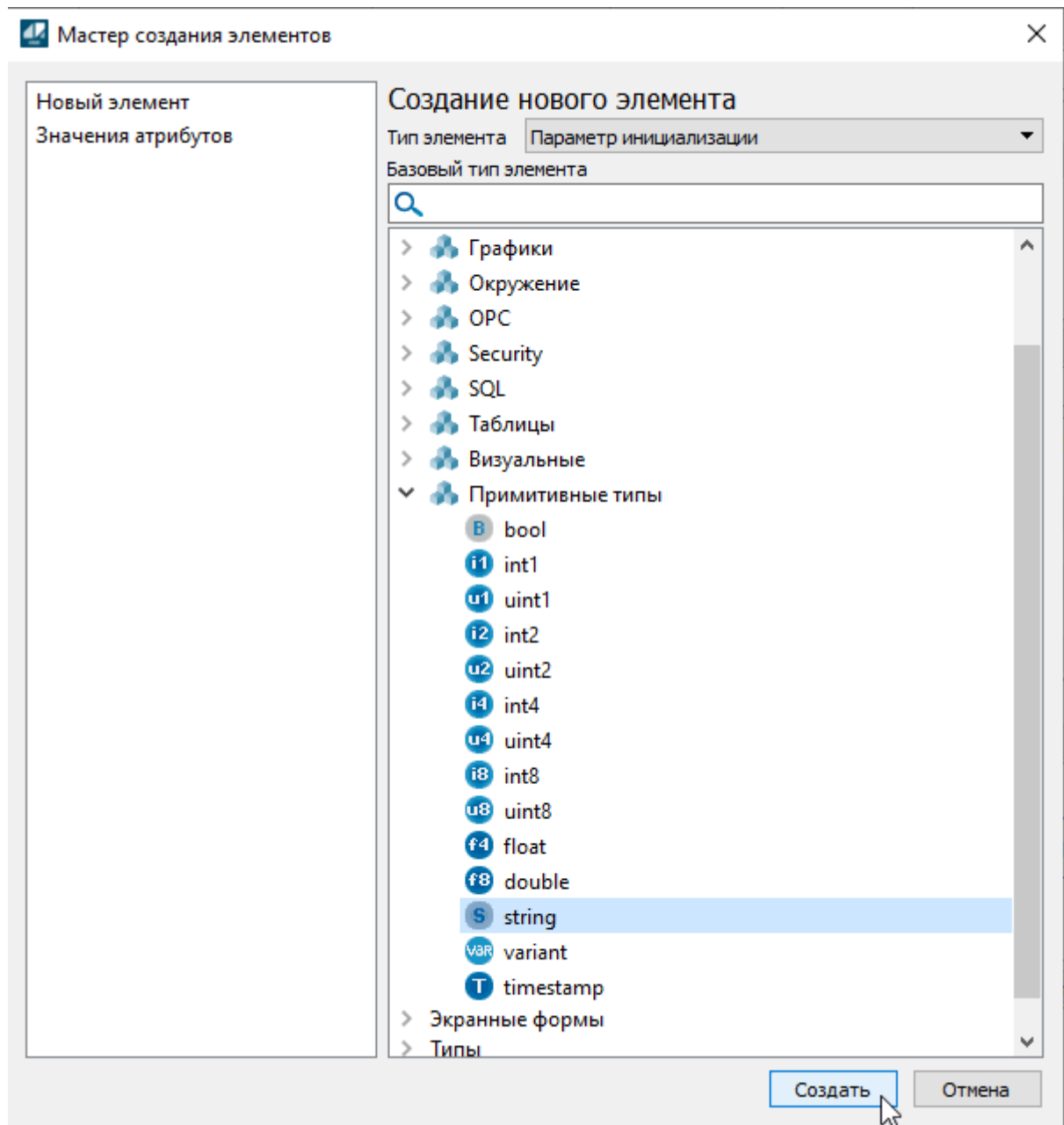
8. В структуре объекта появится вкладка "Параметры инициализации". Переименуйте созданный тип при помощи клавиши "F2" в "init\_Source".



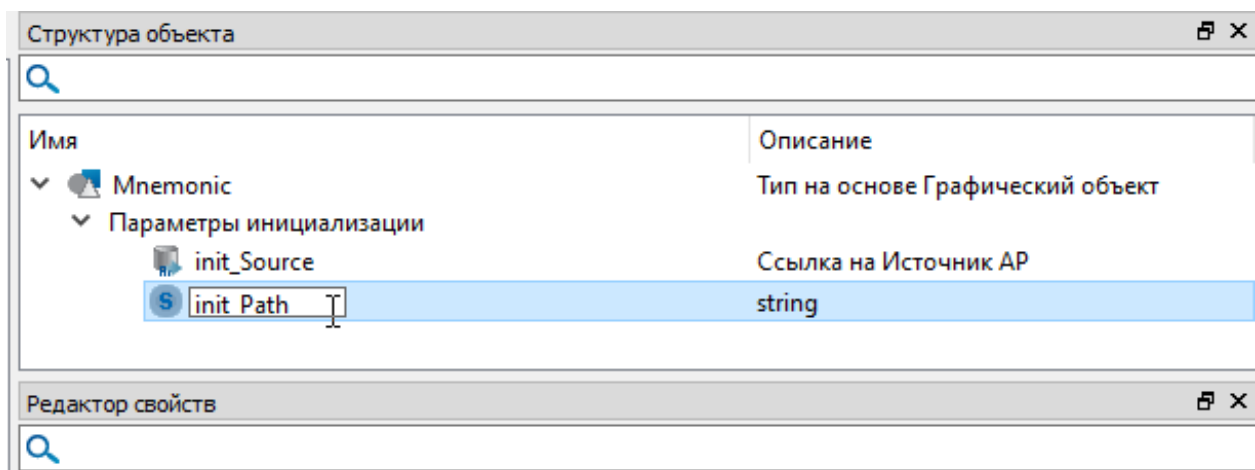
9. Кликните правой кнопкой мыши в структуре объекта по типу "Mnemonic" и из контекстного меню выберите команду "Создать.."

10. В мастере создания элементов в выпадающем списке вверху смените тип элемента на "Параметр инициализации", раскройте папку "Примитивные типы" и выберите базовый тип "string". Нажмите кнопку "Создать".

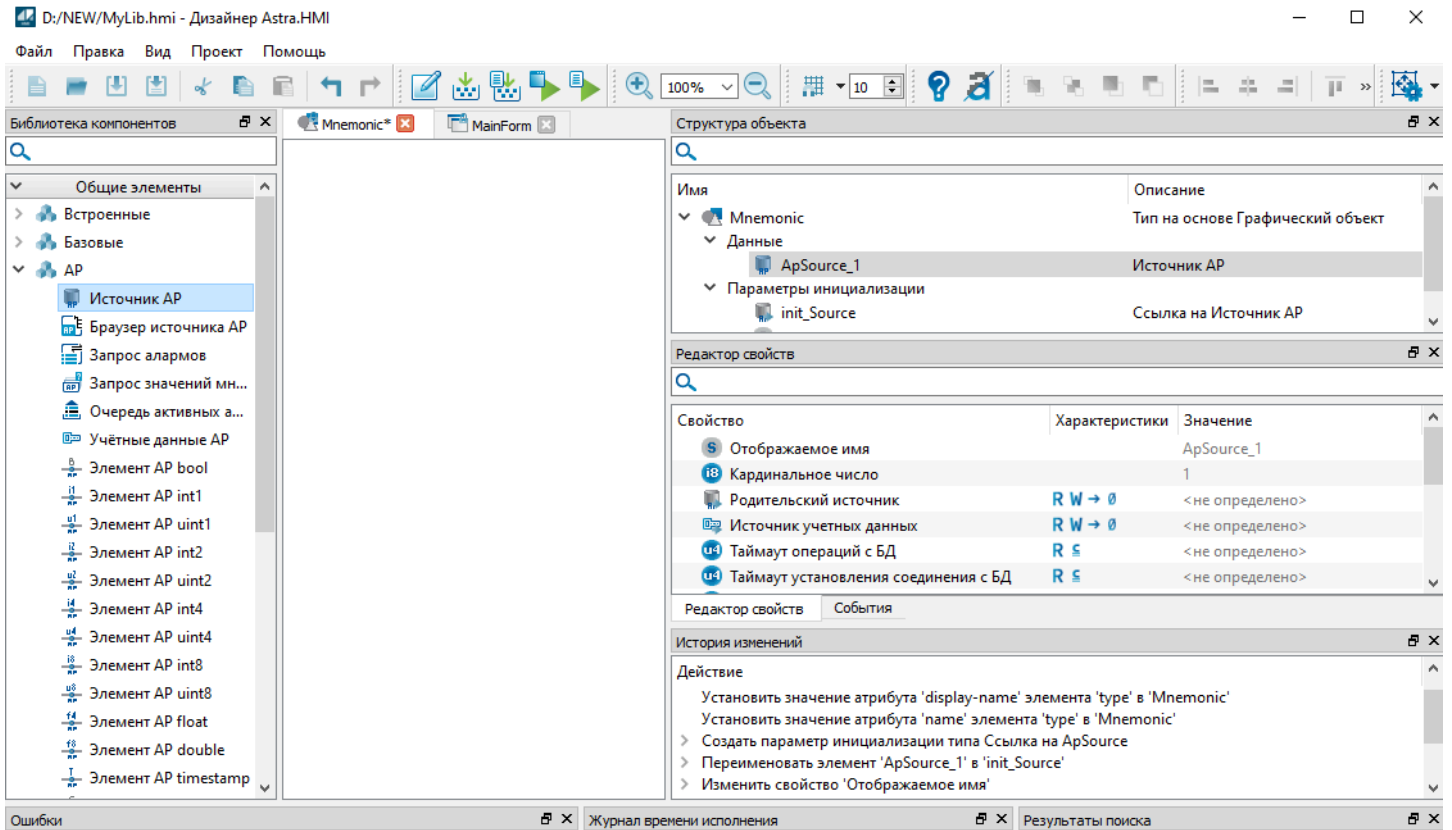




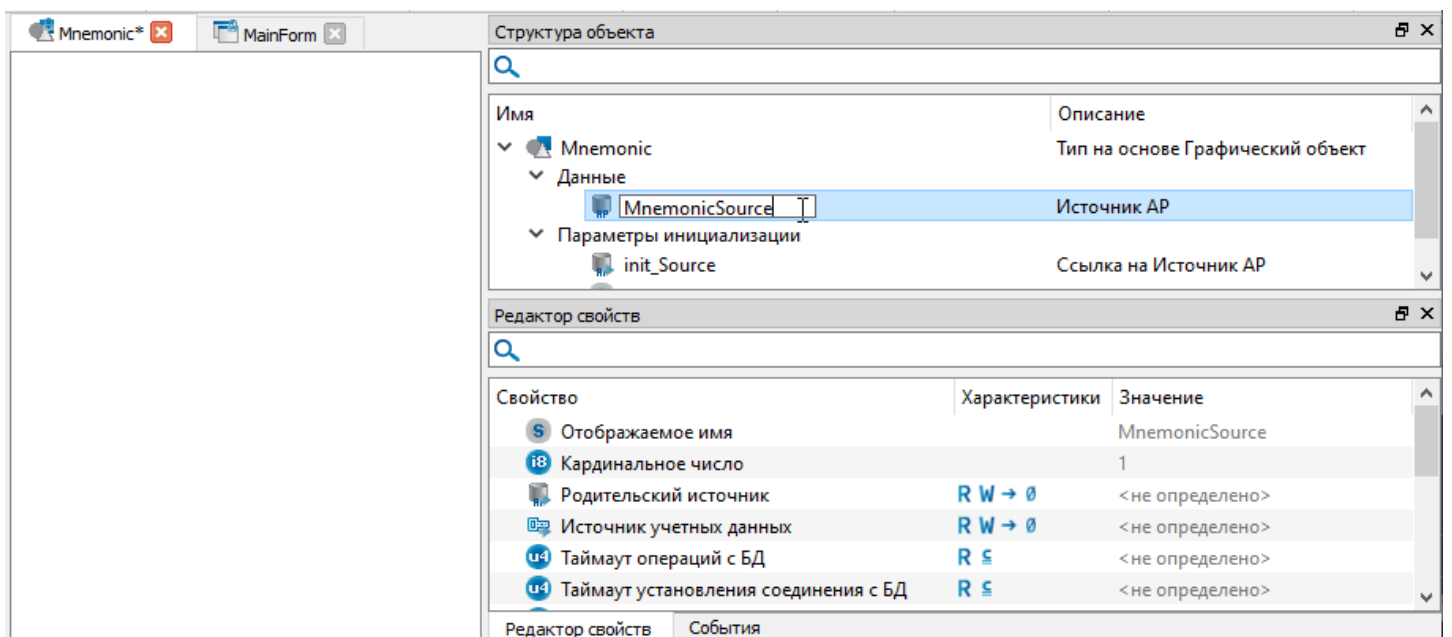
11. Во вкладке "Параметры инициализации" Переименуйте созданный тип "string\_1" при помощи клавиши "F2" в "init\_Path".



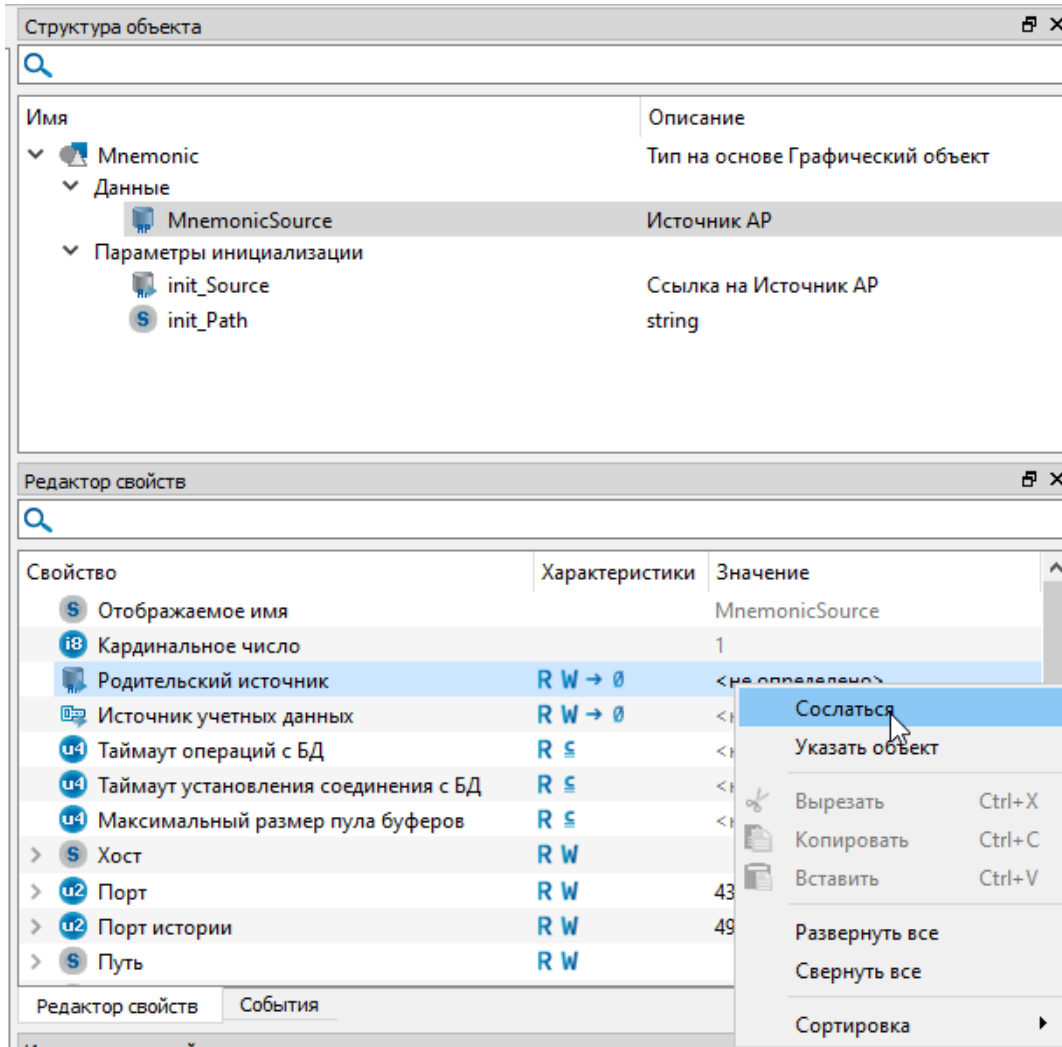
12. Во вкладке "Общие элементы" раскройте вкладку "АР" и путем перетаскивания добавьте элемент "Источник АР".



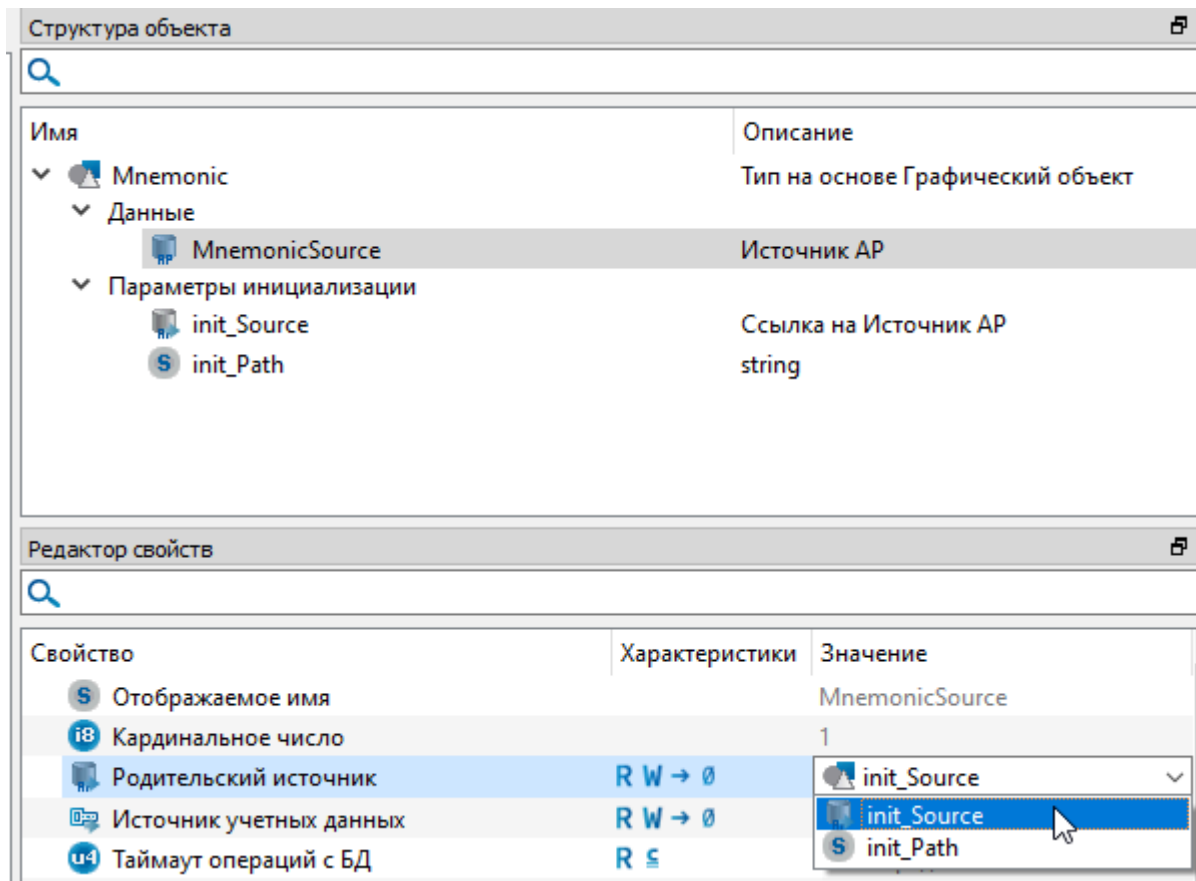
13. Во вкладке "Данные" окна "Структура объекта" переименуйте созданный тип "ApSource\_1" при помощи клавиши "F2" в "MnemonicSource".



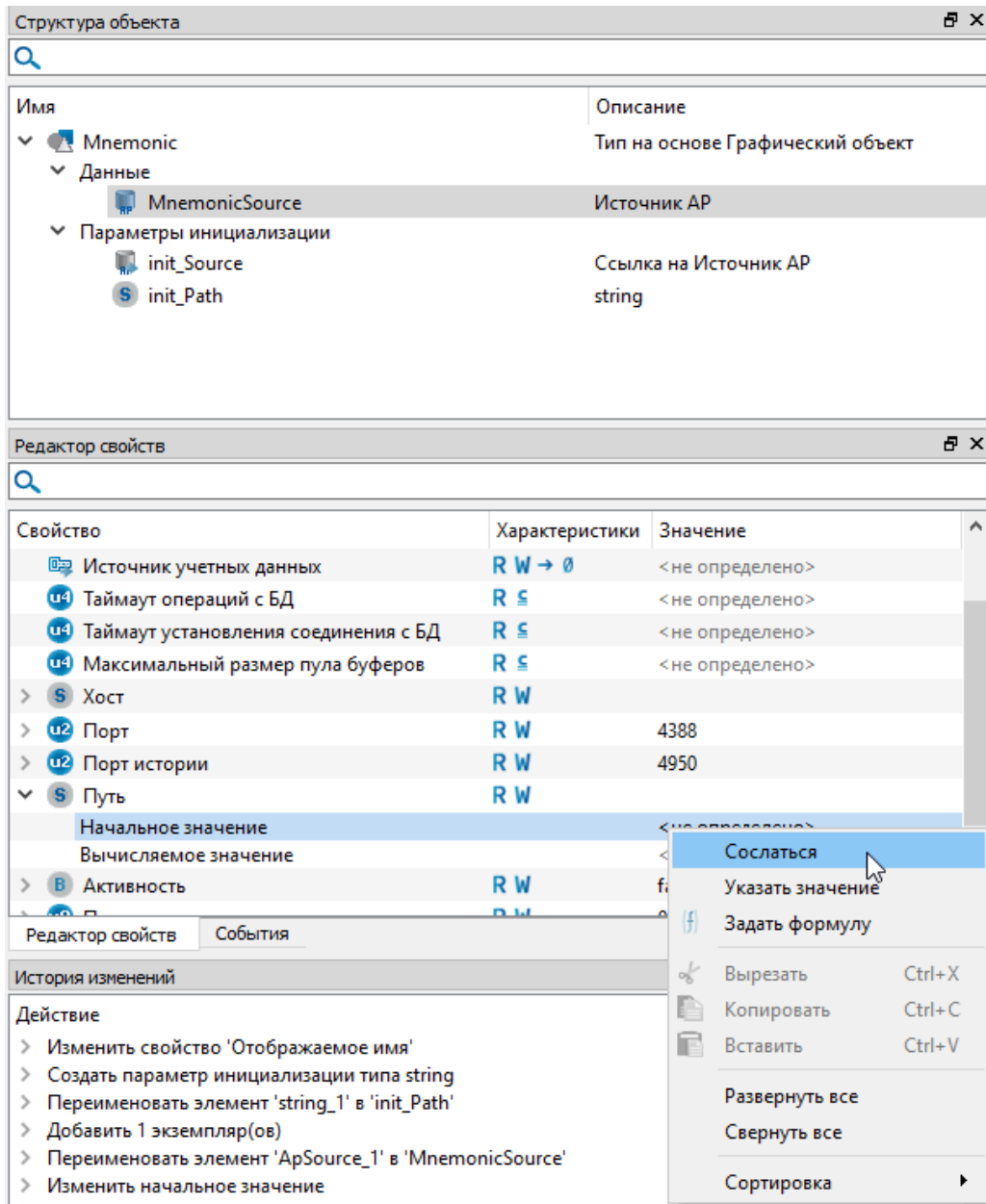
14. Выделите тип "MnemonicSource" нажав левой кнопкой мыши по нему. В окне "Редактор свойств" кликните правой кнопкой мыши по свойству "Родительский источник" в поле "Значение" и из контекстного меню выберите команду "Сослаться".



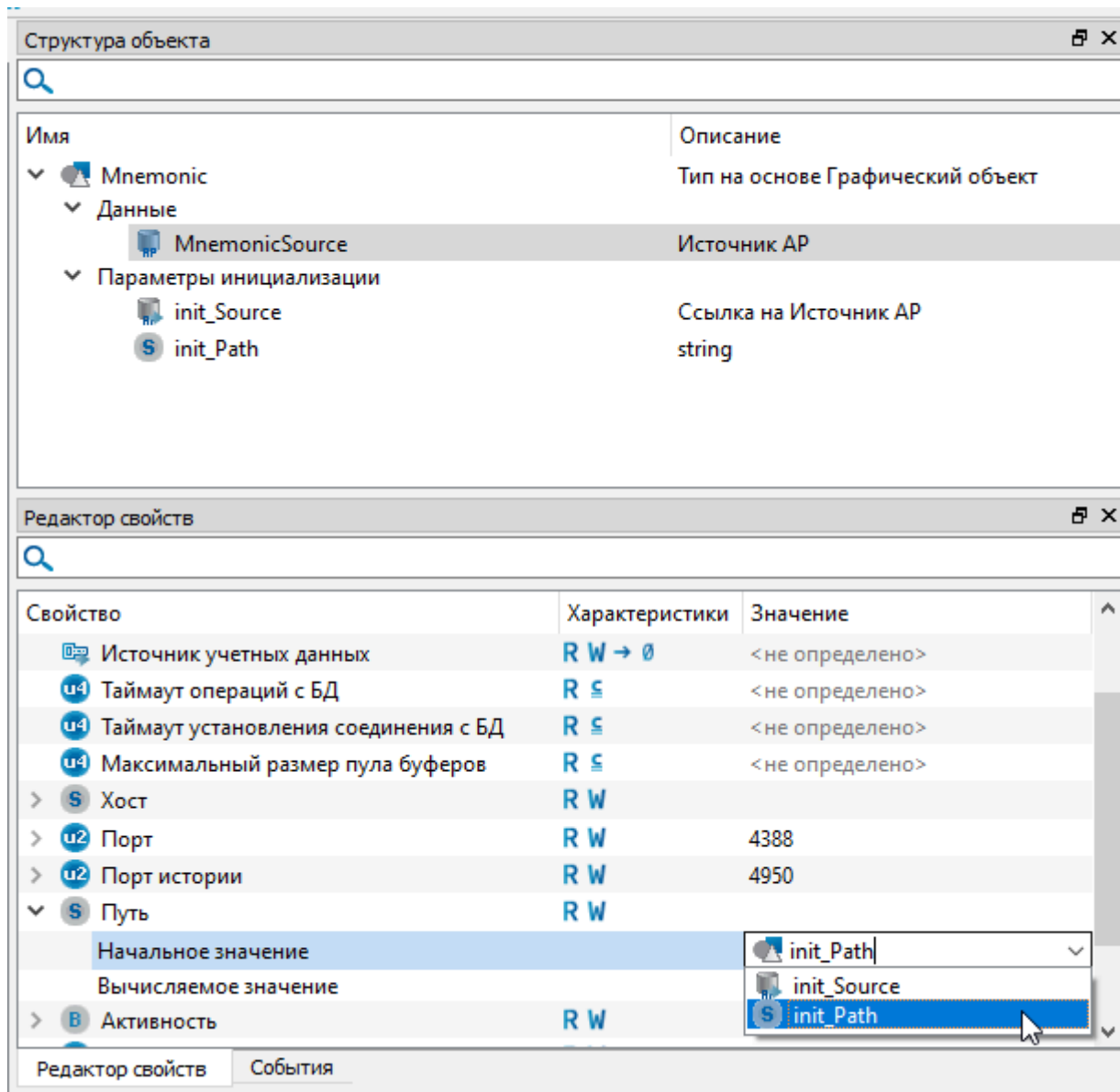
15. В появившемся поле ввода введите название параметра инициализации "init\_Source".



16. Раскройте свойство "Путь" кликнув по крыжику слева. Нажмите правой кнопкой мыши по строке "Начальное значение" и из контекстного меню выберите команду "Сослаться".



17. В появившемся поле ввода введите название параметра инициализации "init\_Path".



18. Установите свойство "Активность" в значение "true" и свойство "Период переподписки" в значение "1000."

Структура объекта

Имя | Описание

- Мnemonic
  - Данные
    - MnemonicSource | Источник AP
  - Параметры инициализации
    - init\_Source | Ссылка на Источник AP
    - init\_Path | string

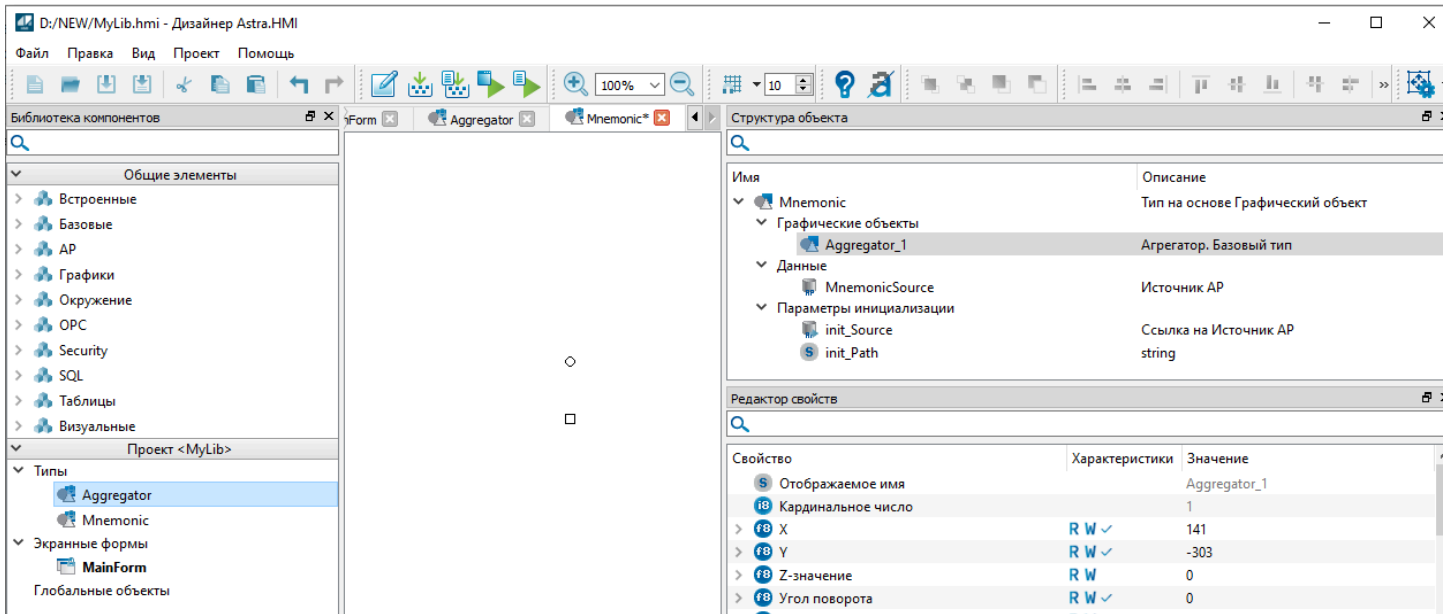
Редактор свойств

Свойство | Характеристики | Значение

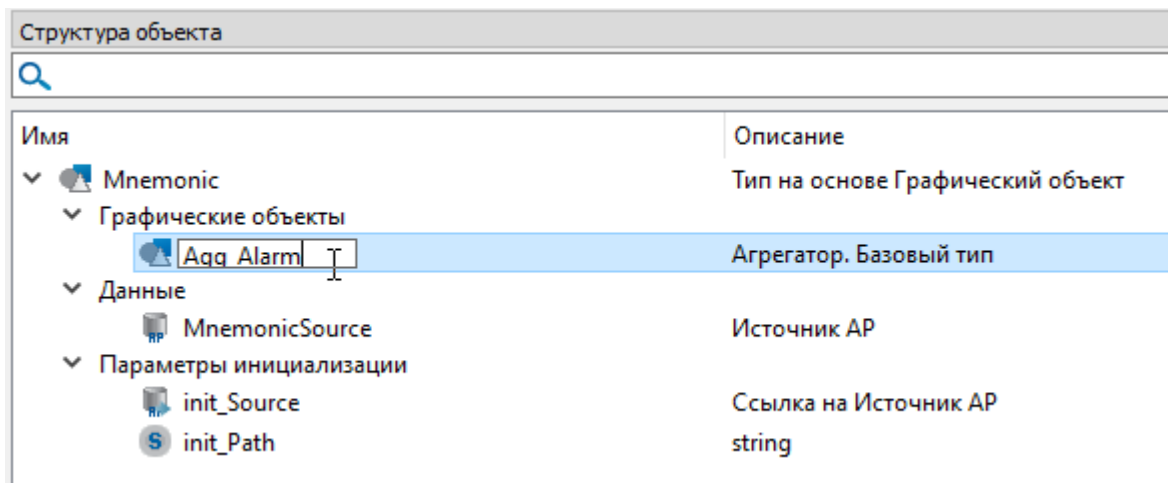
Свойство	Характеристики	Значение
Максимальный размер пула буферов	R E	<не определено>
Хост	R W	
Порт	R W	4388
Порт истории	R W	4950
Путь	R W	
Начальное значение	✓	init_Path
Вычисляемое значение		<не определено>
Активность	R W	true
Период переподписки	R W	1000
Имя клиента	R W	HMI
Идентификатор клиента	R W	HMI

Редактор свойств | События

19. Путем перетаскивания добавьте в "Mnemonic" ранее созданный базовый тип агрегатора "Aggregator" из вкладки "Типы".

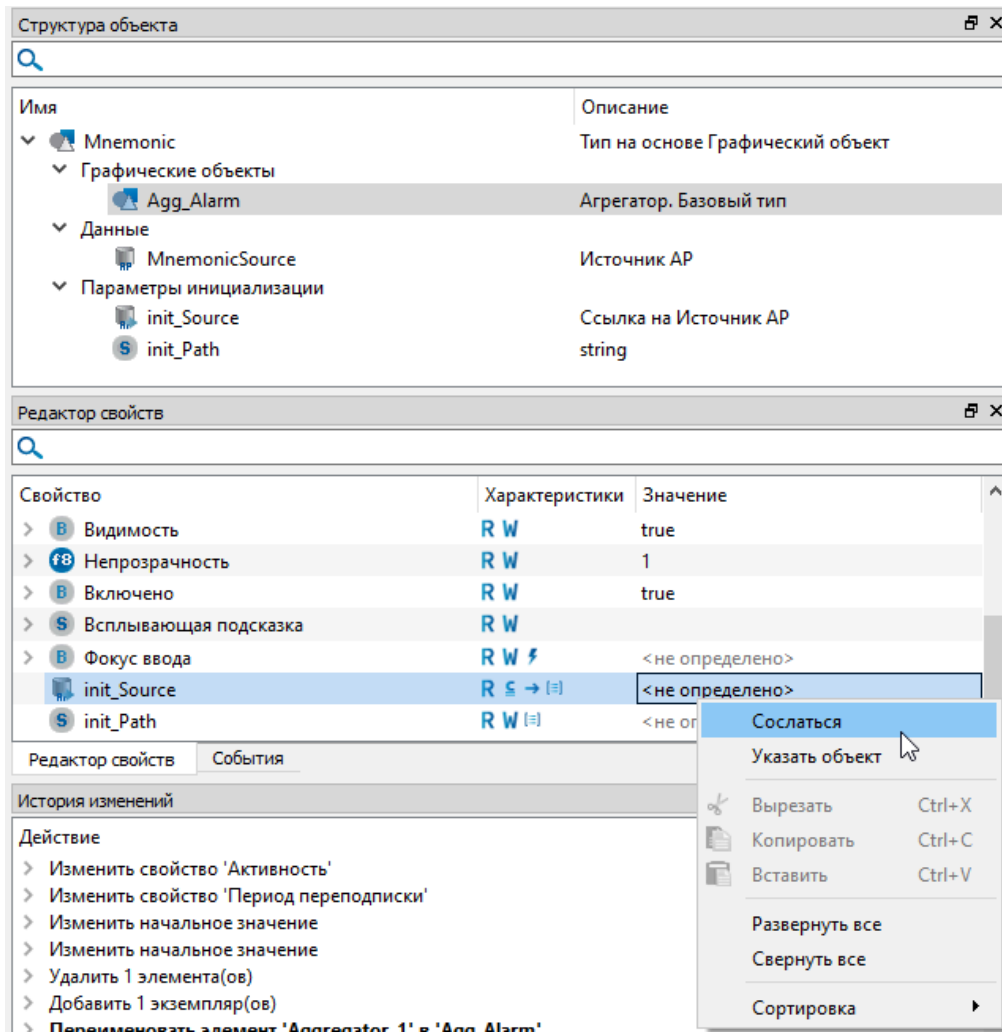


20. Во вкладке "Графические объекты" окна "Структура объекта" переименуйте добавленный тип "Aggregator\_1" при помощи клавиши "F2" в "Agg\_Alarm".

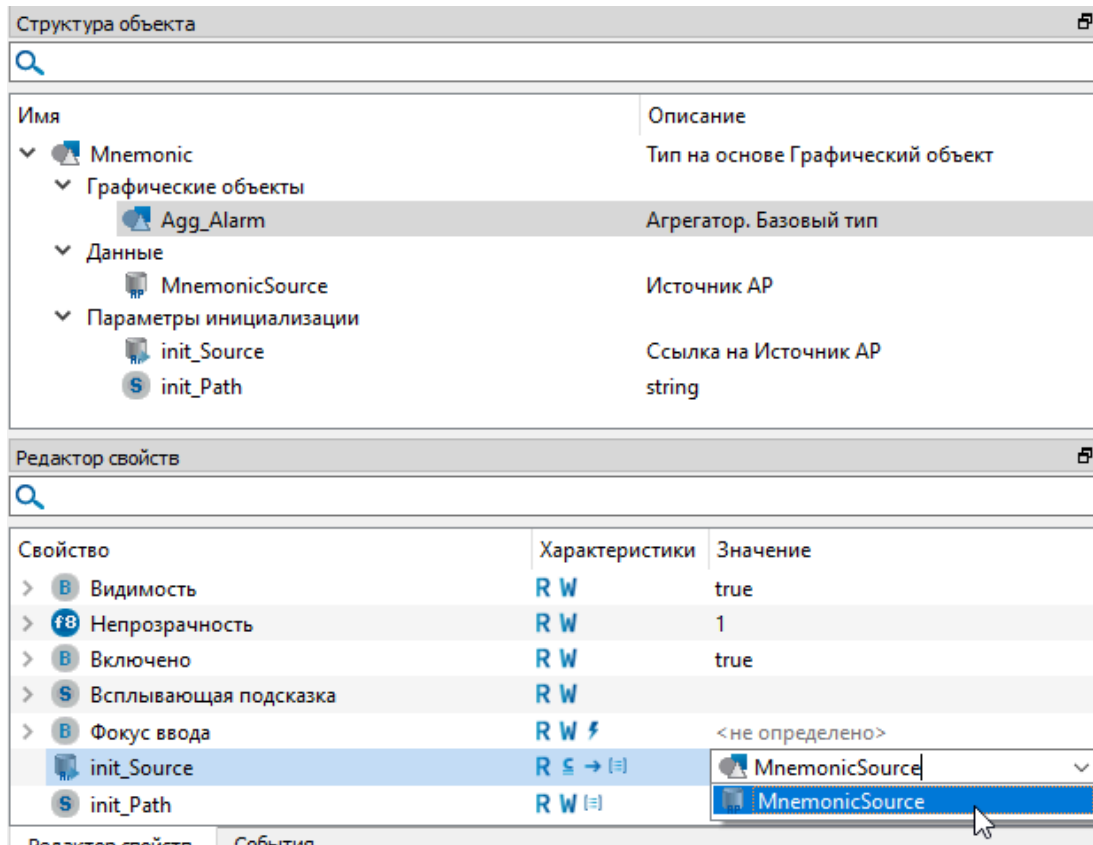


21. Выделите тип "Agg\_Alarm" левой кнопкой мыши и в окне "Редактор свойств" нажмите правой кнопкой мыши по свойству "init\_Source". В контекстном меню выберите команду "Сослаться".

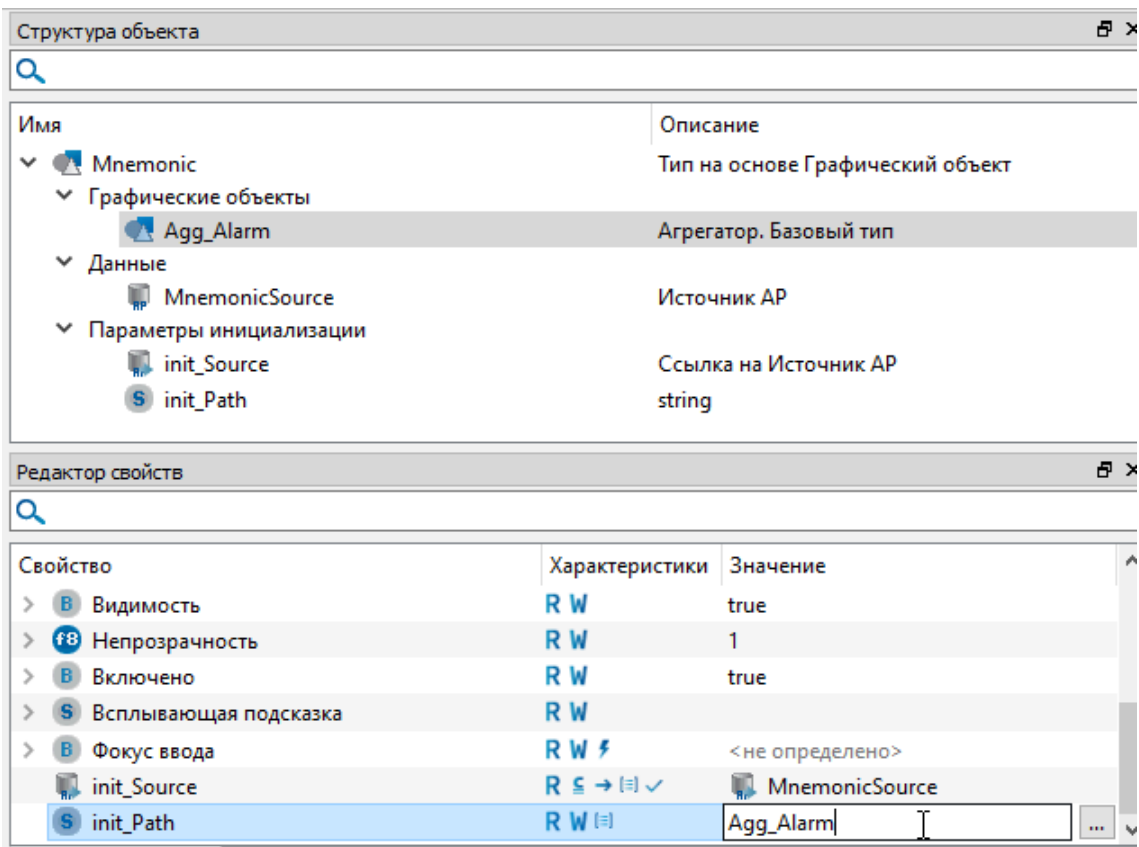




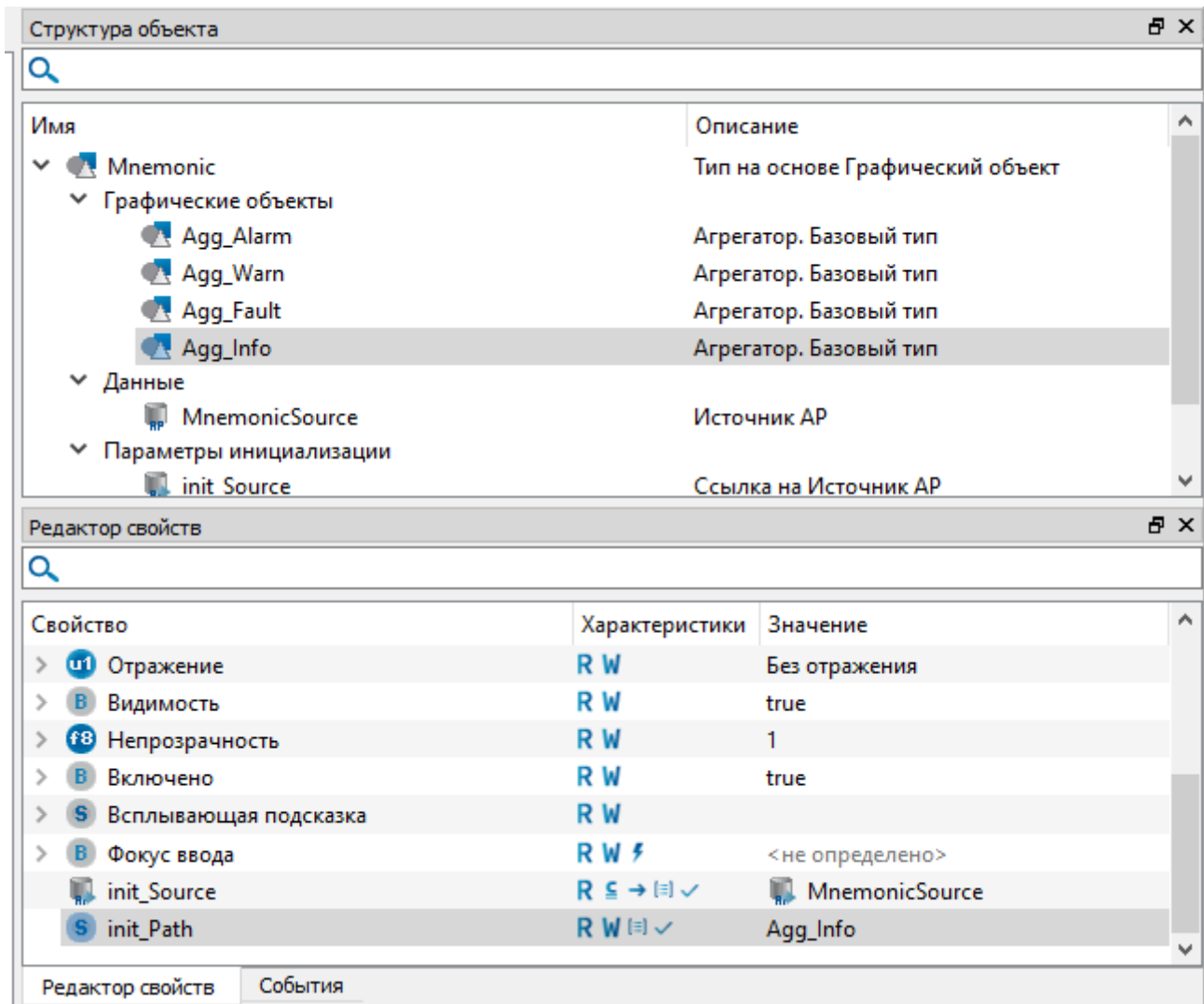
22. В появившемся поле ввода введите название источника данных "MnemonicSource".



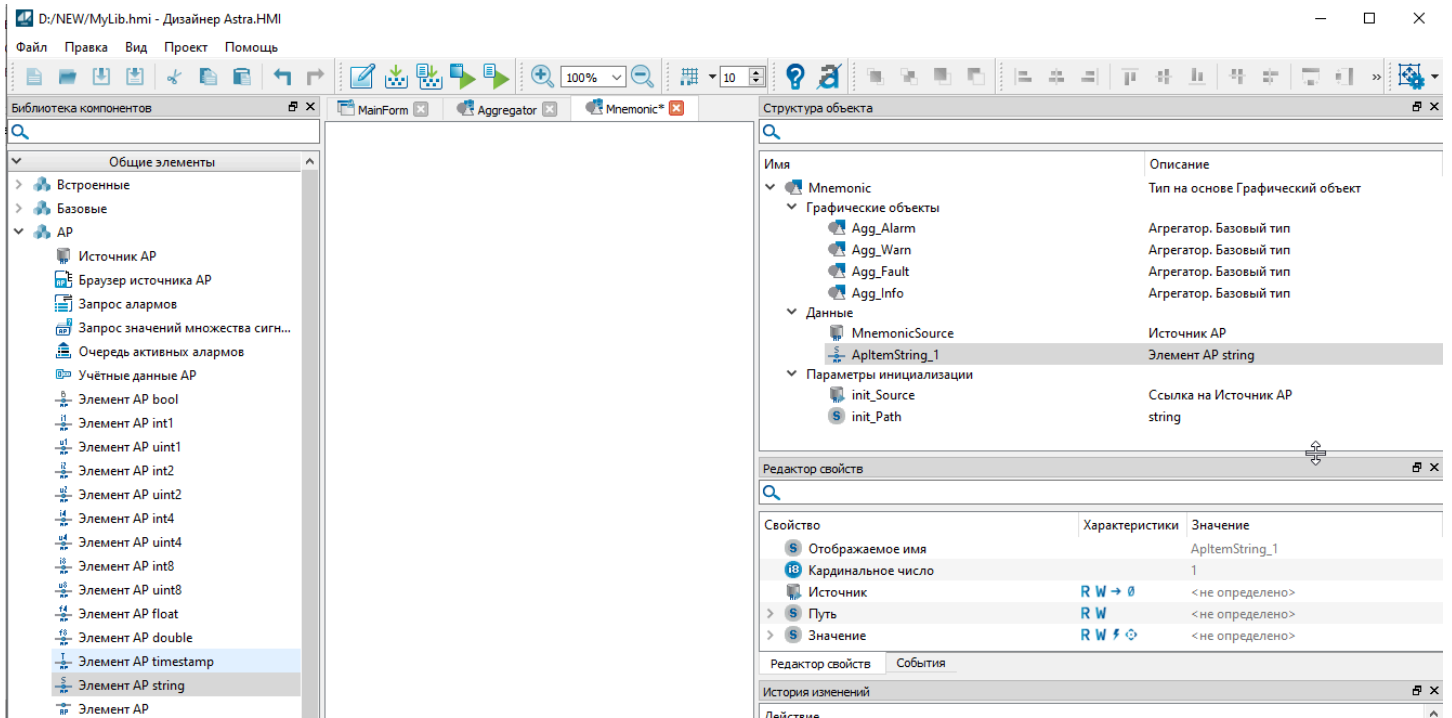
23. Кликните двойным кликом по свойству "init\_Path" в поле значение. В появившемся поле ввода укажите путь до агрегатора аварийных событий "Agg\_Alarm", который агрегирует аварийные события по важности.



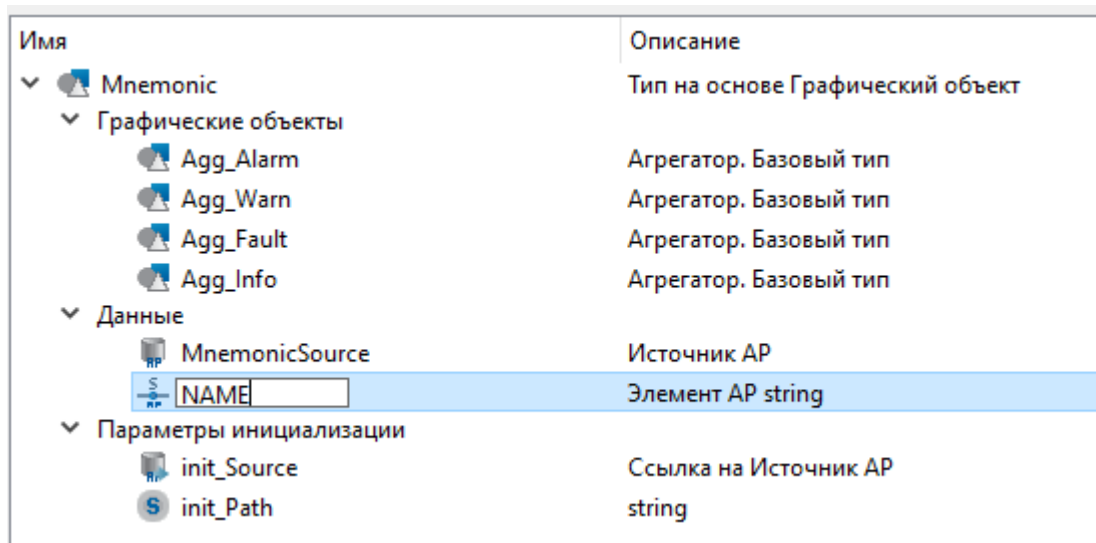
24. Аналогичным образом добавьте агрегаторы для других типов событий.



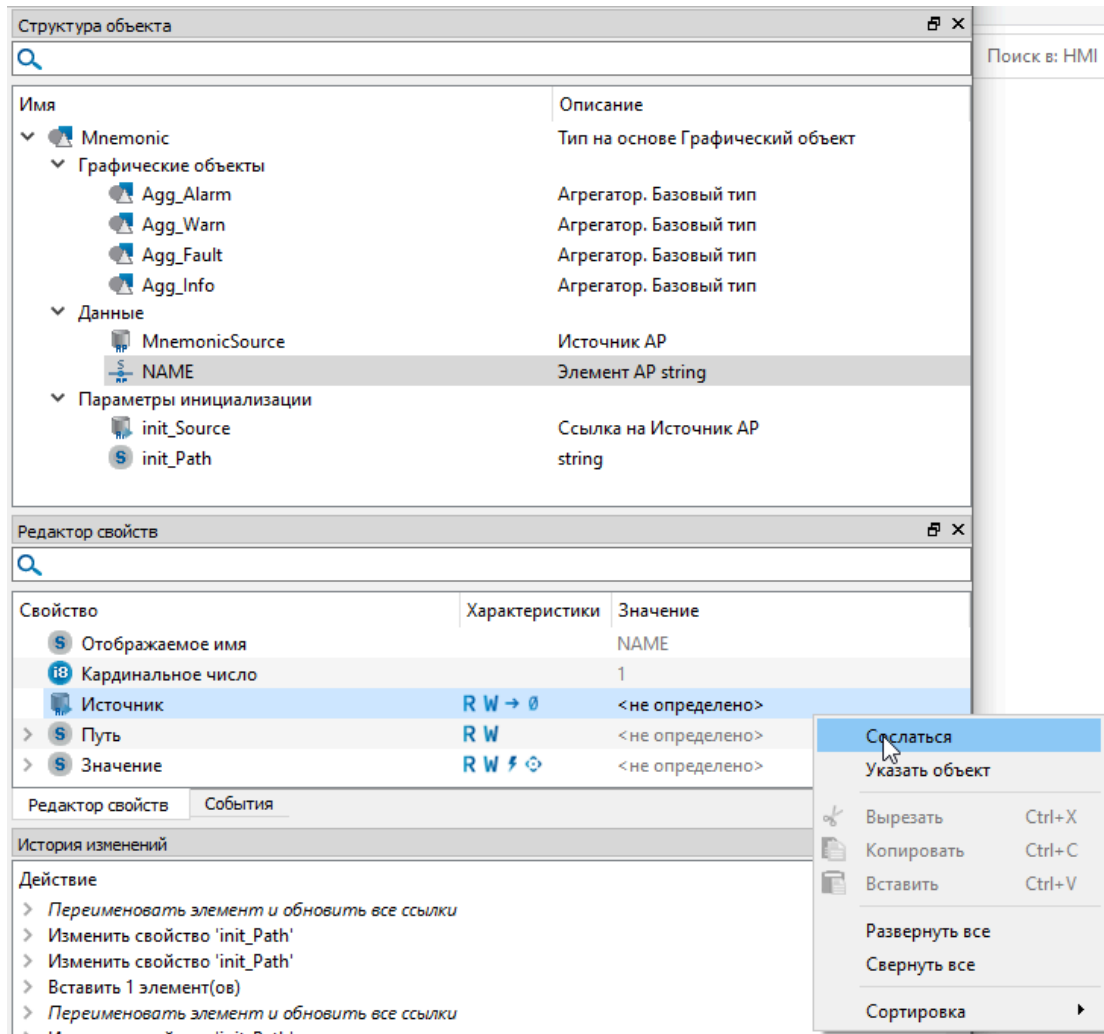
25. При необходимости в базовый тип вы можете добавить отдельные переменные, которые будут являться общими для всех мнемосимволов. Для этого во вкладке "Общие элементы" раскройте папку "AP" и путем перетаскивания добавьте "Элемент AP string".



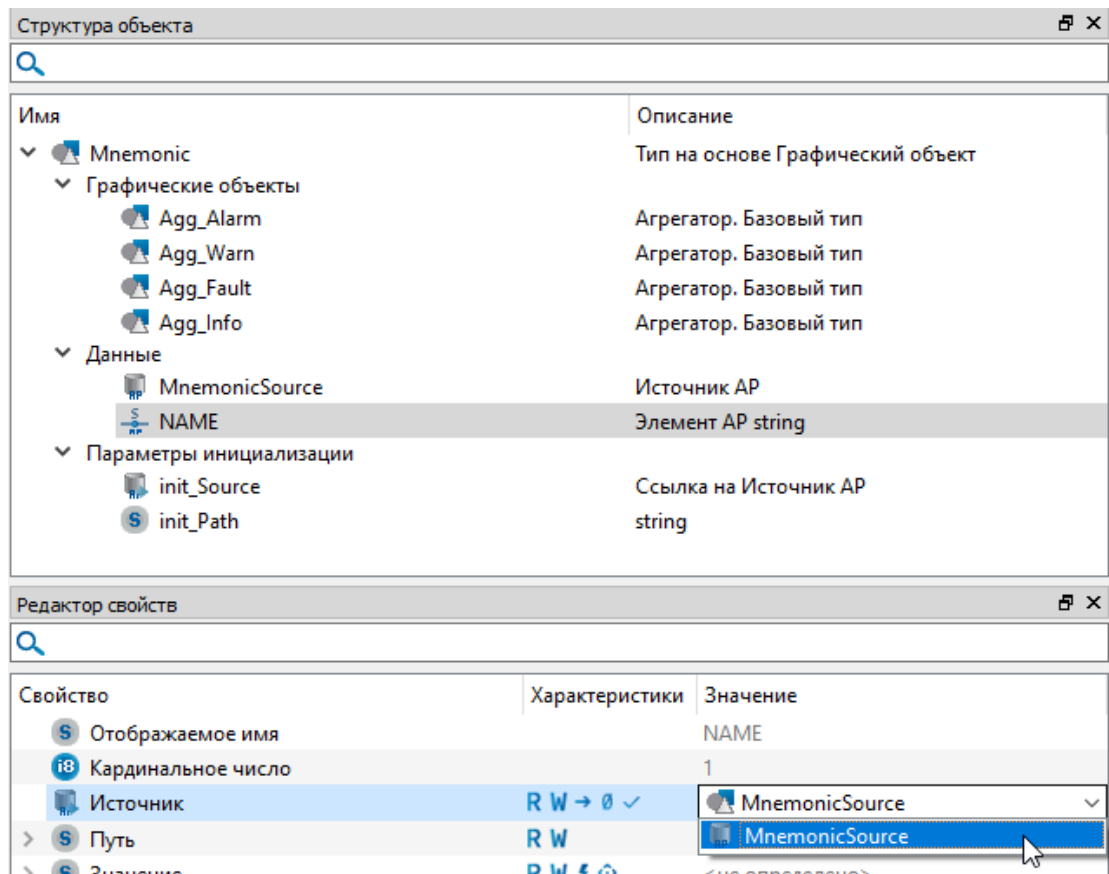
26. Во вкладке "Данные" окна "Структура объекта" переименуйте созданный тип "ApltemString\_1" при помощи клавиши "F2" в "NAME".



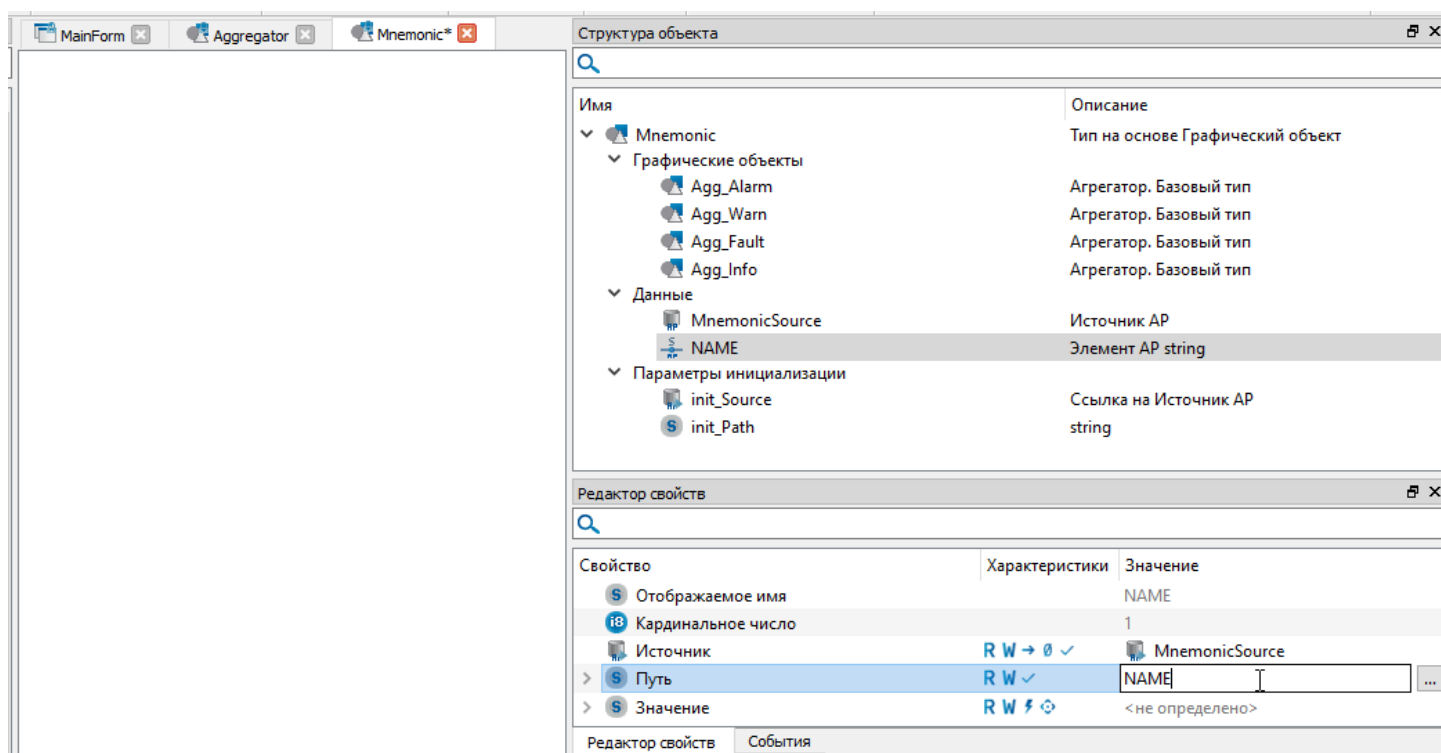
27. Выделите элемент "NAME" левой кнопкой мыши. В окне "Редактор свойств" кликните правой кнопкой мыши по свойству "Источник" в поле "Значение" и из контекстного меню выберите команду "Сослаться".



28. В появившемся поле ввода введите название источника "MnemonicSource".



29. Кликните двойным кликом по свойству "Путь" в поле значение. В появившемся поле ввода укажите путь до параметра "NAME", который отображает значение атрибута "Имя" у экземпляров в Astra.AStudio.

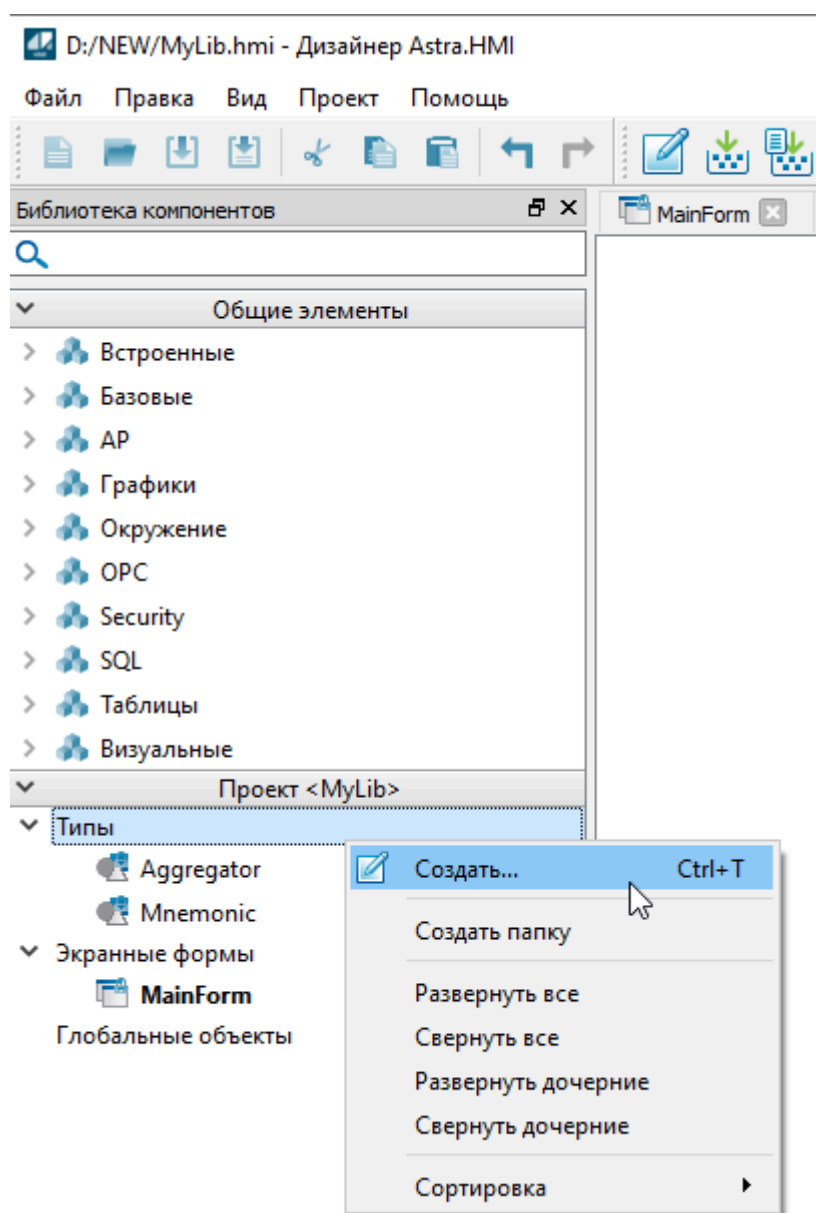


Создание базового типа мнемосимвола завершено.

## 1.2.4.4. Создание базового типа аналогового датчика

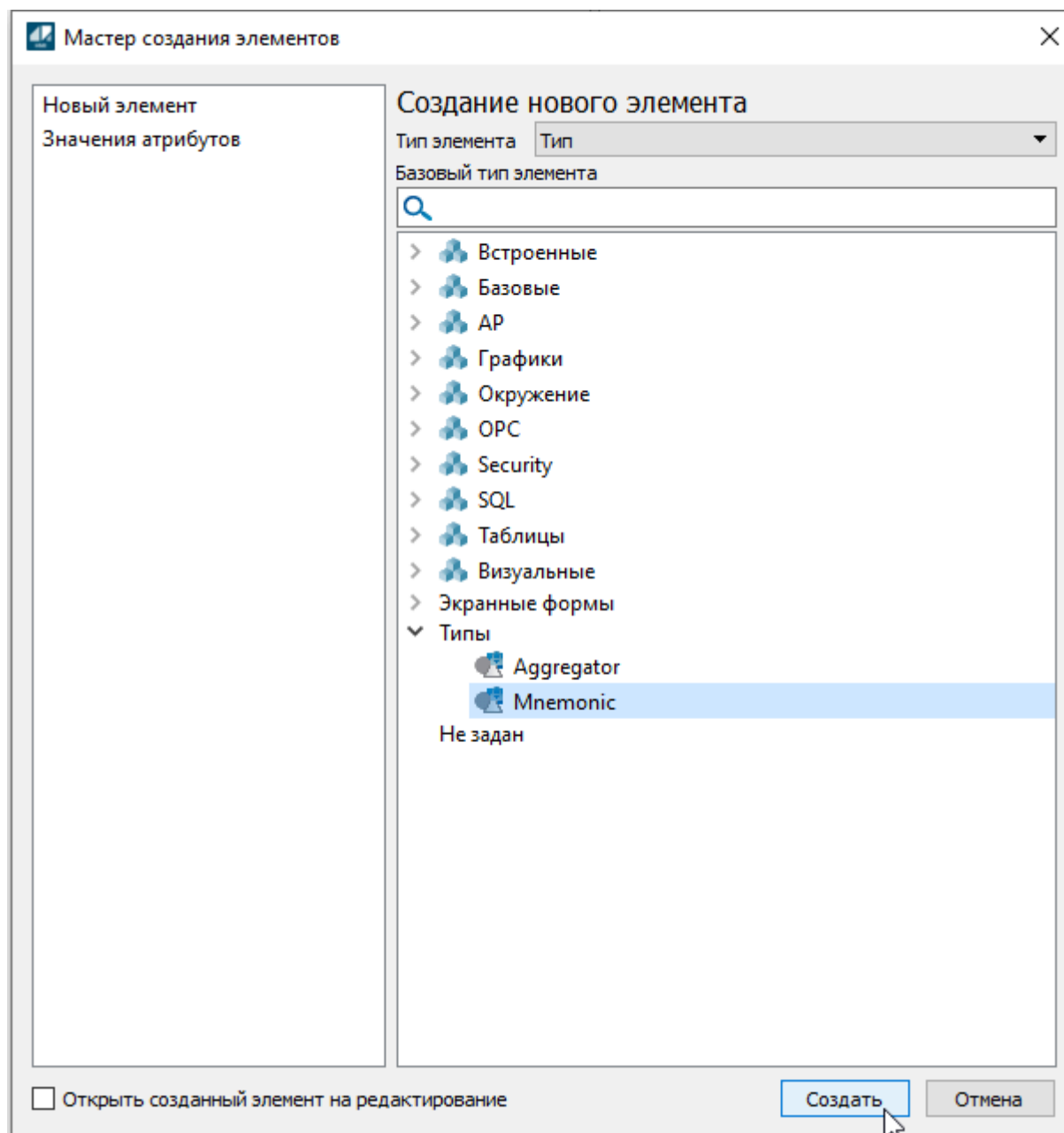
Для того, чтобы создать базовый тип аналогового датчика "AI1", необходимо выполнить следующие действия:

1. Нажмите правой кнопкой мыши по вкладке "Типы" и в контекстном меню выберите команду "Создать...".

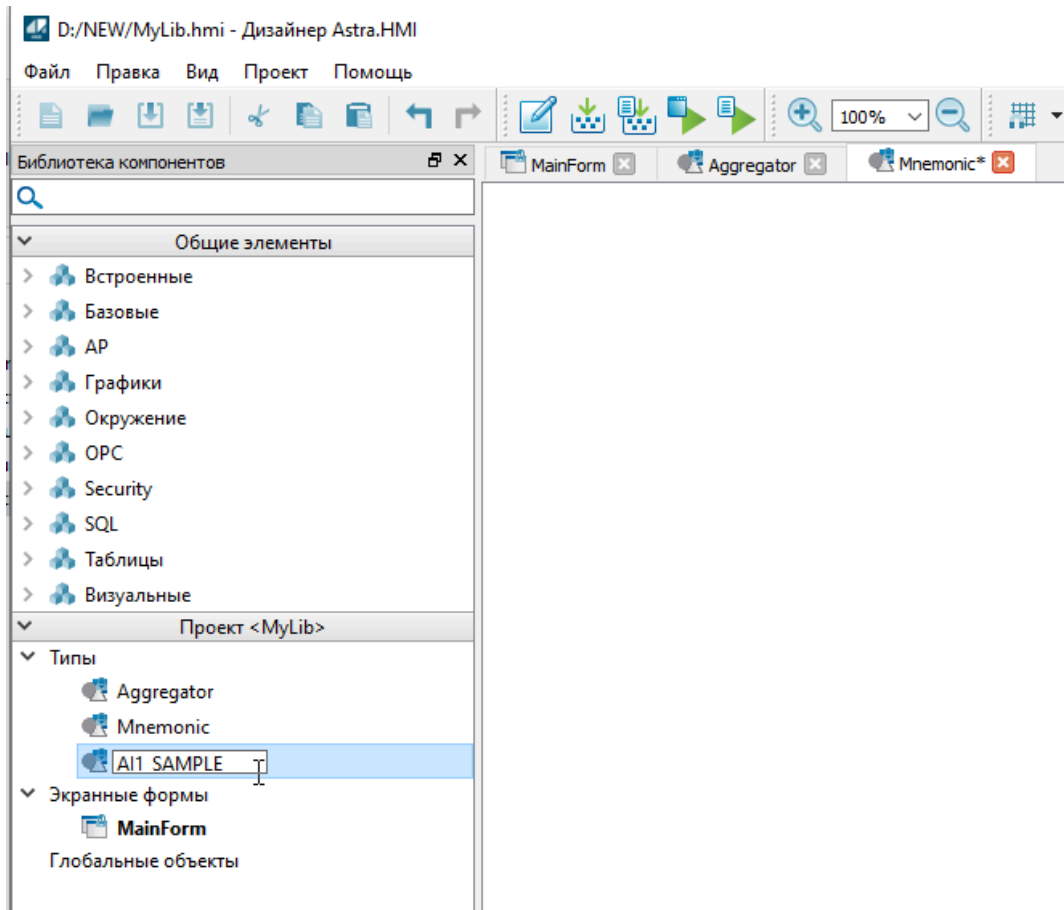


2. В мастере создания элементов раскройте вкладку "Типы" и выберите базовый тип элемента "Mnemonic". Нажмите кнопку "Создать".

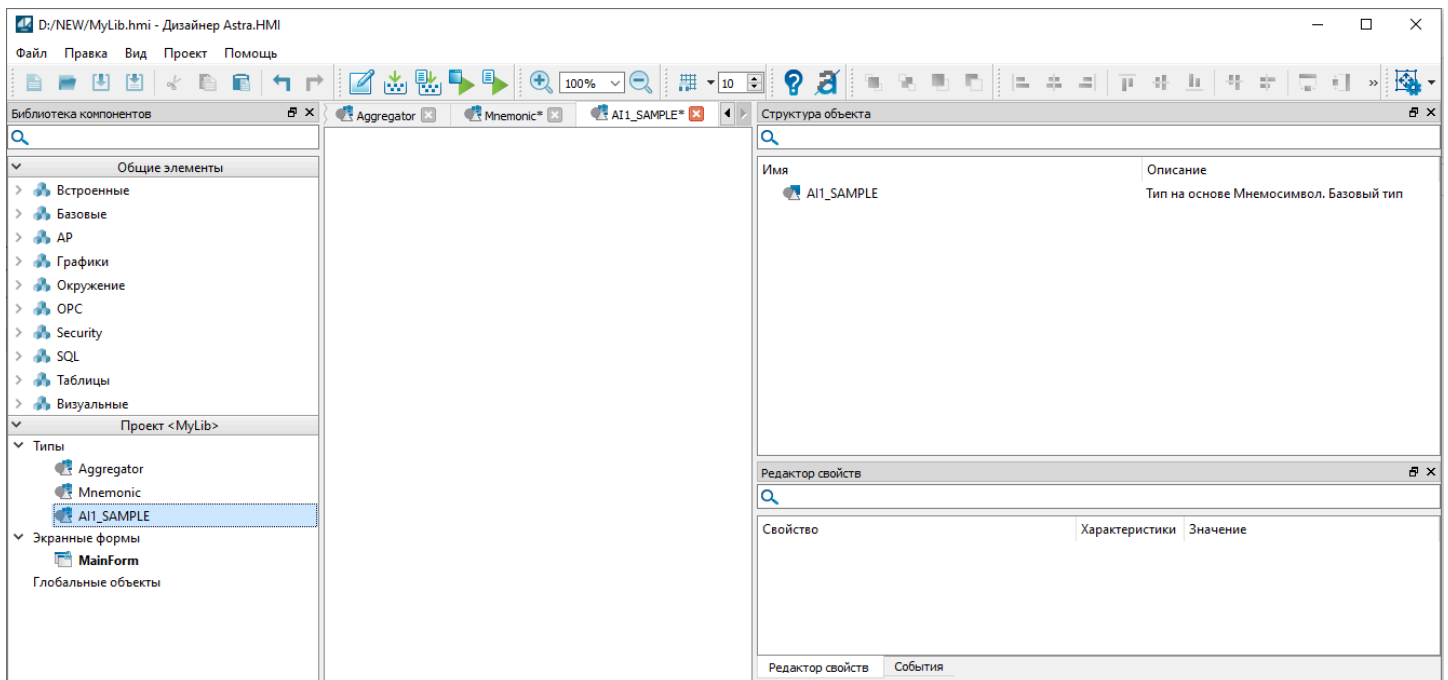




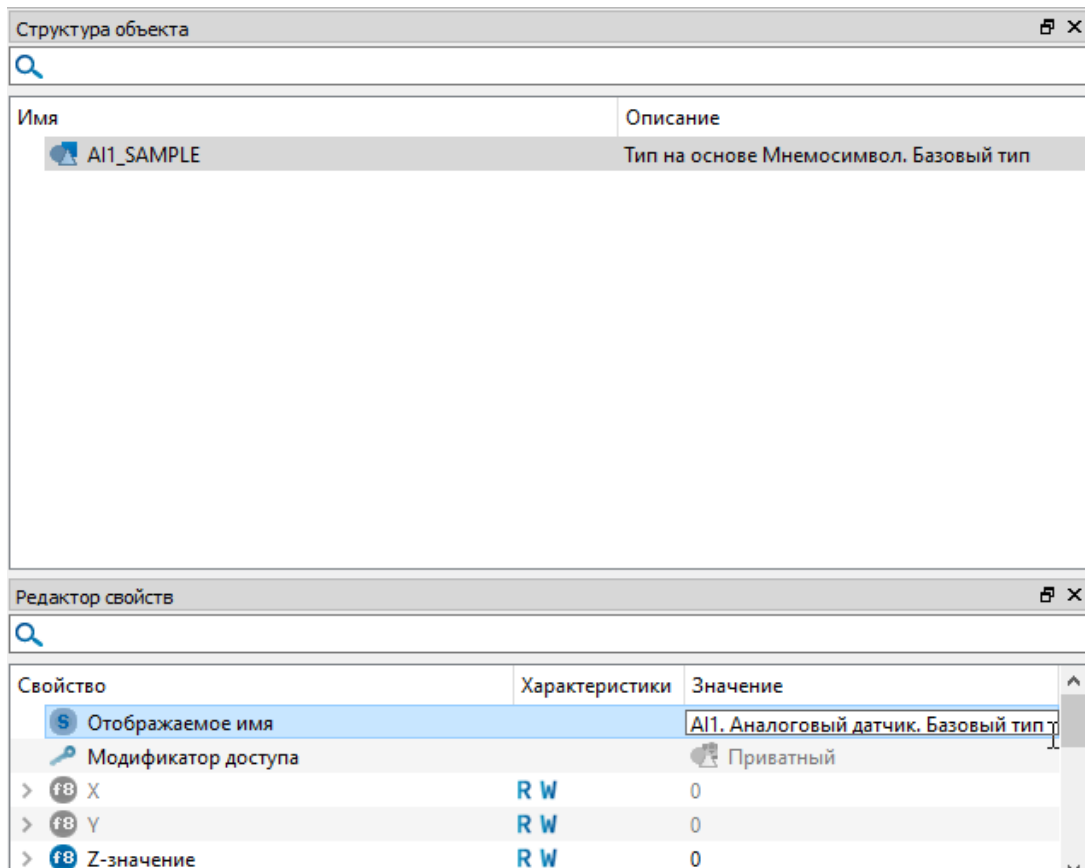
3. Во вкладке "Типы" появится новый графический объект с именем "Mnemonic\_1". Выделите тип левой кнопкой мыши, нажмите клавишу "F2" и задайте имя "AI1\_SAMPLE"



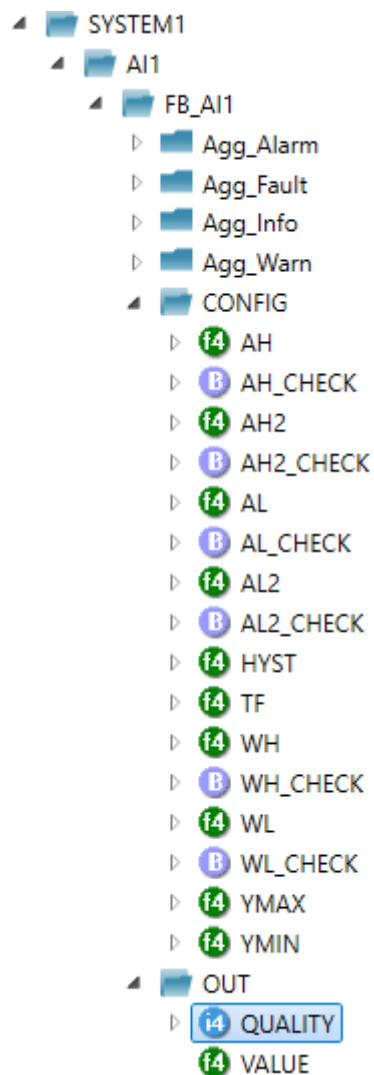
4. Перейдите в тип "AI1\_SAMPLE" двойным кликом левой кнопки мыши. В окне справа отобразится структура объекта.



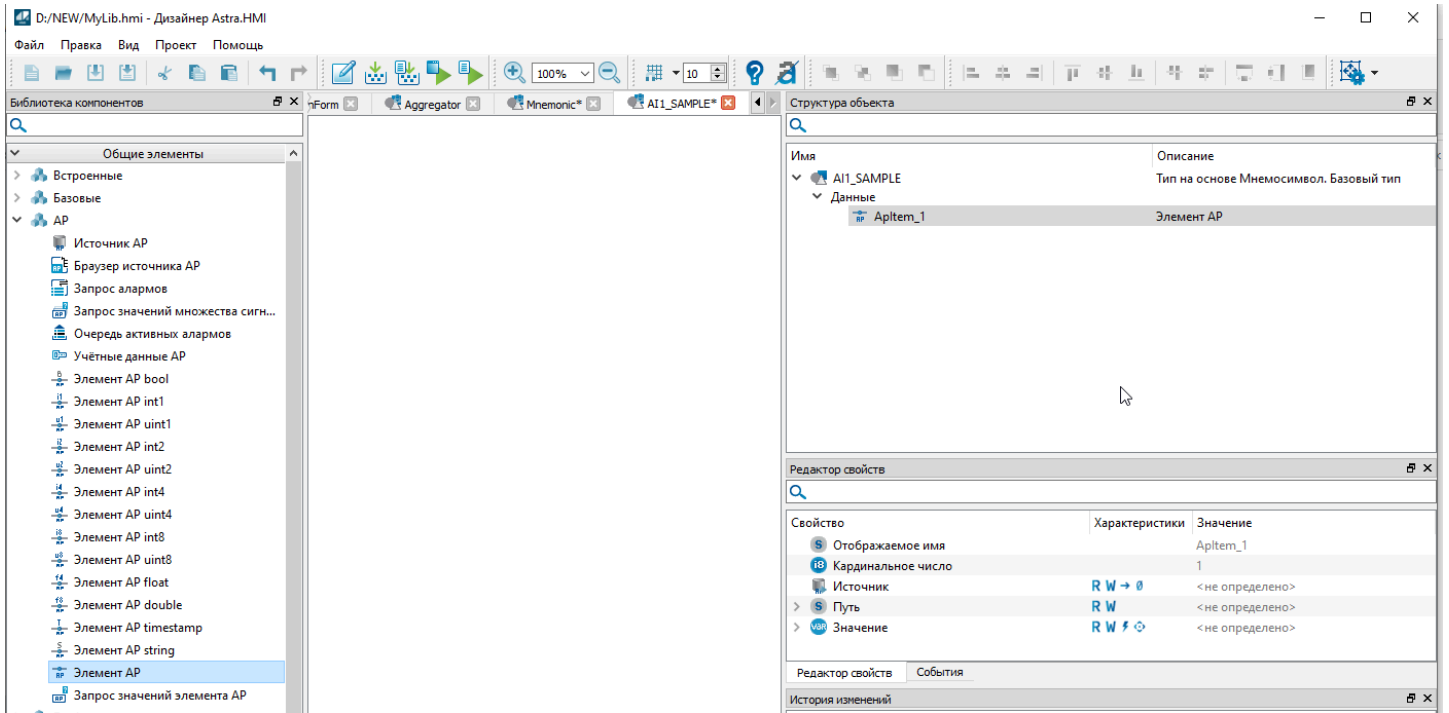
5. Выделите тип "AI1\_SAMPLE" левой кнопкой мыши и в окне "Редактор свойств" задайте отображаемое имя "AI1. Аналоговый датчик. Базовый тип".



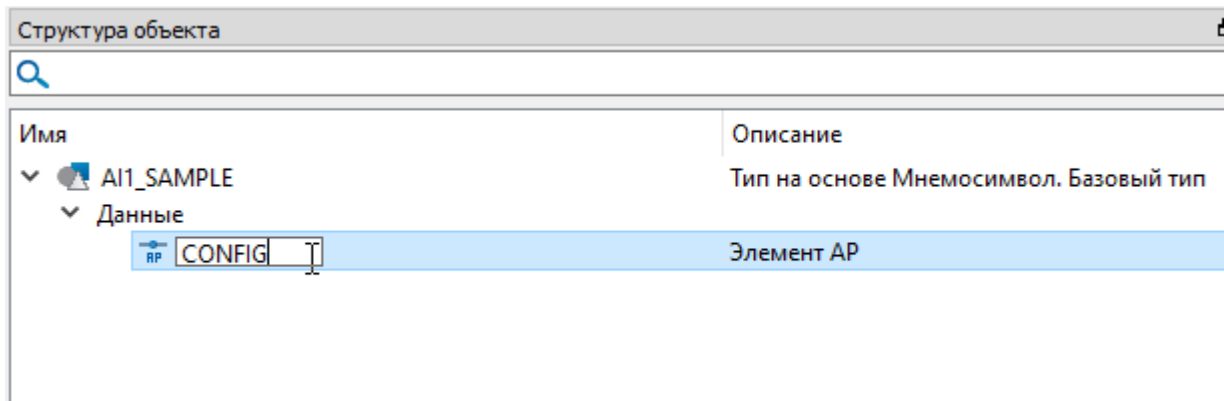
6. В базовый тип аналогового датчика необходимо добавить переменные из представления сервера ввода/вывода. Для таких целей удобнее всего использовать OrcExplorer, так как в нем можно сразу же увидеть вложенные объекты, а также папки, которые соответствуют сокетам в среде разработки Astra.AStudio.



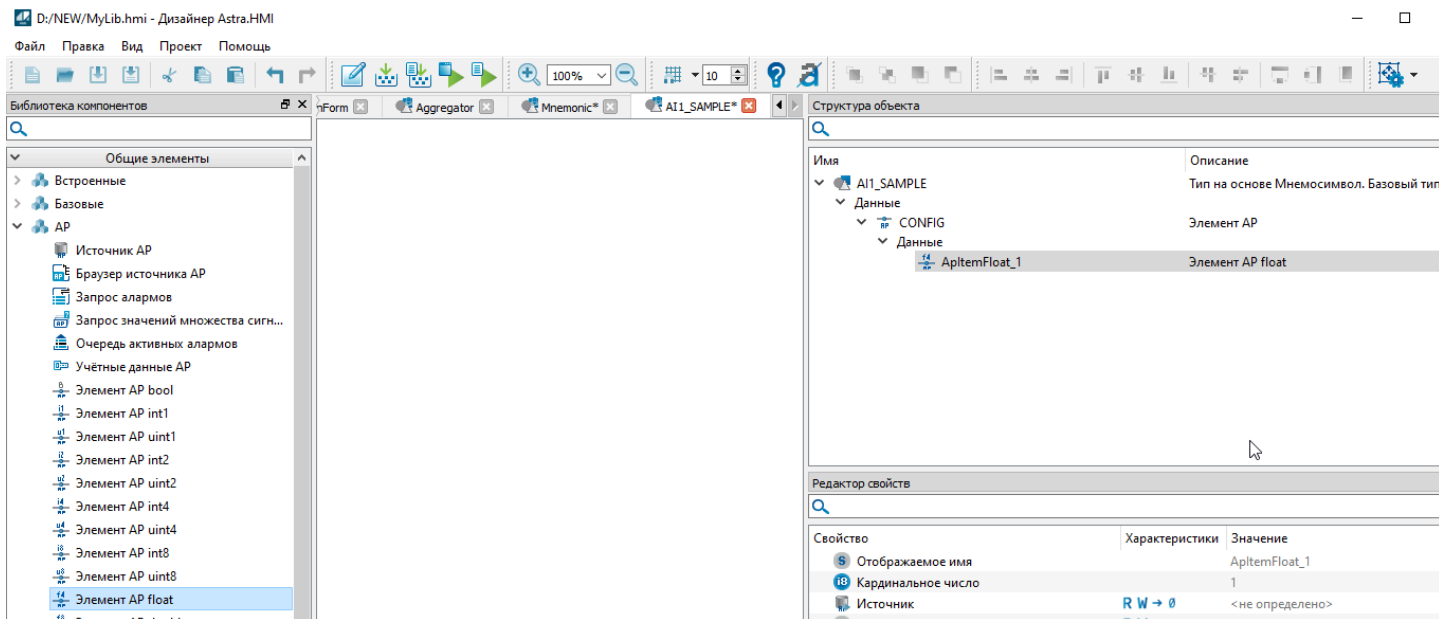
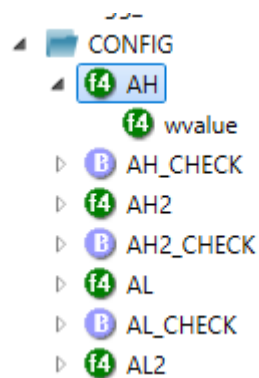
7. Добавим данные из папки "CONFIG" в базовый тип AI1. Для создания папки раскройте папку "Общие элементы", далее перейдите во вкладку "AP" и добавьте элемент "Элемент AP" путем перетаскивания.



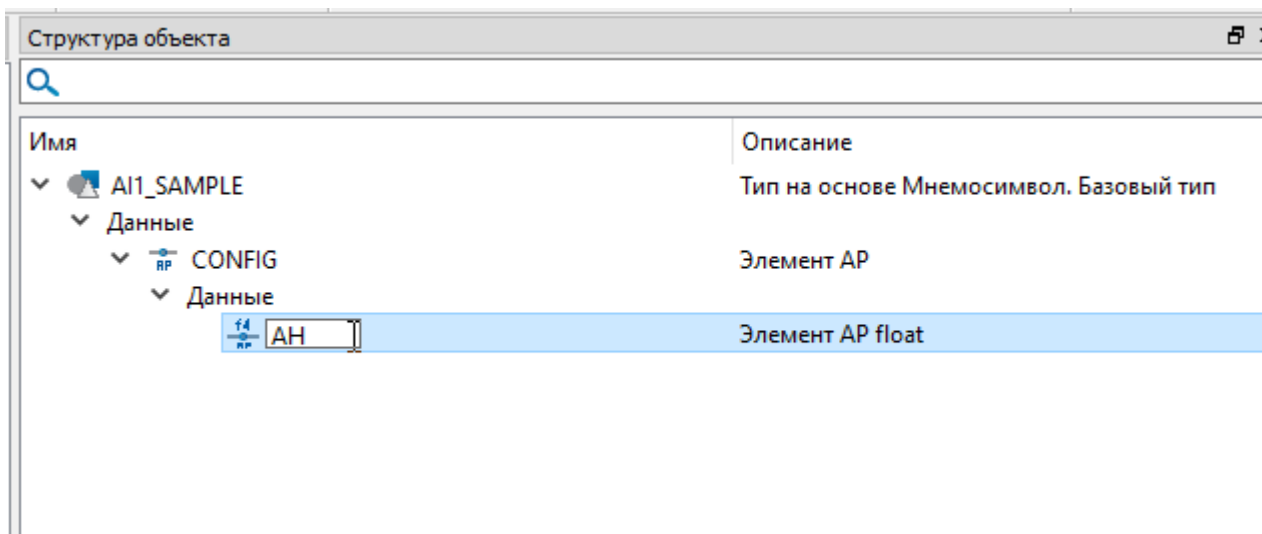
8. Во вкладке "Данные" окна "Структура объекта" переименуйте созданный тип "Apltem" при помощи клавиши "F2" в "CONFIG".



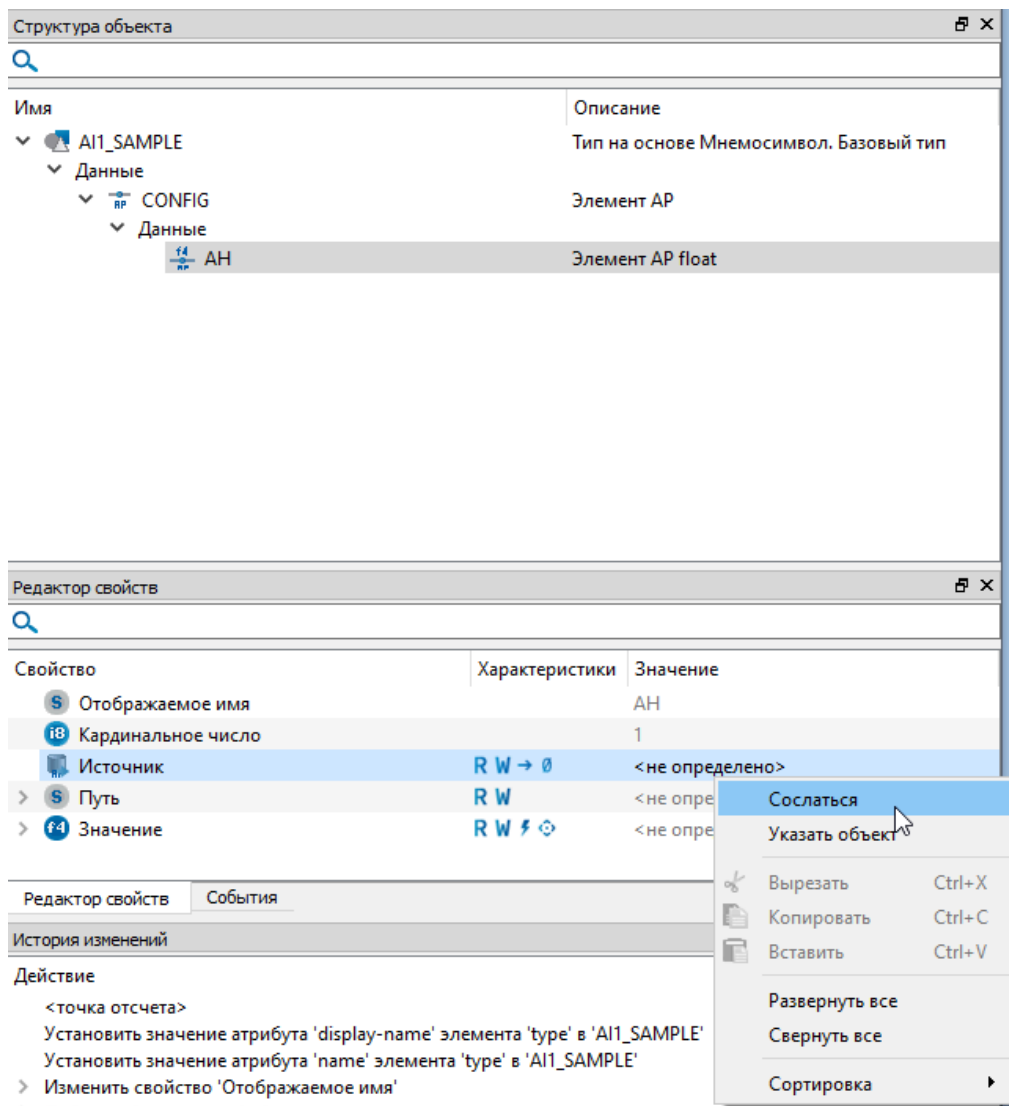
9. Для добавления переменных в папку "CONFIG" путем перетаскивания в тип "Элемент AP" добавьте элементы тех типов, которые отображаются в OrcExplorer, например "АН" (типа float).



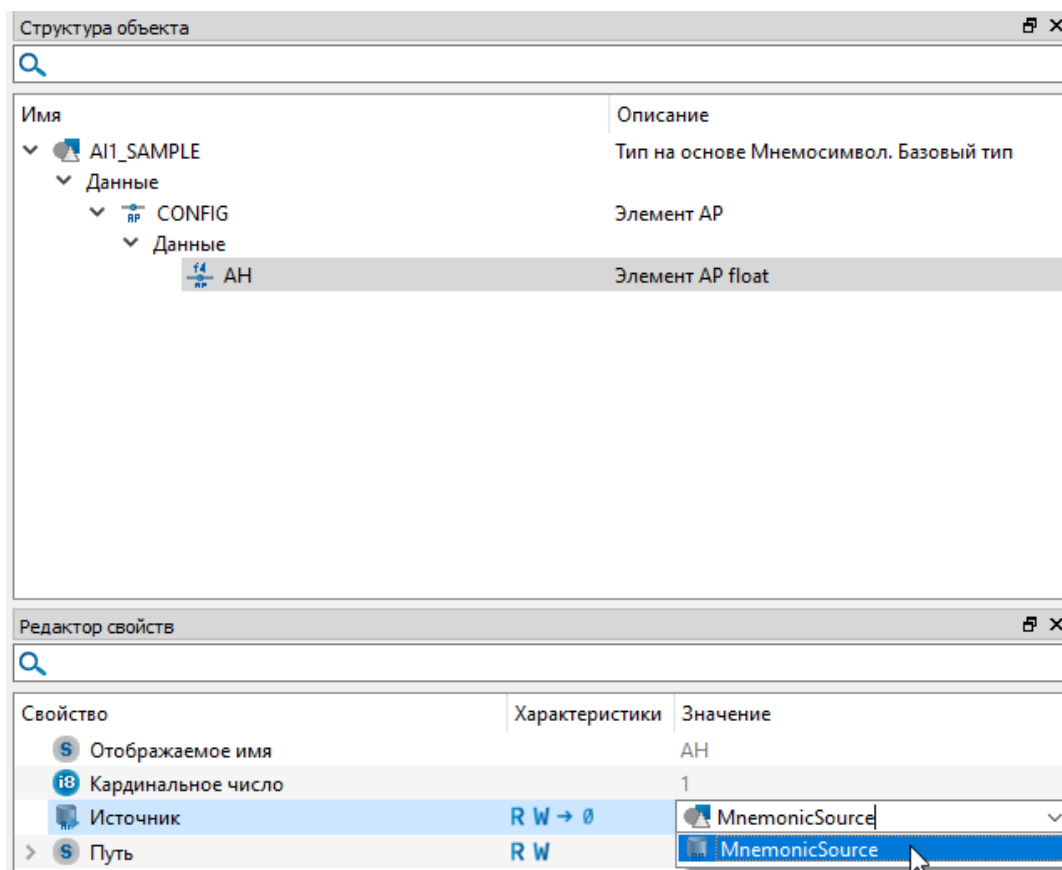
10. Переименуйте созданный тип "ApItem\_Float1" при помощи клавиши "F2" в "AH".



11. Выделите элемент "АН" левой кнопкой мыши. В окне "Редактор свойств" кликните правой кнопкой мыши по свойству "Источник" в поле "Значение" и из контекстного меню выберите команду "Сослаться".



12. В появившемся поле ввода введите название источника базового типа "MnemonicSource".



13. Кликните двойным кликом по свойству "Путь" в поле значение. В появившемся поле ввода укажите путь до параметра "АН", который отображает значение верхней аварийной уставки аналогового датчика.



Структура объекта

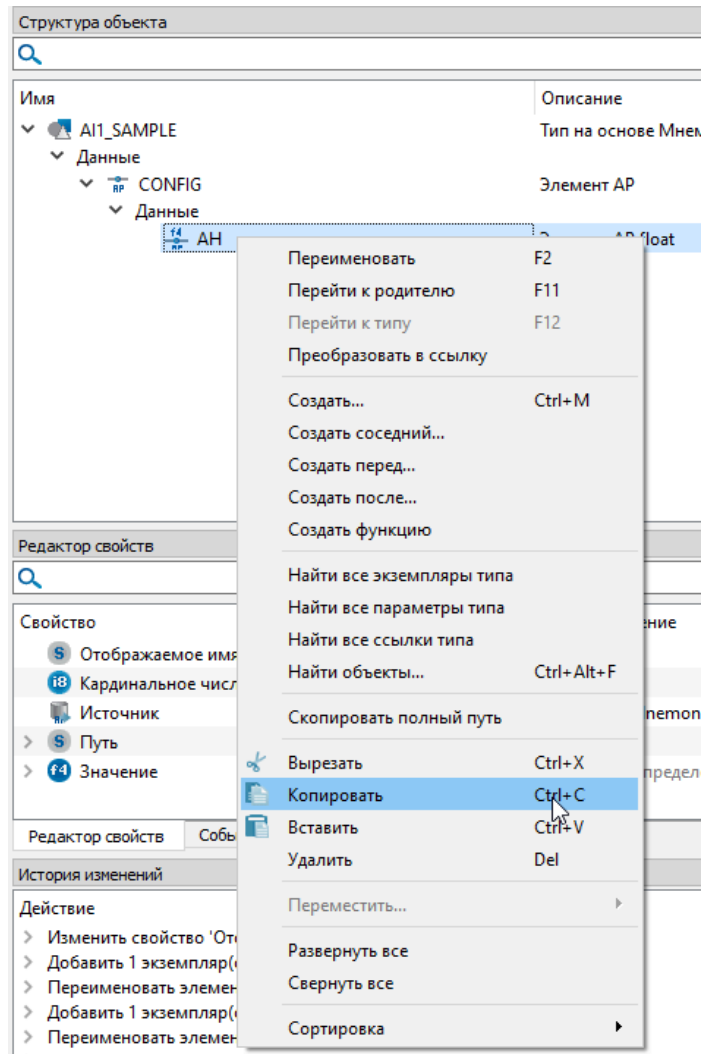
Имя

Имя	Описание
AI1_SAMPLE	Тип на основе Мнемосимвол. Базовый ти
Данные	
CONFIG	Элемент AP
Данные	
AH	Элемент AP float

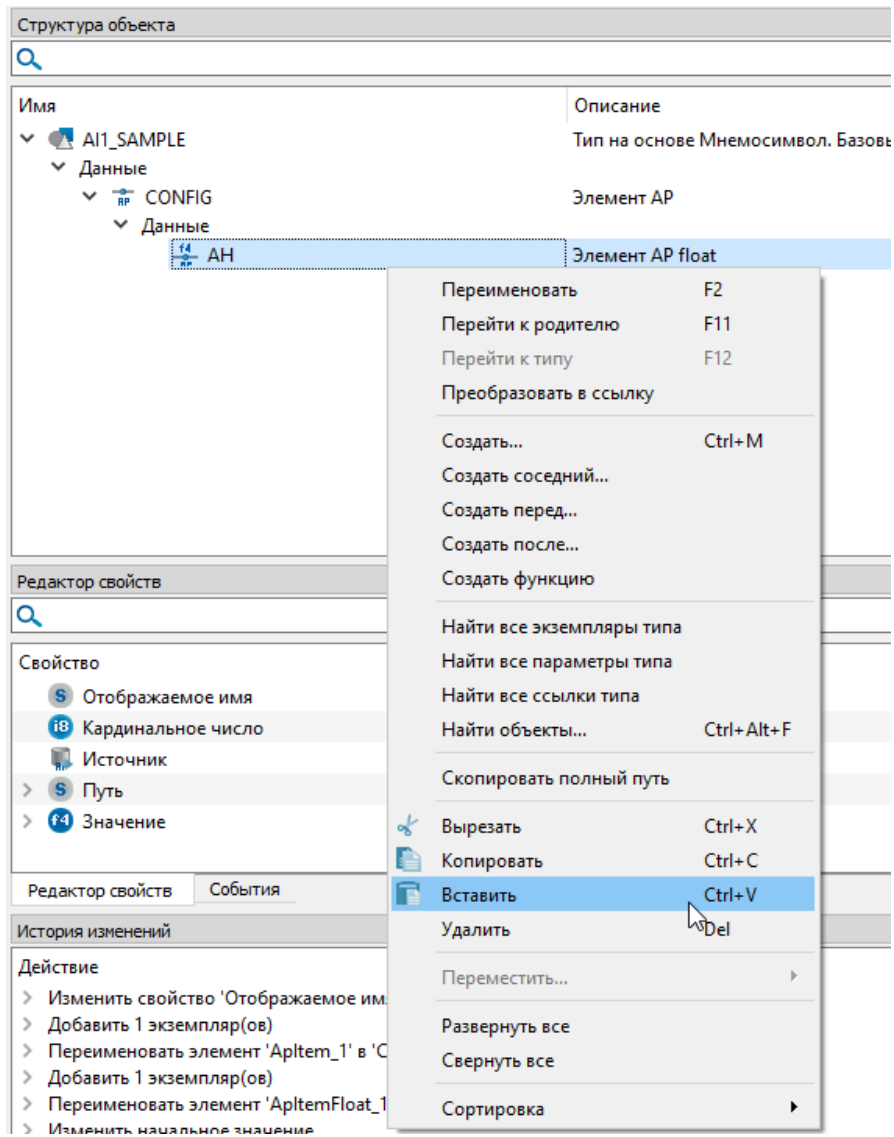
Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		AH
Кардинальное число		1
Источник	R W → 0 ✓	MnemonicSource
Путь	R W ✓	AH
Значение	R W ↻	<не определено>

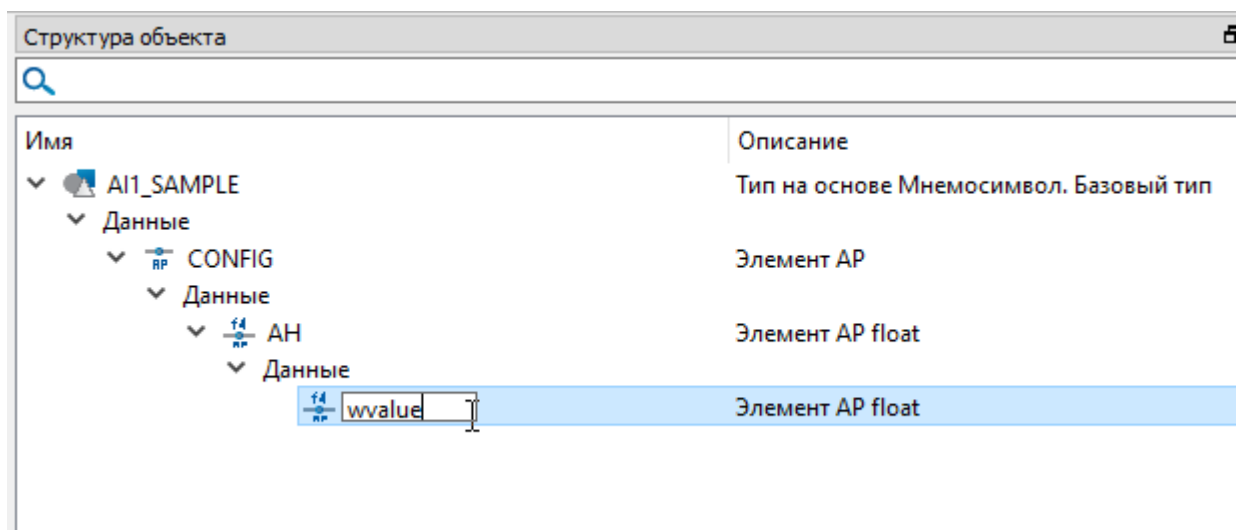
14. У всех уставок есть фактическое значение и значение задания. Поэтому необходимо повторить структуру вложенности параметров, аналогично как в OrcExplorer. Нажмите правой кнопкой мыши по уставке "AH" и в контекстном меню выберите команду "Копировать".



15. Еще раз нажмите правой кнопкой мыши по уставке "АН" и в контекстном меню выберите команду "Вставить".



16. Переименуйте вложенный элемент в "wvalue", как это выглядит в OrcExplorer, воспользовавшись клавишей F2.



17. Выделите элемент "wvalue" и в свойстве "Путь" задайте следующее значение:

The image shows two windows from a software application. The top window, titled "Структура объекта" (Object Structure), displays a tree view of the object hierarchy. The path is: AI1\_SAMPLE > Данные > CONFIG > Данные > АН > Данные > wvalue. The 'wvalue' element is highlighted. The bottom window, titled "Редактор свойств" (Properties Editor), shows the properties for the selected 'wvalue' element. The 'Путь' (Path) property is selected and highlighted in blue.

Имя	Описание
AI1_SAMPLE	Тип на основе Мнемосимвол. Базовь
Данные	
CONFIG	Элемент AP
Данные	
АН	Элемент AP float
Данные	
wvalue	Элемент AP float























  

Свойство	Характеристики	Значение
Отображаемое имя		wvalue
Кардинальное число		1
Источник	R W → 0 ✓	MnemonicSource
Путь	R W ✓	АН.wvalue
Значение	R W ↯ ⚙	<не определено>

18. Аналогичным образом добавьте остальные параметры в папку "CONFIG":

Структура объекта	
Имя	Описание
AI1_SAMPLE	Тип на основе Мнемосимвол. Базовый тип
Данные	
CONFIG	Элемент AP
Данные	
> AH	Элемент AP float
> YMIN	Элемент AP float
> YMAX	Элемент AP float
> TF	Элемент AP float
> HYST	Элемент AP float
> AH2	Элемент AP float
> WH	Элемент AP float
> WL	Элемент AP float
> AL	Элемент AP float
> AL2	Элемент AP float
> SIGN_CHECK	Элемент AP uint2

19. Если параметр находится не в папке, то его можно добавить аналогично тем, что находятся в папке, для аналогового датчика должна получиться следующая структура параметров на уровне HMI:

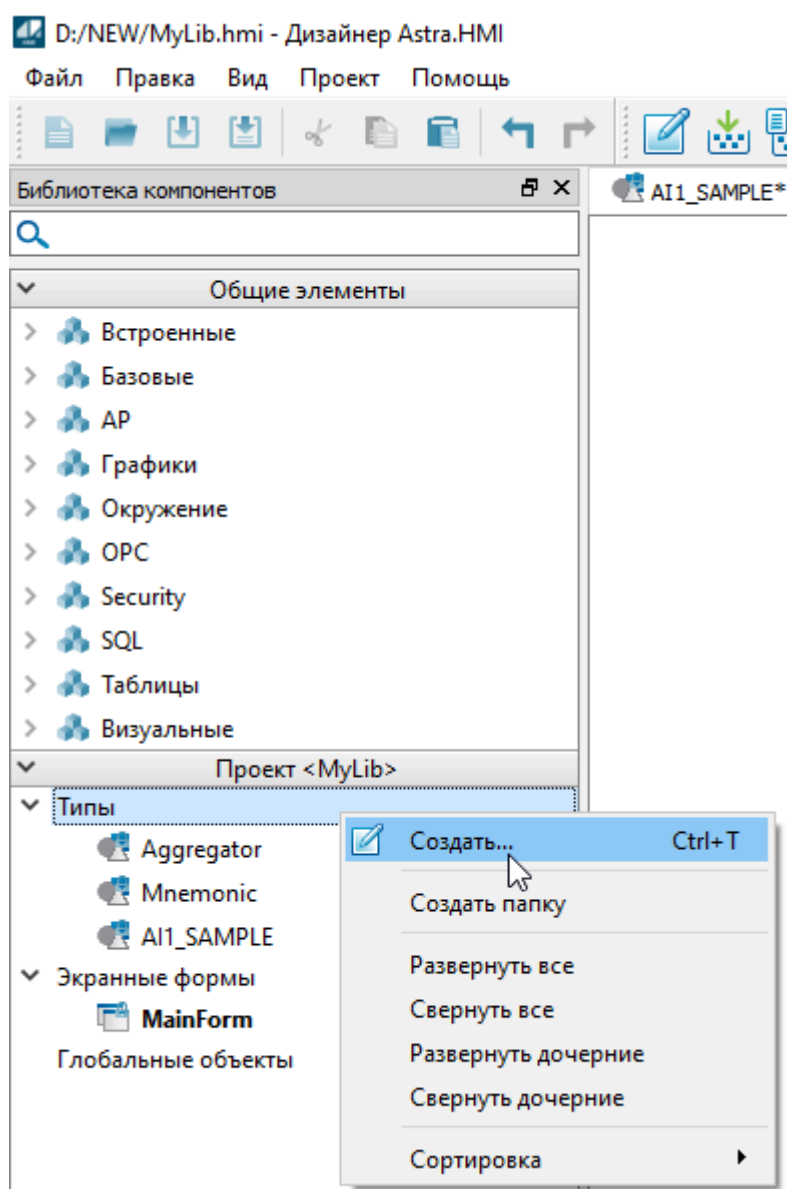
Структура объекта	
Имя	Описание
<ul style="list-style-type: none"> <li>▼  CONFIG               <ul style="list-style-type: none"> <li>▼ Данные                   <ul style="list-style-type: none"> <li>&gt;  AH</li> <li>&gt;  YMIN</li> <li>&gt;  YMAX</li> <li>&gt;  TF</li> <li>&gt;  HYST</li> <li>&gt;  AH2</li> <li>&gt;  WH</li> <li>&gt;  WL</li> <li>&gt;  AL</li> <li>&gt;  AL2</li> <li>&gt;  SIGN_CHECK</li> </ul> </li> <li>▼  MAN_ON                   <ul style="list-style-type: none"> <li>▼ Данные                       <ul style="list-style-type: none"> <li> wvalue</li> </ul> </li> </ul> </li> <li>▼  MAN_VALUE                   <ul style="list-style-type: none"> <li>▼ Данные                       <ul style="list-style-type: none"> <li> wvalue</li> </ul> </li> </ul> </li> </ul> </li> <li> VALUE</li> <li> QUALITY</li> <li> FACT_VALUE</li> <li> SIGN</li> <li> MODE</li> <li> DIAGN</li> </ul>	<ul style="list-style-type: none"> <li>Элемент AP</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP uint2</li> <li>Элемент AP bool</li> <li>Элемент AP bool</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP float</li> <li>Элемент AP int4</li> <li>Элемент AP float</li> <li>Элемент AP int4</li> <li>Элемент AP bool</li> <li>Элемент AP int4</li> </ul>

20. Создание базового типа аналогового датчика "AI1" завершено.

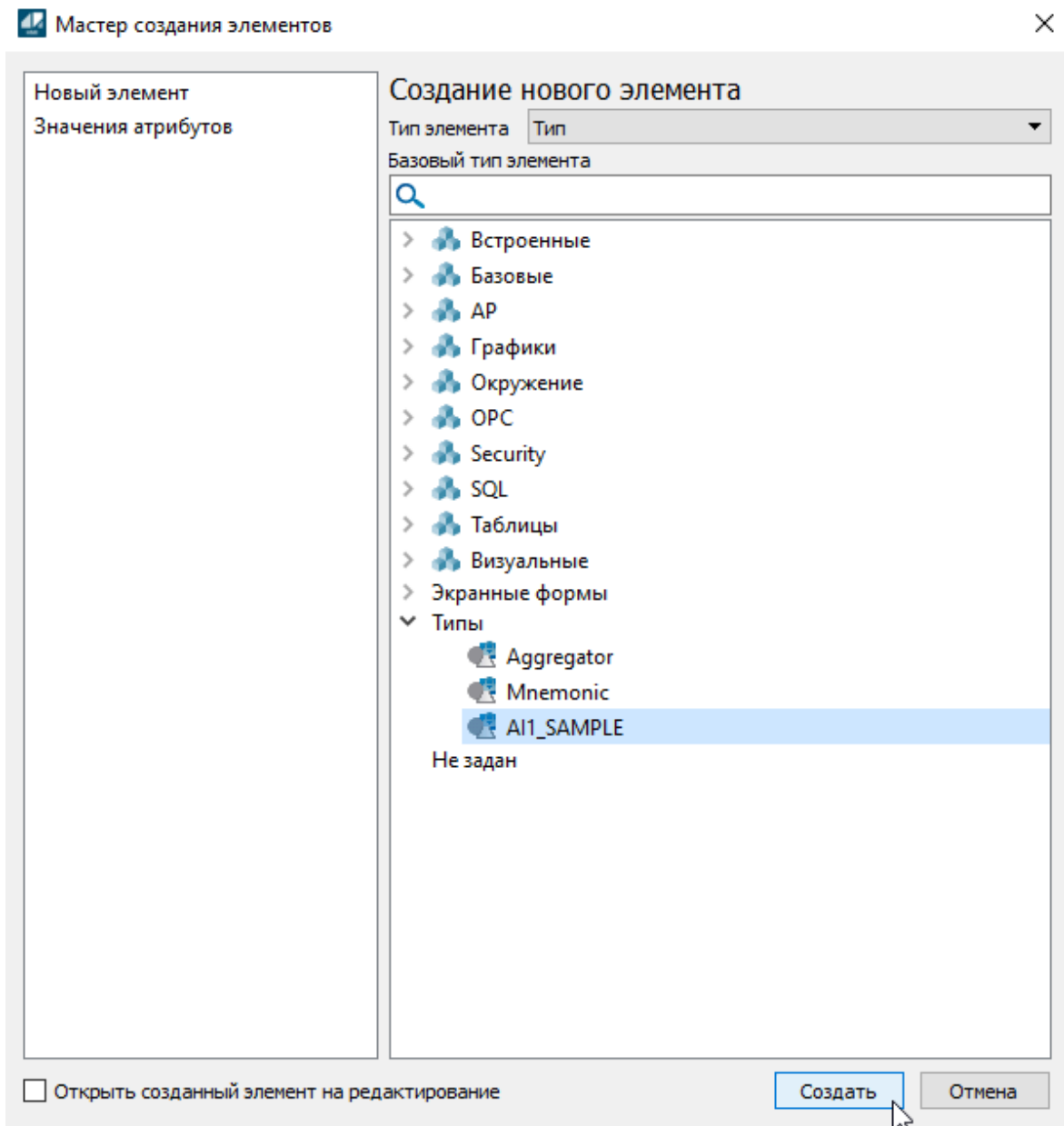
## 1.2.4.5. Создание графического представления типа аналогового датчика

Для того, чтобы создать графическое представление типа аналогового датчика AI1, необходимо выполнить следующие действия:

1. Нажмите правой кнопкой мыши по вкладке "Типы" и в контекстном меню выберите команду "Создать...".



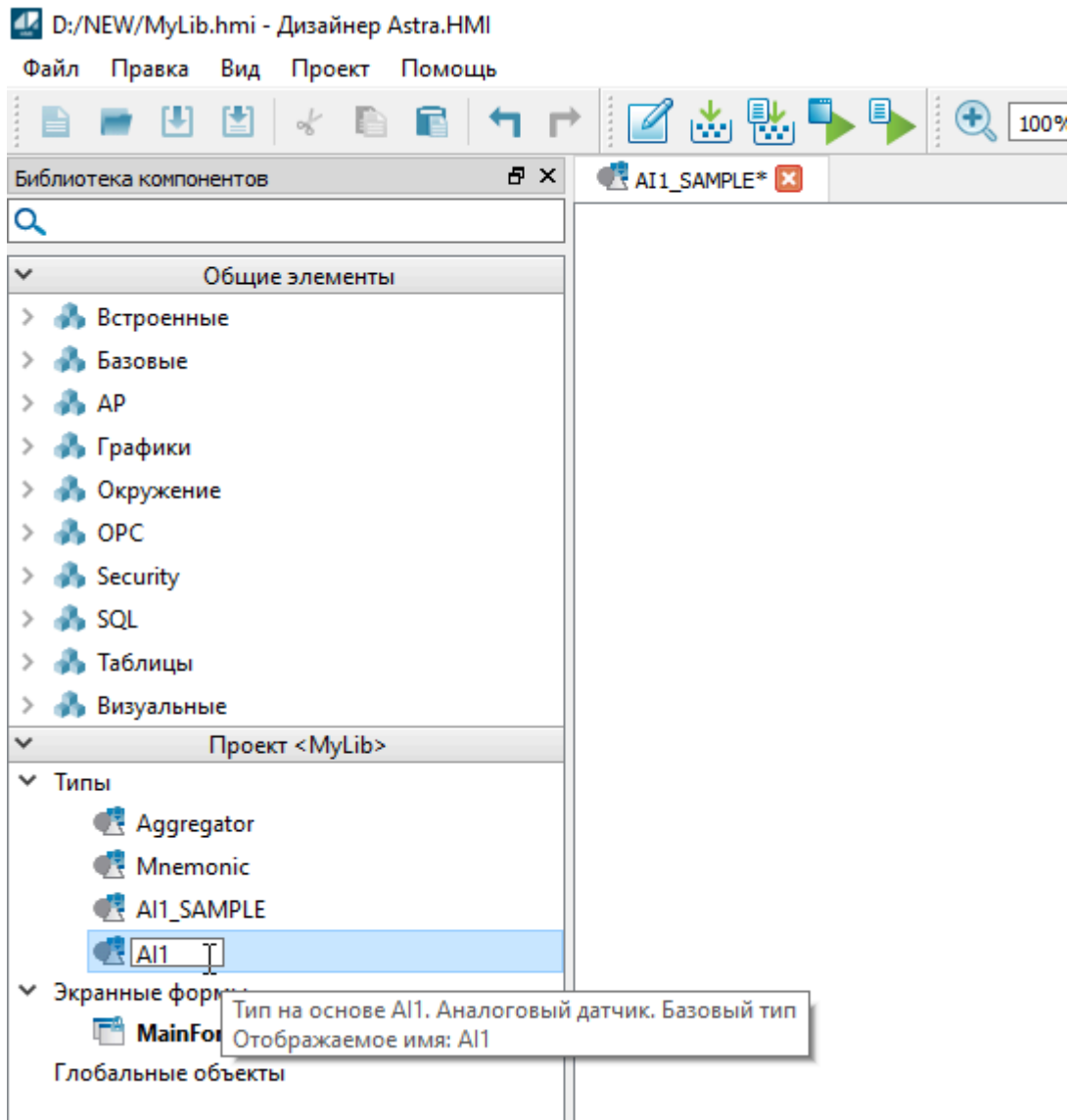
2. В мастере создания элементов раскройте вкладку "Типы" и выберите базовый тип элемента "AI1\_SAMPLE". Нажмите кнопку "Создать".



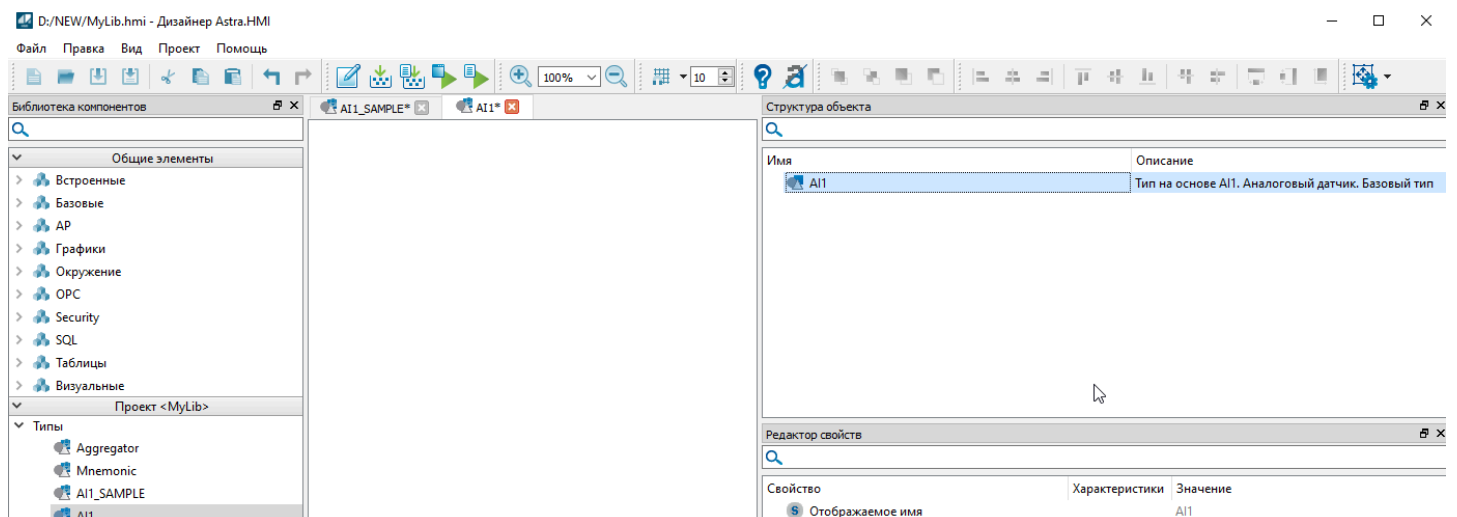
На основе базового типа можно создавать несколько графических представлений. Данные, добавленные в базовый тип будут доступны для каждого отнаследованного типа.

3. Во вкладке "Типы" появится новый графический объект с именем "AI1\_SAMPLE\_1". Выделите тип левой кнопкой мыши, нажмите клавишу "F2" и задайте имя "AI1"

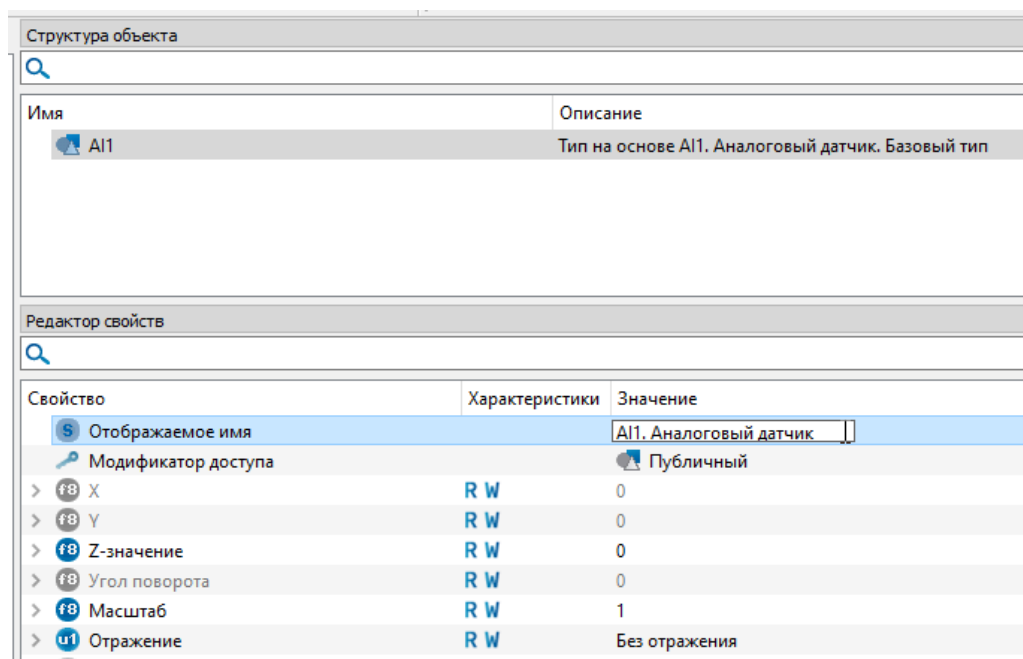




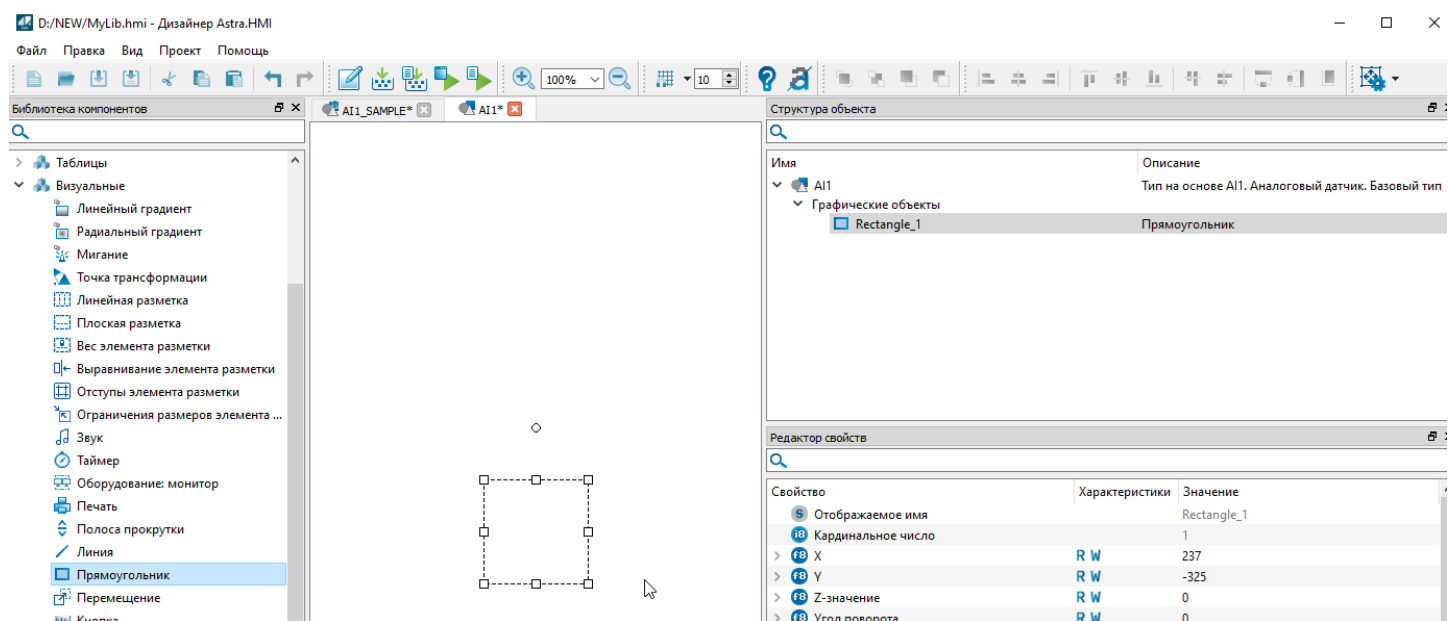
4. Перейдите в тип "AI1" двойным кликом левой кнопки мыши. В окне справа отобразится структура объекта.



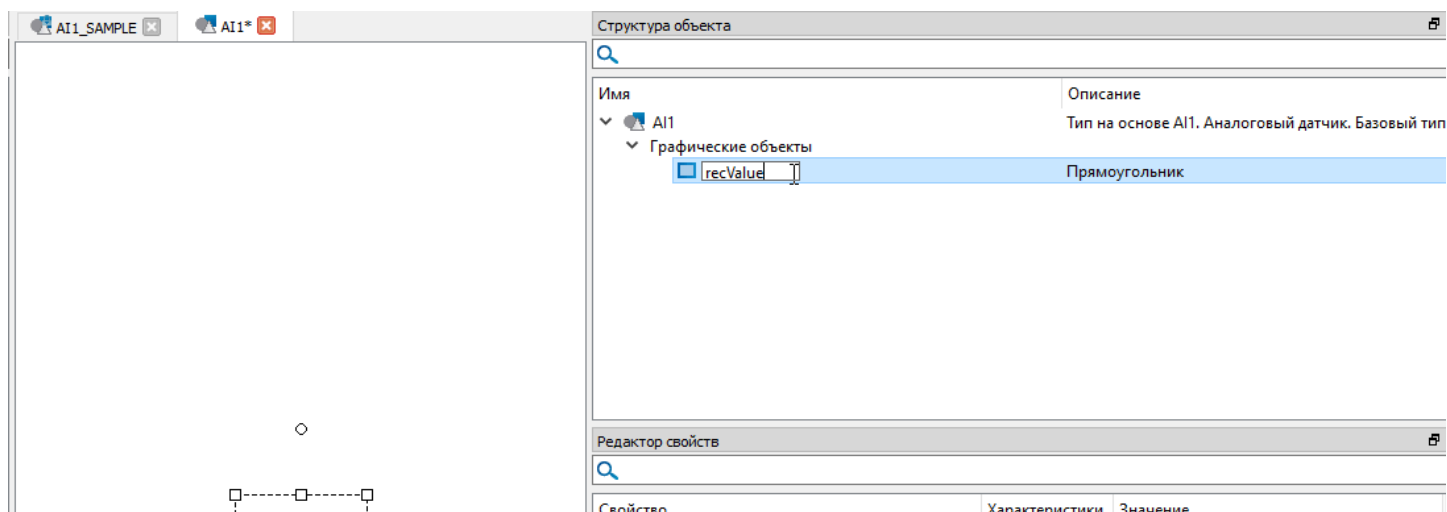
5. Выделите тип "AI1" кликнув по нему. В окне "Редактор свойств" измените значение свойства "Модификатор доступа" на "Публичный" и установите в свойстве "Отображаемое имя" значение "AI1. Аналоговый датчик". Модификатор доступа отвечает за доступность(видимость) тех или иных объектов после добавления библиотеки в проект.



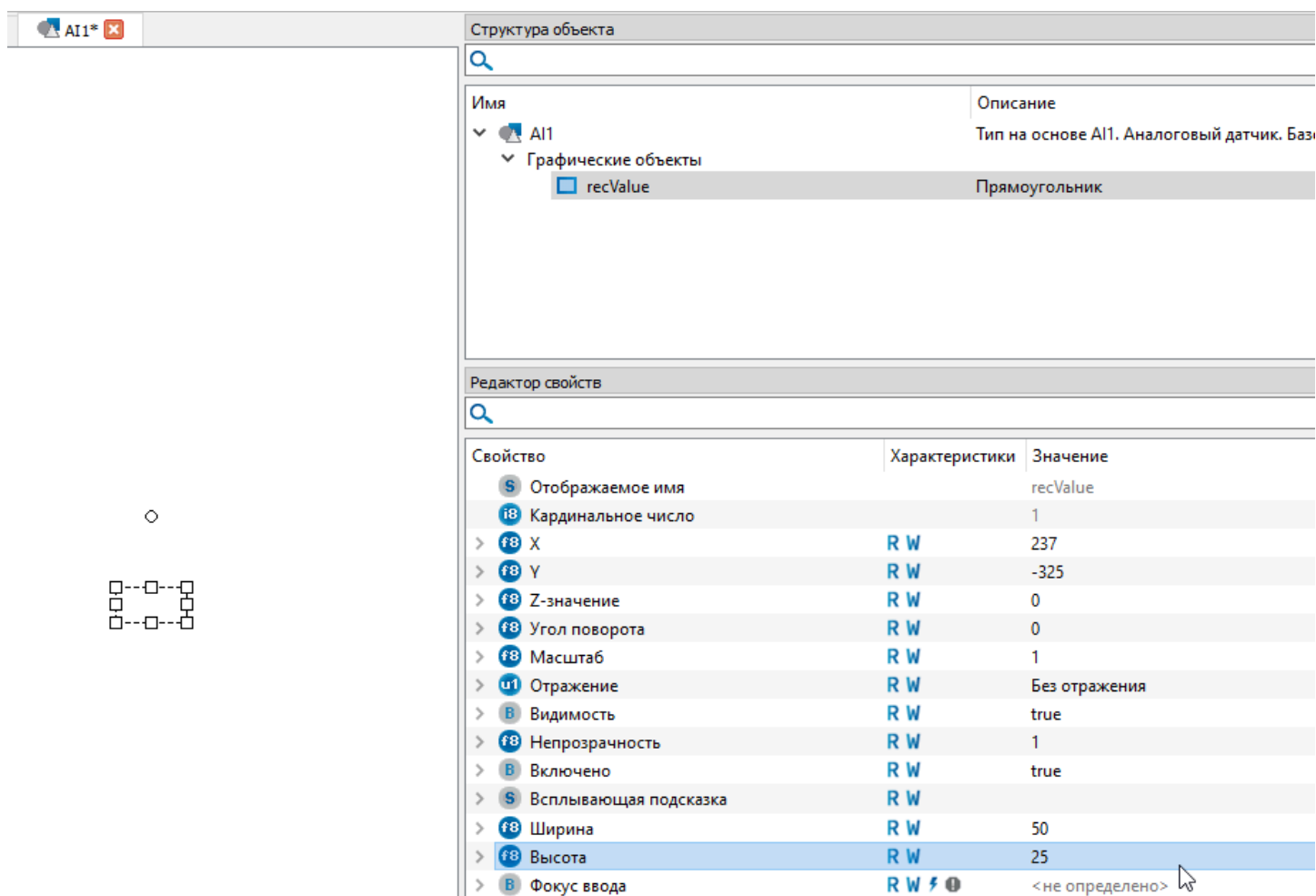
6. Во вкладке "Общие элементы" раскройте вкладку "Визуальные" и добавьте путем перетаскивания элемент "Прямоугольник".



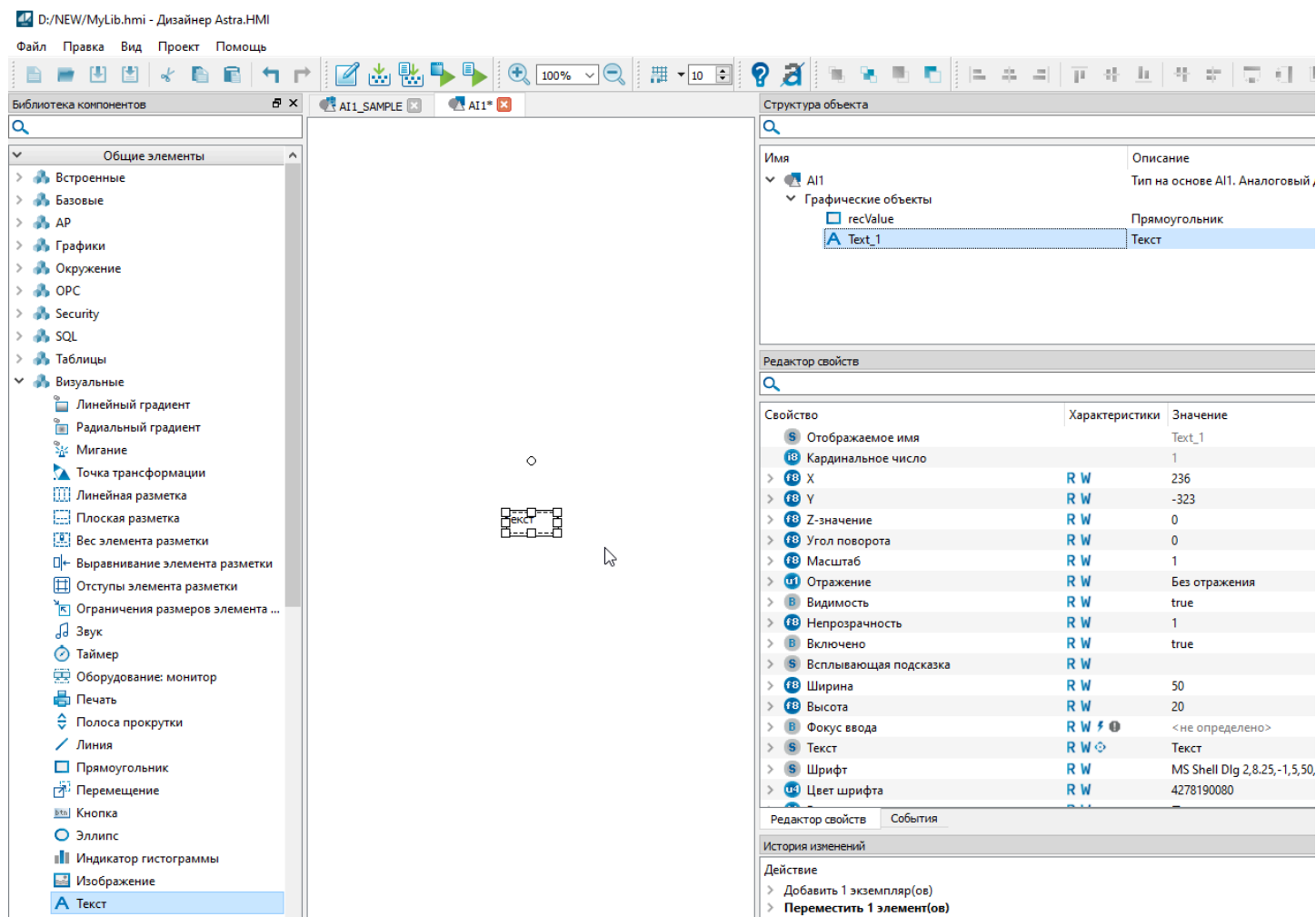
7. Выделите графический объект "Rectangle\_1" левой кнопкой мыши и переименуйте его в "recValue", воспользовавшись клавишей "F2".



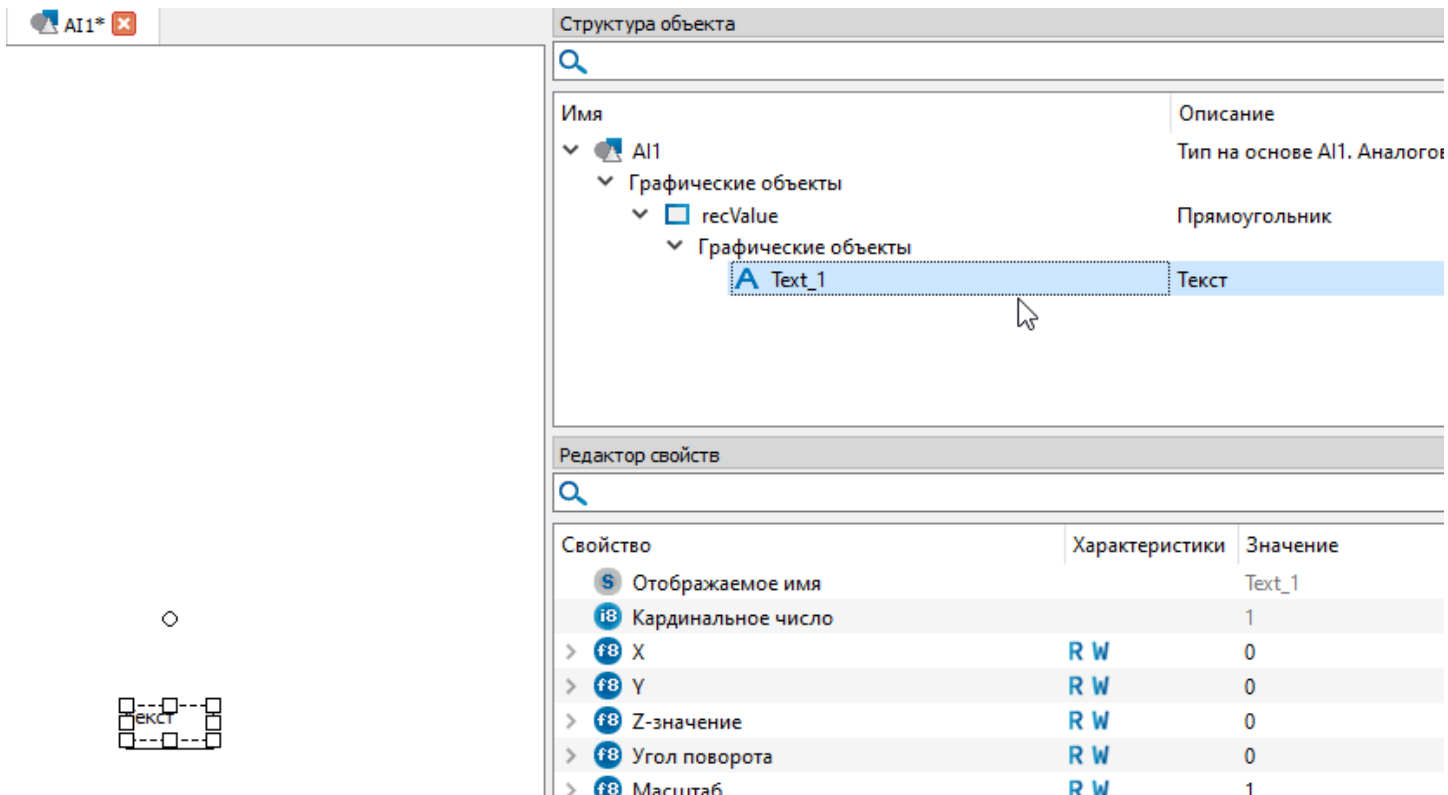
8. В окне "Редактор свойств" задайте свойству "Ширина" значение 50, а свойству "Высота" значение 25.



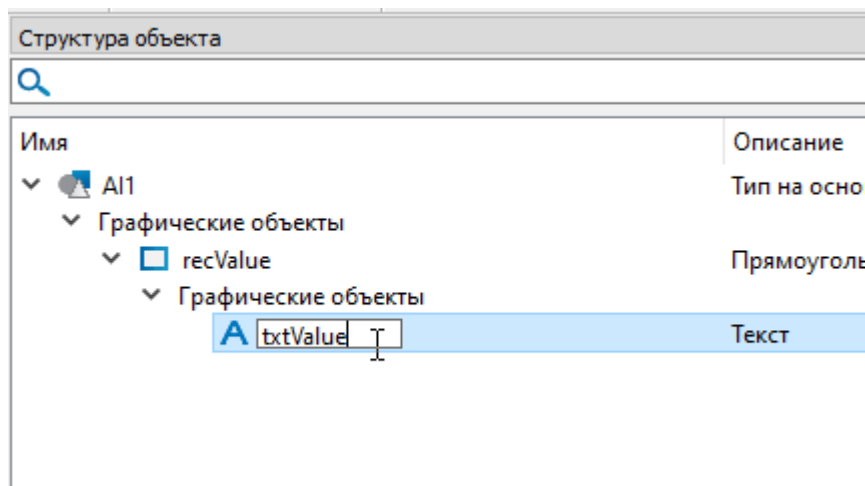
9. Во вкладке "Общие элементы" раскройте вкладку "Визуальные" и добавьте путем перетаскивания элемент "Текст".



10. Текст можно разместить внутри графического объекта "Прямоугольник" для этого необходимо перетащить тип "Text\_1" в тип "recValue".



11. Выделите графический объект "Text\_1" левой кнопкой мыши и переименуйте его в "txtValue", воспользовавшись клавишей "F2".

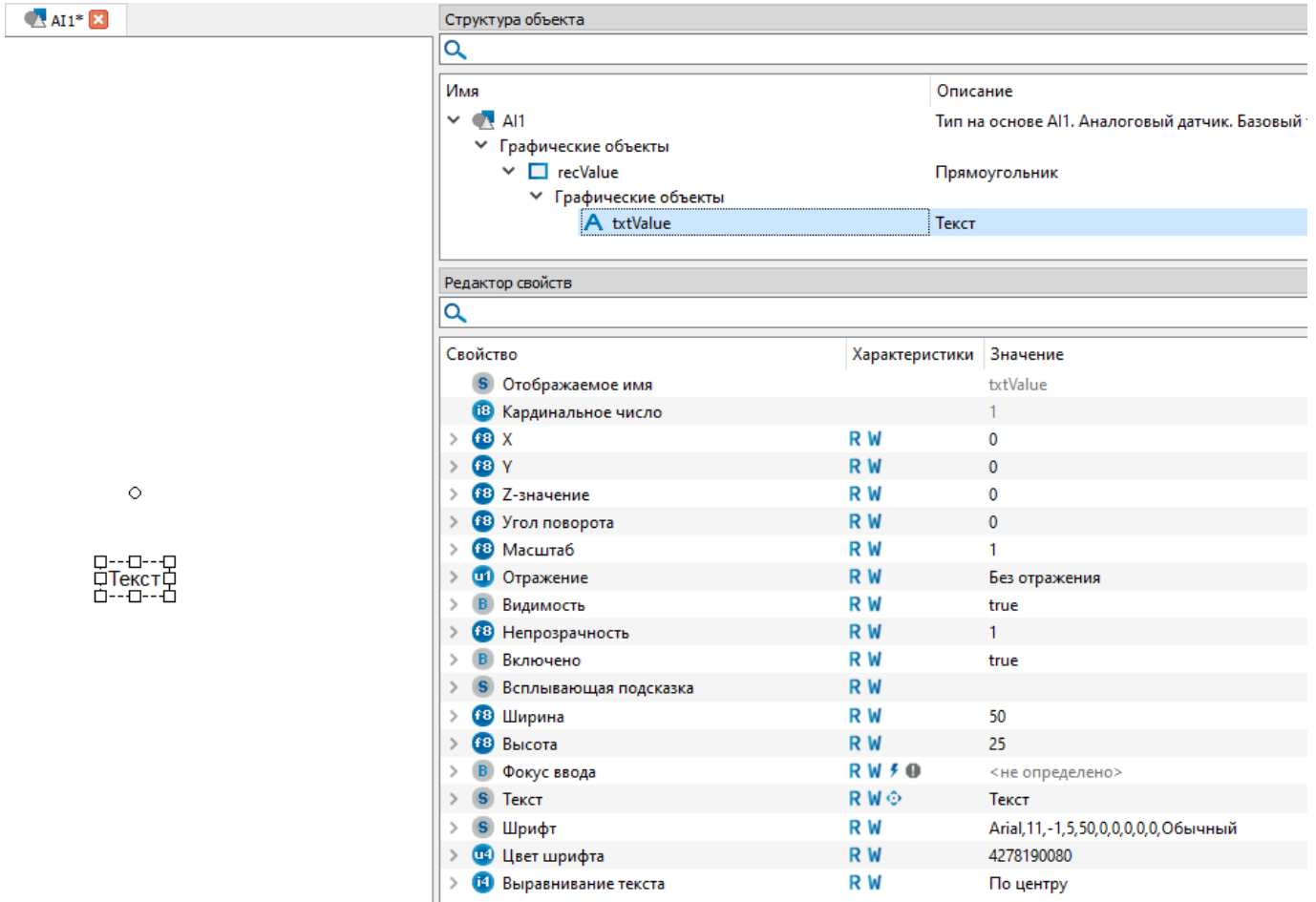


12. В окне "Редактор свойств" измените значения у следующих свойств:

Свойство	Значение
Высота	25
Шрифт	Arial,11,-1,5,50,0,0,0,0,0,Обычный

Выравнивание  
текста

По центру



Структура объекта

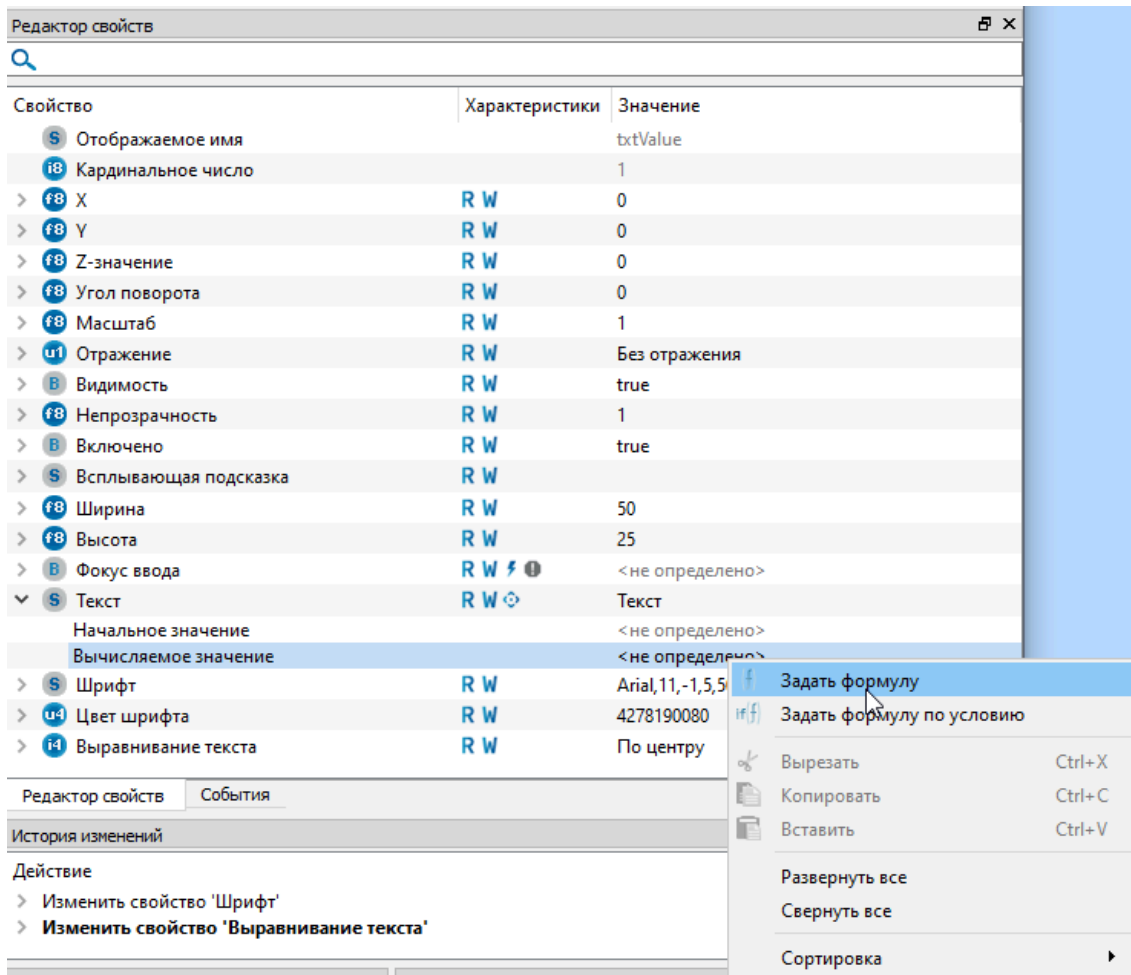
Имя: AI1  
Описание: Тип на основе AI1. Аналоговый датчик. Базовый

Графические объекты  
recValue  
Графические объекты  
txtValue  
Текст

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		txtValue
Кардинальное число		1
X	R W	0
Y	R W	0
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	50
Высота	R W	25
Фокус ввода	R W	<не определено>
Текст	R W	Текст
Шрифт	R W	Arial, 11, -1, 5, 50, 0, 0, 0, 0, Обычный
Цвет шрифта	R W	4278190080
Выравнивание текста	R W	По центру

13. Посредством визуальных типов можно отображать состояние алгоритма или его текущие значения, для того чтобы в поле текст отображалось значение с выхода алгоритма аналогового датчика (AI1) раскройте крыжик у свойства текст и нажмите правой кнопкой мыши по строчке "Вычисляемое значение". Из контекстного меню выберите команду "Задать формулу".



14. В появившемся окне для ввода впишите следующую формулу "VALUE.ValueAsString" и после нажмите сочетание клавиш "Ctrl+Enter". Таким образом, в режиме исполнения вместо значения "Текст" будет отображаться значение с выхода алгоритма аналогового датчика. Так как свойство имеет тип "string" необходимо обратиться к свойству параметра "VALUE" - "ValueAsString", так как оно представляет значение типа "float" в формате строки.

Структура объекта

Имя | Описание

- AI1 | Тип на основе AI1. Аналоговый датчик. Баз
- Графические объекты
  - recValue | Прямоугольник
    - Графические объекты
      - txtValue | Текст

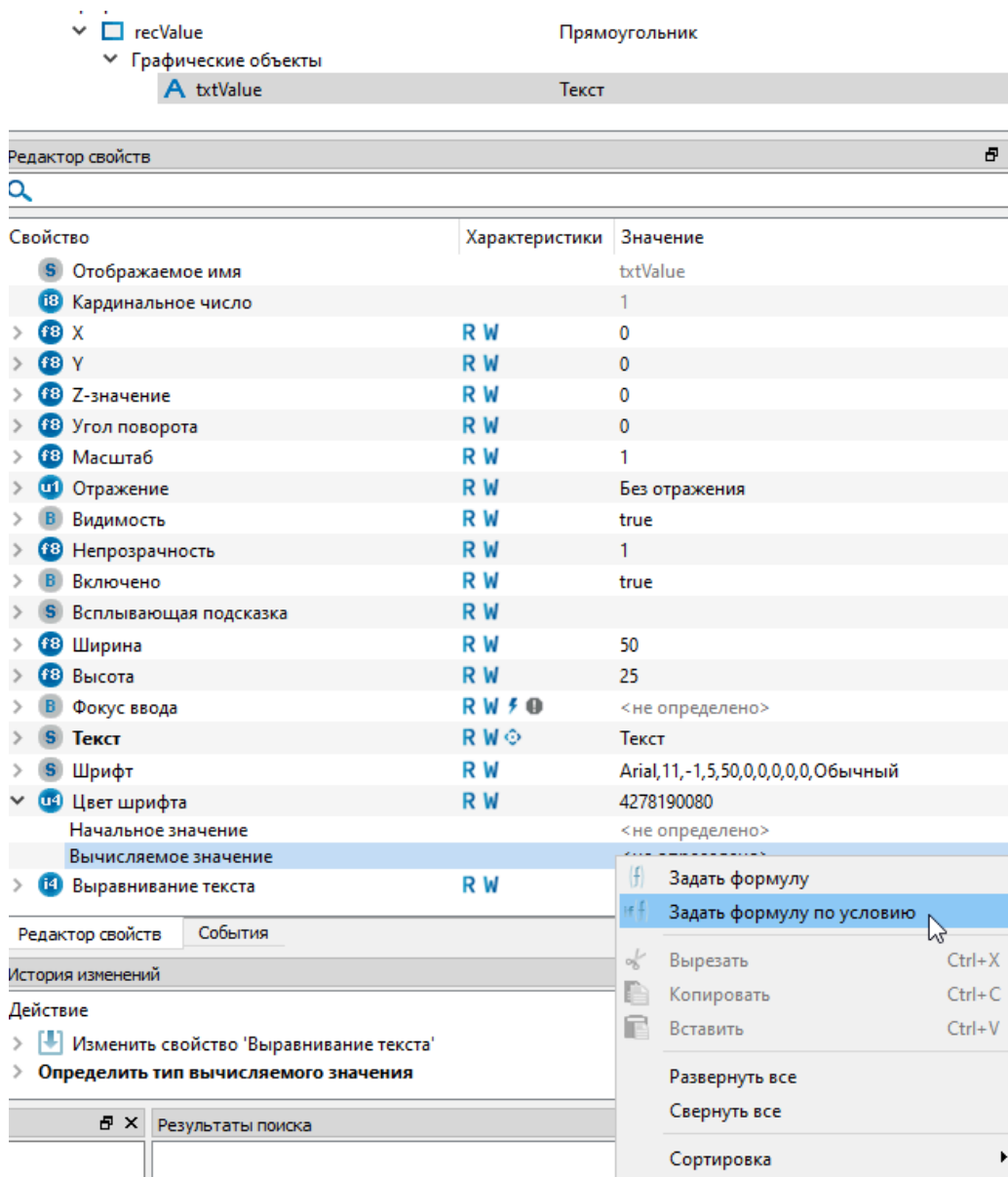
Редактор свойств

Свойство	Характеристики	Значение
> В Включено	R W	true
> S Всплывающая подсказка	R W	
> f8 Ширина	R W	50
> f8 Высота	R W	25
> В Фокус ввода	R W ⚡ ⓘ	< не определено >
▼ S Текст	R W ⚙	Текст
Начальное значение		< не определено >
		Формула
		VALUE.ValueAsString

**i** Для задания формул используется проприетарный язык программирования Astra.Om. Подробнее об этом языке можно узнать из документации.

15. Задайте цветовую раскраску шрифта в зависимости от качества выходного сигнала "VALUE". Для этого раскройте крыжик у свойства "Цвет шрифта" и нажмите правой кнопкой мыши по строке "Вычисляемое значение". Из контекстного меню выберите команду "Задать формулу по условию".





16. Нажмите кнопку "Редактировать...". В появившемся окне откроется таблица с двумя столбцами "Условие" и "Значение". Соответственно, в столбец условие мы запишем условие для переменной "QUALITY" типа "int4", а в столбец значение запишем значение цвета. Например при значении переменной "QUALITY" меньше 6 цвет шрифта становится белым, а по умолчанию черный. После задания условий нажмите кнопку "OK".

Редактор формул по условию

Условие	Значение
QUALITY < 6	0xffffffff
Нет условия	Нет действия
По умолчанию	0xff000000

0 Ошибок | 0 Предупреждений | 0 Сообщений

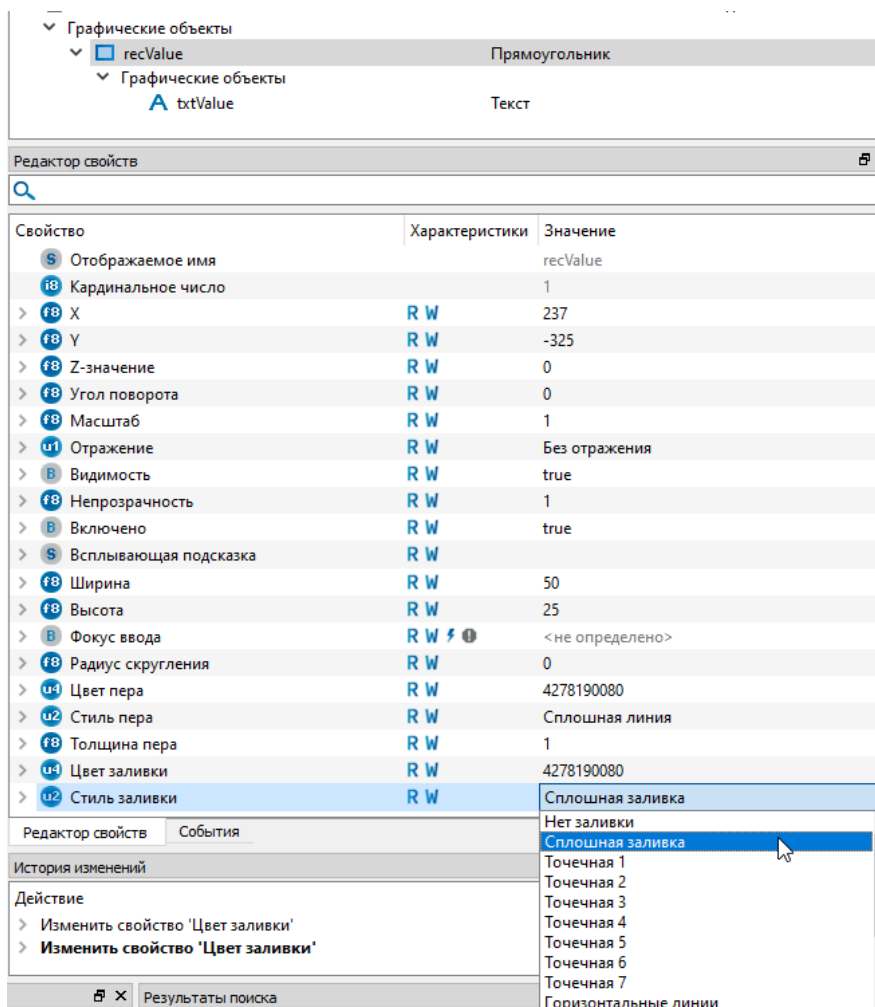
Описание	Положение

OK Отмена

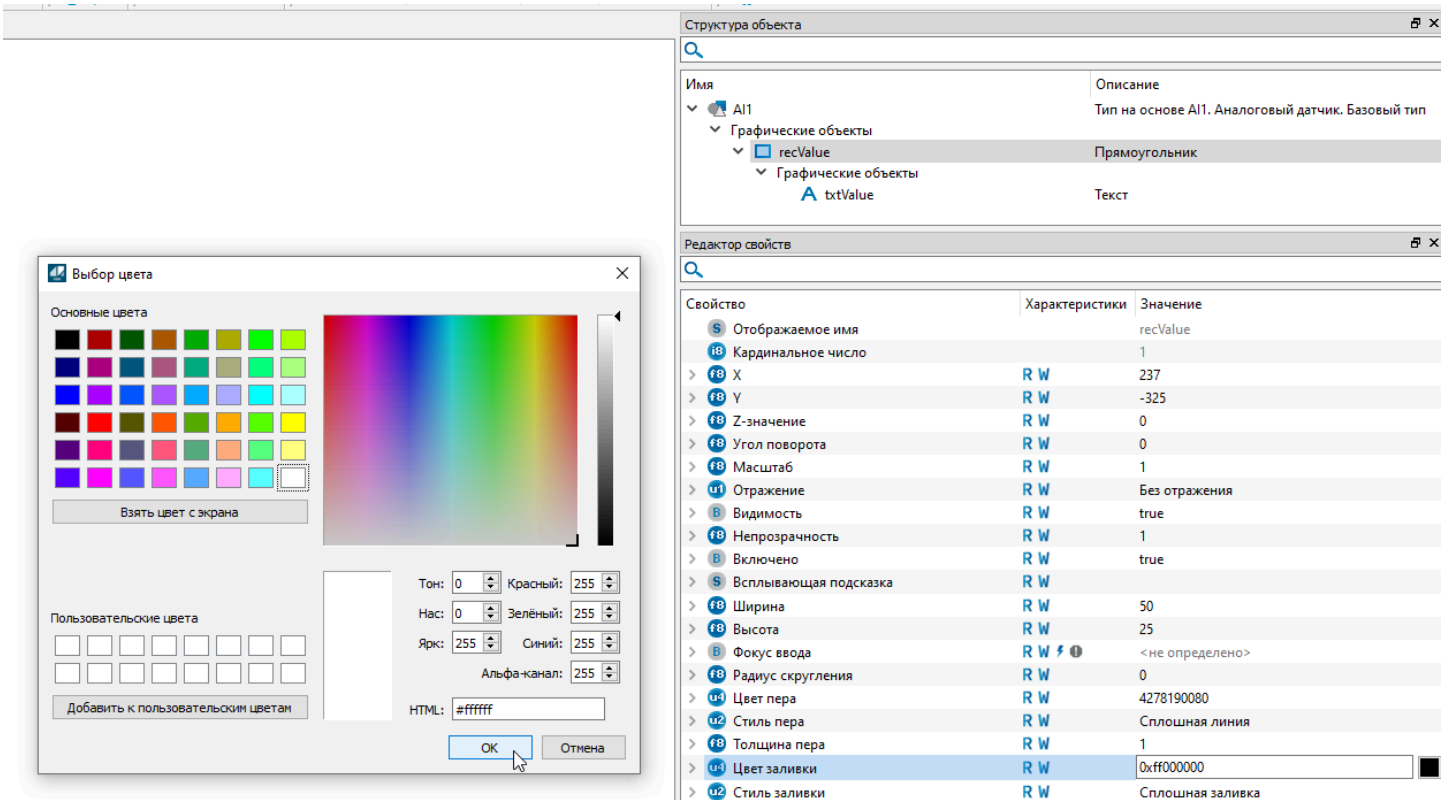


Для задания формул по условию используется проприетарный язык программирования Astra.Om. Подробнее об этом языке можно узнать из документации.

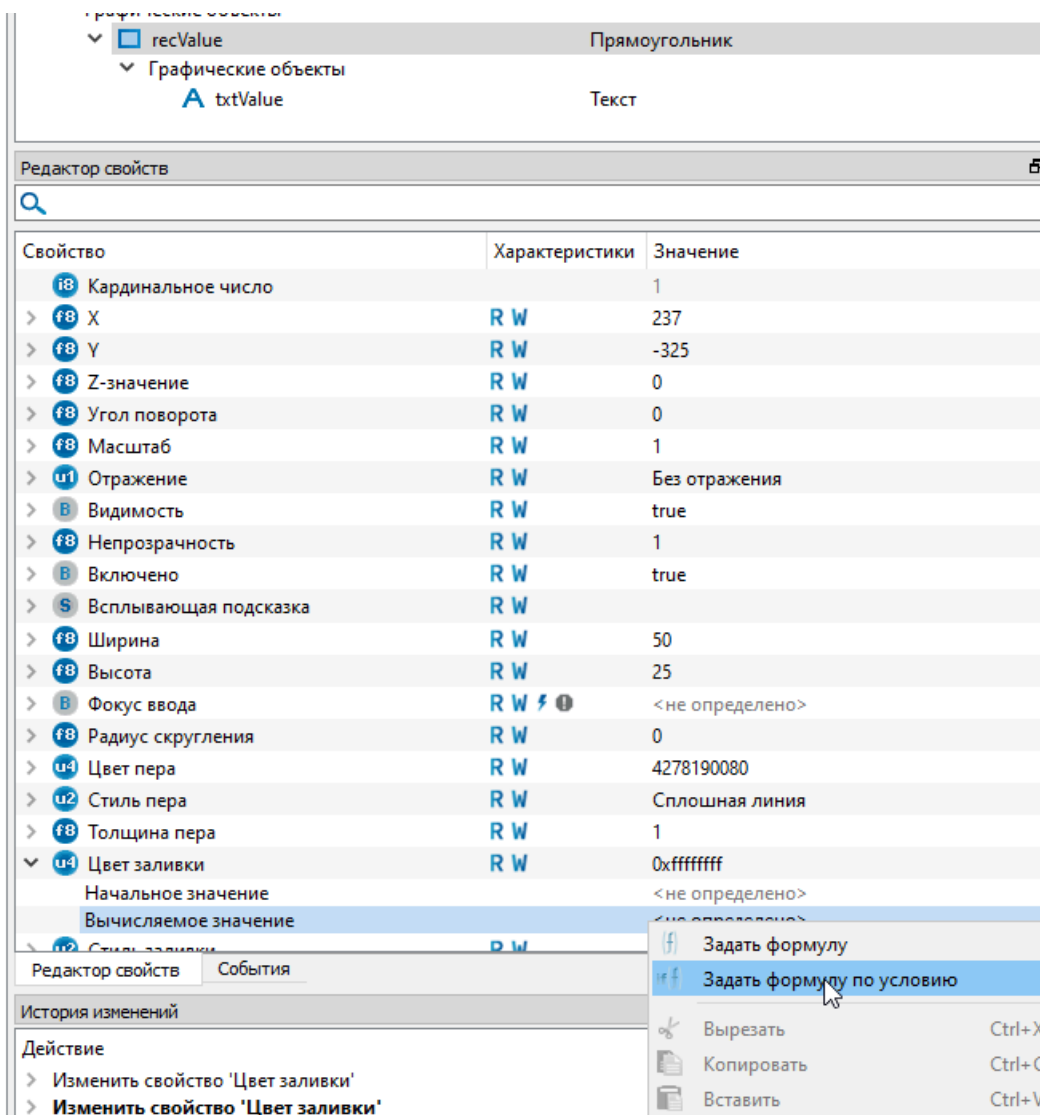
17. Измените стиль заливки для типа "recValue" по умолчанию. Для этого выделите тип "recValue" и задайте свойству "Стиль заливки" значение "Сплошная заливка"



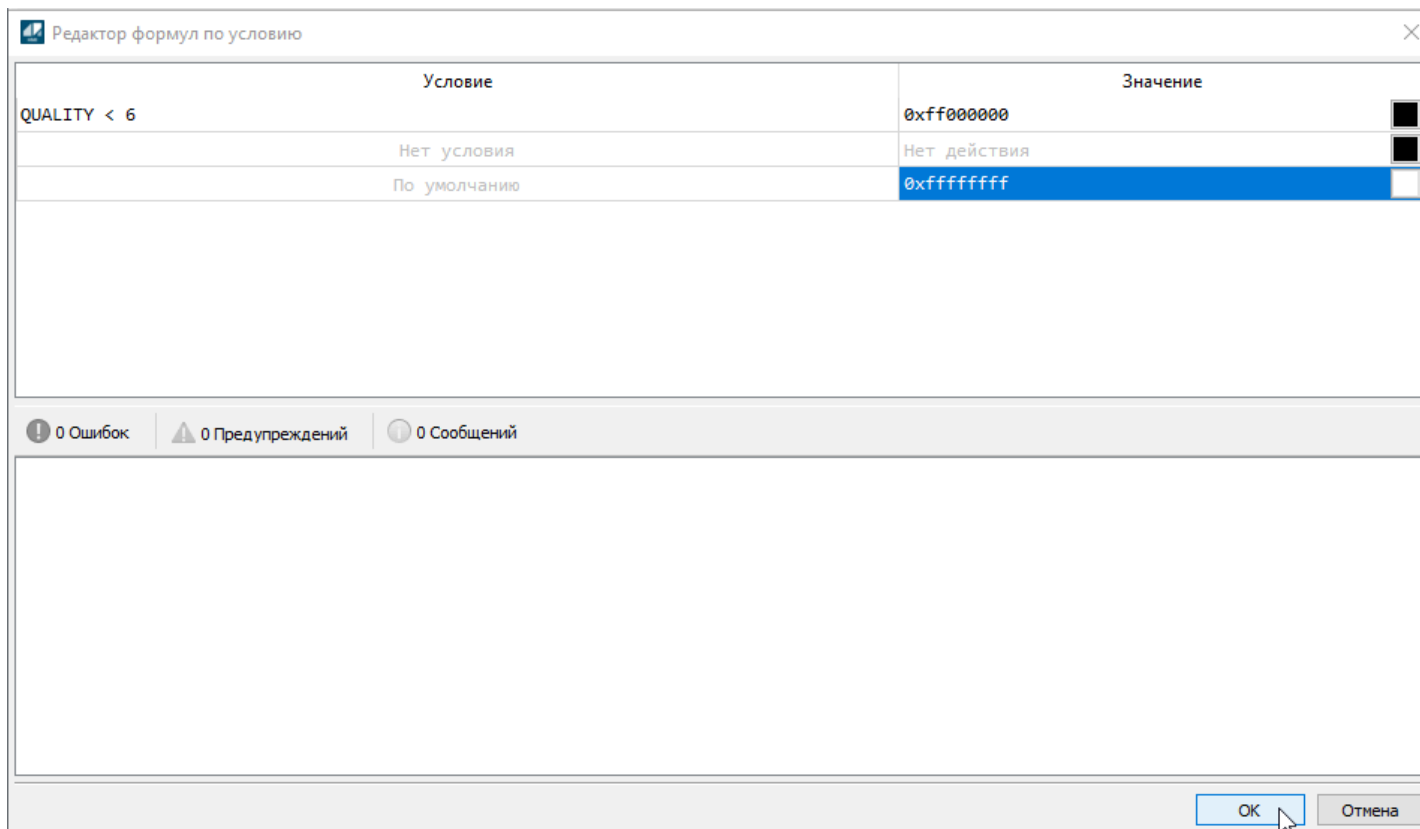
18. Установите по умолчанию белый цвет для заливки, для этого кликните двойным щелчком мыши по полю "Значение" в строке свойства "Цвет заливки". В открывшемся окне выберите цвет в удобном формате и нажмите кнопку "ОК".



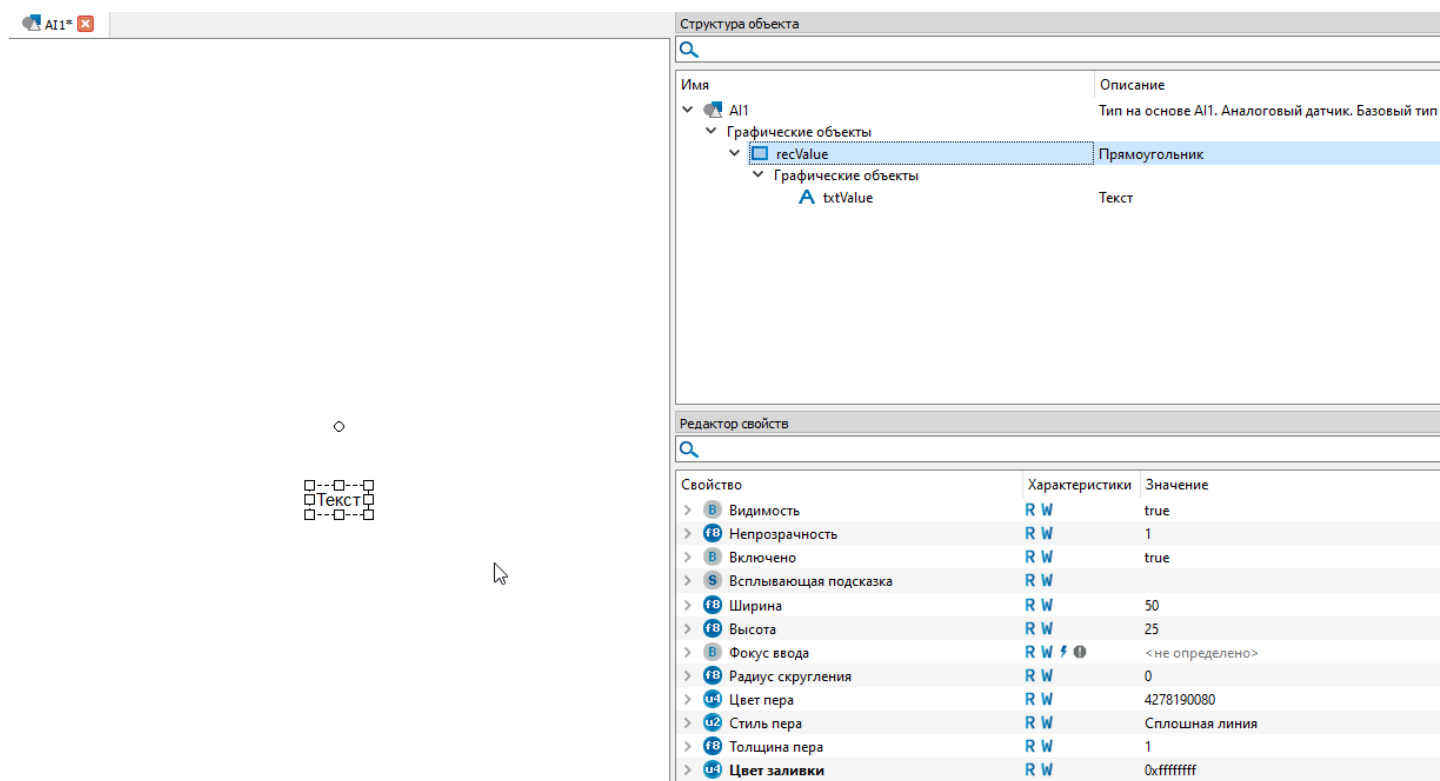
19. Задайте цветовую раскраску заливки типа "recValue" в зависимости от качества выходного сигнала "VALUE". Для этого раскройте крыжик у свойства "Цвет заливки" и нажмите правой кнопкой мыши по строке "Вычисляемое значение". Из контекстного меню выберите команду "Задать формулу по условию".



20. Нажмите кнопку "Редактировать...". В появившемся окне откроется таблица с двумя столбцами "Условие" и "Значение". Соответственно, в столбец условие мы запишем условие для переменной "QUALITY" типа "int4", а в столбец значение запишем значение цвета. Например, при значении переменной QUALITY меньше 6 цвет заливки становится черным, а по умолчанию белый. После задания условий нажмите кнопку "OK".



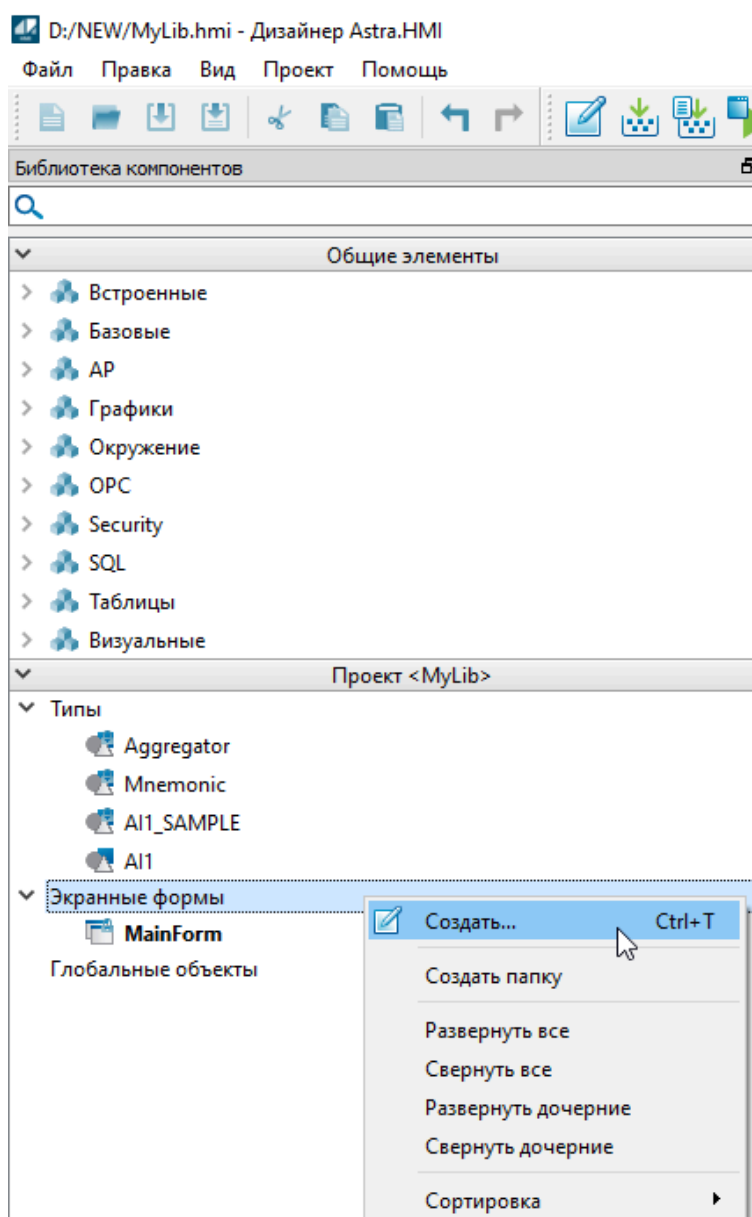
Таким образом, у нас получилось графическое представление с некоторой динамикой для аналогового датчика "AI1", в зависимости от значения переменной "QUALITY".



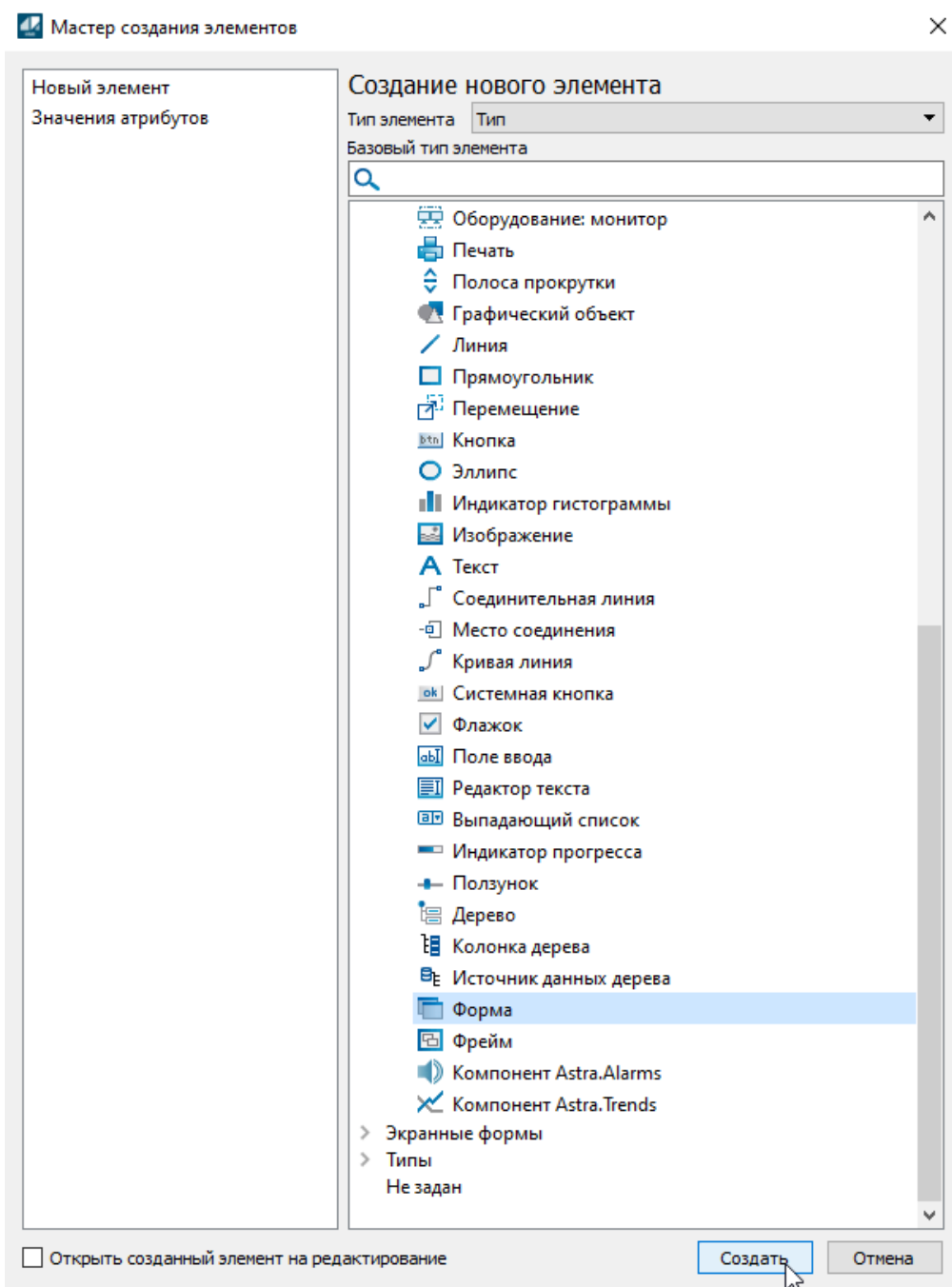
## 1.2.4.6. Создание рабочего окна для типа аналогового датчика

Для того, чтобы создать рабочее окно для типа аналогового датчика "AI1", необходимо выполнить следующие действия:

1. Нажмите правой кнопкой мыши по вкладке "Экранные формы" и в контекстном меню выберите команду "Создать...".

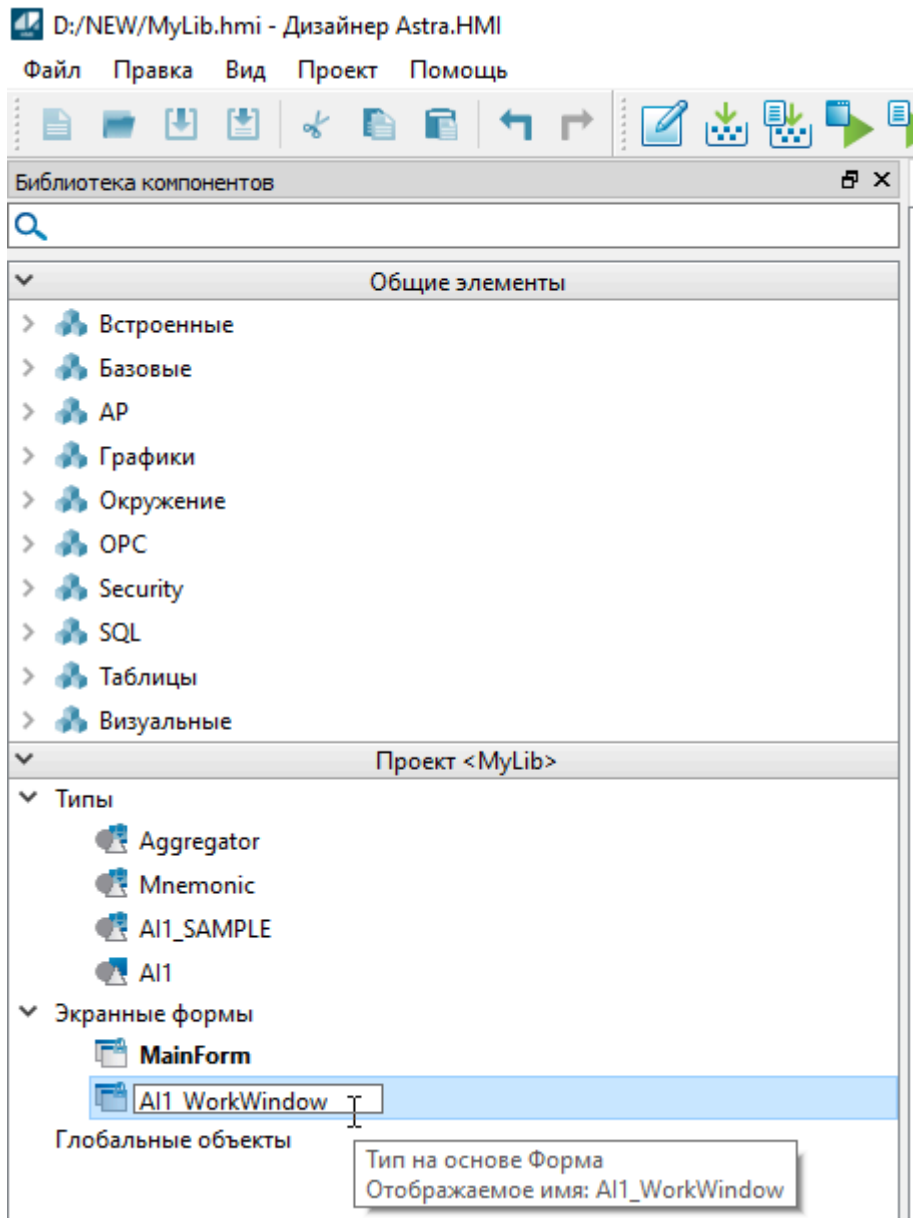


2. В мастере создания элементов раскройте вкладку "Визуальные" и выберите тип "Форма".

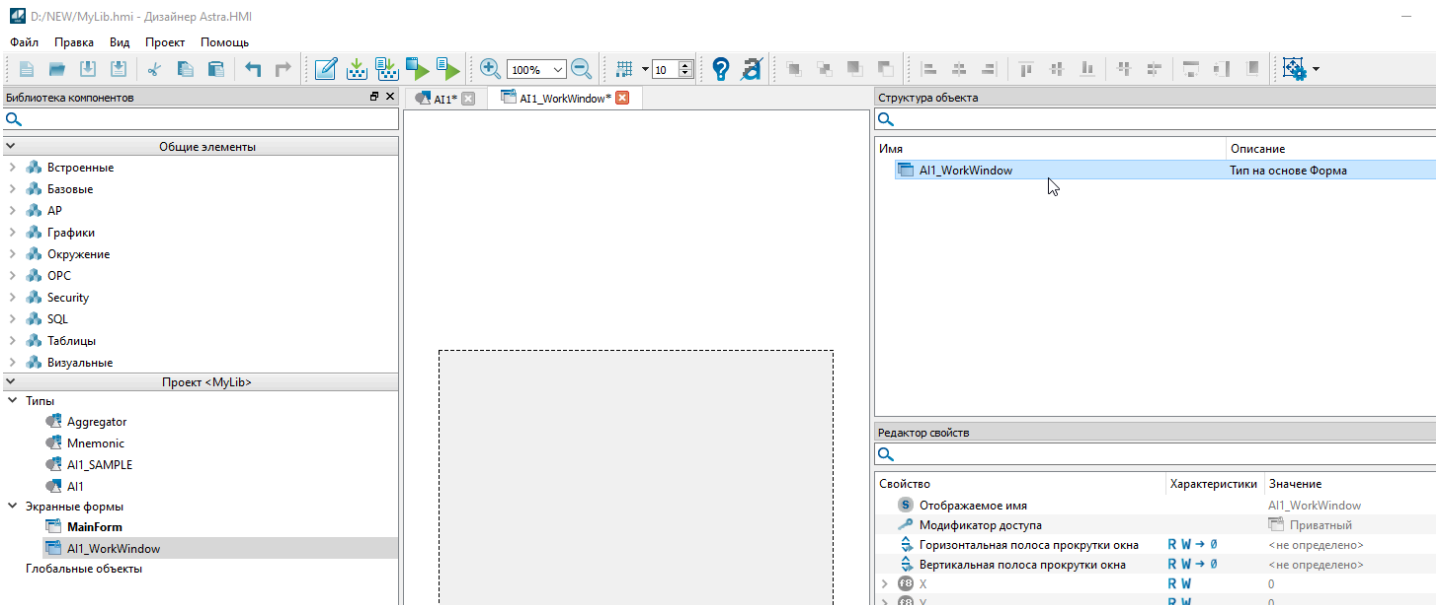


3. Во вкладке "Экранные формы" появится новый графический объект с именем "Form\_1". Выделите тип левой кнопкой мыши, нажмите клавишу "F2" и задайте имя "AI1\_WorkWindow"

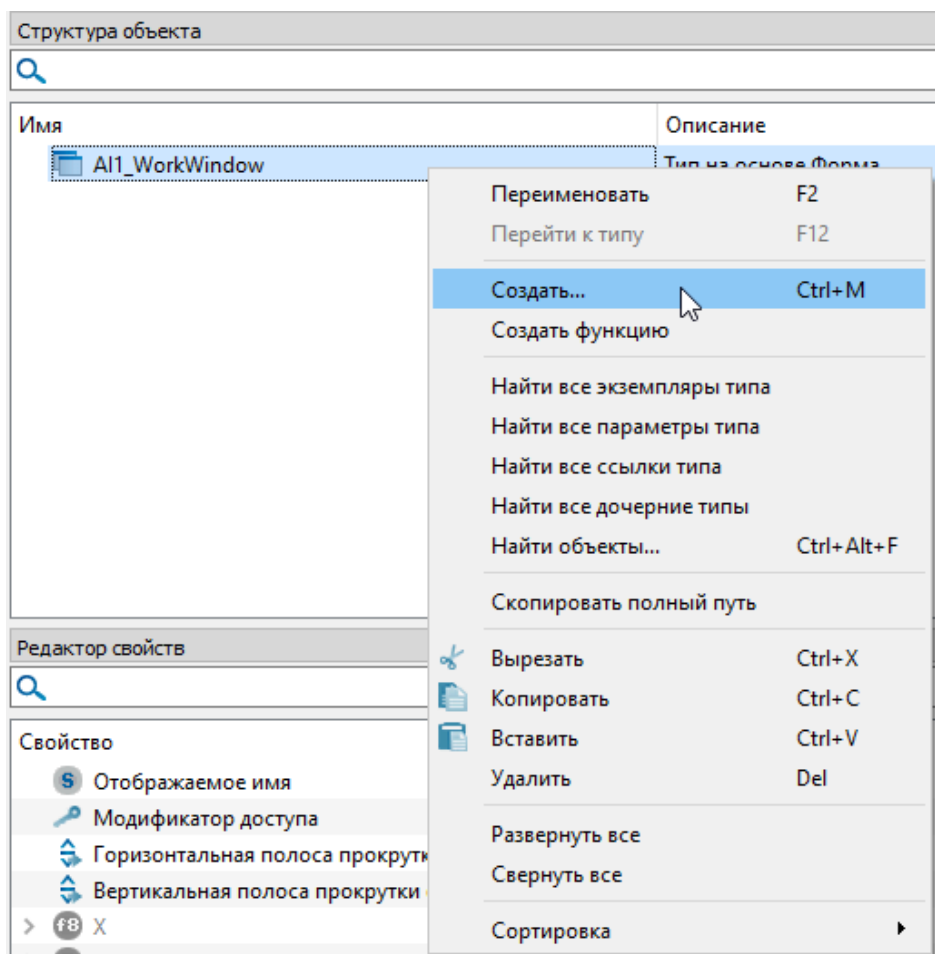




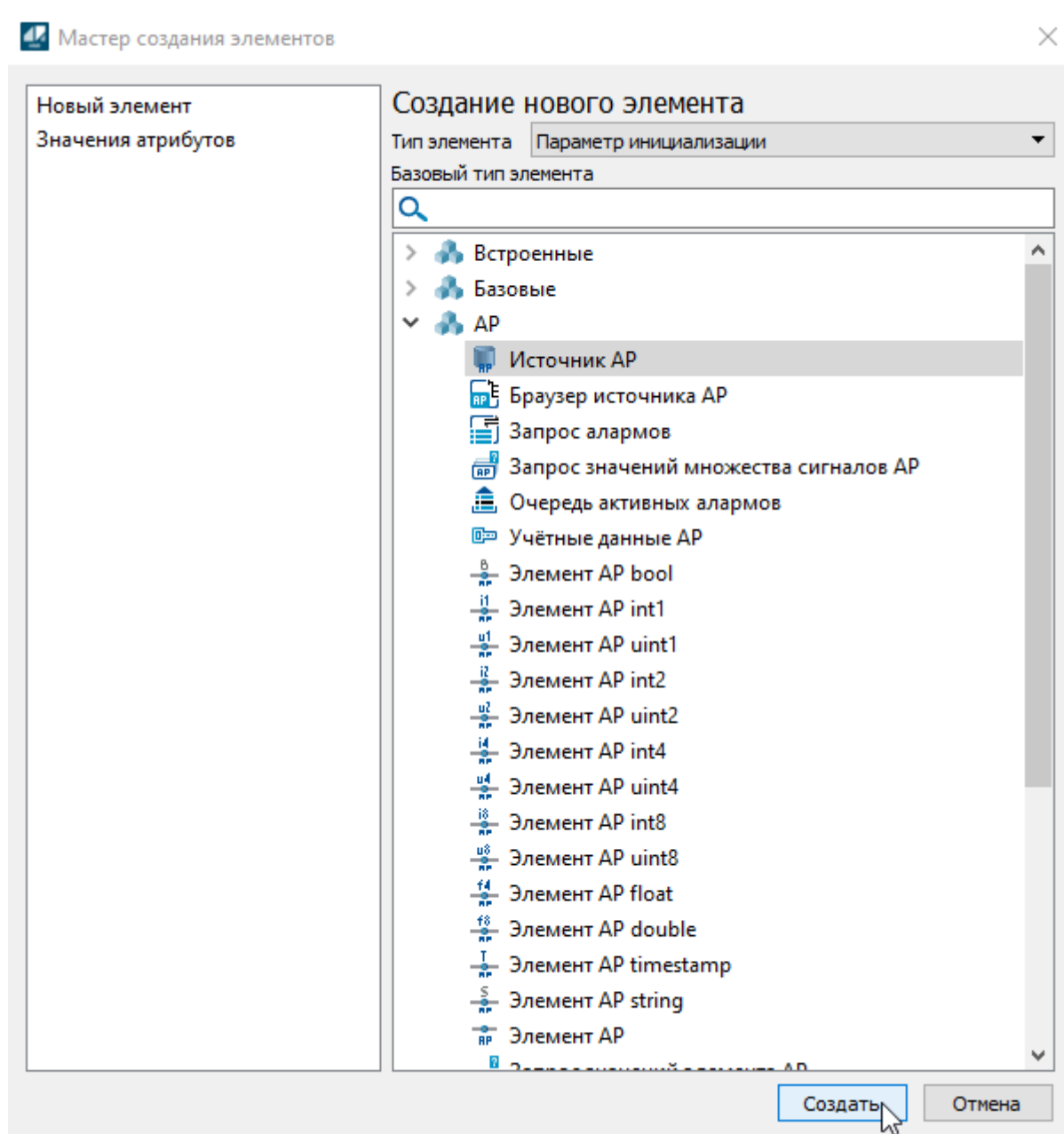
4. Перейдите в тип "AI1\_WorkWindow" двойным кликом левой кнопки мыши. В окне справа отобразится структура объекта.



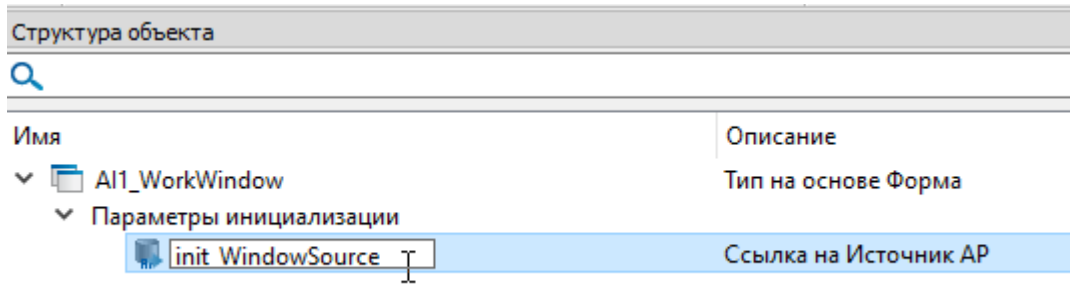
5. Кликните правой кнопкой мыши в структуре объекта по типу "AI1\_WorkWindow" и из контекстного меню выберите команду "Создать...".



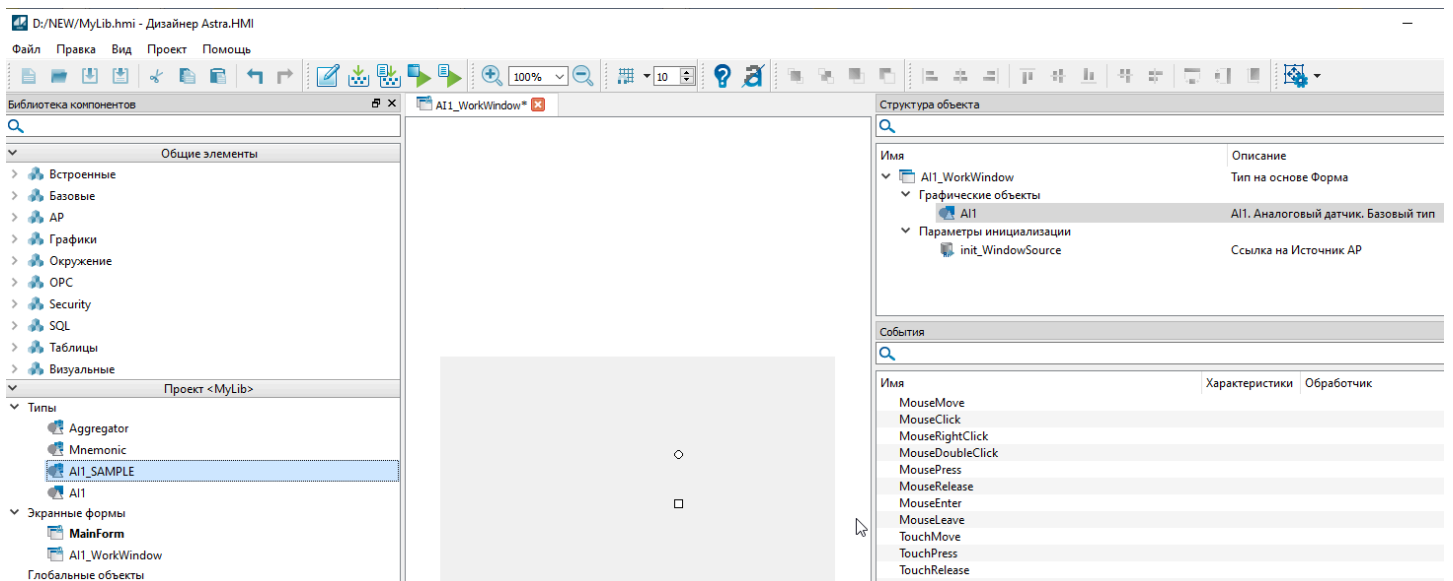
6. В мастере создания элементов в выпадающем списке вверху смените тип элемента на "Параметр инициализации", раскройте папку "AP" и выберите базовый тип "Источник AP". Нажмите кнопку "Создать".



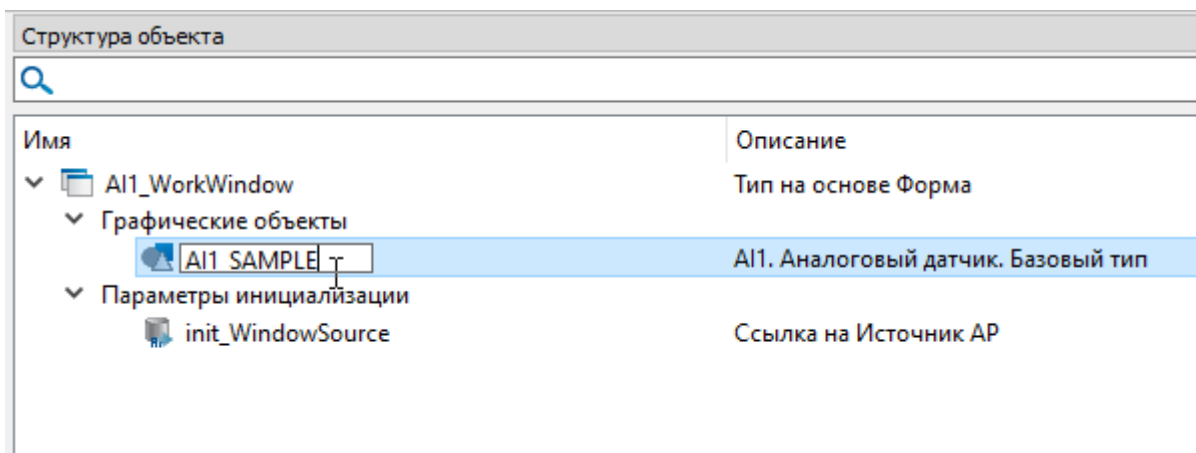
7. В структуре объекта появится вкладка "Параметры инициализации". Переименуйте созданный тип при помощи клавиши "F2" в "init\_WindowSource".



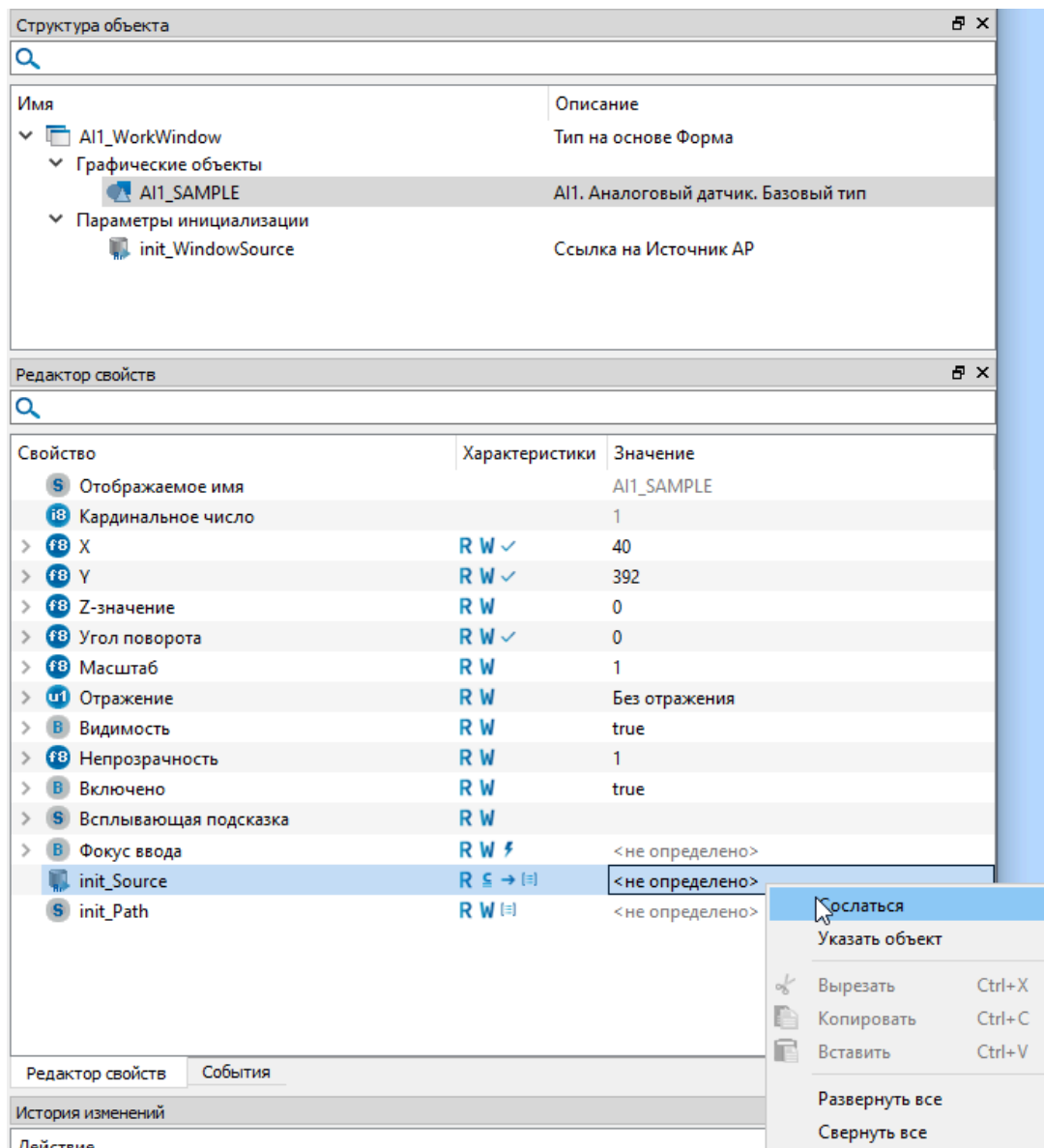
8. Добавьте путем перетаскивания на форму "AI1\_WorkWindow" базовый тип аналогового датчика "AI1\_SAMPLE".



9. Переименуйте добавленный тип в "AI1\_SAMPLE" с помощью клавиши "F2".



10. Выделите тип "AI1\_SAMPLE" нажав левой кнопкой мыши по нему. В окне "Редактор свойств" кликните правой кнопкой мыши по свойству "init\_Source" в поле "Значение" и из контекстного меню выберите команду "Сослаться".



15. В появившемся поле ввода введите название параметра инициализации "init\_WindowSource".

Структура объекта

Имя	Описание
AI1_WorkWindow	Тип на основе Форма
Графические объекты	
AI1_SAMPLE	AI1. Аналоговый датчик. Базовый тип
Параметры инициализации	
init_WindowSource	Ссылка на Источник AP

Редактор свойств

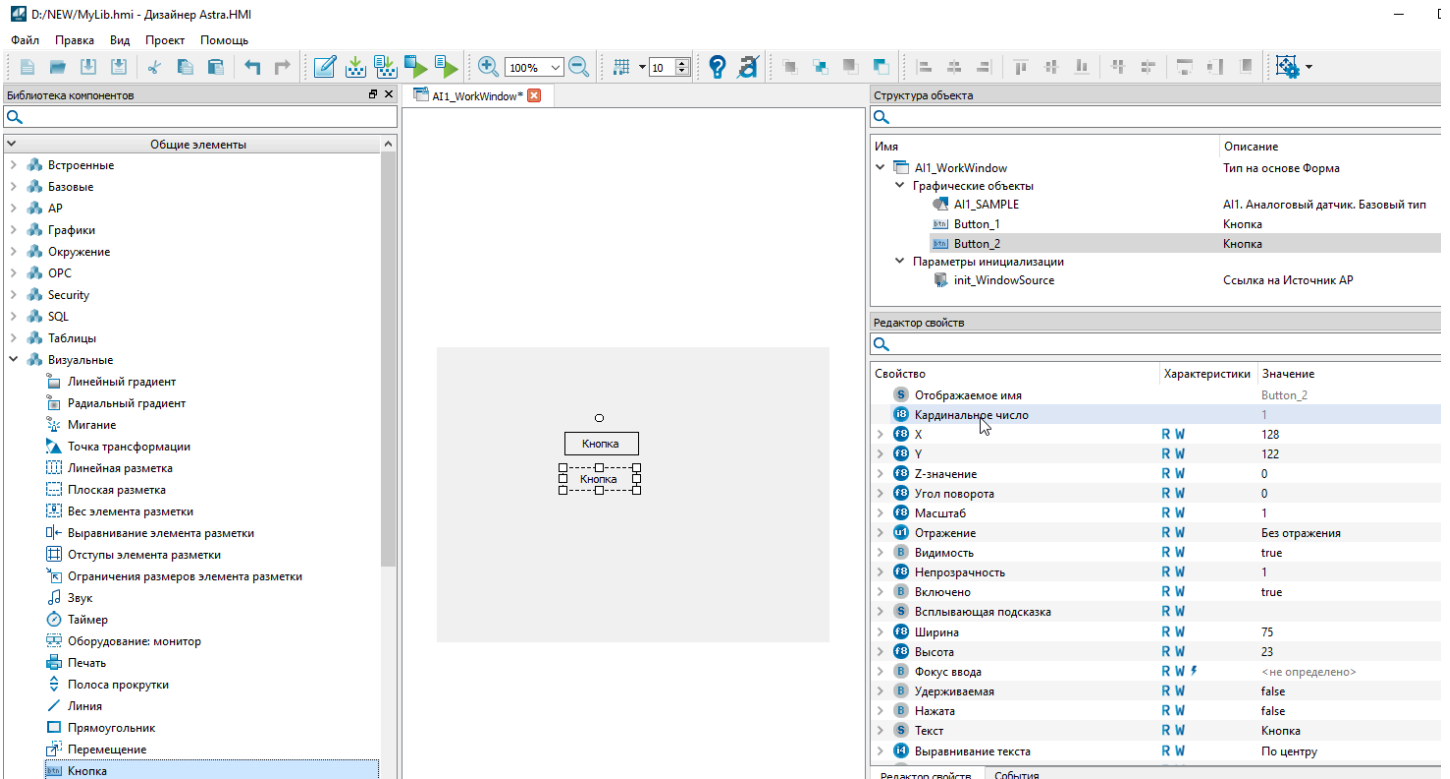
Свойство	Характеристики	Значение
Отображаемое имя		AI1_SAMPLE
Кардинальное число		1
X	R W ✓	40
Y	R W ✓	392
Z-значение	R W	0
Угол поворота	R W ✓	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Фокус ввода	R W ⚡	< не определено >
init_Source	R S → ✓	init_WindowSource



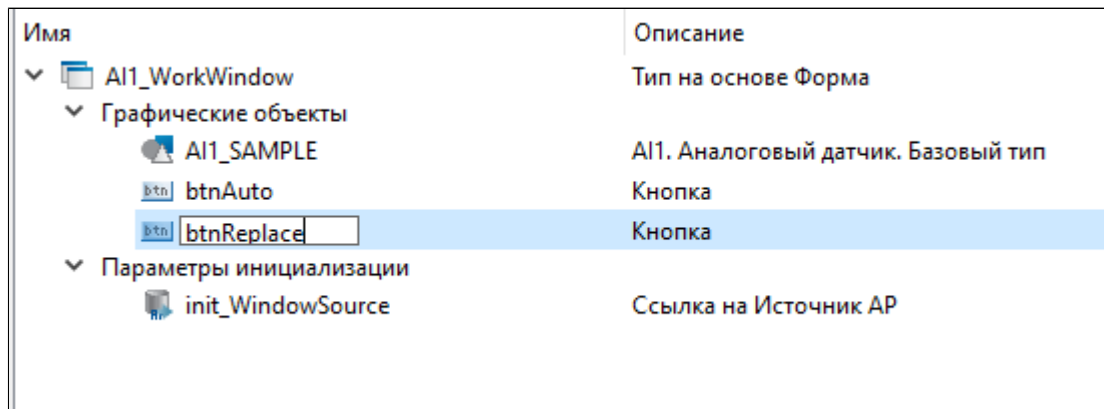
Свойство "init\_Path" в данном случае заполнять не нужно, путь будет взят из источника данных.

После добавления экземпляра типа аналогового датчика на форму и задания значения источника данных можно обращаться к параметрам аналогового датчика, добавленных в базовом типе (AI1\_SAMPLE). В качестве примера создадим 2 кнопки управления переменной MAN\_ON (Перевод в аналогового датчика в режим подмены).

16. Для добавления графического элемента "Кнопка" раскройте вкладку общие элементы и перейдите во вкладку "Визуальные". Путем перетаскивания добавьте и разместите на форме "AI1\_WorkWindow" две кнопки.



17. Задайте название типу "Button\_1" имя "btnAuto", а типу "Button\_2" имя "btnReplace" при помощи клавиши "F2".



18. Выделите тип "btnAuto" левой кнопкой мыши и в окне "Редактор свойств" задайте свойству "Текст" значение "Штатный".

Структура объекта

Имя

Имя	Описание
AI1_WorkWindow	Тип на основе Форма
Графические объекты	
AI1_SAMPLE	AI1. Аналоговый датчик. Базовый тип
btnAuto	Кнопка
btnReplace	Кнопка
Параметры инициализации	
init_WindowSource	Ссылка на Источник AP

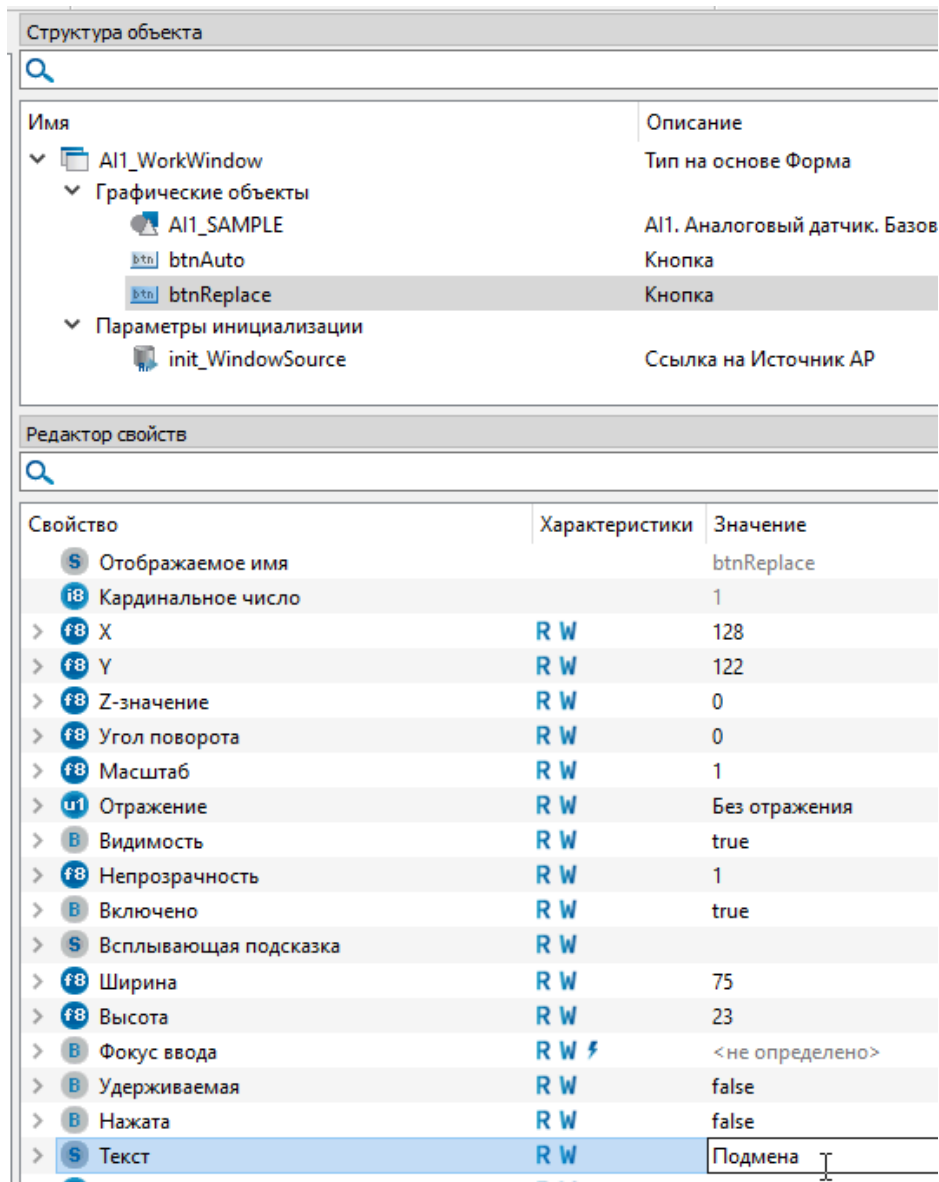
Редактор свойств

Свойство

Свойство	Характеристики	Значение
Отображаемое имя		btnAuto
Кардинальное число		1
X	R W	130
Y	R W	86
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	75
Высота	R W	23
Фокус ввода	R W ⚡	<не определено>
Удерживаемая	R W	false
Нажата	R W	false
Текст	R W	Штатный

19. Выделите тип "btnReplace" левой кнопкой мыши и в окне "Редактор свойств" задайте свойству "Текст" значение "Подмена".





20. В окне "Редактор свойств" перейдите во вкладку "События".

Структура объекта

Имя Описание

- AI1\_WorkWindow
  - Графические объекты
    - AI1\_SAMPLE AI1. Аналоговый датчик. Базовый тип
    - btnAuto Кнопка
    - btnReplace Кнопка
  - Параметры инициализации
    - init\_WindowSource Ссылка на Источник AP

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		btnReplace
Кардинальное число		1
X	R W	128
Y	R W	122
Z-значение	R W	0
Угол поворота	R W	0
Масштаб	R W	1
Отражение	R W	Без отражения
Видимость	R W	true
Непрозрачность	R W	1
Включено	R W	true
Всплывающая подсказка	R W	
Ширина	R W	75
Высота	R W	23
Фокус ввода	R W f	<не определено>
Удерживаемая	R W	false
Нажата	R W	false
Текст	R W	Подмена
Выравнивание текста	R W	По центру

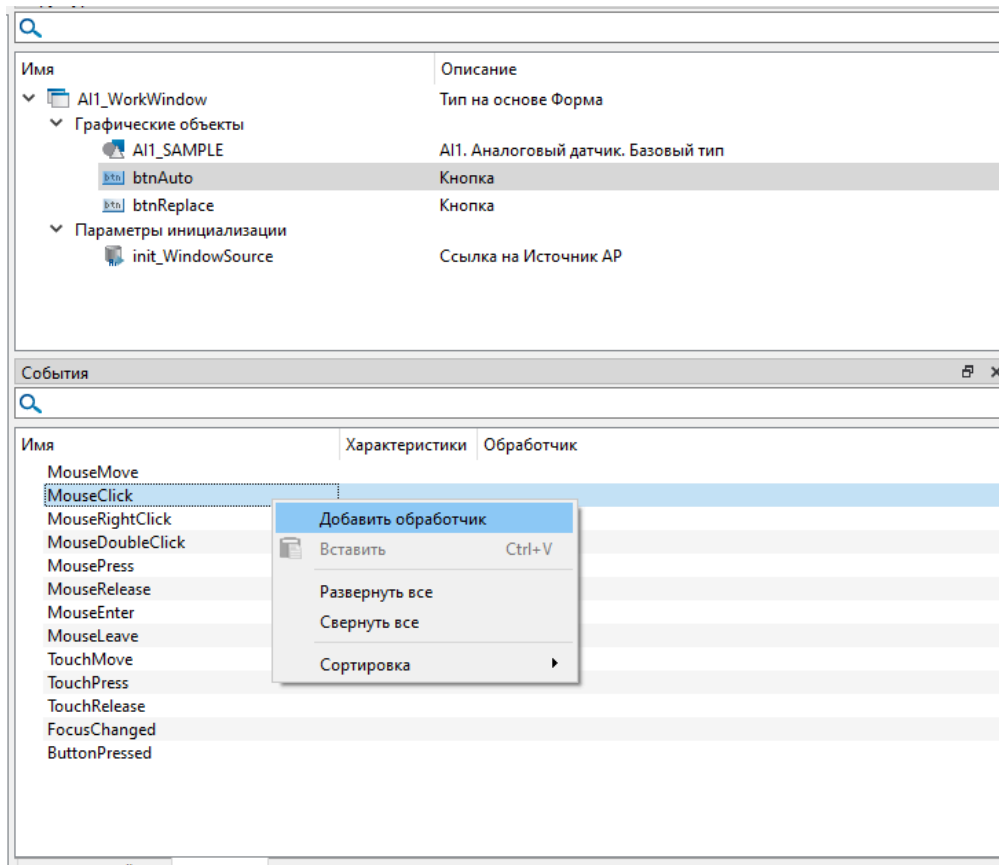
Редактор свойств    События

История изменений

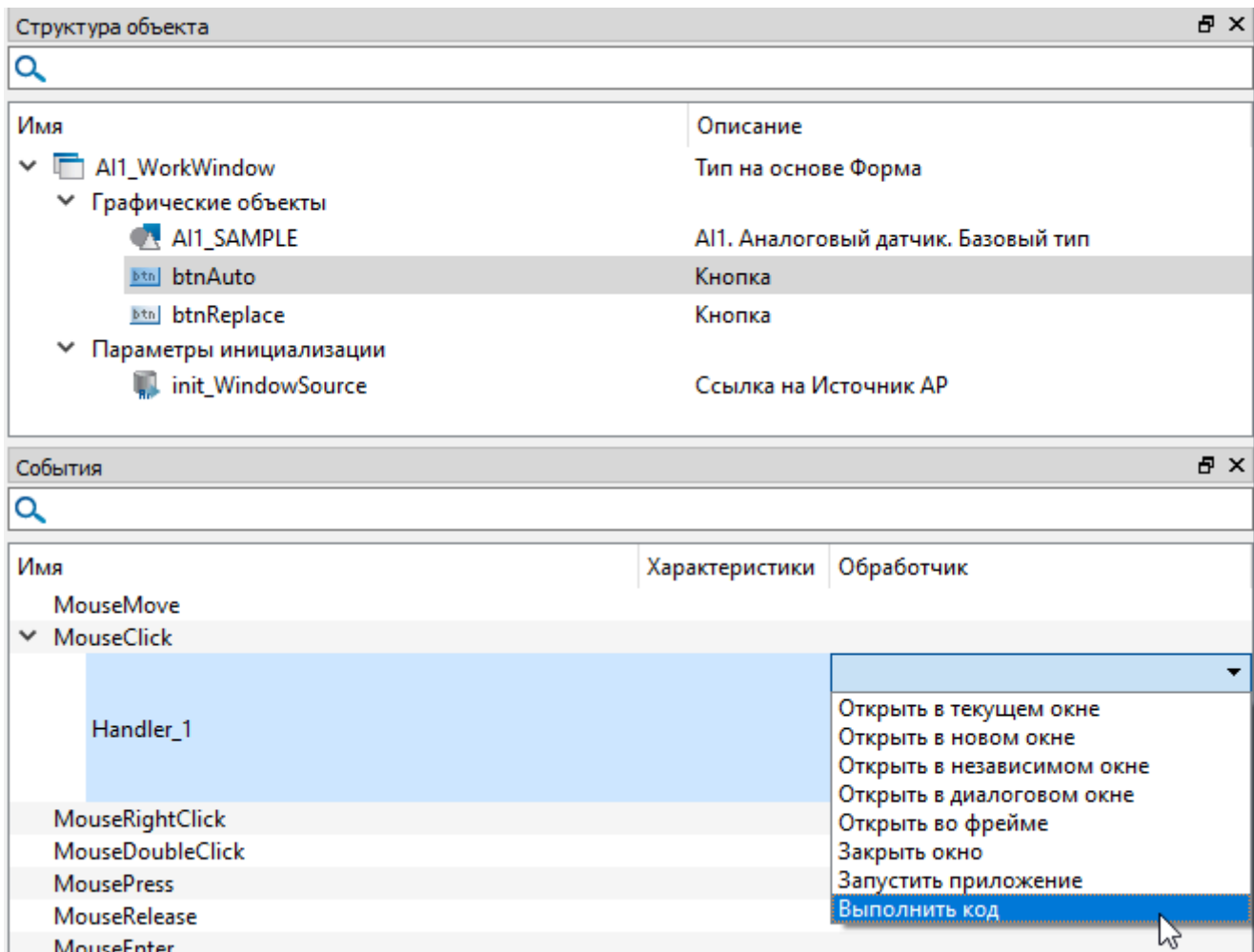


Во вкладке "События" можно задавать различные скрипты при возникновении перечисленных событий. Описание действия каждого события можно посмотреть в документации.

19. Во вкладке события нажмите правой кнопкой мыши по строке "MouseClicked" (Клик мыши) и выберите команду "Добавить обработчик".



20. Двойным кликом мыши нажмите на строчку "<неправильный обработчик>". В выпадающем списке выберите действие "Выполнить код".



21. Нажмите кнопку "Редактировать". В открывшемся окне необходимо вписать код, который будет выполняться по клику на данную кнопку. Для того, чтобы управлять переменной "MAN\_ON" в базовом типе, необходимо обратиться к ней через название добавленного базового типа.



Для записи значения необходимо записывать значение в переменную задания "wvalue".

Пример кода для кнопки "btnAuto":



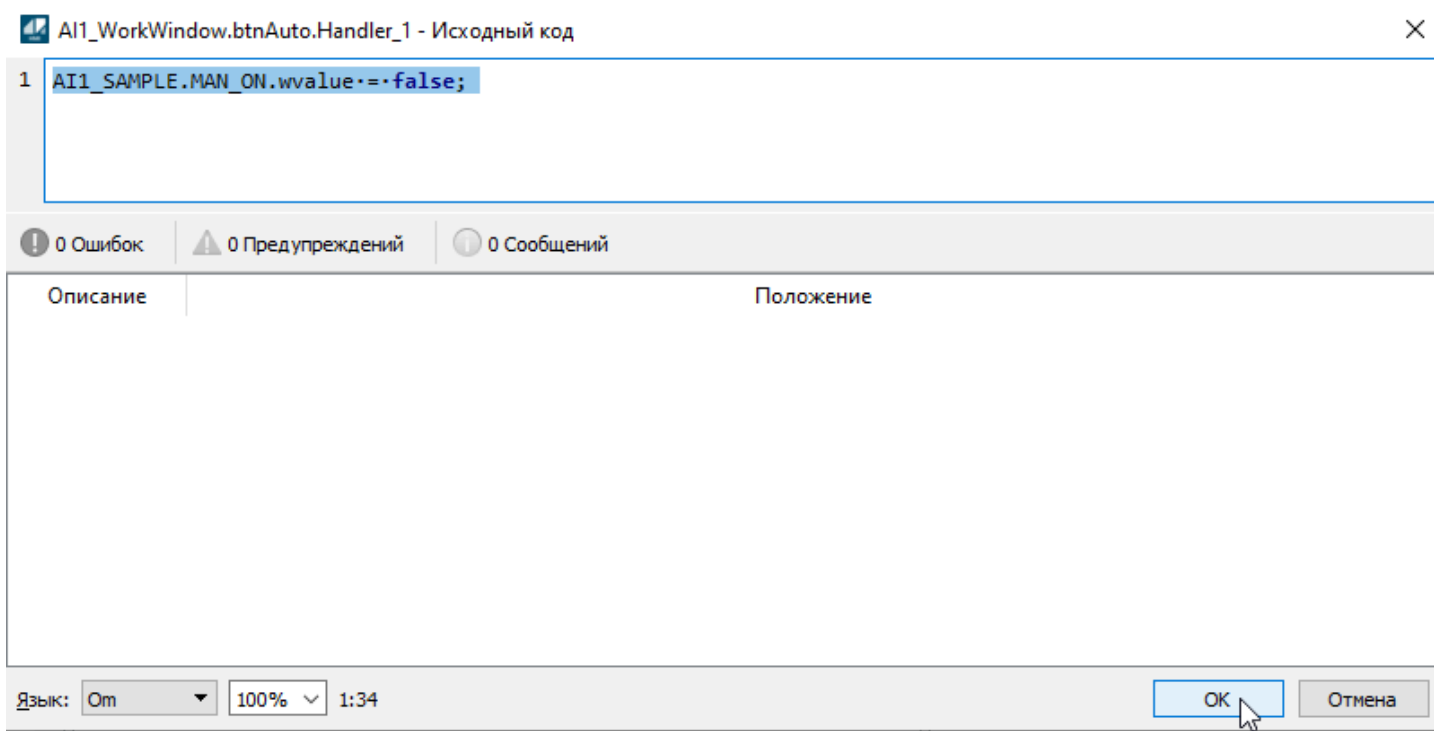
```
AI1_SAMPLE.MAN_ON.wvalue = false;
```



В обработчике событий при написании исходного кода можно выбрать язык программирования в поле "Язык": Astra.Om или

JavaScript. Обратите внимание, что язык JavaScript не компилируется средой разработки Astra.HMI.

В конце написания кода нажмите кнопку "OK".



Аналогичным образом, добавьте обработчик событий по событию "MouseClicked" для кнопки "btnReplace"

Пример кода для кнопки "btnAuto":



```
AI1_SAMPLE.MAN_ON.wvalue = true;
```

На данном этапе при нажатии кнопки "Штатный" в переменную "MAN\_ON" записывается значение "false", а при нажатии кнопки "Подмена" в переменную "MAN\_ON" записывается значение "true".

22. Выделите экранную форму "AI1\_WorkWindow" левой кнопкой мыши. Задайте свойству "Ширина окна" значение 500, а свойству "Высота окна" значение 400.

Структура объекта

Имя

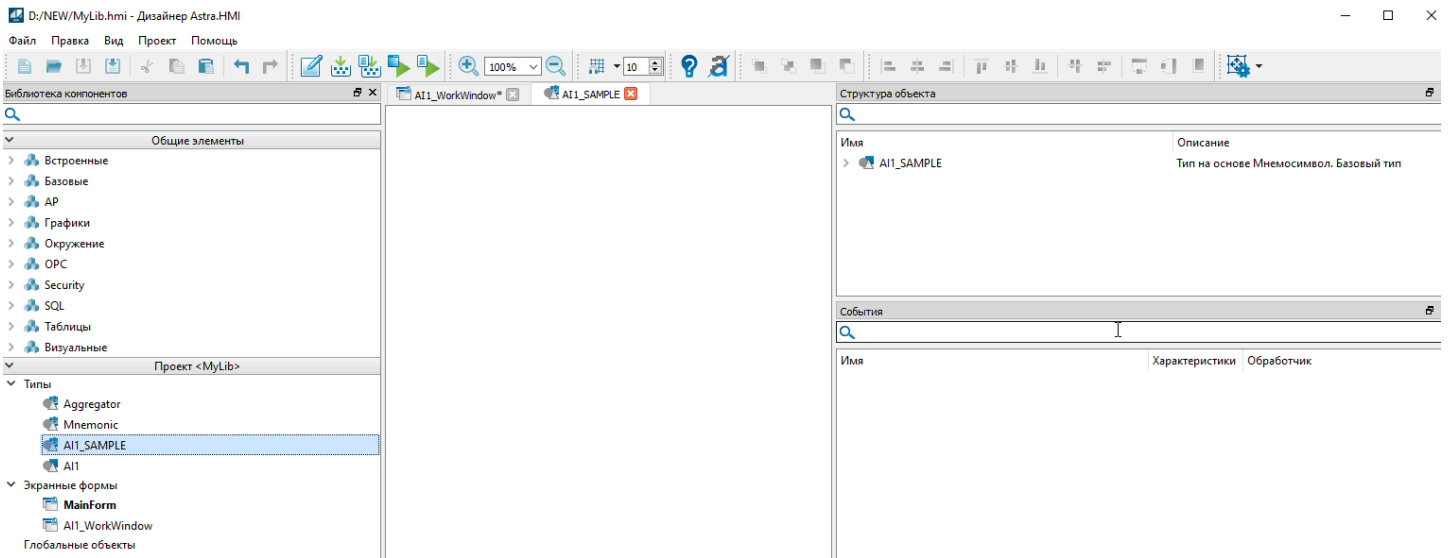
Имя	Описание
AI1_WorkWindow	Тип на основе Форма
Графические объекты	
AI1_SAMPLE	AI1. Аналоговый датчик. Базовый тип
btnAuto	Кнопка
btnReplace	Кнопка
Параметры инициализации	
init_WindowSource	Ссылка на Источник AP

Редактор свойств

Свойство	Характеристики	Значение
u2 Стиль пера	R W	Нет пера
f8 Толщина пера	R W	1
u4 Цвет заливки	R W	4293980400
u2 Стиль заливки	R W	Сплошная заливка
i4 X окна	R W	0
i4 Y окна	R W	0
i4 Ширина окна	R W	500
i4 Высота окна	R W	<input type="text" value="400"/>
s Заголовок окна	R W	Form_1

Необходимо добавить вызов рабочего окна по двойному клику мыши по графическому представлению "AI1".

23. Чтобы настроить вызов рабочего окна аналогового датчика перейдите в базовый тип "AI1\_SAMPLE" кликнув по нему двойным кликом мыши.



24. Выделите тип "AI1\_SAMPLE" левым кликом мыши и перейдите во вкладку "События".

Структура объекта

Имя	Описание
AIT_SAMPLE	Тип на основе Мнемосимвол. Базовый тип
Данные	
CONFIG	Элемент AP
MAN_ON	Элемент AP bool
MAN_VALUE	Элемент AP float
VALUE	Элемент AP float
QUALITY	Элемент AP int4
FACT_VALUE	Элемент AP float
SIGN	Элемент AP int4
MODE	Элемент AP bool
DIAGN	Элемент AP int4

Редактор свойств

Свойство	Характеристики	Значение
Отображаемое имя		AI1. Аналоговый датчик. Базовый тип
Модификатор доступа		Приватный
X	RW	0
Y	RW	0
Z-значение	RW	0
Угол поворота	RW	0
Масштаб	RW	1
Отражение	RW	Без отражения
Видимость	RW	true
Непрозрачность	RW	1
Включено	RW	true
Всплывающая подсказка	RW	
Фокус ввода	RW	<не определено>
init_Source	R S →	<не определено>

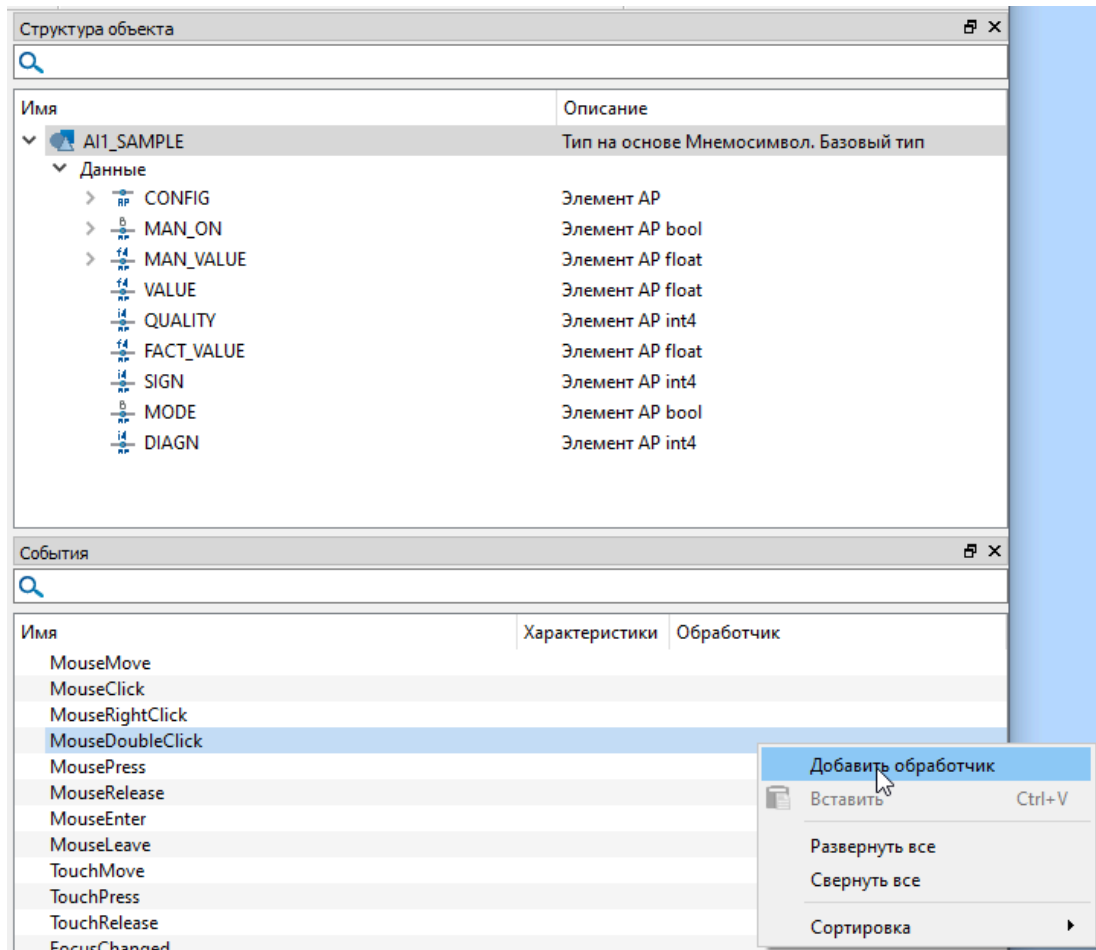
Редактор свойств    События



Во вкладке "События" можно задавать различные скрипты при возникновении перечисленных событий. Описание действия каждого события можно посмотреть в документации.

25. Во вкладке события нажмите правой кнопкой мыши по строке "MouseDownClick" (Двойной щелчок мыши) и выберите команду "Добавить обработчик".





26. Двойным кликом мыши нажмите на строчку "<неправильный обработчик>". В выпадающем списке выберите действие "Открыть в новом окне".

Структура объекта

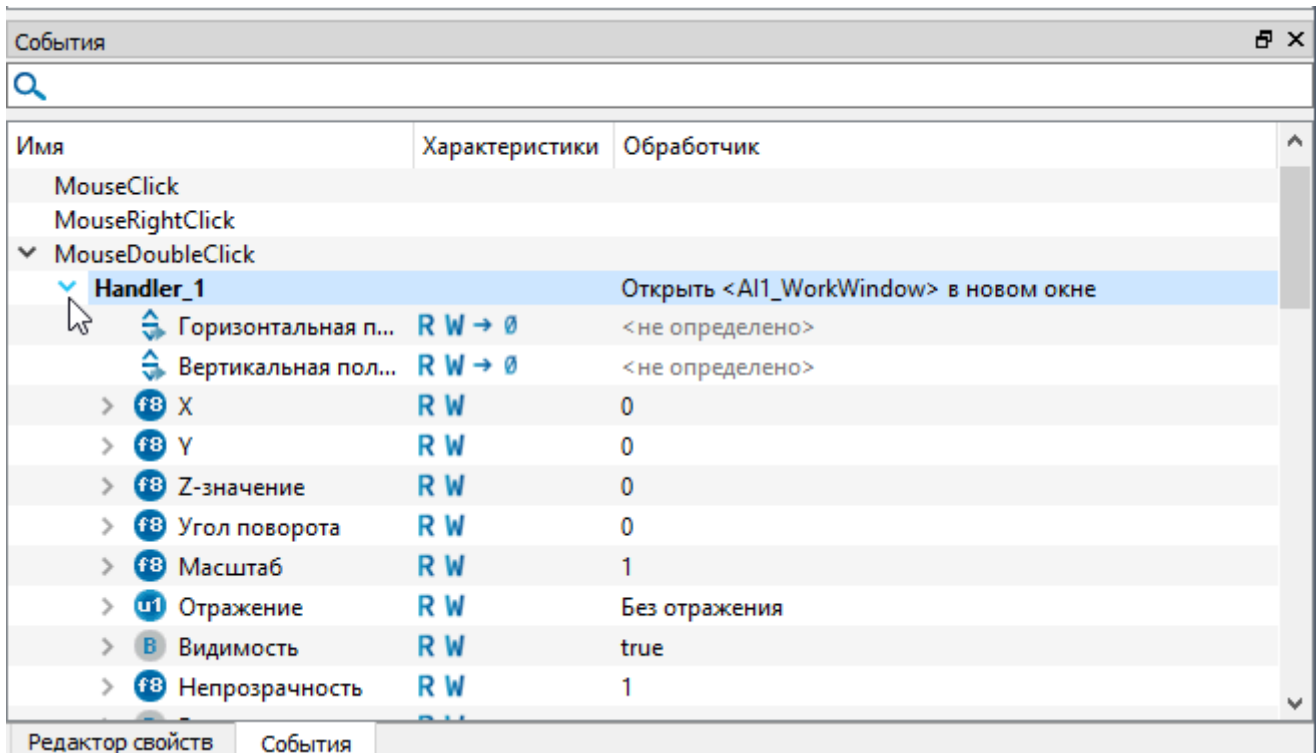
Имя	Описание
AI1_SAMPLE	Тип на основе Мнемосимвол. Базовый тип
Данные	
CONFIG	Элемент AP
MAN_ON	Элемент AP bool
MAN_VALUE	Элемент AP float
VALUE	Элемент AP float
QUALITY	Элемент AP int4
FACT_VALUE	Элемент AP float
SIGN	Элемент AP int4
MODE	Элемент AP bool
DIAGN	Элемент AP int4

События

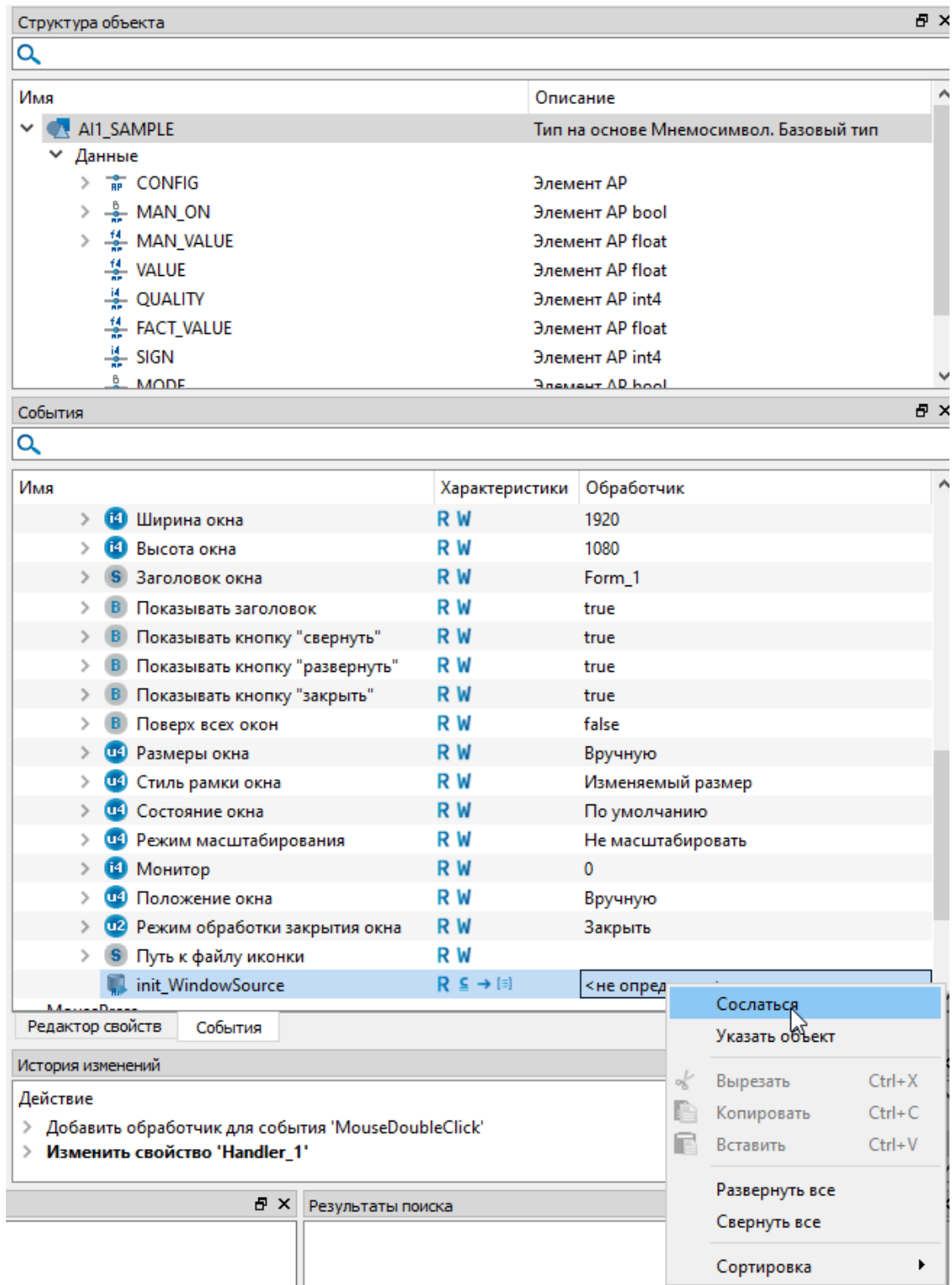
Имя	Характеристики	Обработчик
MouseMove		
MouseClicked		
MouseRightClick		
▼ MouseDoubleClick		
Handler_1		<ul style="list-style-type: none"> <li>Открыть в текущем окне</li> <li><b>Открыть в новом окне</b></li> <li>Открыть в независимом окне</li> <li>Открыть в диалоговом окне</li> <li>Открыть во фрейме</li> <li>Закрыть окно</li> <li>Запустить приложение</li> <li>Выполнить код</li> </ul>
MousePress		
MouseRelease		
MouseEnter		
MouseLeave		

27. Во втором выпадающем списке выберите рабочее окно "AI1\_WorkWindow".

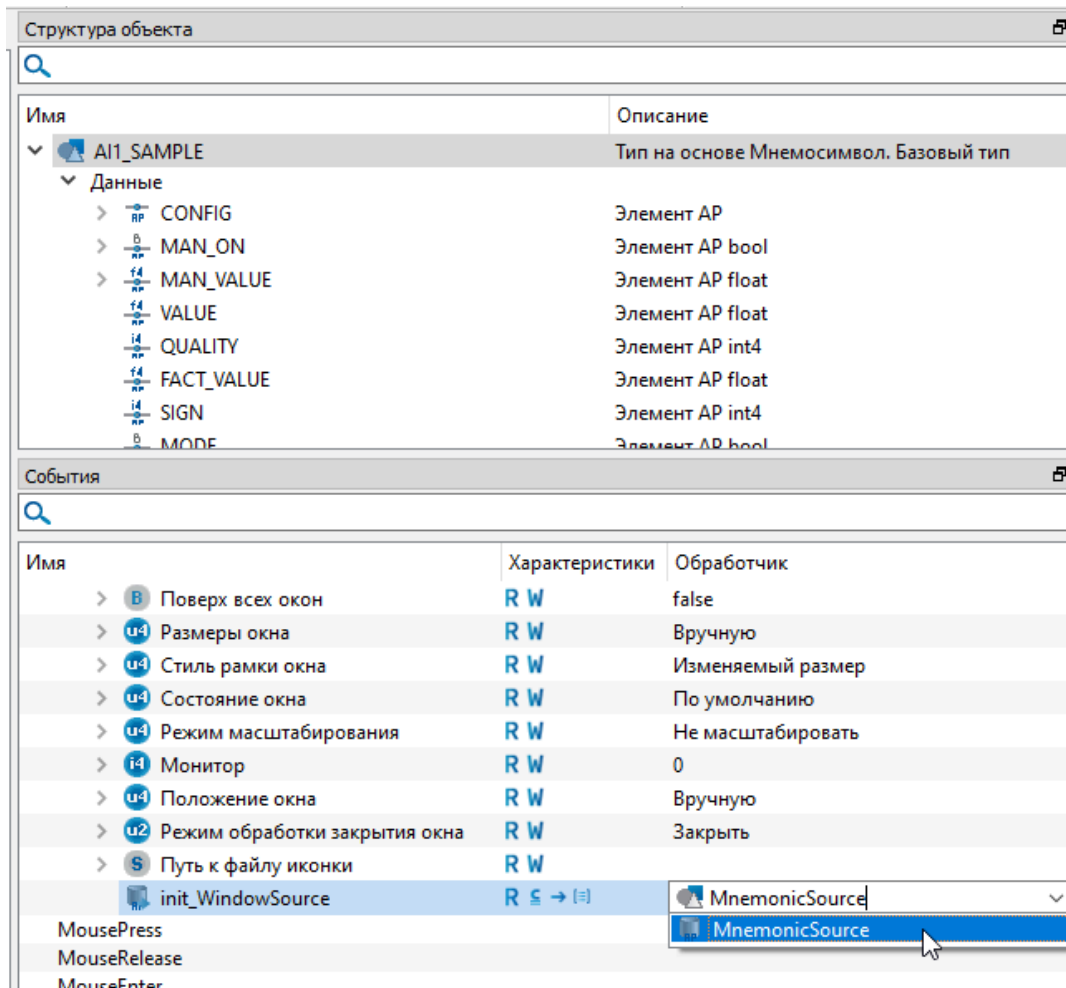




29. Пролитайте вниз до свойства "init\_WindowSource", нажмите правой кнопкой мыши по полю ввода и из контекстного меню выберите команду "Сослаться".



30. В появившемся поле ввода укажите значение источника мнемосимвола "MnemonicSource".

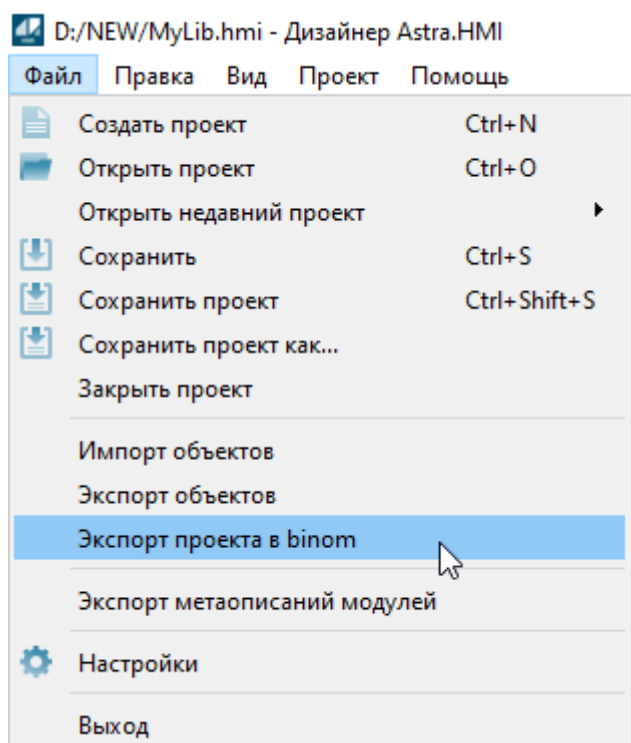


Таким образом, можно создавать и добавлять экранные формы для библиотечных типов.

## 1.2.4.7. Добавление библиотечного типа на мнемосхему проекта

Для добавления библиотечного типа на мнемосхему проекта необходимо выполнить следующие действия:

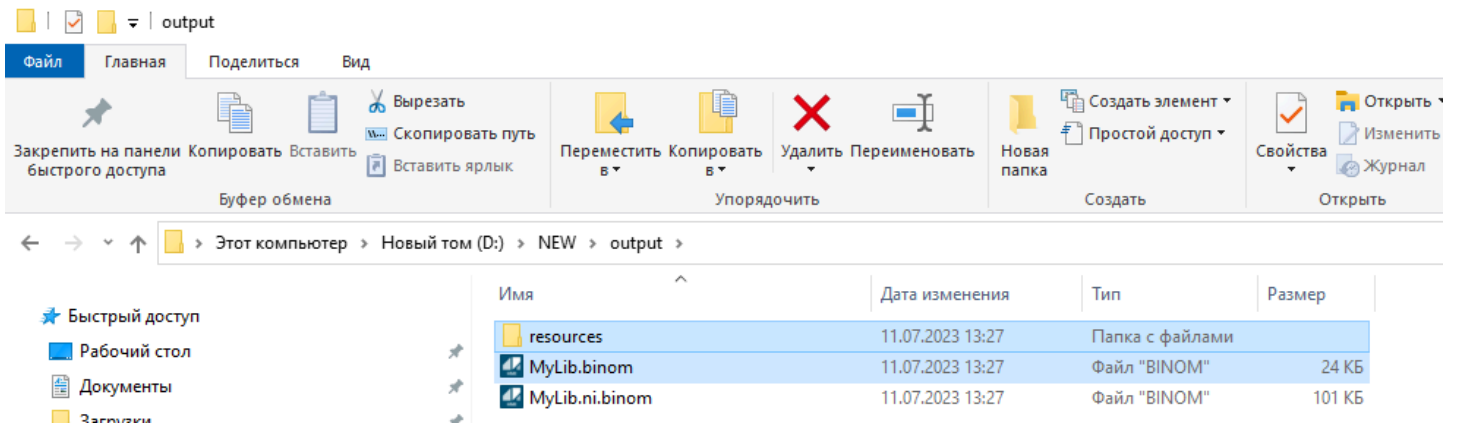
1. Сохраните файл с пользовательской библиотекой в формате \*.binom, воспользовавшись командой "Экспорт проекта в binom" из вкладки "Файл".



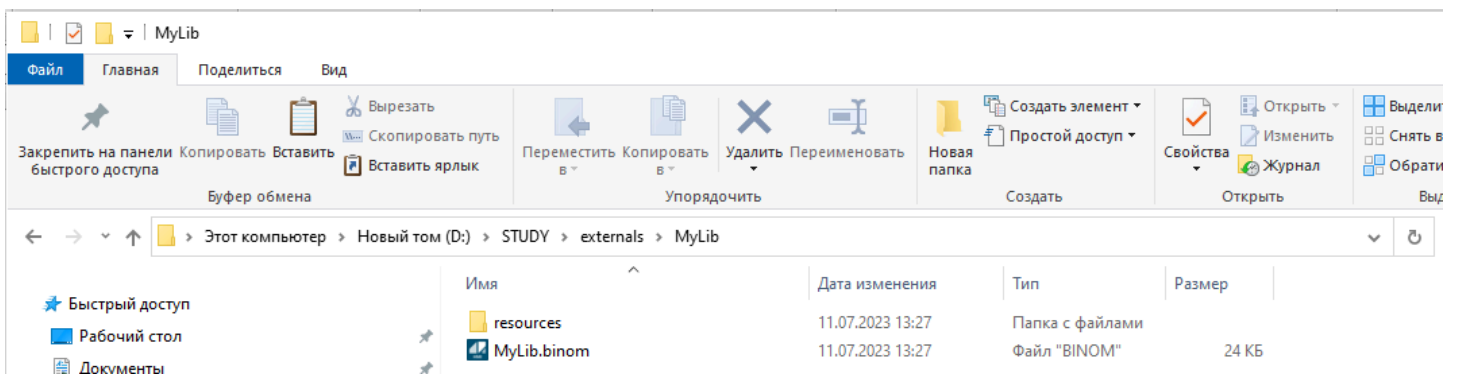
2. Файлы проекта будут сохранены по следующему пути:



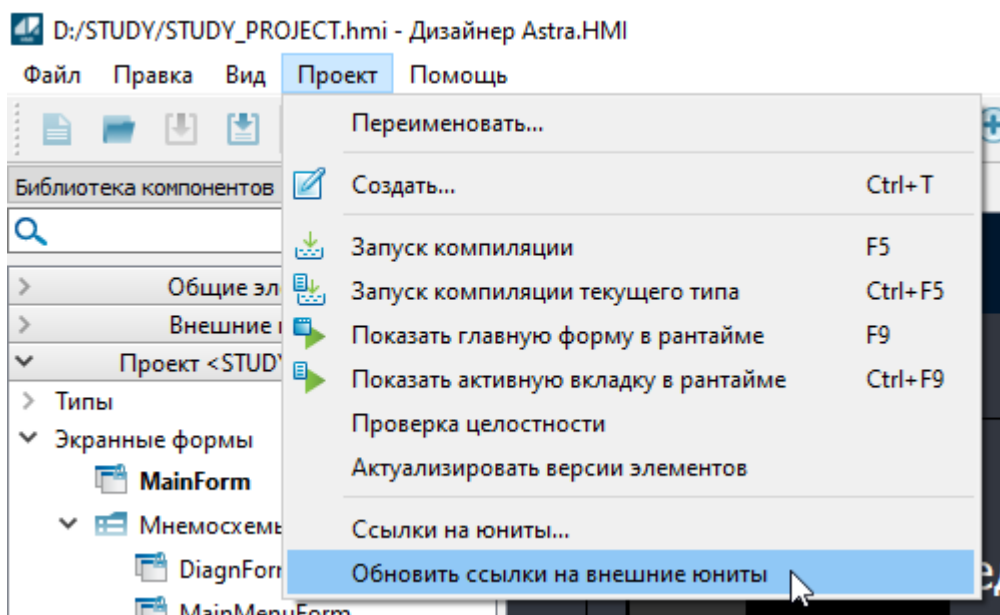
Папка проекта библиотеки/output



3. Скопируйте папку "resources" и файл формата \*.binom и вставьте в папку "externals" проекта.

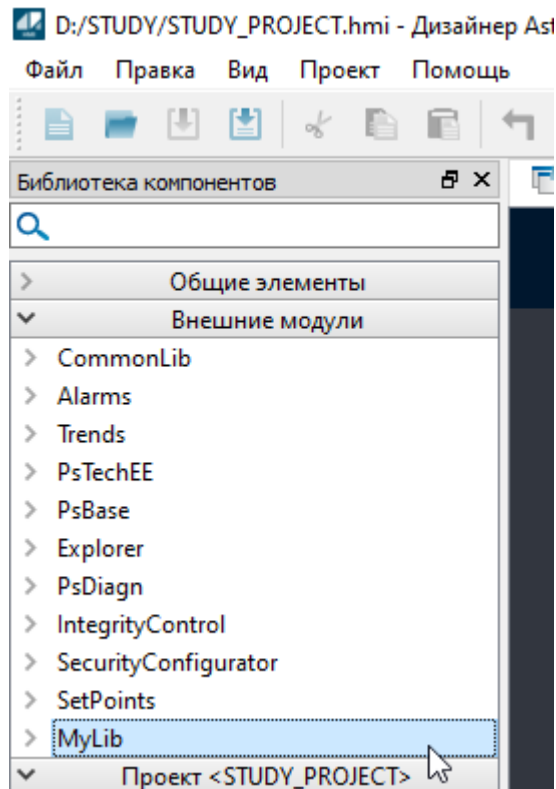


4. Откройте проект в режиме дизайнера и выполните команду "Обновить ссылки на внешние юниты" во вкладке "Проект".

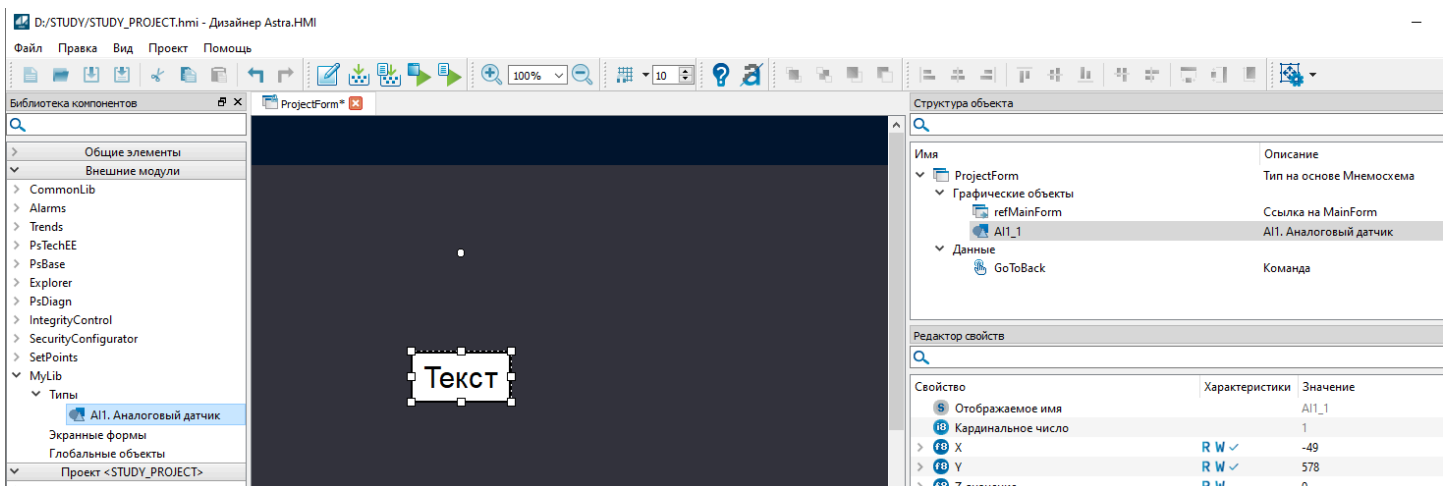




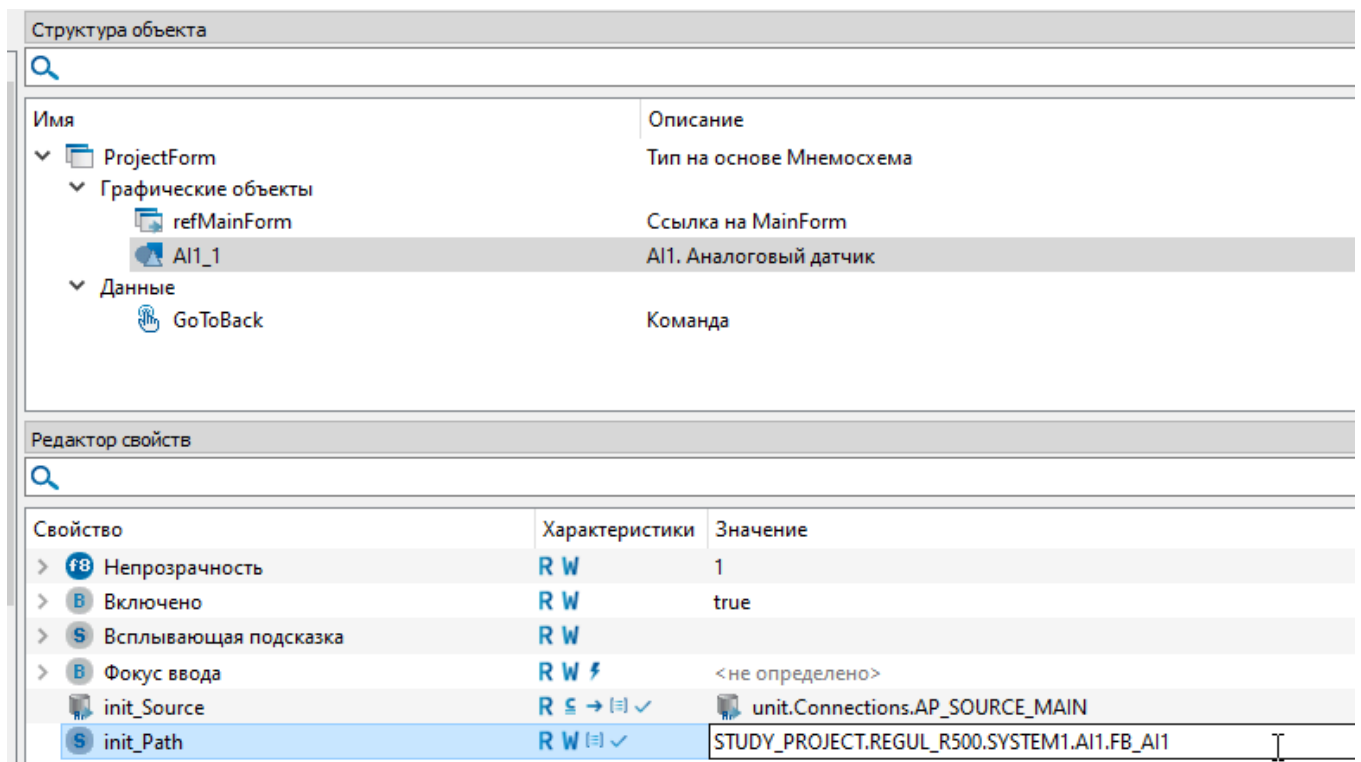
5. Раскройте вкладку "Внешние модули" и убедитесь, что созданная вами библиотека стала отображаться в списке.



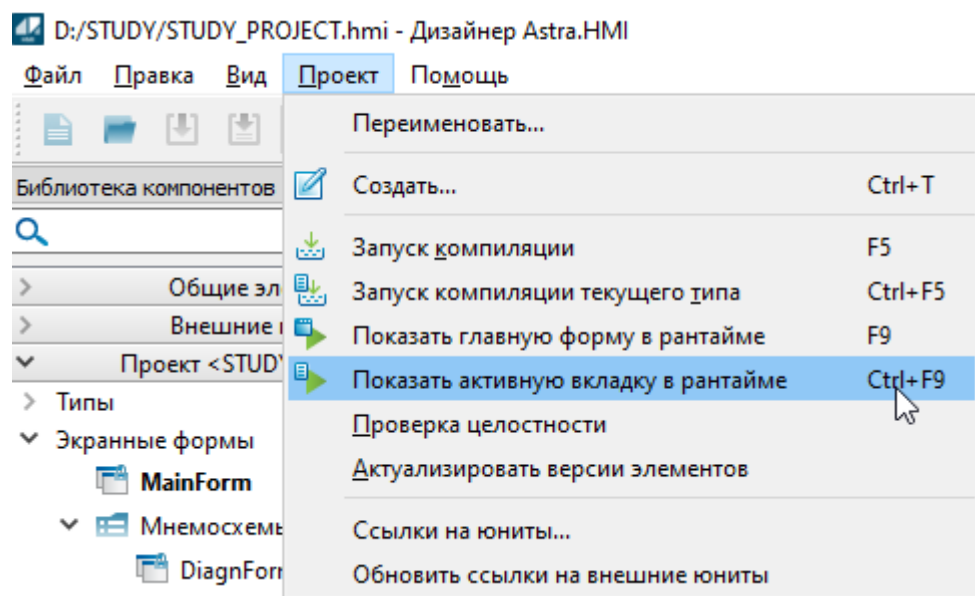
6. Раскройте вкладку MyLib, затем перейдите во вкладку "Типы" и добавьте на мнемосхему путем перетаскивания библиотечный тип "AI1".



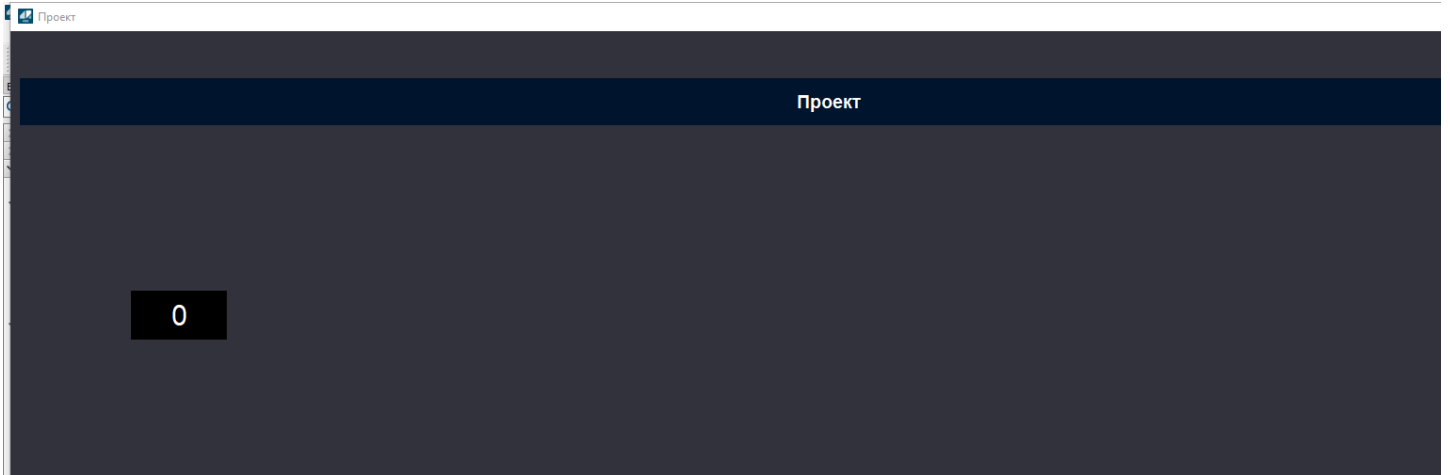
7. Выделите добавленный тип на мнемосхему левым кликом мыши и задайте свойства "init\_Path" и "init\_Source". Где "init\_Source" - ссылка на источник данных, а "init\_Path" - строка инициализации для аналогового датчика (путь до тега).



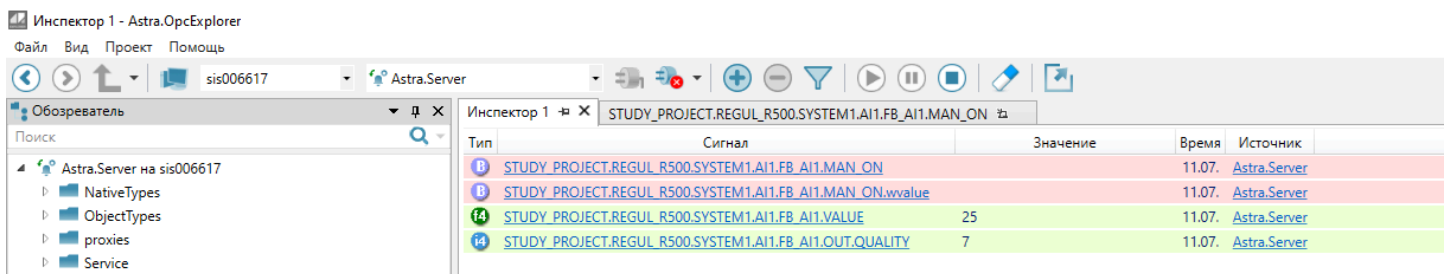
8. Запустите активную вкладку в режим исполнения при помощи команды "Показать активную вкладку в рантайме" из вкладки "Проект".



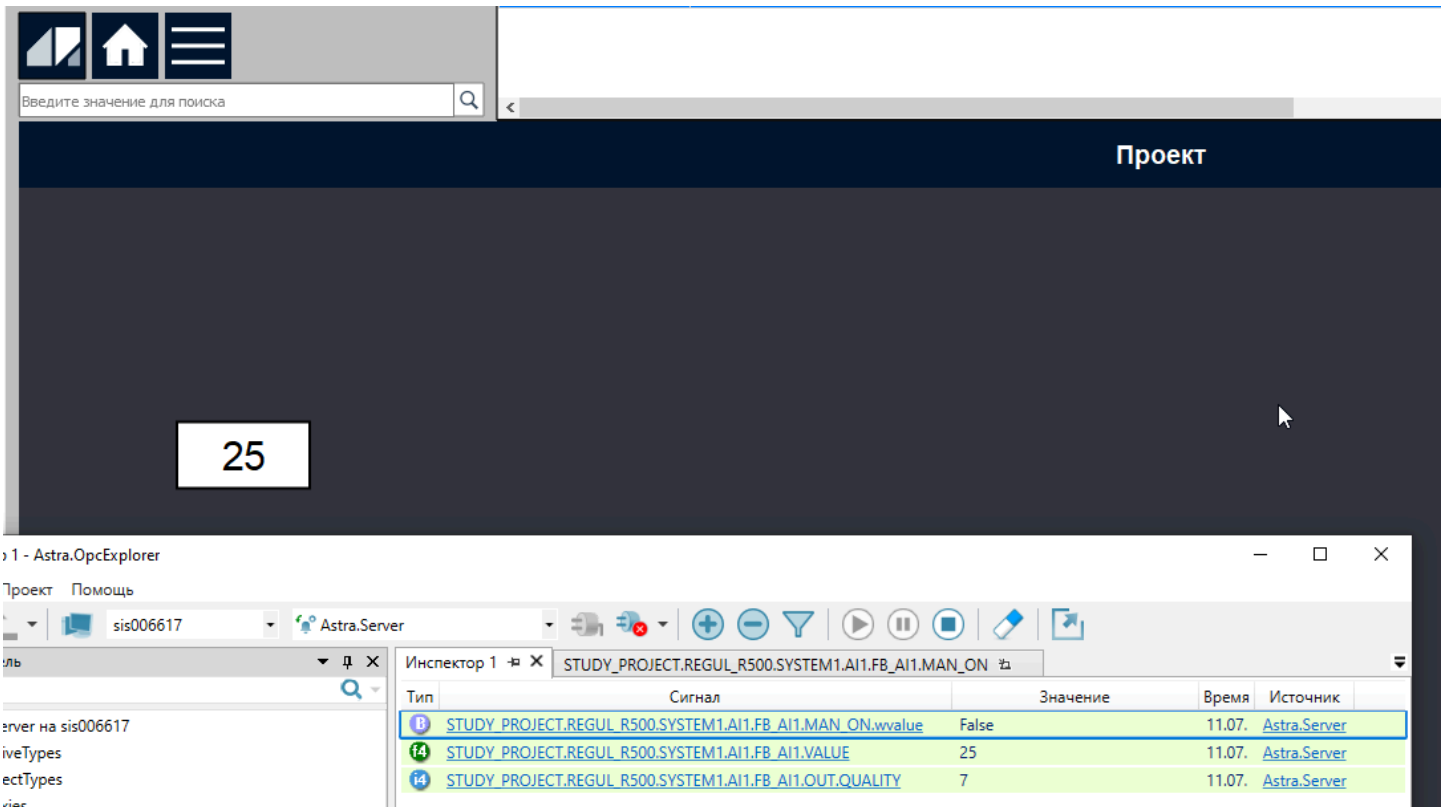
9. Убедитесь, что на запущенной мнемосхеме отображается текущее состояние аналогового датчика (цвет заливки черный, цвет шрифта белый, значение равно нулю).



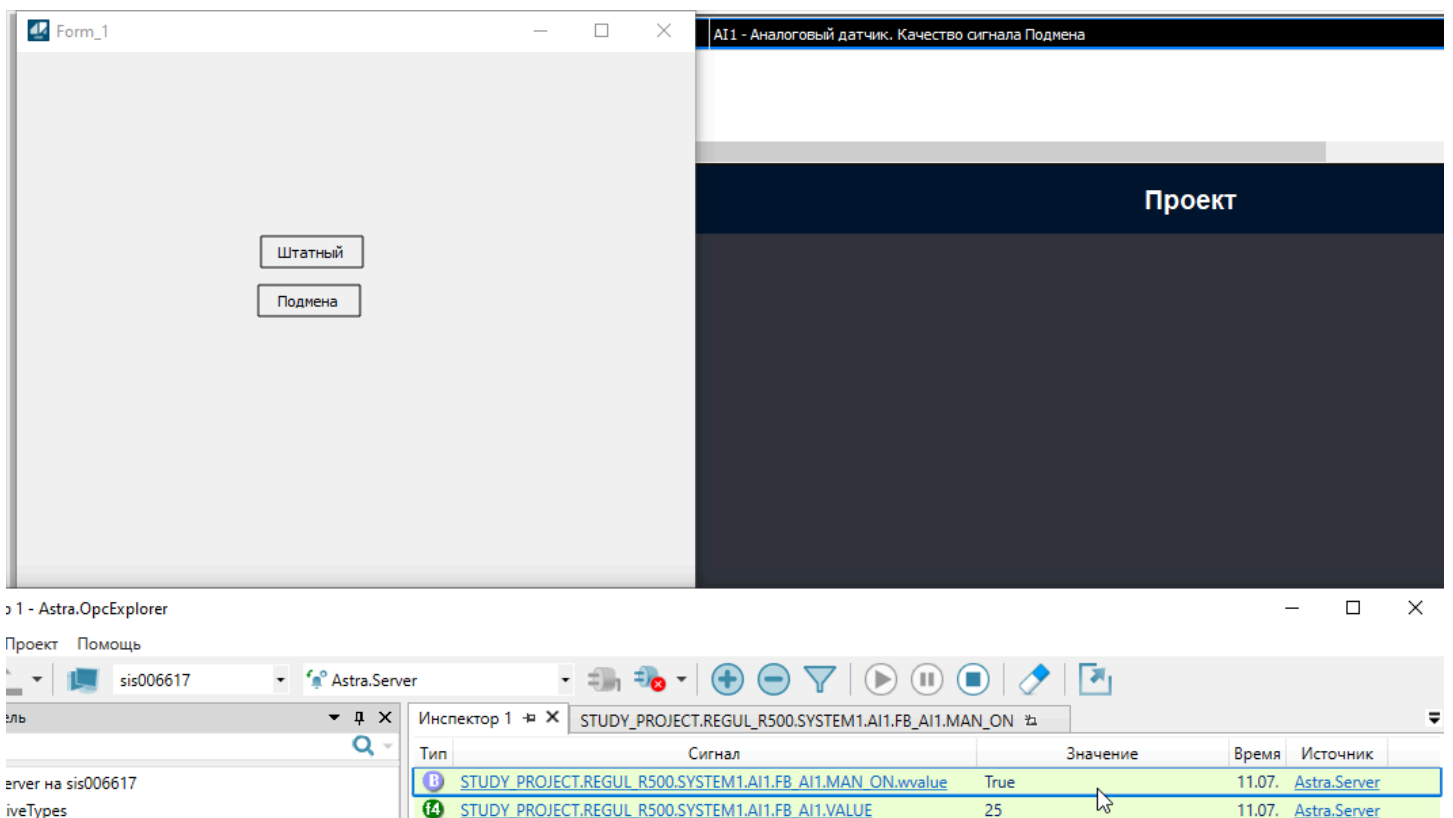
10. В утилите OpсExplorer подключитесь к Astra.Server и добавьте в инспектор сигналы: "VALUE", "QUALITY", "MAN\_ON" данного аналогового датчика.



11. Измените значение переменной "VALUE" на произвольное и установите значение "QUALITY" больше 6. Убедитесь, что состояние датчика изменилось (Цвет заливки стал белым, а цвет шрифта черный, отображение VALUE приняло значение из OpсExplorer).



12. Кликните двойным кликом по мнемосимволу. Откроется рабочее окно, нажмите кнопку "Подмена". Пронаблюдайте, что значение "MAN\_ON.wvalue" изменилось на "true".



13. Нажмите кнопку "Штатный". Пронаблюдайте, что значение "MAN\_ON.wvalue" изменилось на "false".

The screenshot shows the Astra.OpcExplorer interface. On the left, a window titled "Form\_1" contains two buttons: "Штатный" (Normal) and "Подмена" (Override). The main area displays a dark-themed interface with the word "Проект" (Project) at the top. Below the project name is a table with the following data:

Тип	Сигнал	Значение	Время	Источник
	STUDY_PROJECT.REGUL_R500.SYSTEM1.AI1.FB_AI1.MAN_ON.wvalue	False	11.07.	Astra.Server
	STUDY_PROJECT.REGUL_R500.SYSTEM1.AI1.FB_AI1.VALUE	25	11.07.	Astra.Server

Таким образом, можно создавать собственные библиотечные типы и добавлять их на мнемосхему с разработанной динамикой.

## 1.3. Языки программирования

Для исполнения формул, обработчиков функций, обработчиков событий применяется язык программирования Astra.Om.

Для расширения стандартных возможностей языка Astra.Om в Astra.HMI применяется язык программирования JavaScript.



Компилируемые языки программирования работают по принципу компиляции. Это означает, что исходный код программы, написанный на таком языке, сначала преобразуется в машинный код компилятором, а затем полученный исполняемый файл может запускаться непосредственно на компьютере пользователя.

Основное преимущество компилируемых языков заключается в их скорости выполнения. Поскольку код преобразуется в машинный код заранее, время выполнения программы существенно сокращается. Более того, компилируемые языки обеспечивают большую контроль над программой и возможность оптимизации.



JavaScript - является интерпретируемым языком программирования.

Интерпретируемые языки программирования, в отличие от компилируемых, не создают отдельный исполняемый файл, а выполняют программу в режиме интерпретации. При выполнении программы на интерпретируемом языке, исходный код построчно обрабатывается интерпретатором, который понимает команды и выполняет их.

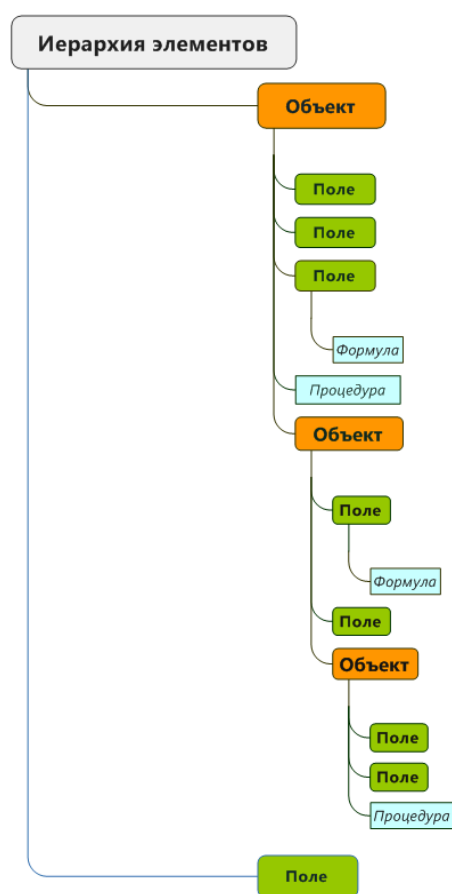
# Язык Astra.Om

Язык Astra.Om позволяет описывать исполняемые элементы скриптов (процедуры и формулы), которые запускаются в процессе работы компонентов ПК AstraRegul.

Скрипты можно создавать в процессе работы со следующими программными компонентами:

- Astra.Server.
- Astra.HMI.
- Astra.AStudio.

Процедуры и формулы в процессе выполнения обращаются к данным. На рисунке ниже приведена схема иерархии элементов. Иерархия элементов – иерархия элементов данных (объекты, поля), к которым могут обращаться исполняемые элементы (формулы, процедуры).



Элементы данных:



- › Поле – атомарный элемент данных, хранящий значение некоторого типа;
- › Объект – составной элемент данных, который может состоять из нескольких вложенных элементов данных (объектов или полей). Объект может сам хранить значение некоторого типа.

Исполняемые элементы:

**Формула** связана с полем, служит для пересчета значения поля. Пересчет формулы начинается автоматически, как только один из аргументов выражения-формулы меняет свое значение.

**Процедура** связана с объектом, служит для выполнения последовательности инструкций после наступления какого либо события, связанного с объектом.

## Относительная адресация

Для работы с иерархией объектов используются следующие ключевые слова и символы:

- me – указатель на текущий объект иерархии;
- @N – обращение к N-му уровню иерархии выше текущего объекта:
  - @0 – текущий объект, аналогично me
  - @1 – родительский объект
  - @2 – родитель родителя



Пусть полный тег объекта: AK.DMN.LU1.NPS1.TANK1.SW1.State:

- me – обращение к самому объекту:  
AK.DMN.LU1.NPS1.TANK1.SW1.State
- @1 – обращение к родительскому объекту:  
AK.DMN.LU1.NPS1.TANK1.SW1;
- @2 – "прыжок" на два уровня вверх по иерархии объектов:  
AK.DMN.LU1.NPS1.TANK1;
- @3 – "прыжок" на три уровня вверх по иерархии объектов:  
AK.DMN.LU1.NPS1.

\$ – символ «доллар» применяется в ситуациях, когда тег объекта противоречит стандартным правилам написания - например, наличие пробелов или запрещенных символов. Символ "\$" указывает компилятору, что написанная после него строка (заклученная в двойные кавычки) является путем до объекта в иерархии.



Путь к атрибуту объекта содержит пробелы, поэтому нужно использовать \$:

X: uint1 = \$"Уровень 1.Уровень 2.Объект А.Атрибут объекта";

## Явное чтение аргумента формулы

При компиляции формул для каждого аргумента создаются привязки, которые отслеживают состояния аргументов и запускают пересчет формулы каждый раз при изменении аргумента. Чтобы компилятор не отслеживал состояние некоторого аргумента при пересчете, используйте оператор явного чтения `read`.



Значение сигнала вычисляется по формуле:  $X + Y * Z$ .  
Необходимо, чтобы значение пересчитывалось при изменениях  $X$  и  $Y$   
и не пересчитывалось при изменениях  $Z$ :

```
X + Y * read Z
```

# Переменные и константы

## Переменные

Переменная – именованное значение, которое может быть изменено в процессе выполнения процедуры. В имени переменной допускается использование символов латинского алфавита (A-Z, a-z), цифр и символа подчеркивания ( \_ ).

Возможно объявление инициализированных и неинициализированных переменных.



X: uint1 = 10; //типизированная переменная с инициализированным значением

T: float; //неинициализированная типизированная переменная

W: var = 5.8; //объявление переменной с автоматическим выбором типа значения

C2: var = 5 + 10; //объявление с вычислением

## Константы

Константа – это значение, которое не может быть изменено в процессе выполнения процедуры. Константа может быть именованной или неименованной. В имени константы допускается использование символов латинского алфавита (A-Z, a-z), цифр и символа подчеркивания ( \_ ).



Различные способы объявления констант:

Q: const = 7; //объявление константы с автоматическим выбором типа

X: 5; //краткий синтаксис объявления константы в автоматическим выбором типа

Y1: uint2 const = 10; // объявление константы с указанием типа

W: 0xF501; //объявление константы - целого числа, заданного в 16-ричном виде

```
P: 0b00001000; //объявление константы - целого числа, заданного в двоичном виде  
M: 1.5; //объявление константы - вещественного числа  
N: 1.3e-10; //объявление константы - вещественного числа (экспоненциальный вид)
```

Вычисляемая константа – это константа, полученная путем вычислений из других констант. Может применяться для придания большей семантики коду процедуры.



```
Const1: 5; // объявим одну константу  
Const2: 10; // объявим вторую константу  
X: Const1 + Const2; // третья константа как результат сложения двух других  
Y: int1 const = X + 10; // типизированная константа как результат вычисления
```

## Область видимости

Область видимости – это часть кода, в рамках которого доступна переменная или константа. Область видимости обозначается фигурными скобками {}. Области видимости могут быть вложенными друг в друга. Константа или переменная, объявленная внутри области видимости не доступна за ними, но доступна во всех вложенных областях видимости.

### Пример



```
{
    X: var = 10;
    B: bool = true;
    if (B)
    {
        Y: X + 1; // допустимое обращение к X
    }
    else
    {
        //недопустимое обращение Y
        //Y объявлен ни в этой области видимости, ни в родительской
        Y = 0;
    }
}
```

# Специальные значения

## NaN

NaN (Not-a-Number) – это особое значение числа с плавающей запятой, которое показывает, что результат операции либо не определен, либо его нельзя представить значением вещественного типа.

При каких условиях возникает NaN:

- › Один из аргументов операции равен NaN.
- › Один из аргументов операции имеет недопустимое значение. Например, извлечение квадратного корня из отрицательного числа.
- › Операция не имеет однозначного результата:
  - › деление нуля на ноль;
  - › деление бесконечности на бесконечность;
  - › умножение нуля на бесконечность;
  - › сложение положительной и отрицательной бесконечностей.

NaN не равен ни одному другому значению, даже самому себе. Чтобы проверить, является ли число значением NaN, используйте функцию `Math.isNaN`.

NaN может быть как положительным, так и отрицательным. Чтобы определить знак, используйте функцию `Math.SignBit`.

## Бесконечность

Infinity (бесконечность) – это особое значение числа с плавающей запятой, которое показывает, что результат операции выходит за границы допустимых значений возвращаемого типа.

Например:  $1.0 / 0$ .

Чтобы проверить, является ли число значением Infinity, используйте функцию `Math.IsInf`.

Бесконечность может быть как положительной ( $+\infty$ ), так и отрицательной ( $-\infty$ ). Чтобы определить знак бесконечности, используйте функцию `Math.SignBit`.

## Отрицательный ноль

Отрицательный ноль ( $-0$ ) – это значение числа с плавающей запятой, которое может появиться в результате математических операций. Отрицательный ноль равен положительному (обычный ноль) и отличается от него только битом знака.

Чтобы определить является ли ноль положительным или отрицательным, нужно вызвать функцию `Math.SignBit`.

В большинстве случаев знак нуля не влияет на результат выполнения операций. Исключение – операции, результат выполнения которых является знаковым нулем или бесконечностью:

- ›  $1/+0 = +\infty$ ;
- ›  $1/-0 = -\infty$
- ›  $-0*1 = -0$ ;
- ›  $-0/1 = -0$



# Типы данных

Тип	Описание	Размер	Допустимые значения
int1	Знаковое целое	1 байт	[-128; 127]
uint1	Беззнаковое целое	1 байт	[0; 255]
int2	Знаковое целое	2 байта	[-32 768; 32 767]
uint2	Беззнаковое целое	2 байта	[0; 65 535]
int4	Знаковое целое	4 байта	[-2 147 483 648; 2 147 483 647]
uint4	Беззнаковое целое	4 байта	[0; 4 294 967 295]
int8	Знаковое целое	8 байт	[-9 223 372 036 854 775 808; 9 223 372 036 854 775 807]
uint8	Беззнаковое целое	8 байт	[0; 18 446 744 073 709 551 615]
float	Значение с плавающей запятой	4 байта	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ Точность: 6-9 цифр
double	Значение с плавающей запятой	8 байт	$[\pm 5.0 \times 10^{-324}; \pm 1.7 \times 10^{308}]$ Точность: 15-17 цифр
bool	Логическое значение	1 байт	true, false
string	Текстовая строка в кодировке UTF16	2 байта × кол-во символов Макс. 2 ГБ	
variant	Универсальный тип данных		Принимает значения любых вышеперечисленных элементарных типов

Чтобы компилятор языка автоматически выбрал необходимый тип данных для переменной, используйте ключевое слово `var` при инициализации значения.



W: `var = 5.8; //` объявление переменной с автоматическим выбором типа

Чтобы компилятор языка автоматически выбрал необходимый тип данных для константы, используйте ключевое слово `const` при инициализации значения.



Q: `const = 7; //` объявление константы с автоматическим выбором типа  
X: `5; //` краткий синтаксис объявления константы с автоматическим выбором типа

## Тип данных `variant`

`Variant` – универсальный тип данных, который может содержать любые виды информации. Способен гибко принимать значения любых элементарных типов:

- целочисленные или вещественные значения;
- логические значения;
- строковые значения;
- неопределенное значение `VT_EMPTY`.

Применяйте тип данных `variant` в следующих случаях:

- если при написании скрипта невозможно заранее точно определить тип данных, хранимый в переменной.
- если известно, что содержимое переменной на этапе исполнения скрипта будет меняться в широких диапазонах.

Чтобы создать переменную `variant` неявным способом, присвойте переменной `variant` значение любого элементарного типа.

## Пример



```
v0: variant; //создать пустую переменную типа variant (VT_EMPTY)
v2: variant = 100; //создать переменную типа variant с
инициализацией значения
s: string = "строка"; //создать переменную типа string
v1: variant = s; //создать переменную типа variant и присвоить ей
значение переменной
```

Также переменную типа `variant` можно создать явным способом, используя набор функций `Variant.From<T>`.

## Неопределенное значение (VT\_EMPTY)

Тип данных `variant` может принимать неопределенное значение `VT_EMPTY` в следующих ситуациях:

› не было инициализировано начальное значение типа `variant`:



```
v0: variant; //создать неинициализированную переменную типа
variant
```

› результат операции будет равен `VT_EMPTY`, если один из аргументов равен `VT_EMPTY`:



```
v0: variant; //создать неинициализированную переменную типа
variant (VT_EMPTY)
v1: variant = v0 + 1; //v1 = VT_EMPTY, т.к. аргумент v0 = VT_EMPTY
```

› результат операции будет равен `VT_EMPTY`, если операция логически-некорректна. Например, попытка сложения числа со строкой приведет к значению `VT_EMPTY`:



```
v0: variant = 1;
```

```
v1: variant = "my string";
```

```
v2: variant = v0 + v1; //недопустимая операция - сложение числа и строки
```

## Набор функций для работы с типом variant

Функции для работы с типом находятся в пространстве имен Variant:

- › Variant.From<T> - набор функций для создания переменной типа variant явным способом;
- › Variant.MayConvertTo<T> - набор функций для проверки возможности конвертации типа variant в какой-либо элементарный тип данных;
- › Variant.To<T> - набор функций для конвертации типа variant в один из элементарных типов данных;
- › Variant.Is<T> - набор функций для проверки типа значения в переменной variant.

## Операции над данными типа variant

В общем случае диапазон возможных операций определяется типом значения, которое содержится в переменной variant. Например, тип variant, содержащий булево значение (true или false) может быть аргументом в логических операциях.

На содержимое переменной типа variant не действует ограничение по мощности. Возможно как расширение по мощности, так и сжатие по мощности. Пределы возможных значений ограничены лишь пределами самых мощных типов.

## Пример



```
//переменная V1 будет содержать 130 (расширение до  
элементарного типа int2)  
X1: int1 = 100;  
Y1: int1 = 30;  
V1: variant = X1 + Y1;  
//Переменная V2 будет содержать 5 (сжатие до элементарного типа  
int1)  
X2: int2 = 32767;
```

```
Y2: int2 = 32762;  
V2: variant = X2 - Y2;  
//переменная R1 = 127, т.к. достигнут предел мощности типа int1  
R1: int1 = X1 + Y1;  
//переменная R2 = 5, но тип переменной по прежнему int2  
R2: int2 = X2 - Y2
```



//Пример некорректной попытки уместить в переменной значение больше максимального

```
X1: uint8 = 18446744073709551615; //максимально-возможное значение uint8
```

```
V1: variant = X1 + 1;
```

# Неявное преобразование типов

Неявное преобразование выполняется, когда операция требует один тип данных, а подставляется значение другого типа.

В этом случае подставленное значение будет автоматически приведено к нужному типу данных.

Тип	Типы приведения
int1	int2, int4, int8, float, double
uint1	int2, int4, int8, uint2, uint4, uint8, float, double
int2	int4, int8, float, double
uint2	int4, int8, uint4, uint8, float, double
int4	int8, float, double
uint4	int8, uint8, float, double
int8	float, double
uint8	float, double
float	double



Следующие преобразования могут привести к потере точности:

- › int4, uint4 → float
- › int8, uint8 → float, double



Неявного преобразования double → float нет.

Поэтому выражения следующего вида компилироваться не будут:

a: float = 5.4; //5.4 - вещественный литерал типа double

# Явное преобразование типов

Производится с помощью набора функций пространства имен `TypeConvert`:

- › `TypeConvert.ToBool`
- › `TypeConvert.ToInt1`
- › `TypeConvert.ToInt2`
- › `TypeConvert.ToInt4`
- › `TypeConvert.ToInt8`
- › `TypeConvert.ToUInt1`
- › `TypeConvert.ToUInt2`
- › `TypeConvert.ToUInt4`
- › `TypeConvert.ToUInt8`
- › `TypeConvert.ToFloat`
- › `TypeConvert.ToDouble`

## Примеры



```
//Явное преобразования типа с обрезкой
```

```
Num: int1 = TypeConvert.ToInt1(129); //Результат: 127 - максимальное  
значение для int1
```



# Синтаксис

## Алфавит

Astra.Om – регистрозависимый язык: переменные названные в разном регистре неэквивалентны друг другу.

Допустимые символы в тексте формул и процедур:

- › латинский алфавит (A-Z, a-z);
- › русский алфавит (А-Я, а-я);
- › цифры (0-9);
- › символы:

`	~	!	@	#	%	^	&
*	(	)	_	-	+	=	\
	{	}	[	]	:	;	"
'	<	>	/	?	.	,	\$

## Ключевые слова

Ключевые слова типов данных:

- › int1
- › int2
- › int4
- › int8
- › uint1
- › uint2
- › uint4
- › uint8
- › float
- › double
- › bool
- › string
- › variant

- › const
- › variant

## Логические значения

- › true
- › false

## Операторы управляющих структур:

- › if;
- › else;
- › break;
- › continue.

## Комментарии

Комментирование строки – символ //.



//Однострочный комментарий

Комментирование фрагмента: начало фрагмента /\*, конец фрагмента \*/.



/\*  
Комментирование  
нескольких  
строк  
\*/

# Литералы

Литералы используются в программном коде для обозначения числовых значений, строк, символов или логических значений. Другими словами литерал представляет собой постоянное значение, у которого нет имени.

## Логические литералы

Литералы означающие истинность или ложность выражения:

- › true
- › false

## Целочисленные литералы

- › 0b<bits> – представление числа в двоичном виде
- › 0x<hex> – представление числа



Число 255 в двоичной системе счисления: 0b11111111

Число 255 в шестнадцатеричной системе счисления: 0xFF

## Вещественные литералы

Вещественные литералы можно записать:

- › В десятичном представлении: X.Y, где X – целая часть числа, а Y – дробная.

Примеры: 2.4, 0.001, 1.0.

Целую часть можно не указывать, в этом случае она считается равной нулю: .1 эквивалентно записи 0.1.

- › В экспоненциальном представлении:  $M \times 10^{\pm P}$ , где M – мантиса, P – порядок.

Значение вычисляется, как  $M \times 10^{\pm P}$ .

Примеры: 1e-2 (равно 0.01), 1e+3 (равно 1000), 314159e-5 (равно 3.14159).

Тип вещественного литерала зависит от наличия суффикса:

- › без суффикса – тип double
- › с суффиксом f – тип float



a: double = 0.1; //Значение типа double  
b: float = 0.1f; //Значение типа float

## Строковые литералы

Строковые литералы заключаются в двойные кавычки и могут содержать любые печатные Unicode символы.

Спецсимволы:

- › \", "" – двойные кавычки
- › \n, \r – перевод строки
- › \t – символ табуляции
- › \\ – обратный слэш

# Операторы

Операторы	Описание
<a href="#">Присваивание</a>	Операция копирования значения и сопутствующей информации из источника в получатель
<a href="#">Арифметические операции</a>	Операции, выполняемые с аргументами числового типа
<a href="#">Логические операции</a>	Операции, выполняемые с аргументами логического типа bool
<a href="#">Битовые операции</a>	Операции, производимые над битами аргументов числового типа
<a href="#">Операции со строками</a>	Операции, производимые над строками
<a href="#">Операторы сравнения</a>	Операторы, позволяющие сравнить два аргумента
<a href="#">Циклы</a>	Управляющие конструкции, предназначенные для организации многократного исполнения набора инструкций
<a href="#">Условные операторы</a>	Операторы, позволяющие организовать ветвление кода в зависимости от того, выполняется ли условие

# Присваивание

Присваивание – операция копирования значения и сопутствующей информации из источника в получатель.



Dest = Source

Dest – получатель значения, Source – источник значения, вычисляемое выражение.

Операция выполняется успешно, если тип значения source приводит к типу значения dest, иначе – ошибка компиляции.

## Пример



```
res = X; //Присваивание
res += X; //Присваивание со сложением (res = res + X)
res -= X; //Присваивание с вычитанием (res = res - X)
res *= X; //Присваивание с умножением (res = res * X)
res /= X; //Присваивание с делением (res = res / X)
res %= X; //Присваивание с делением по модулю (res = res % X)
res <<= X; //Присваивание со сдвигом влево (res = res << X)
res >>= X; //Присваивание со сдвигом вправо (res = res >> X)
res |= X; //Присваивание с дизъюнкцией (res = res | X)
res &= X; //Присваивание с конъюнкцией (res = res & X)
res ^= X; //Присваивание с исключающей дизъюнкцией (res = res ^ X)
res = X cmp Y ? resIfTrue : resIfFalse; // Условное присваивание (cmp – это символ сравнения)
```

# Арифметические операции

Арифметические операции – операции, выполняемые с аргументами числового типа: int1, uint1, int2, uint2, int4, uint4, int8, uint8, float, double.

- › [Отрицание](#)
- › [Сложение](#)
- › [Вычитание](#)
- › [Умножение](#)
- › [Деление](#)
- › [Деление по модулю](#)

**Результат операции** – значение, имеющее тот же тип данных, что и аргументы. Если аргументы имеют разный тип данных, результат будет иметь тип данных наибольшей мощности из аргументов.



Операции отрицание и деление по модулю имеют ограничение на тип аргументов. Допустимые типы аргументов перечислены в описании этих операций.

# Отрицание

Унарная операция, меняет знак значения. Ставится перед константой, переменной или скобкой.



`-X`

Операция применима к следующим типам данных: `int1`, `int2`, `int4`, `int8`, `float`, `double`.

## Пример



```
X1: int1 = 10;
```

```
X2 = -X1; // переменная X2 будет содержать в себе значение -10
```



# Сложение

Позволяет выполнять арифметическое сложение двух аргументов.



X + Y

## Пример



```
X: int1 = 10;
```

```
Y: float = 5.5f;
```

```
Z: var = X + Y; // переменная Z будет содержать значение 15.5 (тип данных float)
```

# Вычитание

Позволяет выполнять операцию вычитания, результатом которой является новое число, получаемое уменьшением значения первого аргумента на значение второго аргумента.



$X - Y$

## Пример



```
X: int1 = 10;
```

```
Y: float = 5.5f;
```

```
Z: var = X - Y; // переменная Z будет содержать значение 4.5 (тип данных float)
```

# Умножение

Позволяет выполнять арифметическое умножение двух аргументов.



$X * Y$

## Пример



```
X: int1 = 10;
```

```
Y: float = 5.5f;
```

```
Z: var = X * Y; // переменная Z будет содержать значение 55 (тип данных float)
```

# Деление

Позволяет выполнять арифметическое деление двух аргументов.



X / Y

## Пример



```
X: int1 = 10;
```

```
Y: float = 2.5f;
```

```
Z: var = X / Y; // переменная Z будет содержать значение 4 (тип данных float)
```

# Деление по модулю

Арифметическая операция, результатом которой является остаток от деления целого числа на другое целое число.



$X \% Y$

Операция применима только к целым типам данных: int1, int2, int4, int8, uint1, uint2, uint4, uint8.

## Пример



```
X: int1 = 10;
```

```
Y: int1 = 3;
```

```
Z: int1 = X % Y; // переменная Z будет содержать значение 1
```

# Логические операции

**Логические операции** – операции, выполняемые с аргументами логического типа bool.

Результат операции: true/false.



Чтобы выполнить логическую операцию с аргументами других типов, необходимо выполнить явное приведение этих аргументов к типу bool, иначе произойдет ошибка компиляции.

В Astra.Оm доступны следующие логические операции:

- › [Отрицание \(инверсия\)](#)
- › [Сложение \(дизъюнкция\)](#)
- › [Умножение \(конъюнкция\)](#)

# Отрицание (инверсия)

Унарная операция. Возвращает true, если аргумент имеет значение false, иначе возвращает false.



!X

## Пример



```
X: bool = true;
```

```
Y = !X; // переменная Y будет содержать в себе значение false
```

# Сложение (дизъюнкция)

Логическое сложение (дизъюнкция) – это бинарная операция, результат которой равен false, если оба аргумента равны false, иначе – true.



$X \ || \ Y$

## Пример



```
Z1: bool = true || true; // Результат – true  
Z2: bool = true || false; // Результат – true  
Z2: bool = false || false; // Результат – true  
Z3: bool = false || false; // Результат – false
```



# Умножение (конъюнкция)

Бинарная операция. Возвращает true, если оба аргумента имеют значение true, иначе возвращает false.



X && Y

## Пример



```
Z1: bool = true && true; // Результат – false  
Z2: bool = true && false; // Результат – false  
Z3: bool = false && false; // Результат – false  
Z4: bool = false && true; // Результат – false
```

# Битовые операции

**Битовые операции** – операции, производимые над битами аргументов числового типа (uint1, uint2, uint4, uint8).

В Astra.Оm доступны следующие битовые операции:

- › [Отрицание \(инверсия\)](#)
- › [Сложение \(дизъюнкция\)](#)
- › [Умножение \(конъюнкция\)](#)
- › [Исключающее сложение](#)
- › [Сдвиг влево](#)
- › [Сдвиг право](#)



У некоторых операций есть ограничение на тип аргументов.

# Отрицание (инверсия)

Побитовое отрицание (инверсия) – унарная операция, изменяющая значение каждого бита аргумента на противоположное: 0 заменяется на 1, 1 – на 0.



$\sim X$

## Пример



X: uint2 = 202; // двоичное значение - 11001010

Y =  $\sim X$ ; // Результат – 53, двоичное значение - 00110101

# Сложение (дизъюнкция)

Побитовое сложение (дизъюнкция) – бинарная операция, которая побитово сравнивает каждый бит первого аргумента со вторым: в двоичное значение операции запишется 0, если оба бита аргументов равны 0, иначе – 1.



X | Y

# Умножение (конъюнкция)

Бинарная операция. Каждый бит первого аргумента логически умножается на соответствующий бит второго аргумента. Применяется для типов uint1, uint2, uint4, uint8.



X & Y

# Исключающее сложение

Бинарная операция. Каждый бит первого аргумента логически складывается с исключением с соответствующим битом второго аргумента. Исключающее сложение – результат равен 0 (false), если оба аргумента равны, иначе – 1 (true). Применяется для типов uint1, uint2, uint4, uint8.



$X \wedge Y$

# Сдвиг влево

Бинарная операция. Результат равен сдвигу влево первого аргумента на количество бит, указанных во втором аргументе.



$X \ll Y$

$X \text{ shl } Y$

Первый аргумент может быть типов – uint1, uint2, uint4, uint8.

Второй аргумент – uint1.

# Сдвиг вправо

Бинарная операция. Результат равен сдвигу право первого аргумента на количество бит, указанных во втором аргументе.



$X \gg Y$

$X \text{ shr } Y$

Первый аргумент может быть типов – uint1, uint2, uint4, uint8.

Второй аргумент – uint1.



# Операции со строками

Аргументами в операциях со строками могут являться только строки: тип `string`.

В Astra.Оm доступны следующие операции со строками:

- › [конкатенация](#)
- › [операции сравнения](#)



Сравнение строк выполняется с учетом регистра

## Конкатенация

Обозначается знаком `+`. Результат выполнения операции – строка, являющаяся объединением строк-аргументов.



`X + Y`

## Пример



```
s1: string = "Con";  
s2: string = "cat";  
s3: string = s1 + s2; // s3 = "Concat"
```

# Операторы сравнения

Бинарные операции. Применяется для всех типов. Возвращают true или false.

## Равно



$X == Y$

## Не равно



$X != Y$

## Больше



$X > Y$

## Меньше



$X < Y$

## Больше или равно



$X >= Y$

## Меньше или равно



$X \leq Y$

# Циклы

**Циклы** – управляющие конструкции, предназначенные для организации многократного исполнения набора инструкций до тех пор, пока удовлетворяется определенное условие.

## Цикл while

Цикл с предварительной проверкой. В отличие от for, цикл while удобно использовать для заранее неизвестного количества итераций.



```
while (condition)
{
    //statement
}
```

Condition – выражение-условие продолжения цикла. Результат выражения должен быть приводим к типу bool. Тело цикла исполняется до тех пор, пока condition равен true. Внутри цикла могут использоваться операторы break (прерывание цикла) и continue (переход на следующую итерацию цикла).

## Пример



```
//Цикл будет выполняться пока переменная Quality больше или  
равна 0  
while(Quality >= 192)
{
    //statement
}
```

## Цикл for

Цикл `for` обычно используется для выполнения определенного количества итераций.



```
for (init; condition; increment)
{
    //statement
}
```

`init` – действие до начала цикла. В большинстве случаев – объявление переменной (счетчика цикла) и присваивание ей начального значения. Объявлять переменную необязательно: присваивать начальное значение можно переменной, объявленной выше.

`condition` – условие продолжения цикла. В большинстве случаев – сравнение счетчика цикла с финальным значением.

`increment` – действие в конце каждой итерации цикла. В большинстве случаев – увеличение счетчика цикла.

Истинность условия проверяется перед каждой итерацией, в том числе перед первой. Если условие не выполняется (возвращает `false`) – цикл прекращается.

Внутри цикла можно использовать операторы `break` (выйти из цикла) и `continue` (перейти на следующую итерацию цикла). Любые части `init`, `condition` или `increment` в описании цикла можно опустить.

## Пример



```
//Сумма всех чисел от 1 до 1000
sum: int4 = 0;
for (i: int4 = 1; i <= 1000; i += 1)
{
    sum += i;
}
```

Внутри цикла можно использовать операторы `break` (выйти из цикла) и `continue` (перейти на следующую итерацию цикла).

Любые части `init`, `condition` или `increment` в описании цикла можно опустить.



Если не указано условие выхода из цикла, оно принимается равным `true`. В этом случае для выхода из цикла необходимо использовать оператор `break`.

## Пример



```
//Поиск первого числа после 1000, кратного 137
value: int4 = 1000;
for (;;) value += 1
    if (value % 137 == 0)
        break;
// после завершения работы цикла: value = 1096
//init – начальное значение присвоено до цикла.
//condition – условие выхода описано в теле цикла.
```

# Условные операторы

Условный оператор `if` позволяет организовать ветвление кода в зависимости от того, выполняется ли условие.



```
if (condition)
{
    //statements
}
else
{
    //statements
}
```

`condition` – выражение-условие, вычисляемое в булево значение. Если выражение-условие истинно, то выполняется блок инструкций ветки `if`, иначе выполняется блок инструкций ветки `else`. Инструкции `if` могут быть вложены в другие инструкции `if`.

## Пример



```
if (Quality >= 192)
{
    // ветка обработки хорошего качества
    if (condition)
    {
        //statements
    }
    else
    {
        //statements
    }
}
```

```
else
{
// ветка обработки плохого качества
}
```

Стандартная конструкция if может быть заменена более компактной тернарной операцией.



```
<condition>?<statement1>:<statement2>
```

Сначала вычисляется логическое выражение <condition>. Если оно истинно, то вычисляется значение <statement1>, в противном случае - значение <statement2>.

## Пример



```
MyString: string;
MyString = (Quality >= 192)?("Хорошее качество"):("Плохое
качество");
```



# Формат даты/времени

В некоторых функциях обработки времени можно указать формат, согласно которому будет формироваться или разбираться строка, описывающая дату/время. Формат представляет собой строку, в которой с помощью флагов форматирования описано расположение компонентов даты/времени в строке.

## Флаги форматирования

Флаг	Описание	Пример
%a	Сокращенное название дня недели	"пн" → понедельник
%A	Название дня недели	"понедельник" → понедельник
%b	Сокращенное название месяца	"янв" → январь
%B	Название месяца	"январь" → январь
%c	Принятое представление даты/времени в текущей локали	
%C	Первые две цифры в записи года	2015 → "20"
%d	Дата месяца от 01 до 31	"01" → 1 число "15" → 15 число
%D	Аналогичен %m%d%y	
%e	Аналогичен %d, но ведущий ноль заменяется на пробел	"1" → 1 число "20" → 20 число
%f	Дробные доли секунды, отображаются даже если равны нулю	"12:30:15.000000"
%F	Дробные доли секунды, отображаются только если не равны нулю	"12:30:15" "01:02:03.012345"
%h	Аналогичен %b	
%H	Час от 00 до 23	

%l	Час от 01 до 12	"20" → 8
%j	Номер дня в году от 001 до 365 (всегда предполагается, что год високосный)	"060" → 29 февраля
%k	Час от 0 до 23	
%l	Час от 1 до 12	
%m	Номер месяца от 01 до 31	"01" → январь
%M	Минуты от 00 до 59	
%s	Секунды, включая дробные доли секунд	"59.000000"
%S	Секунды	"59"
%T	Время в 24-часовой нотации. Эквивалентно %H:%M:%S	"12:30:00"
%u	Номер дня недели от 1 до 7, где 1 – понедельник	"2" → вторник
%U	Номер недели в году от 00 до 53. Первое воскресенье в году считается первым днем недели 01. Если год начинается не с воскресенья, дни до первого воскресенья считаются неделями 00	
%V	Номер недели в формате ISO 8601:1988, значение в диапазоне от 01 до 53. Неделей 01 считается первая неделя января, в которой минимум 4 дня относятся к этому году; если меньше – первой считается следующая за ней неделя	
%w	Номер дня недели от 0 до 6, где 0 – воскресенье	"1" → понедельник
%W	Номер недели в году от 00 до 53, где понедельник – это первый день недели 01	
%x	Формат даты, принятый для текущей локали	en_us → "10/31/2005" ru_ru → "31.10.2005"

%y	Две цифры года	"01" → 2001
%Y	Четыре цифры года	"2001" → 2001

## Примеры



%Y-%B-%d → 2005-апрель-01

%Y%m%d → 20050401

%Y-%m-%d %H:%M:%S%F → 2005-01-02 12:30:00

# Функции

Функция	Описание
<a href="#">Функции обработки времени</a>	Функции с меткой времени, пространство имен DateTime
<a href="#">Математические функции</a>	Функции с основными математическими операциями, пространство имен Math
<a href="#">Строковые функции</a>	Функции со строками, пространство имен String
<a href="#">Функции типа variant</a>	Функции с данными типа variant, пространство имен Variant

# Функции обработки времени



Функций обработки времени вызываются обращением к пространству имен DateTime.

Функции обработки времени получают в качестве аргумента или возвращают метку времени – значение типа uint8, представляющее собой количество 100-наносекундных интервалов прошедших с 1 января 1601 года.

Функция	Описание
<a href="#">AddDays</a>	Возвращает метку времени, смещенную на указанное количество дней
<a href="#">AddHours</a>	Возвращает метку времени, смещенную на указанное количество часов
<a href="#">AddMinutes</a>	Возвращает метку времени, смещенную на указанное количество минут
<a href="#">AddMonths</a>	Возвращает метку времени, смещенную на указанное количество месяцев
<a href="#">AddMSeconds</a>	Возвращает метку времени, смещенную на указанное количество миллисекунд
<a href="#">AddSeconds</a>	Возвращает метку времени, смещенную на указанное количество секунд
<a href="#">AddYears</a>	Возвращает метку времени, смещенную на указанное количество лет
<a href="#">Create</a>	Возвращает метку времени, полученную из набора указанных значений
<a href="#">Day</a>	Для указанной метки времени возвращает число дня
<a href="#">DayOfWeek</a>	Для указанной метки времени возвращает день недели
<a href="#">DaysInMonth</a>	Возвращает количество дней в указанном месяце указанного года
<a href="#">Hour</a>	Для указанной метки времени возвращает час

<a href="#">IsLeapYear</a>	Проверяет, является ли указанный год високосным
<a href="#">Minute</a>	Для указанной метки времени возвращает минуты
<a href="#">Month</a>	Для указанной метки времени возвращает месяц
<a href="#">MSecond</a>	Для указанной метки времени возвращает миллисекунды
<a href="#">Now</a>	Возвращает текущее время компьютера
<a href="#">Parse</a>	Возвращает метку времени, полученную из указанной строки
<a href="#">Second</a>	Для указанной метки времени возвращает секунды
<a href="#">ToLocal</a>	Для указанной метки времени со временем по UTC возвращает метку времени с соответствующим временем в текущем часовом поясе
<a href="#">ToString</a>	Для указанной метки времени возвращает дату-время в виде строки
<a href="#">ToUtc</a>	Для указанной метки времени в текущем часовом поясе возвращает метку времени с соответствующим временем по UTC
<a href="#">UtcNow</a>	Возвращает текущее время по UTC
<a href="#">Year</a>	Для указанной метки времени возвращает год

# AddDays

Возвращает метку времени, смещенную на указанное количество дней.



```
uint8 DateTime.AddDays(uint8 timestamp, double days)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
days	double	Количество дней, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени). Можно указать дробное смещение

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество дней.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddDays(timestamp, 2); // Результат: метка  
времени, соответствующая "08.09.1993 12:00:00"  
b: uint8 = DateTime.AddDays(timestamp, -15); // Результат: метка  
времени, соответствующая "22.08.1993 12:00:00"  
c: uint8 = DateTime.AddDays(timestamp, 0.25); // Результат: метка  
времени, соответствующая "06.09.1993 18:00:00"
```

# AddHours

Возвращает метку времени, смещенную на указанное количество часов.



```
uint8 DateTime.AddHours(uint8 timestamp, double hours)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
hours	double	Количество часов, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени). Можно указать дробное смещение

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество часов.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddHours(timestamp, 2); // Результат: метка  
времени, соответствующая "06.09.1993 14:00:00"  
b: uint8 = DateTime.AddHours(timestamp, -15); // Результат: метка  
времени, соответствующая "05.09.1993 21:00:00"  
c: uint8 = DateTime.AddHours(timestamp, 1.5); // Результат: метка  
времени, соответствующая "06.09.1993 13:30:00"
```



# AddMinutes

Возвращает метку времени, смещенную на указанное количество минут.



```
uint8 DateTime.AddMinutes(uint8 timestamp, double minutes)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
minutes	double	Количество минут, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени). Можно указать дробное смещение

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество минут.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddMinutes(timestamp, 30); // Результат: метка  
времени, соответствующая "06.09.1993 12:30:00"  
b: uint8 = DateTime.AddMinutes(timestamp, -90); // Результат: метка  
времени, соответствующая "06.09.1993 10:30:00"  
c: uint8 = DateTime.AddMinutes(timestamp, 1.5); // Результат: метка  
времени, соответствующая "06.09.1993 12:01:30"
```

# AddMonths

Возвращает метку времени, смещенную на указанное количество месяцев.



```
uint8 DateTime.AddMonths(uint8 timestamp, int4 months)
```



Если при смещении получилась недопустимая дата (например, 30 февраля), то в качестве даты будет взят последний день полученного месяца; остальные поля метки времени (часы, минуты и т.д.) останутся, как у исходной метки времени.

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
minutes	int4	Количество месяцев, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени)

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество месяцев.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddMonths(timestamp, 2); // Результат: метка  
времени, соответствующая "06.11.1993 12:00:00"
```

```
b: uint8 = DateTime.AddMonths(timestamp, 6); // Результат: метка  
времени, соответствующая "06.03.1994 12:00:00"  
c: uint8 = DateTime.AddMonths(timestamp, -3); // Результат: метка  
времени, соответствующая "06.06.1993 12:00:00"
```

# AddMSeconds

Возвращает метку времени, смещенную на указанное количество миллисекунд.



```
uint8 DateTime.AddMSeconds(uint8 timestamp, double mseconds)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
mseconds	double	Количество миллисекунд, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени). Можно указать дробное смещение

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество миллисекунд.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddMSeconds(timestamp, 500); // Результат: метка  
времени, соответствующая "06.09.1993 12:00:00.500"  
b: uint8 = DateTime.AddMSeconds(timestamp, -2500); // Результат:  
метка времени, соответствующая "06.09.1993 11:59:57.500"
```

```
c: uint8 = DateTime.AddMSeconds(timestamp, 1000); // Результат: метка  
времени, соответствующая "06.06.1993 12:00:01.000"
```

# AddSeconds

Возвращает метку времени, смещенную на указанное количество секунд.



```
uint8 DateTime.AddSeconds(uint8 timestamp, double seconds)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
seconds	double	Количество секунд, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени). Можно указать дробное смещение

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество секунд.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddSeconds(timestamp, 30); // Результат: метка  
времени, соответствующая "06.09.1993 12:00:30"  
b: uint8 = DateTime.AddSeconds(timestamp, -90); // Результат: метка  
времени, соответствующая "06.09.1993 11:58:30"  
c: uint8 = DateTime.AddSeconds(timestamp, 50.5); // Результат: метка  
времени, соответствующая "06.09.1993 12:00:50.500"
```

# AddYears

Возвращает метку времени, смещенную на указанное количество лет.



```
uint8 DateTime.AddYears(uint8 timestamp, int4 years)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени, от которой смещаться
years	int4	Количество лет, на которое надо сместиться. Значение может быть отрицательным (смещение назад во времени)

## Возвращаемое значение

Тип – uint8.

Метка времени, смещенная относительно timestamp на указанное количество лет.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
a: uint8 = DateTime.AddYears(timestamp, 7); // Результат: метка  
времени, соответствующая "06.09.2000 12:00:00"  
b: uint8 = DateTime.AddYears(timestamp, -13); // Результат: метка  
времени, соответствующая "06.09.1980 12:00:00"  
c: uint8 = DateTime.AddYears(timestamp, 13); // Результат: метка  
времени, соответствующая "06.09.1980 12:00:00"
```

# Create

Возвращает метку времени, полученную из набора указанных значений.



```
uint8 DateTime.Create(uint2 year, uint1 month, uint1 day, uint1 h, uint1 m, uint1 s)
```

```
uint8 DateTime.Create(uint2 year, uint1 month, uint1 day, uint1 h, uint1 m, uint1 s, uint2 ms)
```

## Параметры

Параметр	Тип	Описание
year	uint2	Год. $1601 \leq \text{year}$
month	uint1	Месяц. $1 \leq \text{month} \leq 12$
day	uint1	Число месяца. $1 \leq \text{day} \leq \text{кол-во дней в месяце}$
h	uint1	Час. $0 \leq h \leq 23$
m	uint1	Минуты. $0 \leq m \leq 59$
s	uint1	Секунды. $0 \leq s \leq 59$
ms	uint2	Опциональный параметр Миллисекунды. $0 \leq \text{ms} \leq 999$ Если не указан, считается равным нулю

## Возвращаемое значение

Тип – uint8.

Метка времени, составленная из набора указанных значений года, месяца, даты, часа, минут, секунд и миллисекунд.





Если какое-то значение принимает некорректное значение, возвращается нулевая метка времени – 1 января 1601 года, 0:00:00.000.

## Примеры



a: uint8 = DateTime.Create(2000, 1, 2, 3, 4, 5); // Результат: метка времени 2 января 2000 года, 3:04:05

b: uint8 = DateTime.Create(1993, 9, 6, 12, 5, 15, 301); // Результат: метка времени 6 сентября 1993 года, 12:05:15.301

c: uint8 = DateTime.Create(1999, 2, 29, 12, 0, 0, 0); // Результат: 1 января 1601 года, 0:00:00.000, т.к. в феврале 1999 года 28 дней

# Day

Для указанной метки времени возвращает число дня.



```
uint1 DateTime.Day(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint1.

Число месяца метки времени timestamp.  $1 \leq \text{значение} \leq 31$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
Result: uint1 = DateTime.Day(timestamp); // Результат: 6
```

# DayOfWeek

Для указанной метки времени возвращает день недели.

Для указанной метки времени возвращает день недели (0-6). 0 – воскресенье, 1 – понедельник и т.д.



```
uint1 DateTime.DayOfWeek(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint1.

День недели метки времени timestamp.  $0 \leq \text{значение} \leq 6$ : 0 – воскресенье, 1 – понедельник и т.д.

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:25:31");  
Result: uint1 = DateTime.DayOfWeek(timestamp); // Результат: 1  
(понедельник)
```

# DaysInMonth

Возвращает количество дней в указанном месяце указанного года.



```
uint1 DateTime.DayInMonth(uint2 year, uint1 month)
```

## Параметры

Параметр	Тип	Описание
year	uint2	Год
month	uint1	Месяц

## Возвращаемое значение

Тип – uint1.

Количество дней в указанном месяце указанного года:

- для февраля – количество дней зависит от того, является ли год високосным.
- для остальных месяцев – количество дней не зависит от года.

## Примеры



```
Result: uint1 = DateTime.DaysInMonth(1995, 5); // Результат: 31
```

# Hour

Для указанной метки времени возвращает час.



```
uint1 DateTime.Hour(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint1.

Час метки времени timestamp.  $0 \leq \text{значение} \leq 23$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
Result: uint1 = DateTime.Hour(timestamp); // Результат: 12
```

# IsLeapYear

Проверяет, является ли указанный год високосным.



```
bool DateTime.IsLeapYear(uint2 year)
```

## Параметры

Параметр	Тип	Описание
year	uint2	Год

## Возвращаемое значение

Тип – bool.

True – если год високосный, иначе – false

## Примеры



```
Result: uint1 = DateTime.DaysInMonth(1992); // Результат: true
```

# Minute

Для указанной метки времени возвращает минуты.



```
uint1 DateTime.Minute(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint1.

Минуты метки времени timestamp.  $0 \leq \text{значение} \leq 59$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:25:00");  
Result: uint1 = DateTime.Minute(timestamp); // Результат: 25
```



# Month

Для указанной метки времени возвращает месяц.



```
uint1 DateTime.Month(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint1.

Месяц метки времени timestamp.  $1 \leq \text{значение} \leq 12$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
Result: uint1 = DateTime.Month(timestamp); // Результат: 9
```

# MSecond

Для указанной метки времени возвращает миллисекунды.



```
uint2 DateTime.MSecond(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint2.

Миллисекунды метки времени timestamp.  $0 \leq \text{значение} \leq 999$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:25:31.155");  
Result: uint2 = DateTime.MSecond(timestamp); // Результат: 155
```

# Now

Возвращает текущее время компьютера.



uint8 DateTime.Now()

## Возвращаемое значение

Тип – uint8.

Метка времени с текущим временем.

## Примеры



```
DateTime.Now(); // результат метка времени с текущим временем на компьютере
```

# Parse

Возвращает метку времени, полученную из указанной строки.



```
uint8 DateTime.Parse(string date_time)
```

```
uint8 DateTime.Parse(string date_time, string format)
```

## Параметры

Параметр	Тип	Описание
date_time	string	Строка, описывающая время
format	string	Опциональный параметр Формат даты/времени, согласно которому будет разбираться строка

## Возвращаемое значение

Тип – uint8.

Метка времени, соответствующая описанным в строке дате/времени. Если в строке указана только дата, возвращает полночь указанной даты: 00:00:00. Если строку не удастся разобрать, возвращает нулевую метку времени: 1 января 1601 года, 0:00:00.000.

## Примеры



```
a: uint8 = DateTime.Parse("06.09.1993 12:01:01");
```

```
// Результат: метка времени, соответствующая указанному в строке  
моменту времени.
```

```
b: uint8 = DateTime.Parse("12:01:01 06.09.1993", "%H:%M:S %d.%m.  
%Y");
```

```
// Результат: метка времени, разобранный согласно указанному  
формату  
с: uint8 = DateTime.Parse("03.11.1999 12:01:01");  
// Результат: метка времени, соответствующая указанному в строке  
моменту времени.
```

# Second

Для указанной метки времени возвращает секунды.



```
uint1 DateTime.Second(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint1.

Секунды метки времени timestamp.  $0 \leq \text{значение} \leq 59$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:25:31");  
Result: uint1 = DateTime.Second(timestamp); // Результат: 31
```

# ToLocal

Для указанной метки времени со временем по UTC возвращает метку времени с соответствующим временем в текущем часовом поясе.



```
uint8 DateTime.ToLocal(uint8 utc_time)
```

## Параметры

Параметр	Тип	Описание
utc_time	uint8	Метка времени по UTC

## Возвращаемое значение

Тип – uint8.

Метка времени в текущем часовом поясе, соответствующая времени utc\_time по UTC.

## Примеры



```
DateTime.ToLocal(DateTime.UtcNow); // результат метка времени в  
текущем часовом поясе соответствующая времени utc_time
```

# ToString

Для указанной метки времени возвращает дату-время в виде строки.



```
string DateTime.ToString(uint8 timestamp)
string DateTime.ToString(uint8 timestamp, string format)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени
format	string	Опциональный параметр Формат даты/времени, согласно которому будет формироваться строка

## Возвращаемое значение

Тип – string.

Строка, описывающая дату-время, в указанном формате или, если формат не указан, в формате %d.%m.%Y %H:%M:%S

## Примеры



```
timestamp: uint8 = DateTime.Create(2000, 1, 2, 3, 4, 5, 0);
a: string = DateTime.ToString(timestamp); // Результат: "02.01.2000
03:04:05"
b: string = DateTime.ToString(timestamp, "%d/%m/%Y-%H:%M:%S.
%f"); // Результат: "02/01/2000-03:04:05.000000" (строка в указанном
формате)
```



# ToUtc

Для указанной метки времени в текущем часовом поясе возвращает метку времени с соответствующим временем по UTC.



```
uint8 DateTime.ToUtc(uint8 local)
```

## Параметры

Параметр	Тип	Описание
local	uint8	Метка времени

## Возвращаемое значение

Тип – uint8.

Метка времени по UTC, соответствующая времени local в текущем часовом поясе.

## Примеры



```
DateTime.ToUtc(DateTime.Now()); // Результат: метка времени по UTC
```

# UtcNow

Возвращает текущее время по UTC.



uint8 DateTime.UtcNow()

## Возвращаемое значение

Тип – uint8.

Метка времени с текущим временем по UTC.

## Примеры



```
DateTime.UtcNow(); // результат метка времени с текущим временем по UTC
```

# Year

Для указанной метки времени возвращает год.



```
uint2 DateTime.Year(uint8 timestamp)
```

## Параметры

Параметр	Тип	Описание
timestamp	uint8	Метка времени

## Возвращаемое значение

Тип – uint2.

Год метки времени timestamp. Значение  $\geq 1601$ .

## Примеры



```
timestamp: uint8 = DateTime.Parse("06.09.1993 12:00:00");  
Result: uint2 = DateTime.Year(timestamp); // Результат: 1993
```

# Математические функции



Вызов математических функций осуществляется обращением к пространству имен `Math`.



Если хотя бы один из параметров равен `NaN`, все функции возвращают `NaN`, если в описании функции не указано иное.

Функция	Описание
<a href="#">Abs</a>	Возвращает модуль числа
<a href="#">Acos</a>	Возвращает арккосинус указанного числа
<a href="#">Acosh</a>	Возвращает обратный гиперболический косинус указанного числа
<a href="#">Asin</a>	Возвращает арксинус указанного числа
<a href="#">Asinh</a>	Возвращает обратный гиперболический синус указанного числа
<a href="#">Atan</a>	Возвращает арктангенс указанного числа
<a href="#">Atan2</a>	Возвращает арктангенс отношения $y/x$
<a href="#">Atanh</a>	Возвращает обратный гиперболический тангенс указанного числа
<a href="#">Cbrt</a>	Возвращает кубический корень указанного числа
<a href="#">Ceil</a>	Возвращает число, округленное до большего целого
<a href="#">Clamp</a>	Возвращает указанное число, ограниченное диапазоном <code>[min; max]</code>
<a href="#">ClearBit</a>	Возвращает число, у которого обнулен указанный бит
<a href="#">CopySign</a>	Возвращает число с величиной $x$ и знаком числа $y$
<a href="#">Cos</a>	Возвращает косинус угла
<a href="#">Cosh</a>	Возвращает гиперболический косинус числа

<a href="#">Dim</a>	Возвращает положительную разность двух чисел
<a href="#">e</a>	Возвращает константу e
<a href="#">Erf</a>	Возвращает значение функции ошибок
<a href="#">Erfc</a>	Возвращает значение дополнительной функции ошибок
<a href="#">Exp</a>	Возвращает число e, возведенное в указанную степень
<a href="#">Exp2</a>	Возвращает число 2, возведенное в указанную степень
<a href="#">ExpM1</a>	Возвращает модуль числа
<a href="#">Floor</a>	Возвращает число, округленное вниз
<a href="#">FusedMultiplyAdd</a>	Возвращает умножение-сложение с однократным округлением: $(x*y)+z$
<a href="#">Gamma</a>	Возвращает значение гамма-функции для указанного числа
<a href="#">Hypot</a>	Для указанных длин катетов возвращает длину гипотенузы
<a href="#">ILogB</a>	Возвращает порядок числа с плавающей запятой в виде знакового целого числа
<a href="#">IsInf</a>	Проверяет, является ли число бесконечностью
<a href="#">IsNaN</a>	Проверяет, является ли число значением NaN
<a href="#">LGamma</a>	Возвращает натуральный логарифм абсолютного значения гамма-функции
<a href="#">Log</a>	Возвращает натуральный логарифм указанного числа
<a href="#">Log1p</a>	Возвращает натуральный логарифм числа $x+1$
<a href="#">Log2</a>	Возвращает двоичный логарифм числа
<a href="#">Log10</a>	Возвращает десятичный логарифм числа
<a href="#">LogB</a>	Возвращает порядок числа с плавающей запятой
<a href="#">Max</a>	Возвращает большее из двух чисел
<a href="#">Min</a>	Возвращает меньшее из двух чисел
<a href="#">NextAfter</a>	Для указанного вещественного числа возвращает ближайшее соседнее число, представимое в рамках типа
<a href="#">pi</a>	Возвращает число Пи

<a href="#">Pow</a>	Возвращает число, возведенное в указанную степень
<a href="#">Remainder</a>	Возвращает вещественный остаток от деления согласно стандарту IEEE-754
<a href="#">Round</a>	Возвращает число, округленное до ближайшего целого или до указанного количества знаков после запятой
<a href="#">ScaleB</a>	Возвращает произведение числа на 2 в указанной степени: $x * 2^n$
<a href="#">SetBit</a>	Возвращает число, у которого значение указанного бита установлено в true или изменено на указанное значение
<a href="#">SignBit</a>	Проверяет, является ли число отрицательным
<a href="#">Sin</a>	Возвращает синус угла
<a href="#">Sinh</a>	Возвращает гиперболический синус числа
<a href="#">Sqr</a>	Возвращает квадрат числа
<a href="#">Sqrt</a>	Возвращает квадратный корень числа
<a href="#">Tan</a>	Возвращает тангенс угла
<a href="#">Tanh</a>	Возвращает гиперболический тангенс числа
<a href="#">TestBit</a>	Для числа возвращает значение указанного бита
<a href="#">ToggleBit</a>	Возвращает число, у которого значение указанного бита изменено на противоположное
<a href="#">Truncate</a>	Возвращает число, у которого отброшена дробная часть

# Abs

Возвращает модуль числа.



```
int1 Math.Abs(int1 value)
int2 Math.Abs(int2 value)
int4 Math.Abs(int4 value)
int8 Math.Abs(int8 value)
float Math.Abs(float value)
double Math.Abs(double value)
```

## Параметры

Параметр	Тип	Описание
value	int1, int2, int4, int8, float, double	Число, для которого надо вычислить модуль

## Возвращаемое значение

Тип – такой же, как у параметра value.

Модуль числа value.

## Примеры



```
Result: uint1 = Math.Abs(-120); // Результат: 120
```

# Аcos

Возвращает арккосинус указанного числа.



float Math.Acos(float8 X)  
double Math.Acos(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Число, являющееся косинусом угла. $-1 \leq x \leq 1$

## Возвращаемое значение

Тип – такой же, как у параметра.

Арккосинус x: угол в радианах, косинус которого равен x.

Значение:

› если  $-1 \leq x \leq 1$ :

$0 \leq \text{результат} \leq \pi$

› иначе: результат:

NaN

## Примеры



```
a: double = Math.Acos(1); // Результат: 0
```



# Acosh

Возвращает обратный гиперболический косинус указанного числа.



```
float Math.Acosh(float8 X)  
double Math.Acosh(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число являющееся гиперболическим косинусом угла. $1 \leq x$

## Возвращаемое значение

Тип – такой же, как у параметра.

Обратный гиперболический косинус x: угол в радианах, гиперболический косинус которого равен x.

Значение:

› если  $1 \leq x$ :

$0 \leq$  результат

› иначе:

результат: NaN

## Примеры



```
a: double = Math.Acosh(1); // Результат: 0
```

# Asin

Возвращает арксинус указанного числа.



```
float Math.Asin(float x)
double Math.Asin(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, являющееся синусом угла. $-1 \leq x \leq 1$

## Возвращаемое значение

Тип – такой же, как у параметра.

Арксинус x: угол в радианах, синус которого равен x.

Значение:

› если  $-1 \leq x \leq 1$ :

$-\pi/2 \leq \text{результат} \leq \pi/2$

› иначе: результат:

NaN

## Примеры



```
a: double = Math.Asin(0); // Результат: 0
```

# Asinh

Возвращает обратный гиперболический синус указанного числа.



```
float Math.Asinh(float x)  
double Math.Asinh(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, являющееся гиперболическим синусом угла.

## Возвращаемое значение

Тип – такой же, как у параметра.

Обратный гиперболический синус x: угол в радианах, гиперболический синус которого равен x.

## Примеры



```
a: double = Math.Asinh(0); // Результат: 0
```

# Atan

Возвращает арктангенс указанного числа.



```
float Math.Atan(float x)
double Math.Atan(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, являющееся тангенсом угла.

## Возвращаемое значение

Тип – такой же, как у параметра.

Арктангенс  $x$  – угол в радианах, тангенс которого равен  $x$ .

Значение:

- › если  $-\infty < x < +\infty$ :  
 $-\pi/2 < \text{результат} < \pi/2$
- › если  $x = -\infty$ :  
результат:  $-\pi/2$
- › если  $x = +\infty$ :  
результат:  $\pi/2$

## Примеры



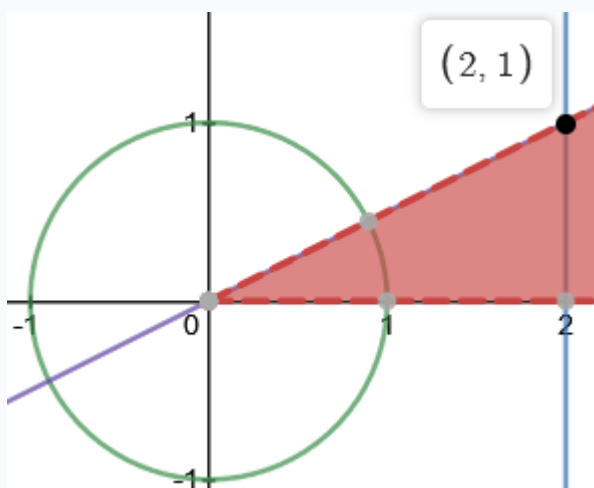
```
a: double = Math.Atan(0); // Результат: 0
```

# Atan2

Возвращает арктангенс отношения  $y/x$ .



Геометрический смысл: функция возвращает угол между осью абсцисс и вектором, ведущим из центра координат в точку  $(x, y)$ .



float Math.Atan2(float y, float x)  
double Math.Atan2(double y, double x)

## Параметры

Параметр	Тип	Описание
y	float, double	Ордината точки
x	float, double	Абсцисса точки, $x \neq 0$

## Возвращаемое значение

Тип – такой же, как у параметров.

Значение:

- > если  $x, y \neq 0, x, y \neq \pm\infty$ :  
 результат – арктангенс отношения  $y/x$ : угол в радианах, тангенс которого равен  $y/x$ .  
 $-\pi \leq \text{результат} \leq \pi$
- > если  $y = \pm 0, x < 0$  или  $x = -0$ :  
 результат:  $\pm\pi$
- > если  $y = \pm 0, x > 0$  или  $x = +0$ :  
 результат:  $\pm 0$
- > если  $y = \pm\infty, -\infty < x < +\infty$ :  
 результат:  $\pm\pi/2$
- > если  $y = \pm\infty, x = -\infty$ :  
 результат:  $\pm 3\pi/4$
- > если  $y = \pm\infty, x = +\infty$ :  
 результат:  $\pm\pi/4$
- > если  $x = \pm 0, y < 0$ :  
 результат:  $-\pi/2$
- > если  $x = \pm 0, y > 0$ :  
 результат:  $\pi/2$
- > если  $x = -\infty, 0 < y < +\infty$ :  
 результат:  $\pi$
- > если  $x = -\infty, -\infty < y < 0$ :  
 результат:  $-\pi$
- > если  $x = +\infty, 0 < y < +\infty$ :  
 результат:  $+0$
- > если  $x = +\infty, -\infty < y < 0$ :  
 результат:  $-0$

## Примеры



```
a: double = Math.Atan2(0, 1); // Результат: 0
```

# Atanh

Возвращает обратный гиперболический тангенс указанного числа.



float Math.Atanh(float x)  
double Math.Atanh(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Число, представляющее гиперболический тангенс. $-1 \leq x \leq 1$

## Возвращаемое значение

Тип – такой же, как у параметра.

Обратный гиперболический тангенс x: угол в радианах, гиперболический тангенс которого равен x.

Значение:

- › если  $-1 < x < 1$ :  
     $-\infty < \text{результат} < +\infty$
- › если  $x = -1$ :  
    результат:  $-\infty$
- › если  $x = 1$ :  
    результат:  $+\infty$
- › если  $x < -1$  или  $x > 1$ :  
    результат: NaN

## Примеры



```
a: double = Math.Atanh(0); // Результат: 0
```



# Cbrt

Возвращает кубический корень указанного числа.



```
float Math.Cbrt(float x)  
double Math.Cbrt(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, из которого нужно извлечь кубический корень

## Возвращаемое значение

Тип – такой же, как у параметра.

Кубический корень из x.

## Примеры



```
a: double = Math.Cbrt(-125); // Результат: -5
```

# Ceil

Возвращает число, округленное до большего целого.



float Math.Ceil(float value)  
double Math.Ceil(double value)

## Параметры

Параметр	Тип	Описание
value	float, double	Округляемое число

## Возвращаемое значение

Тип – такой же, как у параметра value.

Наименьшее целое число, которое больше или равно value.

## Примеры



```
a: double = Math.Ceil(2.11); // Результат: 3  
b: double = Math.Ceil(-4.9); // Результат: -4
```

# Clamp

Возвращает указанное число, ограниченное диапазоном [min; max].



```
int1 Math.Clamp(int1 Value, int1 Min, int1 Max)
int2 Math.Clamp(int2 Value, int2 Min, int2 Max)
int4 Math.Clamp(int4 Value, int4 Min, int4 Max)
int8 Math.Clamp(int8 Value, int8 Min, int8 Max)
uint1 Math.Clamp(uint1 Value, uint1 Min, uint1 Max)
uint2 Math.Clamp(uint2 Value, uint2 Min, uint2 Max)
uint4 Math.Clamp(uint4 Value, uint4 Min, uint4 Max)
uint8 Math.Clamp(uint8 Value, uint8 Min, uint8 Max)
float Math.Clamp(float Value, float Min, float Max)
double Math.Clamp(double Value, double Min, double Max)
```

## Параметры

Параметр	Тип	Описание
Value	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Ограничиваемое значение
Min	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Нижняя граница ограничения
Max	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Верхняя граница ограничения

## Возвращаемое значение

Тип – такой же, как у параметров.

Число Value, ограниченное диапазоном:

› если  $\text{min} \leq \text{value} \leq \text{max}$ :

результат: `value`

› если  $\text{value} < \text{min}$ :

результат: `min`

› если  $\text{value} > \text{max}$ :

результат: `max`

## Примеры



```
a: int4 = Math.Clamp(5, 0, 10); // Результат: 5
```

```
b: int4 = Math.Clamp(15, 0, 10); // Результат: 10
```

```
c: int4 = Math.Clamp(-5, 0, 10); // Результат: 0
```

# ClearBit

Возвращает число, у которого обнулен указанный бит.



```
uint1 Math.ClearBit(uint1 value, uint1 n)
uint2 Math.ClearBit(uint2 value, uint1 n)
uint4 Math.ClearBit(uint4 value, uint1 n)
uint8 Math.ClearBit(uint8 value, uint1 n)
```

## Параметры

Параметр	Тип	Описание
value	uint1, uint2, uint4, uint8	Число, над которым будет выполняться функция
n	uint1	Номер бита, нумерация начинается с нуля

## Возвращаемое значение

Тип – такой же, как у параметра value.

Значение параметра value, у которого обнулен n-й бит. Если n больше размерности типа value, возвращает значение value без изменений.

## Примеры



```
//У числа 30 (11110 в двоичной записи) обнулить бит под номером
1:
a: uint4 = Math.ClearBit(30, 1)
// Результат: 28 (11100 в двоичной записи).
```

# CopySign

Возвращает число с величиной X и знаком числа y.



```
float Math.CopySign(float X, float Y)  
double Math.CopySign(double X, double Y)
```

## Параметры

Параметр	Тип	Описание
X	float, double	Число, величина которого используется в возвращаемом значении
Y	float, double	Число, знак которого используется в возвращаемом значении

## Возвращаемое значение

Тип – такой же, как у параметра value.

Величина x со знаком, как у Y.

## Примеры



```
a: double = Math.CopySign(5, 1); // Результат: 5  
b: double = Math.CopySign(5, -3); // Результат: -5  
c: double = Math.CopySign(-5, 10); // Результат: 5  
d: double = Math.CopySign(-5, -4); // Результат: -5
```

# Cos

Возвращает косинус угла.



float Math.Cos(float value)  
double Math.Cos(double value)

## Параметры

Параметр	Тип	Описание
value	float, double	Угол в радианах

## Примечание

Чтобы преобразовать градусы в радианы, можете использовать следующую формулу:



$\text{radians} = \text{degrees} * \text{Math.pi}()/180$

## Возвращаемое значение

Тип – такой же, как у параметра value.

Косинус угла value.

## Примеры



```
Result: double = Math.Cos(1); // Результат: 0.54030230586814
```

# Cosh

Возвращает гиперболический косинус числа.



```
float Math.Cosh(float x)  
double Math.Cosh(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Угол в радианах

## Возвращаемое значение

Тип – такой же, как у параметра.

Гиперболический косинус x.

Значение:

- › если  $-\infty < x < +\infty$   
1 ≤ результат
- › если  $x = \pm\infty$ :  
результат:  $+\infty$

## Примеры



```
a: double = Math.Cosh(0); // Результат: 1
```



# Dim

Возвращает положительную разность двух чисел.



```
float Math.Dim(float x, float y)
double Math.Dim(double x, double y)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Уменьшаемое
y	float, double	Вычитаемое

## Возвращаемое значение

Тип – такой же, как у параметров.

Положительная разность x и y.

Значение:

- › если  $x \geq y$ :  
результат:  $x - y$
- › если  $x < y$ :  
результат: 0



Функция аналогична функции `Math.Max(x-y, 0)` и отличается от неё тем, что возвращает NaN, если один из аргументов равен NaN.

## Примеры



```
a: double = Math.Dim(5, 1); // Результат: 4  
b: double = Math.Dim(5, 6); // Результат: 0
```

# e

Возвращает константу e.



double Math.e()

## Возвращаемое значение

Тип – double.

Число 2.7182818284590451.

## Примеры



```
e: double = Math.e(); // Результат: 2.7182818284590451
```

# Erf

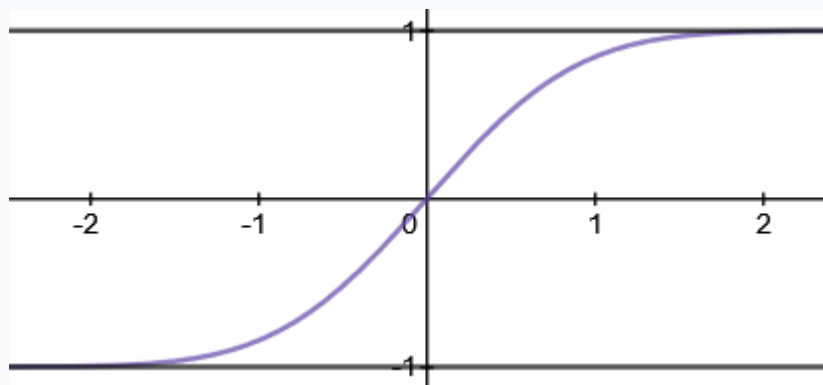
Возвращает значение функции ошибок.



Функция ошибок – неэлементарная функция в теории вероятности и статистике, которая вычисляется по формуле:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

График функции:



float Math.Erf(float x)  
double Math.Erf(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Аргумент функции

## Возвращаемое значение

Тип – такой же, как у параметра.

Значение:

› если  $-\infty < x < +\infty$ :

результат – значение функции ошибок для  $x$ .

$-1 < \text{результат} < 1$

› если  $x = -\infty$ :

результат:  $-1$

› если  $x = +\infty$ :

результат:  $1$

## Примеры



```
Result: float = Math.Erf(0); // Результат: 0
```

# Erfc

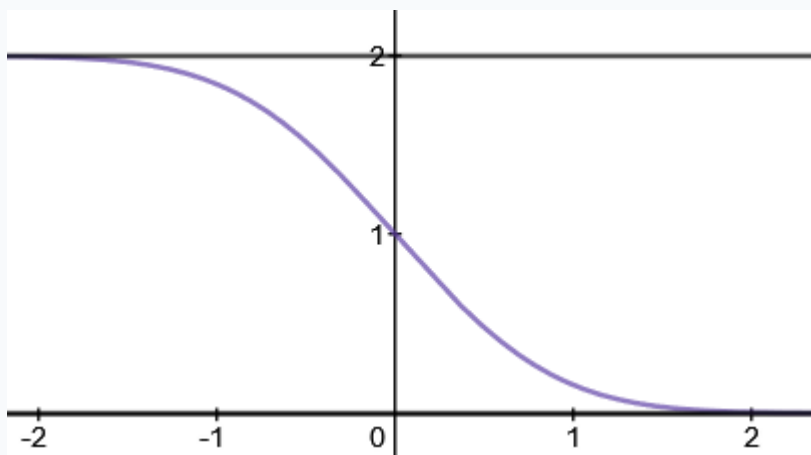
Возвращает значение дополнительной функции ошибок.



Функция ошибок – неэлементарная функция в теории вероятности и статистике, которая вычисляется по формуле:

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$$

График функции:



float Math.Erfc(float x)  
double Math.Erfc(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Аргумент функции

## Возвращаемое значение

Тип – такой же, как у параметра.

Значение:

- › если  $-\infty < x < +\infty$ :  
результат – значение обратной функции ошибок для  $x$ .  
 $0 < \text{результат} < 2$
- › если  $x = -\infty$ :  
результат: 2
- › если  $x = +\infty$ :  
результат: 0

## Примеры



```
Result: float = Math.Erfc(0); // Результат: 1
```

# Exp

Возвращает число  $e$ , возведенное в указанную степень.



float Math.Exp(float p)  
double Math.Exp(double p)

## Параметры

Параметр	Тип	Описание
p	float, double	Показатель степени

## Возвращаемое значение

Тип – такой же, как у параметра p.

Число  $e$ , возведенное в степень p.

## Примеры



Возвести число  $e$  в степень 2.5:  
Result: double = Math.Exp(2.5); // Результат: 12.1824939607035



# Exp2

Возвращает число 2, возведенное в указанную степень.



```
float Math.Exp2(float x)  
double Math.Exp2(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Показатель степени

## Возвращаемое значение

Тип – такой же, как у параметра.

Число 2 в степени x.

## Примеры



```
a: double = Math.Exp2(5); // Результат: 32  
b: double = Math.Exp2(0); // Результат: 1  
c: double = Math.Exp2(-1); // Результат: 0.5
```

# Exp1

Возвращает значение  $e^x-1$ .



float Math.Exp1(float x)  
double Math.Exp1(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Показатель степени

## Возвращаемое значение

Тип – такой же, как у параметра.

Значение:

- › если  $-\infty < x < +\infty$ :  
результат – значение  $e^{(x-1)}$ .  
 $-1 < \text{результат} < +\infty$
- › если  $x = -\infty$ :  
результат: -1
- › если  $x = +\infty$ :  
результат:  $+\infty$

## Примеры



```
a: double = Math.Exp1(5); // Результат: 147.413162
```

# Floor

Возвращает число, округленное вниз.



float Math.Floor(float value)  
double Math.Floor(double value)

## Параметры

Параметр	Тип	Описание
value	float, double	Округляемое число

## Возвращаемое значение

Тип – такой же, как у параметра value.

Наибольшее целое число, которое меньше или равно value.

## Примеры



```
a: double = Math.Floor(7.56); // Результат: 7  
b: double = Math.Floor(-5.1); // Результат: -6
```

# FusedMultiplyAdd

Возвращает умножение-сложение с однократным округлением:  $(x*y)+z$ .



```
float Math.FusedMultiplyAdd(float x, float y, float z)  
double Math.FusedMultiplyAdd(double x, double y, double z)
```



Вычисление выполняется быстрее и более точно, чем при выполнении тех же операций по отдельности, т.к. выполняется через аппаратные инструкции процессора.

## Параметры

Параметр	Тип	Описание
x	float, double	Первый множитель
y	float, double	Второй множитель
z	float, double	Слагаемое

## Возвращаемое значение

Тип – такой же, как у параметров.

Результат вычисления  $(x*y)+z$ .

## Примеры



```
Result: float = Math.FusedMultiplyAdd(5,5,5); // Результат: 30
```

# Gamma

Возвращает значение гамма-функции для указанного числа.



float Math.Gamma(float x)  
double Math.Gamma(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Аргумент функции

## Возвращаемое значение

Тип – такой же, как у параметра.

Значение гамма-функции для x.

## Примеры



```
Result: float = Math.Gamma(1); // Результат: 1
```

# Нупот

Для указанных длин катетов возвращает длину гипотенузы.



```
float Math.Hypot(float x, float y)
double Math.Hypot(double x, double y)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Длина первого катета Может иметь отрицательное значение, т.к. при расчете возводится в квадрат
y	float, double	Длина второго катета Может иметь отрицательное значение, т.к. при расчете возводится в квадрат

## Возвращаемое значение

Тип – такой же, как у параметров.

Длина гипотенузы: корень из суммы квадратов x и y.

## Примеры



```
Result: float = Math.Hypot(3, 4); // Результат: 5
```

# ILogB

Возвращает порядок числа с плавающей запятой в виде знакового целого числа. Порядок — целое число, которое задает нужную степень двойки. Обычно это не истинная величина порядка, а сдвинутая на некоторую константу таким образом, чтобы число было неотрицательным. Так, наименьший возможный порядок (он отрицательный) представлен числом 0.



int4 Math.ILogB(float x)

int4 Math.ILogB(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Число

## Возвращаемое значение

Тип – int4.

Порядок числа x в виде знакового целого.

## Примеры



```
Result: int1 = Math.ILogB(55.5); // Результат: 5
```

# IsInf

Проверяет, является ли число бесконечностью.



```
bool Math.IsInf(float x)
bool Math.IsInf(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число

## Возвращаемое значение

Тип – bool.

Значение:

- Если  $x = \pm\infty$ :  
результат: true;
- Иначе:  
результат: false.

## Примеры



```
Result: bool = Math.IsInf(1); // Результат: false
```



# IsNaN

Проверяет, является ли число значением NaN.



```
bool Math.IsNaN(float x)  
bool Math.IsNaN(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число

## Возвращаемое значение

Тип – bool.

Значение:

- › Если  $x = \text{NaN}$ :  
результат: true;
- › иначе:  
результат: false.

## Примеры



```
Result: bool = Math.IsNaN(1); // Результат: false
```

# LGamma

Возвращает натуральный логарифм абсолютного значения гамма-функции.



```
float Math.LGamma(float x)  
double Math.LGamma(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Аргумент функции

## Возвращаемое значение

Тип – такой же, как у параметра.

Возвращает натуральный логарифм значения гамма-функции для x.

## Примеры



```
Result: float = Math.LGamma(1); // Результат: 0
```

# Log

Возвращает натуральный логарифм указанного числа.



```
float Math.Log(float x)
float Math.Log(float x, float base)
double Math.Log(double x)
double Math.Log(double x, float base)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, натуральный логарифм которого нужно найти. $0 \leq x$
base	float, double	Опциональный параметр. Основание логарифма. Если не указан, используется значение e: берется натуральный логарифм числа, $0 < \text{base}; \text{base} \neq 1$

## Возвращаемое значение

Тип – такой же, как у параметра x.

Логарифм x по основанию base: степень, в которую надо возвести base, чтобы получить x, если base не указан, то натуральный логарифм x: степень, в которую надо возвести число e, чтобы получить x.

Значение:

› если base не указан:

результат – натуральный логарифм x: степень, в которую надо возвести число e, чтобы получить x.

› если base указан:

результат – логарифм  $x$  по основанию  $base$ : степень, в которую надо возвести  $base$ , чтобы получить  $x$ .

$-\infty < \text{результат} < +\infty$

## Примеры



```
Result: float = Math.Log(2,2); // Результат: 1
```

# Log1p

Возвращает натуральный логарифм числа  $x+1$ .



Функция возвращает более точный результат, чем `Math.Log(x+1)` для значений  $x$  близких к 0.



`float Math.Log1p(float x)`  
`double Math.Log1p(double x)`

## Параметры

Параметр	Тип	Описание
<code>x</code>	<code>float</code> , <code>double</code>	Число

## Возвращаемое значение

Тип – такой же, как у параметра.

Натуральный логарифм числа  $x+1$ : логарифм числа  $x+1$  по основанию  $e$ .

$-\infty < \text{результат} < +\infty$

## Примеры



```
Result: float = Math.Log1p(15); // Результат: 2.772589
```

# Log2

Возвращает двоичный логарифм числа.



float Math.Log2(float x)

double Math.Log2(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Число, двоичный логарифм которого нужно найти. $0 < x$

## Возвращаемое значение

Тип – такой же, как у параметра.

Двоичный логарифм  $x$ : степень, в которую нужно возвести 2, чтобы получить  $x$ .

Значение:

- › если  $x > 0$ :  
результат – двоичный логарифм  $x$ : степень, в которую нужно возвести 2, чтобы получить  $x$ .
- › если  $x = 0$ :  
результат:  $-\infty$
- › если  $x = +\infty$ :  
результат:  $+\infty$
- › если  $x < 0$ :  
результат: NaN

## Примеры



```
Result: float = Math.Log2(-1); // Результат: NaN
```

```
Result: double = Math.Log2(16); // Результат: 4
```

# Log10

Возвращает десятичный логарифм числа.



```
loat Math.Log10(float x)
double Math.Log10(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, десятичный логарифм которого нужно найти. $0 < x$

## Возвращаемое значение

Тип – такой же, как у параметра.

Десятичный логарифм x: степень, в которую нужно возвести 10, чтобы получить x.

Значение:

- › если  $x > 0$ :  
результат – десятичный логарифм x: степень, в которую нужно возвести 10, чтобы получить x.
- › если  $x = 0$ :  
результат:  $-\infty$
- › если  $x = +\infty$ :  
результат:  $+\infty$
- › если  $x < 0$ :  
результат: NaN

## Примеры





```
a: double = Math.Log10(10); // Результат: 1  
b: double = Math.Log10(100); // Результат: 2  
c: double = Math.Log10(1); // Результат: 0  
d: double = Math.Log10(0.1); // Результат: -1
```

# LogB

Возвращает порядок числа с плавающей запятой. Порядок — целое число, которое задает нужную степень двойки. Обычно это не истинная величина порядка, а сдвинутая на некоторую константу таким образом, чтобы число было неотрицательным. Так, наименьший возможный порядок (он отрицательный) представлен числом 0.



float Math.LogB(float x)  
double Math.LogB(double x)

## Параметры

Параметр	Тип	Описание
x	float, double	Число

## Возвращаемое значение

Тип — такой же, как у параметра.

Порядок числа  $x$  в виде знакового числа с плавающей запятой.

Значение:

- › если  $x \neq 0$  и  $x \neq \pm\infty$ :  
результат — порядок числа  $x$  в виде знакового числа с плавающей запятой.
- › если  $x = 0$ :  
результат:  $-\infty$
- › если  $x = \pm\infty$ :  
результат:  $+\infty$

## Примеры



```
Result: float = Math.LogB(5.33); // Результат: 2
```

# Max

Возвращает большее из двух чисел.



```
int1 Math.Max(int1 X, int1 Y)
int2 Math.Max(int2 X, int2 Y)
int4 Math.Max(int4 X, int4 Y)
int8 Math.Max(int8 X, int8 Y)
uint1 Math.Max(uint1 X, uint1 Y)
uint2 Math.Max(uint2 X, uint2 Y)
uint4 Math.Max(uint4 X, uint4 Y)
uint8 Math.Max(uint8 X, uint8 Y)
float Math.Max(float X, float Y)
double Math.Max(double X, double Y)
```

## Параметры

Параметр	Тип	Описание
X	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Первое число
Y	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Второе число

## Возвращаемое значение

Тип – совпадает с типом параметров.

Большее из чисел X и Y.

## Примеры



```
result: uint1 = Math.Max(43, 47); // Результат: 47
```

# Min

Возвращает меньшее из двух чисел.



```
int1 Math.Min(int1 X, int1 Y)
int2 Math.Min(int2 X, int2 Y)
int4 Math.Min(int4 X, int4 Y)
int8 Math.Min(int8 X, int8 Y)
uint1 Math.Min(uint1 X, uint1 Y)
uint2 Math.Min(uint2 X, uint2 Y)
uint4 Math.Min(uint4 X, uint4 Y)
uint8 Math.Min(uint8 X, uint8 Y)
float Math.Min(float X, float Y)
double Math.Min(double X, double Y)
```

## Параметры

Параметр	Тип	Описание
X	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Первое число
Y	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Второе число

## Возвращаемое значение

Тип – совпадает с типом параметров.

Меньшее из чисел X и Y.

## Примеры



```
result: uint1 = Math.Min(43, 47); // Результат: 43
```

# NextAfter

Для указанного вещественного числа возвращает ближайшее соседнее число, представимое в рамках типа.



```
float Math.NextAfter(float from, float to)  
double Math.NextAfter(double from, float to)
```

## Параметры

Параметр	Тип	Описание
from	float, double	Число, для которого будет искаться сосед
to	float, double	Число, по направлению к которому будет искаться сосед

## Возвращаемое значение

Тип – такой же, как у параметров.

Число, ближайшее к from в направлении to и представимое в рамках типа.

## Примеры



```
Result: float = Math.NextAfter(1.1, 2.2); // Результат: 1.1
```



# pi

Возвращает число Пи.



double Math.pi()

## Возвращаемое значение

Тип – double.

Число 3.1415926535897931.

## Примеры



```
Result: double = Math.pi(); // Результат: 3.1415926535897931
```

# Pow

Возвращает число, возведенное в указанную степень.



```
float Math.Pow(float X, float Y)  
double Math.Pow(double X, double Y)
```

## Параметры

Параметр	Тип	Описание
X	float, double	Первое число
Y	float, double	Второе число

## Возвращаемое значение

Тип – совпадает с типом параметров.

Число X, возведенное в степень Y.

## Примеры



```
//Возвести число 1.5 в степень 4  
Result: double = Math.Pow(1.5, 4); // Результат: 5.0625
```

# Remainder

Возвращает вещественный остаток от деления согласно стандарту IEEE-754.



```
float Math.Remainder(float x, float y)
double Math.Remainder(double x, float y)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Делимое
y	float, double	Делитель

## Возвращаемое значение

Тип – такой же, как у параметров.

Возвращает остаток от деления  $x/y$  согласно определению в стандарте IEEE-754:  $result = x - (y * Q)$ , где  $Q$  – частное  $x/y$ , округленное до ближайшего целого.

## Примеры



```
Result: float = Math.Remainder(5,2); // Результат: 1
```

# Round

Возвращает число, округленное до ближайшего целого или до указанного количества знаков после запятой.



```
float Math.Round(float value)
float Math.Round(float value, uint4 precision)
double Math.Round(double value)
double Math.Round(double value, uint4 precision)
```

## Параметры

Параметр	Тип	Описание
value	float, double	Округляемое число
precision	uint4	Опциональный параметр Количество знаков после запятой. Если не указан, принимается равным нулю (округление до целого)

## Возвращаемое значение

Тип – такой же, как у параметра value.

Число value, округленное до ближайшего целого или до указанного количества знаков после запятой.

## Примеры



```
//Округлить до целого
a: double = Math.Round(1.1); // Результат: 1
b: double = Math.Round(7.5); // Результат: 8
c: double = Math.Round(-1.5); // Результат: -2
```



//Округлить до указанного количества знаков после запятой

x: double = Math.Round(4.666666, 2); // Результат: 4.67

my\_pi: double = Math.Round(Math.pi(), 5); // Результат: 3,14159

# ScaleB

Возвращает произведение числа на 2 в указанной степени:  $x \cdot 2^n$ .



Вычисление выполняется быстрее, чем при выполнении тех же операций по-отдельности, т.к. выполняется через аппаратные инструкции процессора.



float Math.ScaleB(float x, int4 exp)  
double Math.ScaleB(double x, int4 exp)

## Параметры

Параметр	Тип	Описание
x	float, double	Множитель
exp	int4	Показатель степени, в которую надо возвести 2

## Возвращаемое значение

Тип – такой же, как у параметра x.

Значение  $x \cdot 2^{\text{exp}}$ .

## Примеры



```
Result: float = Math.ScaleB(1,1); // Результат: 2
```

# SetBit

Возвращает число, у которого значение указанного бита установлено в true или изменено на указанное значение.



```
uint1 Math.SetBit(uint1 value, uint1 n)
uint2 Math.SetBit(uint2 value, uint1 n)
uint4 Math.SetBit(uint4 value, uint1 n)
uint8 Math.SetBit(uint8 value, uint1 n)
uint1 Math.SetBit(uint1 value, uint1 n, bool bitValue)
uint2 Math.SetBit(uint2 value, uint1 n, bool BitValue)
uint4 Math.SetBit(uint4 value, uint1 n, bool bitValue)
uint8 Math.SetBit(uint8 value, uint1 n, bool BitValue)
```

## Параметры

Параметр	Тип	Описание
value	fuint4, uint8	Число, над которым будет выполняться функция
n	uint1	Номер бита, нумерация начинается с нуля
bitValue	bool	Опциональный параметр. Новое значение n-го бита: true (1) или false (0) Если не указан, значение бита будет установлено в true

## Возвращаемое значение

Тип – такой же, как у параметра value.

Значение параметра value, у которого n-му биту установлено значение bitValue или true, если bitValue не указан. Если n больше размерности типа value, возвращает значение value без изменений.

## Примеры



//У числа 22 (10110 в двоичной записи) установить бит под номером 3

a: uint4 = Math.SetBit(22, 3) // Результат: 30 (11110 в двоичной записи).



//У числа 22 (10110 в двоичной записи) изменить бит под номером 5

b: uint4 = Math.SetBit(22, 5) // Результат: 54 (110110 в двоичной записи).



//У числа 19 (10011 в двоичной записи) обнулить значение бита под номером 0

c: uint4 = Math.SetBit(19, 0, false) // Результат: 18 (10010 в двоичной записи).



# SignBit

Проверяет, является ли число отрицательным.



```
bool Math.SignBit(float x)  
bool Math.SignBit(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Число, знак которого проверяем

## Возвращаемое значение

Тип – bool.

Значение:

- › если  $x < 0$ ,  $x = -0$ ,  $x = -\infty$  или  $x$  – отрицательный NaN:  
результат: true
- › если  $x > 0$ ,  $x = +0$ ,  $x = +\infty$  или  $x$  – положительный NaN:  
результат: false

## Примеры



```
Result: bool = Math.SignBit(-120); // Результат: true
```

# Sin

Возвращает синус угла.



float Math.Sin(float value)  
double Math.Sin(double value)

## Параметры

Параметр	Тип	Описание
value	float, double	Угол в радианах



Чтобы преобразовать градусы в радианы, можете использовать следующую формулу:  
 $\text{radians} = \text{degrees} * \text{Math.pi}()/180$

## Возвращаемое значение

Тип – такой же, как у параметра value.

Синус угла value.

## Примеры



```
Result: double = Math.Sin(1); // Результат: 0.841470984807897
```

# Sinh

Возвращает гиперболический синус числа.



```
loat Math.Sinh(float x)  
double Math.Sinh(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Угол в радианах

## Возвращаемое значение

Тип – такой же, как у параметра.

Значение:

- › если  $-\infty < x < +\infty$ :  
результат – гиперболический синус  $x$ .
- › если  $x = -\infty$ :  
результат:  $-\infty$
- › если  $x = +\infty$ :  
результат:  $+\infty$

## Примеры



```
a: double = Math.Sinh(0); // Результат: 0
```

# Sqr

Возвращает квадрат числа.



```
int1 Math.Sqr(int1 value)
int2 Math.Sqr(int2 value)
int4 Math.Sqr(int4 value)
int8 Math.Sqr(int8 value)
uint1 Math.Sqr(uint1 value)
uint2 Math.Sqr(uint2 value)
uint4 Math.Sqr(uint4 value)
uint8 Math.Sqr(uint8 value)
float Math.Sqr(float value)
double Math.Sqr(double value)
```

## Параметры

Параметр	Тип	Описание
value	int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Возводимое в квадрат число

## Возвращаемое значение

Тип – такой же, как у параметра value.

Число value, возведенное в квадрат. Для операции применяются правила переполнения типа операции умножения.

## Примеры



```
Result: int4 = Math.Sqrt(12); // Результат: 144
```

# Sqrt

Возвращает квадратный корень числа.



```
float Math.Sqrt(float value)  
double Math.Sqrt(double value)
```

## Параметры

Параметр	Тип	Описание
value	float, double	Число, квадратный корень которого нужно найти. Должно быть неотрицательным

## Возвращаемое значение

Тип – такой же, как у параметра value.

Квадратный корень из числа value.

Значение:

- › если  $x \geq 0$ :  
результат – квадратный корень из числа value.
- › если  $x < 0$ :  
результат: NaN

## Примеры



```
a: double = Math.Sqrt(1.69); // Результат: 1.3
```

# Tan

Возвращает тангенс угла.

```
float Math.Tan(float x)
```

```
double Math.Tan(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Угол в радианах



Чтобы преобразовать градусы в радианы, можете использовать следующую формулу:

```
radians = degrees*Math.pi()/180
```

## Возвращаемое значение

Тип – такой же, как у параметра x.

Значение:

› если  $-\infty < x < +\infty$ :

результат – тангенс угла x;

› если  $x = \pm\infty$ :

результат: NaN.

## Примеры



```
a: double = Math.Tan(0); // Результат: 0
```

```
b: double = Math.Tan(Math.pi()/4); // Результат: 1
```

# Tanh

Возвращает гиперболический тангенс числа.



```
float Math.Tanh(float x)
double Math.Tanh(double x)
```

## Параметры

Параметр	Тип	Описание
x	float, double	Угол в радианах

## Возвращаемое значение

Тип – такой же, как у параметра.

Гиперболический тангенс x.

Значение:

- › если  $-\infty < x < +\infty$ :  
результат – гиперболический тангенс x.  
 $-1 < \text{результат} < 1$
- › если  $x = -\infty$ :  
результат: -1
- › если  $x = +\infty$ :  
результат: 1

## Примеры



```
Result: float = Math.Tanh(0); // Результат: 0
```



# TestBit

Для числа возвращает значение указанного бита.

Проверить бит X. Возвращает true, если в бите с номером bitNo установлена единица, иначе false. Если bitNo больше размерности типа X, возвращает false.



```
bool Math.TestBit(uint1 value, uint1 bitNo)
```

```
bool Math.TestBit(uint2 value, uint1 bitNo)
```

```
bool Math.TestBit(uint4 value, uint1 bitNo)
```

```
bool Math.TestBit(uint8 value, uint1 bitNo)
```

## Параметры

Параметр	Тип	Описание
value	fuint1, uint2, uint4, uint8	Число, над которым будет выполняться функция
bitNo	uint1	Номер бита, нумерация начинается с нуля

## Возвращаемое значение

Тип – bool.

Значение бита под номером bitNo в числе value.

## Примеры



```
//Для числа 56 (111000 в двоичной записи) определить значение битов:
```

```
a: bool = Math.TestBit(56, 0); // Результат: false
```

```
b: bool = Math.TestBit(56, 3); // Результат: true
```

```
c: bool = Math.TestBit(56, 6); // Результат: false
```

# ToggleBit

Возвращает число, у которого значение указанного бита изменено на противоположное.



```
uint1 Math.ToggleBit(uint1 value, uint1 n)
uint2 Math.ToggleBit(uint2 value, uint1 n)
uint4 Math.ToggleBit(uint4 value, uint1 n)
uint8 Math.ToggleBit(uint8 value, uint1 n)
```

## Параметры

Параметр	Тип	Описание
value	fuint1, uint2, uint4, uint8	Число, над которым будет выполняться функция
n	uint1	Номер бита, нумерация начинается с нуля

## Возвращаемое значение

Тип – такой же, как у параметра value.

Значение параметра value, у которого значение n-го бита изменено на противоположное.

## Примеры



```
//У числа 22 (10110 в двоичной записи) изменить бит под номером 3:
a: uint1 = Math.ToggleBit(22, 3); // Результат: 30 (11100 в двоичной записи)
```



```
//У числа 22 (10110 в двоичной записи) изменить бит под номером 1:
b: uint1 = Math.ToggleBit(22, 1); // Результат: 20 (10100 в двоичной записи)
```



//У числа 22 (10110 в двоичной записи) изменить бит под номером 5:  
b: uint1 = Math.ToggleBit(22, 5); // Результат: 54 (110110 в двоичной записи)

# Truncate

Возвращает число, у которого отброшена дробная часть.



float Math.Truncate(float value)  
double Math.Truncate(double value)

## Параметры

Параметр	Тип	Описание
value	float, double	Усекаемое число

## Возвращаемое значение

Тип – такой же, как у параметра value.

Целая часть числа value.

## Примеры



```
a: double = Math.Truncate(18.7585); // Результат: 18  
a: double = Math.Truncate(-7.2); // Результат: -7
```

# Строковые функции



Вызов строковых функций осуществляется обращением к пространству имен String.

Пространство имен Str является устаревшим и оставлено для обратной совместимости.

Функция	Описание
<a href="#">Add</a>	Возвращает объединение двух строк
<a href="#">Concat</a>	Возвращает объединение двух строк
<a href="#">Contains</a>	Проверяет, есть ли в строке указанная подстрока
<a href="#">EndsWith</a>	Проверяет, заканчивается ли строка на указанную подстроку
<a href="#">EQ</a>	Лексикографически сравнивает две строки аналогично оператору "равно"
<a href="#">Equals</a>	Проверяет две строки на равенство
<a href="#">GE</a>	Лексикографически сравнивает две строки аналогично оператору "больше или равно"
<a href="#">GT</a>	Лексикографически сравнивает две строки аналогично оператору "больше"
<a href="#">IndexOf</a>	Возвращает индекс первого вхождения подстроки в указанную строку
<a href="#">Insert</a>	Возвращает строку, в указанное место которой вставлена другая строка
<a href="#">IsValidFormat</a>	Проверяет, является ли указанная строка корректной строкой формата printf
<a href="#">LastIndexOf</a>	Возвращает индекс последнего вхождения подстроки в указанную строку
<a href="#">LE</a>	Лексикографически сравнивает две строки аналогично оператору "меньше или равно"
<a href="#">Length</a>	Возвращает количество символов в строке

<a href="#">LT</a>	Лексикографически сравнивает две строки аналогично оператору "меньше"
<a href="#">NE</a>	Лексикографически сравнивает две строки аналогично оператору "не равно"
<a href="#">Remove</a>	Возвращает строку, в которой начиная с указанной позиции удалено указанное количество символов
<a href="#">Replace</a>	Возвращает строку, в которой все вхождения одной подстроки заменены на другую подстроку
<a href="#">Reserve</a>	Возвращает пустую строку с местом, зарезервированным для хранения указанного количества символов
<a href="#">StartsWith</a>	Проверяет, начинается ли строка с указанной подстроки
<a href="#">SubString</a>	Для указанной строки возвращает подстроку
<a href="#">ToBool</a>	Конвертирует указанную строку в значение типа bool
<a href="#">ToDouble</a>	Конвертирует указанную строку в число типа double
<a href="#">ToFloat</a>	Конвертирует указанную строку в число типа float
<a href="#">ToInt1</a>	Конвертирует указанную строку в число типа int1
<a href="#">ToInt2</a>	Конвертирует указанную строку в число типа int2
<a href="#">ToInt4</a>	Конвертирует указанную строку в число типа int4
<a href="#">ToInt8</a>	Конвертирует указанную строку в число типа int8
<a href="#">ToLocalizedString</a>	Возвращает строковое представление указанного значения с учетом системного языка
<a href="#">ToLower</a>	Возвращает строку, в которой все символы приведены к нижнему регистру
<a href="#">ToString</a>	Возвращает строковое представление указанного значения
<a href="#">ToUInt1</a>	Конвертирует указанную строку в число типа uint1
<a href="#">ToUInt2</a>	Конвертирует указанную строку в число типа uint2
<a href="#">ToUInt4</a>	Конвертирует указанную строку в число типа uint4

<a href="#">ToUint8</a>	Конвертирует указанную строку в число типа uint8
<a href="#">ToUpper</a>	Возвращает строку, в которой все символы приведены к верхнему регистру
<a href="#">Trim</a>	Возвращает строку, в которой удалены пробелы в начале и в конце строки

# Add

Возвращает объединение двух строк.



```
string String.Add(string x, string y)
```

Функция является полным аналогом функции [String.Concat](#).

## Примеры



```
a: string = String.Add("Note: ", "Message"); // Результат: "Note:  
Message"
```



# Concat

Возвращает объединение двух строк.



```
string String.Concat(string x, string y)
```

## Параметры

Параметр	Тип	Описание
x	string	Первая строка
y	string	Вторая строка

## Возвращаемое значение

Тип – string.

Строка x, в конец которой добавлена строка y.



Функция аналогична сложению двух строк:  $s1 + s2$ .

## Примеры



```
a: string = String.Concat("Note: ", "Message"); // Результат: "Note: Message"
```

# Contains

Проверяет, есть ли в строке указанная подстрока.



```
bool String.Contains(string x, string y)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, в которой будем проверять наличие подстроки
y	string	Подстрока, которую будем искать

## Возвращаемое значение

Тип – bool.

Значение:

- › true, если в строке x есть подстрока y,
- › false – если подстроки нет,
- › Если y – пустая подстрока, то возвращается true.

## Примеры



```
a: bool = String.Contains("Hello, World!", "World"); // Результат: true  
b: bool = String.Contains("Hello, World!", "hello"); // Результат: false  
c: bool = String.Contains("Hello, World!", ""); // Результат: true
```

# EndsWith

Проверяет, заканчивается ли строка на указанную подстроку.



```
bool String.EndsWith(string x, string y)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем проверять
y	string	Подстрока, которую будем искать в конце строки

## Возвращаемое значение

Тип – bool.

Значение:

- › true, если строка x заканчивается на подстроку y;
- › иначе – false,
- › Если y – пустая строка, вернется true.

## Примеры



```
a: bool = String.EndsWith("The first valve is opened", "opened"); //  
Результат: true
```

# EQ

Лексикографически сравнивает две строки аналогично оператору "равно".



```
bool String.EQ(string s1, string s2)
```

## Параметры

Параметр	Тип	Описание
s1	string	Первая строка
s2	string	Вторая строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если строки s1 и s2 совпадают,
- › иначе – false.



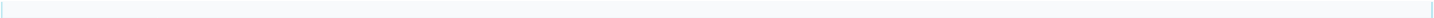
Альтернативные способы сравнения строк на равенство:

- › Оператор сравнения: `s1 == s2`. Полностью аналогичен функции `String.EQ`.
- › Функция `String.Equals`. Может сравнивать строки как с учетом регистра, так и без учета регистра

## Примеры



```
s1: string = "Opened";  
s2: string = "opened";  
res: string = Variant.EQ(s1, s2); //Результат: false
```



# Equals

Проверяет две строки на равенство.



```
bool String.Equals(string x, string y)
```

```
bool String.Equals(string x, string y, bool caseSensitive)
```

## Параметры

Параметр	Тип	Описание
x	string	Первая строка
y	string	Вторая строка
caseSensitive	bool	Опциональный параметр. Чувствительность к регистру символов: <ul style="list-style-type: none"><li>&gt; true – символы сравниваются с учетом регистра.</li><li>&gt; false – строки сравниваются без учета регистра.</li></ul> Если параметр не указан, строки сравниваются с учетом регистра.

## Возвращаемое значение

Тип – bool.

Значение:

- > true, если строки x и y совпадают,
- > false – если не совпадают.

## Примеры



```
a: bool = String.Equals("Opened", "opened"); // Результат: false
```

```
b: bool = String.Equals("Opened", "opened", false); // Результат: true
```

# GE

Лексикографически сравнивает две строки аналогично оператору "больше или равно".



```
bool String.GE(string s1, string s2)
```

## Параметры

Параметр	Тип	Описание
s1	string	Первая строка
s2	string	Вторая строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если строка s1 больше или равна s2,
- › иначе – false.



Функция аналогична оператору сравнения строк: `s1 >= s2`

## Примеры



```
s1: string = "Opened";  
s2: string = "opened";  
res: string = Variant.GE(s1, s2); //Результат: false
```



# GT

Лексикографически сравнивает две строки аналогично оператору "больше".



```
bool String.GT(string s1, string s2)
```

## Параметры

Параметр	Тип	Описание
s1	string	Первая строка
s2	string	Вторая строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если строка s1 больше s2,
- › иначе – false.



Функция аналогична оператору сравнения строк: `s1 > s2`

## Примеры



```
s1: string = "Opened";  
s2: string = "opened";  
res: string = Variant.GT(s1, s2); //Результат: false
```

# IndexOf

Возвращает индекс первого вхождения подстроки в указанную строку.



```
int4 String.IndexOf(string x, string y)
int4 String.IndexOf(string x, string y, int4 start)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, в которой будем искать подстроку
y	string	Подстрока, которую ищем
start	int4	Опциональный параметр. Стартовая позиция, с которой начинается поиск. Нумерация начинается с нуля. Если указано значение меньше нуля, принимается равным нулю

## Возвращаемое значение

Тип – int4.

Номер символа в строке x, с которого начинается первое вхождение подстроки y в строку x. Нумерация символов начинается с нуля.

Если подстрока y не входит в x, вернется -1.

Поиск начинается с позиции start или с начала строки, если start не указано.

Если подстрока не найдена, вернётся -1.

## Примеры



```
a: int4 = String.IndexOf("Warning! Check valve state", "Check"); //
Результат: 9
```

```
b: int4 = String.IndexOf("one and one more", "one"); // Результат: 0
c: int4 = String.IndexOf("one and one more", "one", 1); // Результат: 8
d: int4 = String.IndexOf("one and one more", "one", 10); // Результат: -1
e: int4 = String.IndexOf("Warning!", "warning"); // Результат: -1
```

# Insert

Возвращает строку, в указанное место которой вставлена другая строка.



```
string String.Insert(string x, int4 start, string y)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
start	int4	Позиция, в которую надо вставить строку. Если значение меньше нуля – считается равной нулю. Если значение больше длины строки x – считается равной длине строки x
y	string	Строка, которую нужно вставить

## Возвращаемое значение

Тип – string.

Строка x, в которую начиная с позиции start вставлена строка y.

## Примеры



```
a: string = String.Insert("default string", 0, "my "); // Результат: "my default string"
```

```
b: string = String.Insert("default string", 7, " modified"); // Результат: "default modified string"
```

```
c: string = String.Insert ("default string", 100, " is short"); // Результат: "default string is short"
```

# IsValidFormat

Проверяет, является ли указанная строка корректной строкой формата printf. Описание формата printf: [https://en.wikipedia.org/wiki/Printf\\_format\\_string](https://en.wikipedia.org/wiki/Printf_format_string)



```
bool String.IsValidFormat(string x)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true, если строка x является корректной строкой в формате printf,
- › иначе – false.

## Примеры



```
x: string = "Warning! Check valve state";  
res: bool = String.IsValidFormat(x); //Результат: true
```

# LastIndexOf

Возвращает индекс последнего вхождения подстроки в указанную строку.



```
int4 String.LastIndexOf(string x, string y)
```

```
int4 String.LastIndexOf(string x, string y, int4 start)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, в которой будем искать подстроку
y	string	Подстрока, которую ищем
start	int4	Опциональный параметр. Стартовая позиция, с которой начинается поиск. Нумерация начинается с нуля. Если указано значение меньше нуля, принимается равным нулю

## Возвращаемое значение

Тип – int4.

Номер символа в строке x, с которого начинается последнее вхождение подстроки y в строку x. Нумерация символов начинается с нуля.

Поиск начинается с позиции start или с конца строки, если start не указано, и идёт в направлении начала строки.

Если start указано, искомая будет такое вхождение, которое заканчивается не позднее позиции start. Иными словами, поиск будет выполняться так, будто после позиции start символов нет.

Если подстрока не найдена, вернётся -1.

## Примеры



```
a: int4 = String.LastIndexOf("Warning! Check valve state", "Check"); //
```

Результат: 9

```
b: int4 = String.LastIndexOf("one and one more", "one"); // Результат: 8
```

```
c: int4 = String.LastIndexOf("one and one more", "one", 2); // Результат: 0
```

```
d: int4 = String.LastIndexOf("one and one more", "one", 1); // Результат:
```

-1

```
e: int4 = String.LastIndexOf("Warning!", "warning"); // Результат: -1
```

# LE

Лексикографически сравнивает две строки аналогично оператору "меньше или равно".



```
bool String.LE(string s1, string s2)
```

## Параметры

Параметр	Тип	Описание
s1	string	Первая строка
s2	string	Вторая строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если строка s1 меньше или равна s2,
- › иначе – false.



Функция аналогична оператору сравнения строк: `s1 <= s2`

## Примеры



```
s1: string = "Opened";  
s2: string = "opened";  
res: string = Variant.LE(s1, s2); //Результат: true
```



# Length

Возвращает количество символов в строке.



int4 String.Length(string x)

## Параметры

Параметр	Тип	Описание
x	string	Строка

## Возвращаемое значение

Тип – int4.

Количество символов в строке x.

## Примеры



```
a: int4 = String.Length("Warning!"); // Результат: 8  
b: int4 = String.Length(""); // Результат: 0
```

# LT

Лексикографически сравнивает две строки аналогично оператору "меньше".



```
bool String.LT(string s1, string s2)
```

## Параметры

Параметр	Тип	Описание
s1	string	Первая строка
s2	string	Вторая строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если строка s1 меньше s2,
- › иначе – false.



Функция аналогична оператору сравнения строк: `s1 < s2`

## Примеры



```
s1: string = "Opened";  
s2: string = "opened";  
res: string = Variant.LT(s1, s2); //Результат: true
```

# NE

Лексикографически сравнивает две строки аналогично оператору "не равно".



```
bool String.NE(string s1, string s2)
```

## Параметры

Параметр	Тип	Описание
s1	string	Первая строка
s2	string	Вторая строка

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если строки s1 и s2 различаются;
- › если равны – вернется false.



Функция аналогична оператору сравнения строк: s1 != s2

## Примеры



```
s1: string = "Opened";  
s2: string = "opened";  
res: string = Variant.NE(s1, s2); //Результат: true
```

# Remove

Возвращает строку, в которой начиная с указанной позиции удалено указанное количество символов.



```
string String.Remove(string x, int4 start, int4 length)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
start	int4	Позиция, начиная с которой будем удалять символы. Если значение меньше нуля – считается равной нулю. Если значение больше длины строки – символы не будут удаляться
length	int4	Количество удаляемых символов. Если значение меньше или равно нулю – символы не будут удаляться. Может быть больше количества символов в строке после позиции start – в этом случае будут удалены все символы до конца строки

## Возвращаемое значение

Тип – string.

Строка x, из которой удалено length символов, начиная с позиции start.

## Примеры



```
a: string = String.Remove("my default string", 3, 8); // Результат: "my string"
```

```
b: string = String.Remove("Note: message", -100, 6); // Результат:
"message"
c: string = String.Remove("Note: message", 4, 100); // Результат: "Note"
```

# Replace

Возвращает строку, в которой все вхождения одной подстроки заменены на другую подстроку.



```
string String.Replace(string x, string oldStr, string newStr)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
oldStr	string	Подстрока, которую надо заменить. Если значение – пустая строка, то вернется исходная строка
newStr	string	Подстрока, которую надо подставить вместо заменяемой

## Возвращаемое значение

Тип – string.

Строка x, в которой каждое вхождение подстроки oldStr заменено на newStr.

## Примеры



```
a: string = String.Replace("Let's start!", "start", "continue"); // Результат:  
"Let's continue!"
```

```
b: string = String.Replace("Switch 1 turns off device 1.", "1", "2"); //  
Результат: "Switch 2 turns off device 2."
```

```
c: string = String.Replace("Note: Message", "Note: ", ""); // Результат:  
"Message"
```

# Reserve

Возвращает пустую строку с местом, зарезервированным для хранения указанного количества символов. Используется, если внешней функции нужно на вход подать строку определенного размера для записи выходного значения.



```
string String.Reserve(int4 capacity)
```

## Параметры

Параметр	Тип	Описание
capacity	int4	Количество символов. Должно быть неотрицательным

## Возвращаемое значение

Тип – string.

Пустая строка с местом, зарезервированным под хранение capacity символов.

## Примеры



```
a: string = String.Reserve(10); // Результат: "" (пустая строка)
b: int4 = String.Length(a); // Результат: 10 - количество символов, под
которые зарезервировано место.
```

# StartsWith

Проверяет, начинается ли строка с указанной подстроки.



```
bool String.StartsWith(string x, string y)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем проверять
y	string	Подстрока, которую будем искать в начале строки

## Возвращаемое значение

Тип – bool.

Значение:

- › true, если строка x начинается с подстроки y;
- › иначе – false,
- › Если y – пустая строка, вернется true.

## Примеры



```
a: bool = String.StartsWith("Warning! Pump is broken", "Warning!"); //  
Результат: true
```



# SubString

Для указанной строки возвращает подстроку.



```
string String.SubString(string x, int4 start)
string String.SubString(string x, int4 start, int4 length)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, в которой будет искаться подстрока
start	int4	Номер символа, с которого извлекать подстроку. Нумерация начинается с нуля. Если указано значение меньше нуля, принимается равным нулю
length	int4	Опциональный параметр. Длина подстроки. Если указан 0 – вернется пустая подстрока. Если указано значение меньше нуля, принимается равным нулю

## Возвращаемое значение

Тип – string.

Подстрока строки x, начиная с позиции start длины length (если length указан) или до конца строки (если length не указан).

## Примеры



```
a: string = String.SubString("Hello, World!", 0, 5); // Результат: "Hello"
b: string = String.SubString("Hello, World!", 5); // Результат: ", World!"
```

# ToBool

Конвертирует указанную строку в значение типа bool.



```
float String.ToBool(string x)
float String.ToBool(string x, bool defValue)
float String.ToBool(string x, bool defValue, bool caseSensitive)
```

## Параметры

Параметр	Тип	Описание
x	string	Первая строка
defValue	bool	Опциональный параметр. Значение, которое вернется, если конвертация будет неуспешна. Если значение не указано, считается равным false
caseSensitive	bool	Чувствительность к регистру. Если caseSensitive = false, то регистр символов значения не имеет. По умолчанию: true

## Возвращаемое значение

Тип – bool.

Значение:

- › true – если x – строка true.
- › false – если x – строка false.
- › defValue – если конвертировать не получилось.



Если caseSensitive = false, то регистр символов значения не имеет.

## Примеры



```
a: bool = String.ToBool("true"); // Результат: true
b: bool = String.ToBool("True"); // Результат: false
c: bool = String.ToBool("false"); // Результат: false
d: bool = String.ToBool("FALSE", true); // Результат: true
e: bool = String.ToBool("FALSE", true, false); // Результат: false
f: bool = String.ToBool("1"); // Результат: false
g: bool = String.ToBool("non bool"); // Результат: false
h: bool = String.ToBool("non bool"); // Результат: false
i: bool = String.ToBool("non bool", true); // Результат: true
```

# ToDouble

Конвертирует указанную строку в число типа double.



```
double String.ToDouble(string x, double defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем конвертировать
defValue	double	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – double.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: double = String.ToDouble("100.22", 0); // Результат: 100.22  
b: double = String.ToDouble("314e-2", 0); // Результат: 3.14  
c: double = String.ToDouble("not number", -1); // Результат: -1  
d: double = String.ToDouble("1,2", -1); // Результат: -1
```

# ToFloat

Конвертирует указанную строку в число типа float.



```
float String.ToFloat(string x, float defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем конвертировать
defValue	float	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – float.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: float = String.ToFloat("100.22", 0); // Результат: 100.22  
b: float = String.ToFloat("314e-2", 0); // Результат: 3.14  
c: float = String.ToFloat("not number", -1); // Результат: -1  
d: float = String.ToFloat("1,2", -1); // Результат: -1
```

# ToInt1

Конвертирует указанную строку в число типа int1.



```
int1 String.ToInt1(string x, int1 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем конвертировать
defValue	int1	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – int1.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: int1 = String.ToInt1("100", 0); // Результат: 100  
b: int1 = String.ToInt1("not number", -1); // Результат: -1
```

# ToInt2

Конвертирует указанную строку в число типа int2.



```
int2 String.ToInt2(string x, int2 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем конвертировать
defValue	int2	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – int2.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: int2 = String.ToInt2("100", 0); // Результат: 100  
b: int2 = String.ToInt2("not number", -1); // Результат: -1
```

# ToInt4

Конвертирует указанную строку в число типа int4.



```
int4 String.ToInt4(string x, int4 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем конвертировать
defValue	int4	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – int4.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: int4 = String.ToInt4("100", 0); // Результат: 100  
b: int4 = String.ToInt4("not number", -1); // Результат: -1
```



# ToInt8

Конвертирует указанную строку в число типа int8.



```
int8 String.ToInt8(string x, int8 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Строка, которую будем конвертировать
defValue	int4	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – int8.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: int4 = String.ToInt4("100", 0); // Результат: 100  
b: int4 = String.ToInt4("not number", -1); // Результат: -1
```

# ToLocalizedString

Возвращает строковое представление указанного значения с учетом системного языка. Например, в русском языке у дробных чисел в качестве разделителя будет запятая вместо точки.



```
string String.ToLocalizedString(bool x)
string String.ToLocalizedString(int1 x)
string String.ToLocalizedString(int2 x)
string String.ToLocalizedString(int4 x)
string String.ToLocalizedString(int8 x)
string String.ToLocalizedString(uint1 x)
string String.ToLocalizedString(uint2 x)
string String.ToLocalizedString(uint4 x)
string String.ToLocalizedString(uint8 x)
string String.ToLocalizedString(float x)
string String.ToLocalizedString(double x)
string String.ToLocalizedString(string format, int1 x)
string String.ToLocalizedString(string format, int2 x)
string String.ToLocalizedString(string format, int4 x)
string String.ToLocalizedString(string format, int8 x)
string String.ToLocalizedString(string format, uint1 x)
string String.ToLocalizedString(string format, uint2 x)
string String.ToLocalizedString(string format, uint4 x)
string String.ToLocalizedString(string format, uint8 x)
string String.ToLocalizedString(string format, float x)
string String.ToLocalizedString(string format, double x)
```

## Параметры

Параметр	Тип	Описание
format	string	Оptionальный параметр.

		Правило преобразования чисел в строку в формате printf: <a href="https://en.wikipedia.org/wiki/Printf_format_string">https://en.wikipedia.org/wiki/Printf_format_string</a> Если не указан, используется значение %f
x	bool, int1, int2, int4, int8, uint1, uint2, uint4, uint8, float, double	Значение, которое нужно преобразовать к строке.



Формат вывода %f, который используется по умолчанию, имеет точность 6 знаков после запятой. Если нужно преобразовывать числа в строку с большей точностью, укажите формат вывода с большей точностью. Например, %.10f – выводить дробные числа с точностью 10 знаков после запятой.

## Возвращаемое значение

Тип – string.

Строковое представление указанного значения с учетом системного языка. Логические значения преобразуются в строки true и false вне зависимости от системного языка.

Строковое представление значения x:

- если x имеет тип bool – возвращается строка true или false.
- если x – число, возвращается строковое представление этого числа в формате format с учётом системного языка.

## Примеры



```
a: string = String.ToLocalizedString(7.15);  
// Результат:  
//- "7.15", если системный язык - английский;  
//- "7,15", если системный язык - русский.
```

# ToLower

Возвращает строку, в которой все символы приведены к нижнему регистру.



```
string String.ToLower(string x)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка

## Возвращаемое значение

Тип – string.

Строка x, в которой все символы приведены к нижнему регистру.

## Примеры



```
a: string = String.ToLower("ALARM!"); // Результат: "alarm!"
```

# ToString

Возвращает строковое представление указанного значения.



```
string String.ToString(int8 x)
string String.ToString(uint8 x)
string String.ToString(double x)
string String.ToString(string format, int8 x)
string String.ToString(string format, uint8 x)
string String.ToString(string format, double x)
string String.ToString(bool x)
```

## Параметры

Параметр	Тип	Описание
format	string	Опциональный параметр Правило преобразования чисел в строку в формате printf: <a href="https://en.wikipedia.org/wiki/Printf_format_string">https://en.wikipedia.org/wiki/Printf_format_string</a> Если не указан, используется значение %f
x	int8, uint8, double, bool	Значение, которое нужно преобразовать к строке

## Обратите внимание



Формат вывода %f, который используется по умолчанию, имеет точность 6 знаков после запятой. Если нужно преобразовывать числа в строку с большей точностью, укажите формат вывода с большей точностью. Например, %.10f – выводить дробные числа с точностью 10 знаков после запятой.

## Возвращаемое значение

Тип – string.

Строковое представление значения x:

- если x имеет тип bool – возвращается строка true или false.
- если x – число, возвращается строковое представление этого числа в формате format.

## Примеры



//Преобразование целых чисел:

a: string = String.ToString(777); // Результат: "777"

b: string = String.ToString("%05f", 12); // Результат: "00012"

c: string = String.ToString("%05f", -12); // Результат: "-0012"



//Преобразование вещественных чисел:

a: string = String.ToString(1.1); // Результат: "1.100000"

b: string = String.ToString(-123.0e-2); // Результат: "-1.230000"

c: string = String.ToString("%e", 0.001); // Результат:

"1.000000e-03" (экспоненциальная форма записи)

d: string = String.ToString(0.123456789); // Результат:

"0.123457" (значение округляется до 6 знаков после запятой)

e: string = String.ToString("%.10f", 0.123456789); // Результат:

"0.1234567890" (значение выводится с точностью 10 знаков после запятой)



//Преобразование логических значений:

a: string = String.ToString(true); // Результат: "true"

# ToUint1

Конвертирует указанную строку в число типа uint1.



```
uint1 String.ToUint1(string x, uint1 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
defValue	uint1	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – uint1.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: uint2 = String.ToUint1("100", 0); // Результат: 100  
b: uint2 = String.ToUint1("not number", 0); // Результат: 0
```



# ToUint2

Конвертирует указанную строку в число типа uint2.



```
uint2 String.ToUint2(string x, uint2 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
defValue	uint2	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – uint2.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: uint2 = String.ToUint2("100", 0); // Результат: 100  
b: uint2 = String.ToUint2("not number", 0); // Результат: 0
```

# ToUint4

Конвертирует указанную строку в число типа uint4.



```
uint4 String.ToUint4(string x, uint4 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
defValue	uint4	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – uint4.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: uint4 = String.ToUint4("100", 0); // Результат: 100  
b: uint4 = String.ToUint4("not number", 0); // Результат: 0
```

# ToUint8

Конвертирует указанную строку в число типа uint8.



```
uint8 String.ToUint8(string x, uint8 defValue)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка
defValue	uint8	Значение, которое вернется, если конвертация будет неуспешна

## Возвращаемое значение

Тип – uint8.

Число, полученное из строки x, или defValue, если конвертировать строку в число нельзя.

## Примеры



```
a: uint8 = String.ToUint8("100", 0); // Результат: 100  
b: uint8 = String.ToUint8("not number", 0); // Результат: 0
```

# ToUpper

Возвращает строку, в которой все символы приведены к верхнему регистру.



```
string String.ToUpper(string x)
```

## Параметры

Параметр	Тип	Описание
x	string	Исходная строка

## Возвращаемое значение

Тип – string.

Строка x, в которой все символы приведены к верхнему регистру.

## Примеры



```
a: string = String.ToUpper("alarm!"); // Результат: "ALARM!"
```

# Trim

Возвращает строку, в которой удалены пробелы в начале и в конце строки.



```
string String.Trim(string X)
```

## Параметры

Параметр	Тип	Описание
X	string	Исходная строка

## Возвращаемое значение

Тип: string.

Строка X, в которой удалены все пробелы в начале и в конце строки.

## Примеры



```
a: string = String.Trim(" Warning! "); // Результат: "Warning!"
```

# Функции типа variant



Вызов внутренних функций типа variant осуществляется обращением к пространству имен Variant.

Функция	Описание
<a href="#">Add</a>	Возвращает сумму двух вариантов
<a href="#">And</a>	Возвращает побитовое И двух вариантов
<a href="#">DEC</a>	Возвращает значение варианта, уменьшенное на 1
<a href="#">Div</a>	Возвращает частное двух вариантов
<a href="#">EQ</a>	Сравнивает два варианта аналогично оператору "равно"
<a href="#">From&lt;T&gt;</a>	Группа функций, которые создают значение типа variant из значения элементарного типа
<a href="#">GE</a>	Сравнивает два варианта аналогично оператору "больше или равно"
<a href="#">GT</a>	Сравнивает два варианта аналогично оператору "больше"
<a href="#">INC</a>	Возвращает значение варианта, увеличенное на 1
<a href="#">INV</a>	Возвращает побитовую инверсию варианта
<a href="#">Is&lt;T&gt;</a>	Группа функций, которые проверяют, относится ли значение типа variant к элементарному типу
<a href="#">IsEmpty</a>	Проверяет, является ли значение типа variant пустым
<a href="#">LE</a>	Сравнивает два варианта аналогично оператору "меньше или равно"
<a href="#">LogicalAnd</a>	Возвращает логическое И двух вариантов
<a href="#">LogicalOr</a>	Возвращает логическое ИЛИ двух вариантов
<a href="#">LT</a>	Сравнивает два варианта аналогично оператору "меньше"
<a href="#">MayConvertTo&lt;T&gt;</a>	Группа функций, которые проверяют, можно ли конвертировать значение типа variant в элементарный тип
<a href="#">Mul</a>	Возвращает произведение двух вариантов

<a href="#"><u>NE</u></a>	Сравнивает два варианта аналогично оператору "не равно"
<a href="#"><u>NEG</u></a>	Возвращает значение указанного варианта с противоположным знаком
<a href="#"><u>Not</u></a>	Возвращает логическое отрицание указанного варианта
<a href="#"><u>Or</u></a>	Возвращает побитовое ИЛИ двух вариантов
<a href="#"><u>Sub</u></a>	Возвращает разность двух вариантов
<a href="#"><u>To&lt;T&gt;</u></a>	Группа функций, которые для значения типа variant возвращают ее значение, конвертированное в элементарный тип
<a href="#"><u>Xor</u></a>	Возвращает побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ двух вариантов

# Add

Возвращает сумму двух вариантов.



```
variant Variant.Add(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первое слагаемое
y	variant	Второе слагаемое

## Возвращаемое значение

Тип – variant.

Сумма x и y. Если типы значений несовместимы, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Сумма двух вариантов с целыми значениями  
x: variant = 40;  
y: variant = 10;  
res: variant = Variant.Add(x, y); //Результат: 50
```



```
//Сумма двух вариантов с несовместимыми значениями  
x: variant = 40;  
y: variant = "10";  
res: variant = Variant.Add(x, y); // Результат: VT_EMPTY
```



# And

Возвращает побитовое "И" двух вариантов.



```
variant Variant.And(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант. Должен иметь целочисленное значение
y	variant	Второй вариант. Должен иметь целочисленное значение

## Возвращаемое значение

Тип – variant.

Побитовое И параметров x и y. Если любой из параметров содержит не целочисленное значение, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Побитовое И двух вариантов с целочисленными значениями  
x: variant = 12; //В двоичной записи - 1100  
y: variant = 21; //В двоичной записи - 10101  
res: variant = Variant.And(x, y); //Результат: 4 (в двоичной записи -  
100)
```



```
//Побитовое И двух вариантов с несовместимыми значениями  
x: variant = 12;
```

```
y: variant = 1.0;
```

```
res: variant = Variant.And(x, y); //Результат: VT_EMPTY
```

# DEC

Возвращает значение варианта, уменьшенное на 1.



variant Variant.DEC(variant x)

## Параметры

Параметр	Тип	Описание
x	variant	Вариант (должен иметь целочисленное значение)

## Возвращаемое значение

Тип – variant.

Значение x, уменьшенное на 1. Если x имеет не целочисленный тип, вернется пустое значение (VT\_EMPTY).

## Примеры



```
x: variant = 15;  
res: variant = Variant.DEC(x); //Результат: 14
```



```
x: variant = 2.5;  
res: variant = Variant.DEC(x); //Результат: VT_EMPTY
```

# Div

Возвращает частное двух вариантов.



```
variant Variant.Div(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Делимое
y	variant	Делитель

## Возвращаемое значение

Тип – variant.

Частное x и y. Если значения целочисленные, вернется целая часть от деления. Если типы значений несовместимы, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Частное двух вариантов с целыми значениями  
x: variant = 45;  
y: variant = 10;  
res: variant = Variant.Div(x, y); //Результат: 4
```



```
//Частное двух вариантов с вещественными значениями  
x: variant = 45.0;  
y: variant = 10.0;  
res: variant = Variant.Div(x, y); //Результат: 4.5
```



//Частное двух вариантов с несовместимыми значениями

```
x: variant = 40;
```

```
y: variant = "10";
```

```
res: variant = Variant.Div(x, y); //Результат: VT_EMPTY
```

# EQ

Сравнивает два варианта аналогично оператору "равно".



```
variant Variant.EQ(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант
y	variant	Второй вариант

## Возвращаемое значение

Тип – variant.

Результат сравнения `x == y`:

- › true – если операция истинна для значений.
- › false – если операция ложна для значений.
- › VT\_EMPTY – если значения несравнимы или один из операндов – пустое значение (VT\_EMPTY).

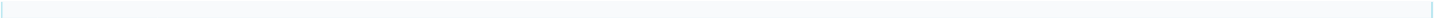
## Примеры



```
x: variant = 40;  
y: variant = 40.0;  
res: variant = Variant.EQ(x, y); //Результат: true
```



```
x: variant = 40;  
y: variant = "40";  
res: variant = Variant.EQ(x, y); //Результат: VT_EMPTY
```



# From<T>

Группа функций, которые создают значение типа `variant` из значения элементарного типа явным способом.



```
variant Variant.FromInt1(int1 x)
variant Variant.FromInt2(int2 x)
variant Variant.FromInt4(int4 x)
variant Variant.FromInt8(int8 x)
variant Variant.FromUInt1(uint1 x)
variant Variant.FromUInt2(uint2 x)
variant Variant.FromUInt4(uint4 x)
variant Variant.FromUInt8(uint8 x)
variant Variant.FromFloat(float x)
variant Variant.FromDouble(double x)
variant Variant.FromBool(bool x)
variant Variant.FromString(string x)
```

## Параметры

Параметр	Тип	Описание
<code>x</code>	все кроме <code>variant</code>	Значение, которое будет присвоено варианту

## Возвращаемое значение

Тип – `variant`.

Вариант (значение типа `variant`), имеющий значение `x`.

## Примеры





```
//Создать переменную типа variant из переменной элементарного  
типа uint1
```

```
X1: uint1 = 100;
```

```
V1: variant = Variant.FromUInt1(X1);
```



```
//Создать переменную типа variant из переменной элементарного  
типа int4
```

```
X2: int4 = 12300655;
```

```
V2: variant = Variant.FromInt4(X2);
```



```
//Создать переменную типа variant из переменной  
элементарного типа double
```

```
X3: double = 345.543333;
```

```
V3: variant = Variant.FromDouble(X3);
```



```
//Создать переменную типа variant из переменной  
элементарного типа string:
```

```
X4: string = "my string";
```

```
V4: variant = Variant.FromString(X4);
```



```
//Создать переменную типа variant из переменной  
элементарного типа bool
```

```
X5: bool = false;
```

```
V5: variant = Variant.FromBool(X5);
```

# GE

Сравнивает два варианта аналогично оператору "больше или равно".



```
variant Variant.GE(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант
y	variant	Второй вариант

## Возвращаемое значение

Тип – variant.

Результат сравнения  $x \geq y$ :

- › true – если операция истинна для значений.
- › false – если операция ложна для значений.
- › VT\_EMPTY – если значения несравнимы или один из операндов – пустое значение (VT\_EMPTY).

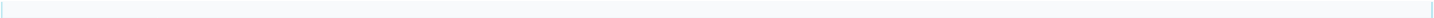
## Примеры



```
x: variant = 40;  
y: variant = 40.0;  
res: variant = Variant.GE(x, y); // Результат: true
```



```
x: variant = 40;  
y: variant = "40";  
res: variant = Variant.GE(x, y); // Результат: VT_EMPTY
```



# GT

Сравнивает два варианта аналогично оператору "больше".



```
variant Variant.GT(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант
y	variant	Второй вариант

## Возвращаемое значение

Тип – variant.

Результат сравнения  $x > y$ :

- › true – если операция истинна для значений.
- › false – если операция ложна для значений.
- › VT\_EMPTY – если значения несравнимы или один из операндов – пустое значение (VT\_EMPTY).

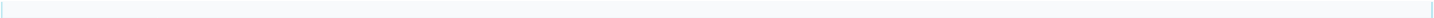
## Примеры



```
x: variant = 40;  
y: variant = 40.0;  
res: variant = Variant.GT(x, y); // Результат: false
```



```
x: variant = 40;  
y: variant = "40";  
res: variant = Variant.GT(x, y); // Результат: VT_EMPTY
```



# INC

Возвращает значение варианта, увеличенное на 1.



variant Variant.INC(variant x)

## Параметры

Параметр	Тип	Описание
x	variant	Вариант. Должен иметь целочисленное значение

## Возвращаемое значение

Тип – variant.

Значение x, увеличенное на 1. Если x имеет не целочисленный тип, вернется пустое значение (VT\_EMPTY).

## Примеры



```
x: variant = 15;  
res: variant = Variant.INC(x); // Результат: 16
```



```
x: variant = 2.5;  
res: variant = Variant.INC(x); // Результат: VT_EMPTY
```

# INV

Возвращает побитовую инверсию варианта.



variant Variant.INV(variant x)

## Параметры

Параметр	Тип	Описание
x	variant	Вариант (должен иметь беззнаковое целое значение)

## Возвращаемое значение

Тип – variant.

Побитовая инверсия варианта x. Если x имеет не беззнаковый целый тип, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Побитовая инверсия беззнакового целого значения:  
x: variant = Variant.FromUInt1(99); // В двоичной записи - 01100011  
res: variant = Variant.INV(x); // Результат: 156 (в двоичной записи -  
10011100)
```



```
//Побитовая инверсия не беззнакового целого значения:  
x: variant = 1.15;  
res: variant = Variant.INV(x); // Результат: VT_EMPTY
```

# Is<T>

Группа функций для проверки типа значения в переменной `variant`.



```
bool Variant.IsInt1(variant x)
bool Variant.IsInt2(variant x)
bool Variant.IsInt4(variant x)
bool Variant.IsInt8(variant x)
bool Variant.IsUInt1(variant x)
bool Variant.IsUInt2(variant x)
bool Variant.IsUInt4(variant x)
bool Variant.IsUInt8(variant x)
bool Variant.IsFloat(variant x)
bool Variant.IsDouble(variant x)
bool Variant.IsBool(variant x)
bool Variant.IsString(variant x)
```

## Параметры

Параметр	Тип	Описание
<code>x</code>	<code>variant</code>	Вариант, который будем проверять

## Возвращаемое значение

Тип – `bool`.

`true` – если значение варианта `x` является значением элементарного типа; `false` – если не является.

## Примеры





```
//Проверить соответствует ли переменная типу uint1  
X1: variant = 25;  
Result1: bool = Variant.IsUint1(X1); // Результат: true
```



```
//Проверить, соответствует ли переменная типу float  
X2: variant = "my string";  
Result2: bool = Variant.IsFloat(X2); // Результат: false
```

# IsEmpty

Проверяет, является ли значение типа variant пустым.



```
bool Variant.IsEmpty(variant x)
```

## Параметры

Параметр	Тип	Описание
x	variant	Вариант, который будем проверять

## Возвращаемое значение

Тип – bool.

- › true – если значение варианта x пустое (VT\_EMPTY);
- › false – если не пустое.

## Примеры



```
x: variant; // переменная типа variant без инициализации  
(VT_EMPTY)  
Result2: bool = Varinat.IsEmpty(x); // Результат: true
```

# LE

Сравнивает два варианта аналогично оператору "меньше или равно".



```
variant Variant.LE(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант
y	variant	Второй вариант

## Возвращаемое значение

Тип – variant.

Результат сравнения  $x \leq y$ :

- › true – если операция истинна для значений.
- › false – если операция ложна для значений.
- › VT\_EMPTY – если значения несравнимы или один из операндов – пустое значение (VT\_EMPTY).

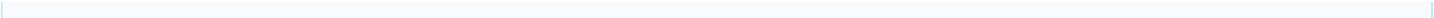
## Примеры



```
x: variant = 40;  
y: variant = 40.0;  
res: variant = Variant.LE(x, y); // Результат: true
```



```
x: variant = 40;  
y: variant = "40";  
res: variant = Variant.LE(x, y); // Результат: VT_EMPTY
```



# LogicalAnd

Возвращает логическое И двух вариантов.



```
variant Variant.LogicalAnd(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант. Должен иметь значение типа bool
y	variant	Второй вариант. Должен иметь значение типа bool

## Возвращаемое значение

Тип – variant.

Логическое И параметров x и y. Если любой из параметров содержит значение не типа bool, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Логическое И двух вариантов с логическими значениями  
x: variant = true;  
y: variant = false;  
res: variant = Variant.LogicalAnd(x, y); // Результат: false
```



```
//Логическое И двух вариантов с несовместимыми значениями  
x: variant = true;  
y: variant = 1;  
sum: variant = Variant.LogicalAnd(x, y); // Результат: VT_EMPTY
```

# LogicalOr

Возвращает логическое ИЛИ двух вариантов.



```
variant Variant.LogicalOr(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант. Должен иметь значение типа bool
y	variant	Второй вариант. Должен иметь значение типа bool

## Возвращаемое значение

Тип – variant.

Логическое ИЛИ параметров x и y. Если любой из параметров содержит значение не типа bool, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Логическое ИЛИ двух вариантов с логическими значениями  
x: variant = true;  
y: variant = false;  
res: variant = Variant.LogicalOr(x, y);  
// Результат: true
```



```
//Логическое ИЛИ двух вариантов с несовместимыми значениями  
x: variant = true;  
y: variant = 1;  
res: variant = Variant.LogicalOr(x, y);
```

// Результат: VT\_EMPTY

# LT

Сравнивает два варианта аналогично оператору "меньше".



```
variant Variant.LT(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант
y	variant	Второй вариант

## Возвращаемое значение

Тип – variant.

Результат сравнения  $x < y$ :

- › true – если операция истинна для значений.
- › false – если операция ложна для значений.
- › VT\_EMPTY – если значения несравнимы или один из операндов – пустое значение (VT\_EMPTY).

## Примеры

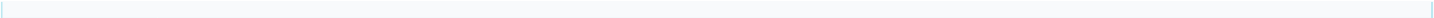


```
x: variant = 40;  
y: variant = 40.0;  
res: variant = Variant.LT(x, y); // Результат: false
```



```
x: variant = 40;  
y: variant = "40";  
res: variant = Variant.LT(x, y); // Результат: VT_EMPTY
```





# MayConvertTo<T>

Группа функций для проверки возможности конвертации типа `variant` в какой-либо элементарный тип данных.



```
uint1 Variant.MayConvertToInt1(variant x)
uint1 Variant.MayConvertToInt2(variant x)
uint1 Variant.MayConvertToInt4(variant x)
uint1 Variant.MayConvertToInt8(variant x)
uint1 Variant.MayConvertToUint1(variant x)
uint1 Variant.MayConvertToUint2(variant x)
uint1 Variant.MayConvertToUint4(variant x)
uint1 Variant.MayConvertToUint8(variant x)
uint1 Variant.MayConvertToFloat(variant x)
uint1 Variant.MayConvertToDouble(variant x)
uint1 Variant.MayConvertToBool(variant x)
uint1 Variant.MayConvertToString(variant x)
```

## Параметры

Параметр	Тип	Описание
<code>x</code>	<code>variant</code>	Значение, для которого будем проверять возможность конвертации

## Возвращаемое значение

Тип – `uint1`.

- › `0x00` (`MAY_CONVERTED`) – конвертация возможна.
- › `0x80` (`CAN_NOT_CONVERTED`) – конвертация невозможна.
- › `0x01` (`MAY_CONVERTED_WITH_LOSS`) – возможна конвертация с потерей точности.
- › `0x02` (`MAY_CONVERTED_WITH_CUT`) – возможна конвертация с обрезкой.

Если одновременно возможна конвертация с обрезкой и потерей точности, то вернется 0x02 – возможна конвертация с обрезкой.

## Примеры



```
//Проверяем, можно ли конвертировать значение в int1
V1: variant = 128;
ResultConvert: uint1 = Variant.MayConvertToInt1(V1); // Результат: 0x02
– возможна конвертация с обрезкой (до 127)
```



```
//Проверяем, можно ли конвертировать значение в int1:
V2: variant = 999.089f;
ResultConvert: uint1 = Variant.MayConvertToInt1(V2); //Результат: 0x01 -
возможна конвертация с потерей точности (до 100)
```



```
//Проверяем, можно ли конвертировать значение в string
V3: variant = "Моя строка";
ResultConvert: uint1 = Variant.MayConvertToString(V3);
// Результат: 0x00 - конвертация возможна
```



```
//Проверяем, можно ли конвертировать значение в uint1:
V3: variant = "Моя строка";
ResultConvert: uint1 = Variant.MayConvertToUint1(V3); // Результат:
0x80 – конвертация невозможна (строку нельзя конвертировать в
число)
```



```
//Проверяем, можно ли конвертировать значение в bool
V4: variant = true;
ResultConvert: uint1 = Variant.MayConvertToBool(V4); // Результат: 0x00
– конвертация возможна
```

# Mul

Возвращает произведение двух варинатов.



```
variant Variant.Mul(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый множитель
y	variant	Второй множитель

## Возвращаемое значение

Тип – variant.

Произведение x и y. Если типы значений несовместимы, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Произведение двух вариантов с целыми значениями  
x: variant = 40;  
y: variant = 10;  
res: variant = Variant.Mul(x, y); // Результат: 400
```



```
//Произведение двух вариантов с несовместимыми значениями  
x: variant = 40;  
y: variant = "10";  
res: variant = Variant.Mul(x, y); // Результат: VT_EMPTY
```

# NE

Сравнивает два варианта аналогично оператору "не равно".



```
variant Variant.NE(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант
y	variant	Второй вариант

## Возвращаемое значение

Тип – variant.

Результат сравнения  $x \neq y$ :

- › true – если операция истинна для значений.
- › false – если операция ложна для значений.
- › VT\_EMPTY – если значения несравнимы или один из операндов – пустое значение (VT\_EMPTY).

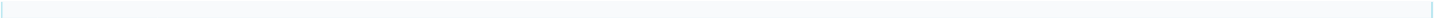
## Примеры



```
x: variant = 40;  
y: variant = 40.0;  
res: variant = Variant.NE(x, y); // Результат: false
```



```
x: variant = 40;  
y: variant = "40";  
res: variant = Variant.NE(x, y); // Результат: VT_EMPTY
```



# NEG

Возвращает значение указанного варианта с противоположным знаком.



variant Variant.NEG(variant x)

## Параметры

Параметр	Тип	Описание
x	variant	Вариант. Должен иметь числовое значение

## Возвращаемое значение

Тип – variant.

Значение x с противоположным знаком. Если x имеет не числовой тип, вернется пустое значение (VT\_EMPTY).

## Примеры



```
x: variant = 0.1;  
res: variant = Variant.NEG(x); // Результат: -0.1
```



```
x: variant = true;  
res: variant = Variant.NEG(x); // Результат: VT_EMPTY
```



# NOT

Возвращает логическое отрицание указанного варианта.



```
variant Variant.Not(variant x)
```

## Параметры

Параметр	Тип	Описание
x	variant	Вариант. Должен иметь значение типа bool

## Возвращаемое значение

Тип – variant.

Логическое отрицание варианта x. Если x имеет тип значения не bool, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Логическое отрицание варианта с логическим значением  
x: variant = true;  
res: variant = Variant.NOT(x); // Результат: false
```



```
//Логическое отрицание варианта не с логическим значением  
x: variant = 1;  
res: variant = Variant.NOT(x); // Результат: VT_EMPTY
```

# Or

Возвращает побитовое ИЛИ двух вариантов.



```
variant Variant.Or(variant x, variant y)
```

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант. Должен иметь целочисленное значение
y	variant	Второй вариант. Должен иметь целочисленное значение

## Возвращаемое значение

Тип – variant.

Побитовое ИЛИ параметров x и y. Если любой из параметров содержит не целочисленное значение, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Побитовое ИЛИ двух вариантов с целочисленными значениями  
x: variant = 12; // В двоичной записи - 1100  
y: variant = 21; // В двоичной записи - 10101  
res: variant = Variant.Or(x, y); // Результат: 29 (в двоичной записи -  
11101)
```



```
//Побитовое ИЛИ двух вариантов с несовместимыми значениями  
x: variant = 12;
```

```
y: variant = 1.0;
```

```
res: variant = Variant.Or(x, y); // Результат: VT_EMPTY
```

# Sub

Возвращает разность двух вариантов.



variant Variant.Sub(variant x, variant y)

## Параметры

Параметр	Тип	Описание
x	variant	Уменьшаемое
y	variant	Вычитаемое

## Возвращаемое значение

Тип – variant.

Разность  $x - y$ . Если типы значений несовместимы, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Разность двух вариантов с целыми значениями  
x: variant = 40;  
y: variant = 10;  
res: variant = Variant.Sub(x, y); // Результат: 30
```



```
//Разность двух вариантов с несовместимыми значениями  
x: variant = 40;  
y: variant = "10";  
res: variant = Variant.Sub(x, y); // Результат: VT_EMPTY
```

## To<T>

Группа функций для конвертации типа `variant` в один из элементарных типов данных.



```
int1 Variant.ToInt1(variant x, int1 def_value)
int2 Variant.ToInt2(variant x, int2 def_value)
int4 Variant.ToInt4(variant x, int4 def_value)
int8 Variant.ToInt8(variant x, int8 def_value)
uint1 Variant.ToUInt1(variant x, uint1 def_value)
uint2 Variant.ToUInt2(variant x, uint2 def_value)
uint4 Variant.ToUInt4(variant x, uint4 def_value)
uint8 Variant.ToUInt8(variant x, uint8 def_value)
float Variant.ToFloat(variant x, float def_value)
double Variant.ToDouble(variant x, double def_value)
bool Variant.ToBool(variant x, bool def_value)
string Variant.ToString(variant x, string def_value)
```

## Параметры

Параметр	Тип	Описание
<code>x</code>	<code>variant</code>	Вариант, который нужно конвертировать
<code>def_value</code>	все кроме <code>variant</code>	Значение, которое вернется, если конвертировать не получится

## Возвращаемое значение

Тип – зависит от функции.

Значение варианта `x`, приведенное к элементарному типу; `def_value` – если конвертировать не получится.

## Примеры



//Конвертировать переменную типа variant в переменную элементарного типа uint1

V1: variant = 120;

Result1: uint1 = Variant.ToUint1(V1, 0);



//Конвертировать переменную типа variant в переменную типа float

V2: variant = 999.089f;

Result1: float = Variant.ToFloat(V2, 0.0f);



//Конвертировать переменную типа variant в переменную типа string

V3: variant = "Моя строка";

Result2: string = Variant.ToString(V3, "");



//Конвертировать переменную типа variant в переменную типа bool

V4: variant = true;

Result3: bool = Variant.ToBool(V4, false);

# Xor

Возвращает побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ двух вариантов.



variant Variant.Xor(variant x, variant y)

## Параметры

Параметр	Тип	Описание
x	variant	Первый вариант. Должен иметь целочисленное значение
y	variant	Второй вариант. Должен иметь целочисленное значение

## Возвращаемое значение

Тип: variant.

Побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ параметров x и y. Если любой из параметров содержит не целочисленное значение, вернется пустое значение (VT\_EMPTY).

## Примеры



```
//Побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ двух вариантов с  
целочисленными значениями  
x: variant = 12; // В двоичной записи - 1100  
y: variant = 21; // В двоичной записи - 10101  
res: variant = Variant.Xor(x, y); // Результат: 25 (в двоичной записи -  
11001)
```



//Побитовое ИЛИ двух вариантов с несовместимыми значениями

x: variant = 12;

y: variant = 1.0;

res: variant = Variant.Xor(x, y); // Результат: VT\_EMPTY



## 1.4. Командная разработка

Проект хранится в виде отдельных папок и файлов. Данная структура хранения даёт возможность нескольким разработчикам одновременно работать над проектом. Рабочая копия каждого файла, расположенная на локальном компьютере, редактируется независимо от других и помещается в единое дерево файлов. В любое время есть возможность открыть ранее сохранённую версию файла. Корректировки и дополнения, внесённые разными разработчиками, могут объединяться и вноситься в одну из версий того или иного файла.



Для командной разработки проекта можно использовать [TortoiseSVN](#).

## 1.5. Решение проблем

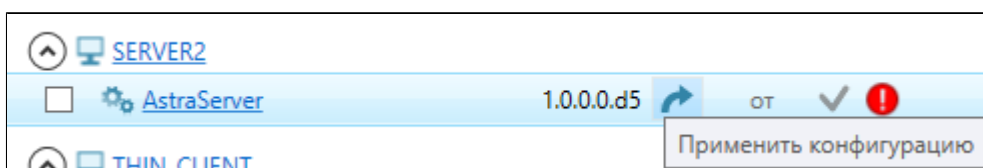
В данном разделе представлено решение наиболее часто встречающихся проблем:

[Не применяются атрибуты при развертывании конфигурации на серверах](#)  
[Пустые сообщения о деактивации в Astra.HMI.Alarms](#)

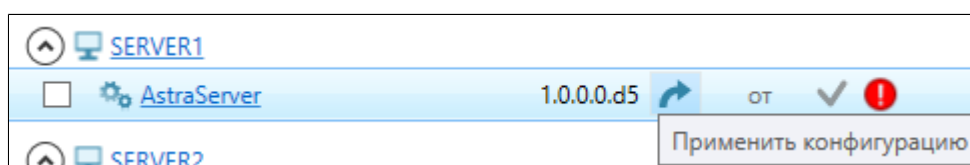
## 1.5.1. Не применяются атрибуты при развертывании конфигурации на серверах

Для безударного применения атрибутов при развертывании конфигурации на серверах необходимо выполнить следующие действия:

1. Обновить конфигурацию на резервном сервере:

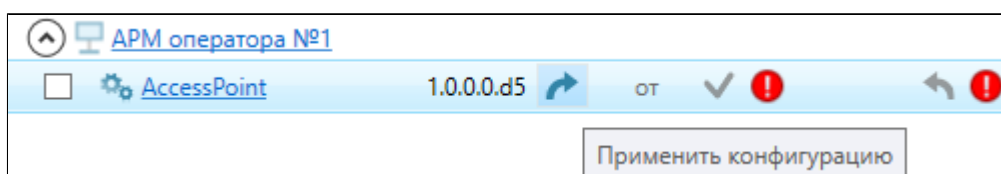


2. Обновить конфигурацию на основном сервере:

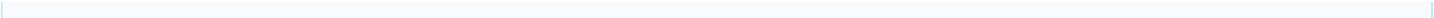


На серверах значения атрибутов изменится

Для того, чтобы на АРМ оператора отображались актуальные атрибуты необходимо также обновить конфигурацию AccessPoint:



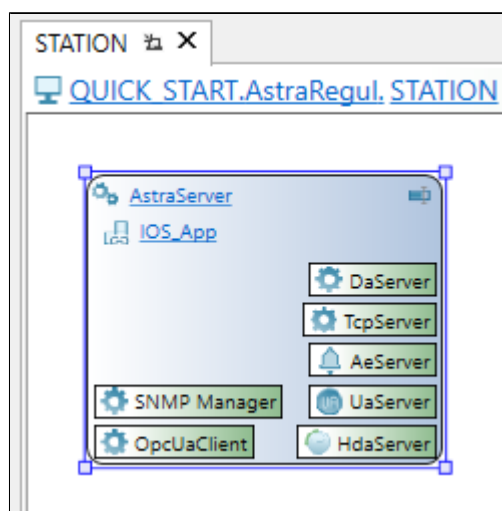
Обновление конфигурации Astra.AccessPoint приведет к краткосрочной потере связи между АРМ и серверами, но удара не будет, так как основной и резервный сервера сконфигурированы и работают. Для полного безударного перехода рекомендуется использовать дополнительный АРМ оператора.



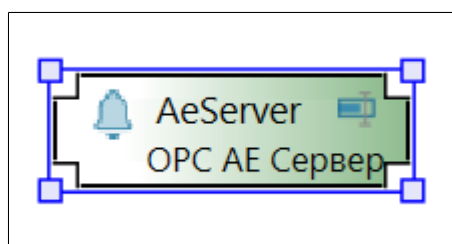
## 1.5.2. Пустые сообщения о деактивации в Astra.HMI.Alarms

Чтобы в журнале не формировались пустые сообщения со временем деактивации события необходимо выполнить следующие действия:

1. Перейдите в компонент "Astra.Server".



2. Нажмите на элемент "OPC AE Сервер".



3. В окне свойств необходимо настроить следующие параметры:

- › Использовать расширение OPC AE: Да.
- › Сохранять события в оперативной таблице событий: Да.
- › Папка для хранения оперативных событий: укажите путь до папки, в которую необходимо сохранять оперативные события.

Свойства	
AeServer OPC AE Сервер	
<b>Общие</b>	
Идентификатор сервера	Astra.OPCAEServer
Имя	AeServer
<b>Параметры модуля</b>	
Правило обработки подавленных событий	На стороне сервера
Отсылать квитирование по подавленным источникам	Нет
Использовать расширения OPC AE	Да
Уведомления о блокировании/подавлении	Отключены
Максимальный размер очереди уведомлений	100000
Передавать события по OPC DA протоколу	Нет
Шаблон сообщения события	
Активность	Да
Отображаемое имя	
<b>Таблица оперативных событий</b>	
Сохранять события в оперативной таблице событий	Да
Папка для хранения оперативных событий	C:\events
Ограничение по давности цепей событий, час	0
Ограничение по количеству цепей событий	0



После настройки необходимо перезапустить Astra.Server.