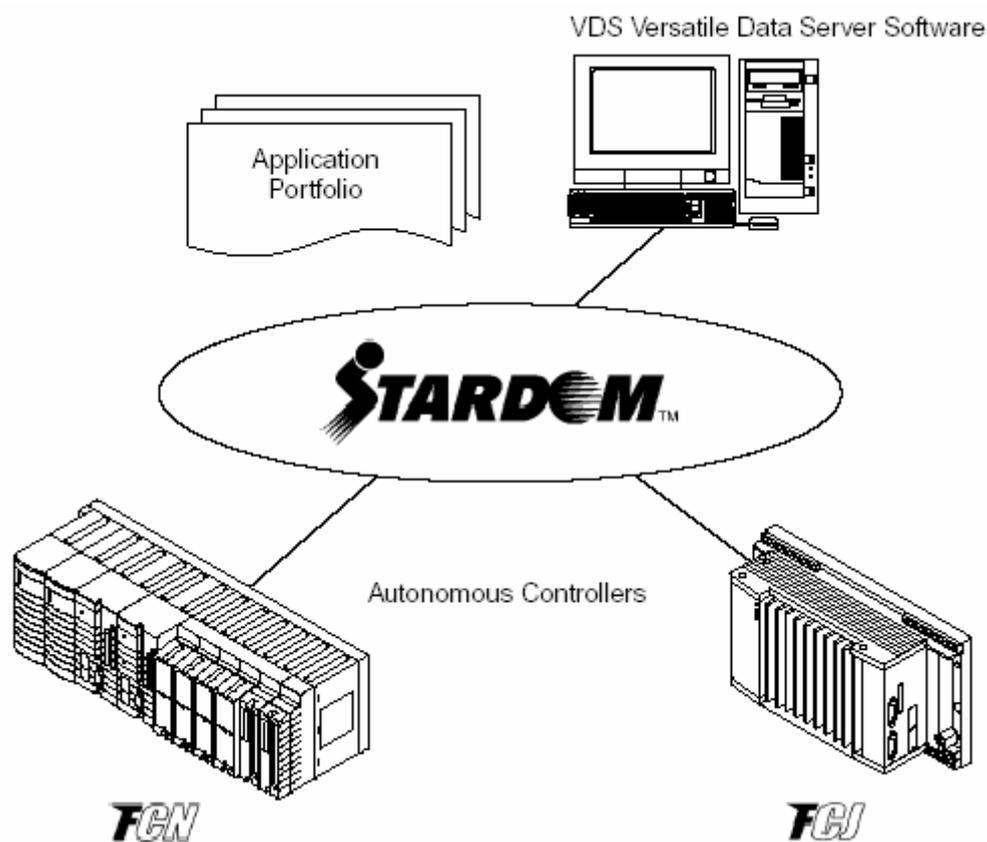


ЧАСТЬ 3



ФУНКЦИИ VDS



СОДЕРЖАНИЕ

| | |
|--|----|
| СОДЕРЖАНИЕ..... | 3 |
| 3.1 Введение..... | 5 |
| 3.1.1 Обзор функций..... | 5 |
| 3.1.2 Структура VDS..... | 7 |
| 3.2 Общие установки..... | 9 |
| 3.2.1 Инсталляция..... | 9 |
| 3.2.2 Учётные записи пользователя..... | 10 |
| 3.2.3 Пусковое системное меню VDS..... | 11 |
| 3.2.4 Подключение коммуникаций..... | 12 |
| 3.2.5 Управление лицензиями (License Management)..... | 14 |
| 3.2.6 Конфигурирование системы..... | 15 |
| 3.2.7 Средство сохранения конфигурации (Backup Tool)..... | 17 |
| 3.3 Построитель объектов (Object Builder)..... | 18 |
| 3.3.1 Использование Object Builder..... | 18 |
| 3.3.2 Импортирование базы данных тегов (Tag Database) FCX..... | 19 |
| 3.3.2.1 Имена реализаций объектов (Instance Names)..... | 19 |
| 3.3.2.2 Классы объектов (Object Classes)..... | 19 |
| 3.3.2.3 Функция импорта объекта..... | 20 |
| 3.3.2.4 Запись в переключатели (Switches)..... | 21 |
| 3.3.3 Создание управляющих групп (Control Groups)..... | 23 |
| 3.3.4 Создание управляющих объектов (Control Object)..... | 24 |
| 3.3.5 Создание объектов ввода/вывода (I/O Object)..... | 25 |
| 3.3.6 Запуск функций сервера данных (Data Server)..... | 26 |
| 3.3.7 Процедуры..... | 27 |
| 3.3.7.1 Импортирование и модификация базы данных тегов (Tag Database)..... | 27 |
| 3.3.7.2 Запуск функций VDS..... | 30 |
| 3.4 Графический редактор (Graphic Builder)..... | 32 |
| 3.4.1 Основные характеристики..... | 32 |
| 3.4.2 Примитивы (Primitives)..... | 33 |
| 3.4.3 Графические объекты (Display Objects)..... | 34 |
| 3.4.4 Функциональные графические объекты (Function Objects)..... | 35 |
| 3.4.5 Модифицирование поведения графических объектов (Graphic Modify)..... | 38 |
| 3.4.6 Библиотечные примитивы (Linked Parts)..... | 40 |
| 3.4.7 Настройка источника данных (Datasource)..... | 43 |
| 3.4.8 HMI Deployment Tool..... | 44 |
| 3.5 Функция архивирования Истории (Historian) и Тренды (Trend)..... | 46 |
| 3.5.1 Структура архива Истории..... | 46 |
| 3.5.2 Функция архивирования Истории..... | 47 |
| 3.5.3 Процедуры..... | 49 |
| 3.6 Функция генерирования Рапортов (Report)..... | 51 |
| 3.6.1 Общий обзор..... | 51 |
| 3.6.2 Функции Рапорта..... | 52 |
| 3.6.3 Сбор данных (Data Collection) и функция обработки (Processing Function)..... | 54 |
| 3.6.4 Функции событийных рапортов (Lot, Batch)..... | 55 |
| 3.6.5 Выбор данных для рапорта..... | 57 |

| | | |
|-------|--|----|
| 3.6.6 | Конфигурирование процесса закрытия рапорта..... | 58 |
| 3.6.7 | Установки для функции генерирования рапортов..... | 59 |
| 3.7 | Функция сообщений (Message)..... | 60 |
| 3.8 | Функция защиты от неправомерного доступа..... | 63 |
| 3.9 | Клиент серверные функции HMI..... | 67 |
| 3.9.1 | Конфигурирование HMI Server..... | 67 |
| 3.9.2 | Функция операторского окна (Operator Display)..... | 69 |

3.1 Введение.

3.1.1 Обзор функций.

Универсальный сервер данных (**Versatile Data Server**) поддерживает две основных функции:

- Функции сервера данных (**Data Server**);
- Функции операторского интерфейса (**HMI Server**).

Основной функцией Data Server является чтение данных из управляющих устройств, таких как FCX или FA-M3 PLC и сохранять эти данные в форме управляющими объектами (**Control Objects**). Эти данные могут быть доступны функциям HMI Server или другим приложениям через OPC интерфейс.

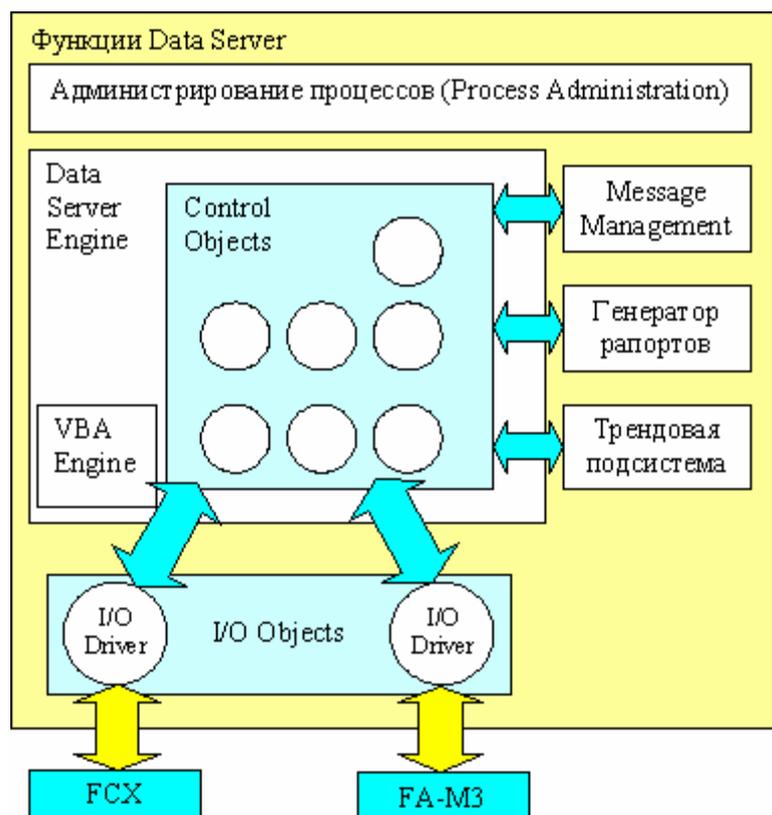
HMI Server читает данные из **Data Server** используя OPC и обслуживает операторский интерфейс в виде страниц Web клиента, т.е. в PC запущен Microsoft Internet Explorer.

Функции VDS:

| | |
|----------------------------|--|
| Функции HMI Client | Обзорные объектные показания (Object View Display) |
| | Графические окна (мнемосхемы) (Graphic Window Display) |
| | Защита от неправомерного доступа (Log On) |
| | Трендовые показания (Trend Displays) |
| | Показания сообщений (Message Display) |
| | Голосовые сообщения (Audible Message) |
| Функции HMI Server | Web сервер (Web Server) |
| | Управление сессиями (Session Management) |
| | Доступ к данным (Data Access) |
| Функции Data Server | Сервер данных (Data Server) с управляющими объектами (Control Objects) |
| | Драйверы ввода/вывода (I/O Drivers) - FCN/FCJ - FA-M3 - OPC Server |
| | Управление сообщениями (Message Management) |
| | Управление задачами (Task Management) |
| | База данных (Database (ISAM)) |
| | Управление защитой (Security Management) |
| | |
| | |

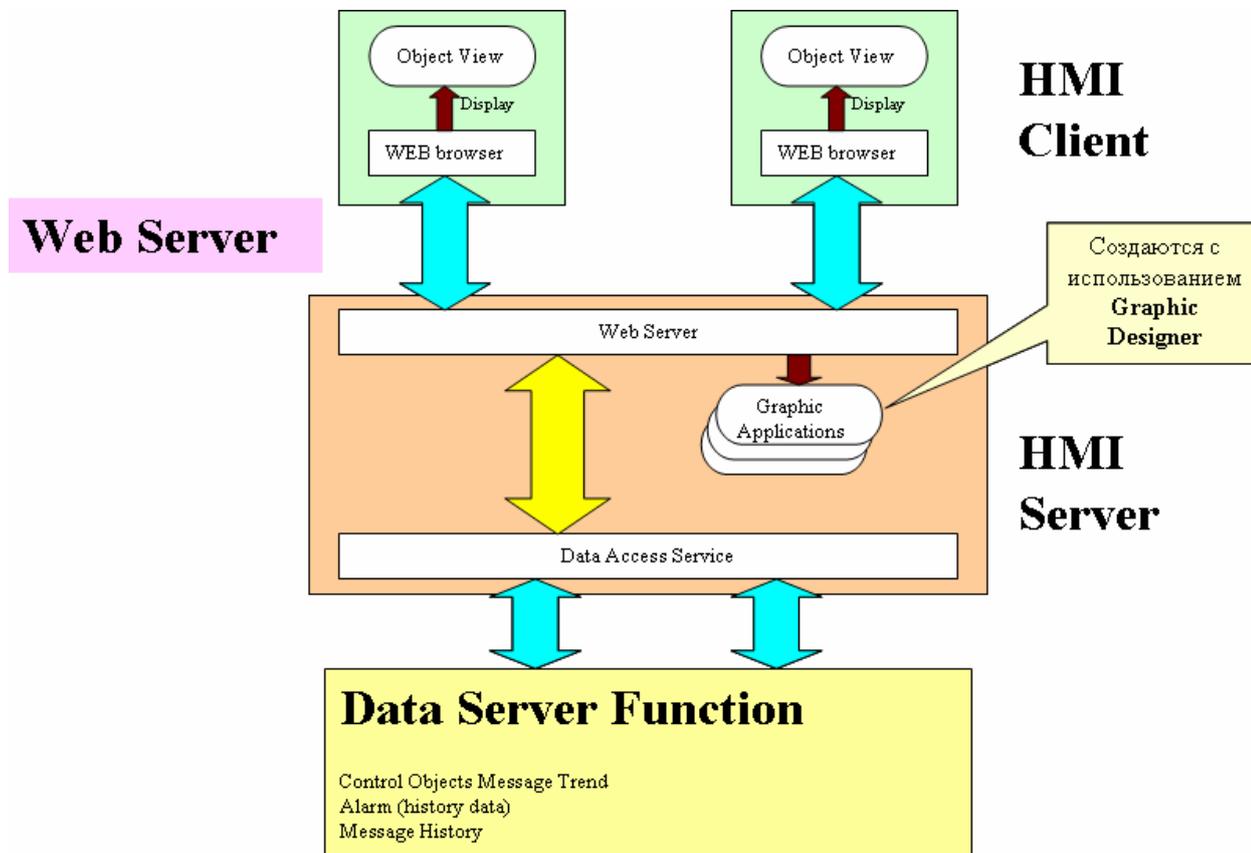
3.1.2 Структура VDS.

Data Server включает в себя драйвер ввода/вывода полевого устройства с использованием объектов ввода/вывода (**I/O Objects**), обработчик сервера (**Data Server Engine**) использующий управляющие объекты (**Control Objects**), и OPC сервер (**OPC Server**), который делает данные доступными со стороны сервера операторского интерфейса (**HMI server**). Обработчик VBA (**VBA Engine**) обеспечивает доступ к управляющим объектам со стороны программ написанных на Visual Basic.



Система конфигурируется с помощью построителя объектов (**Object Builder**).

HMI Server включает в себя сервис доступа к данным (**Data Access Service**) осуществляющий чтение данных из **Data Server** и **Web Server**, обслуживающий графические приложения (**Graphic Applications**) клиентов операторского интерфейса (**HMI Clients**).



3.2 Общие установки.

3.2.1 Инсталляция.

Процесс инсталляции подробно описан в части 1.

3.2.2 Учётные записи пользователя.

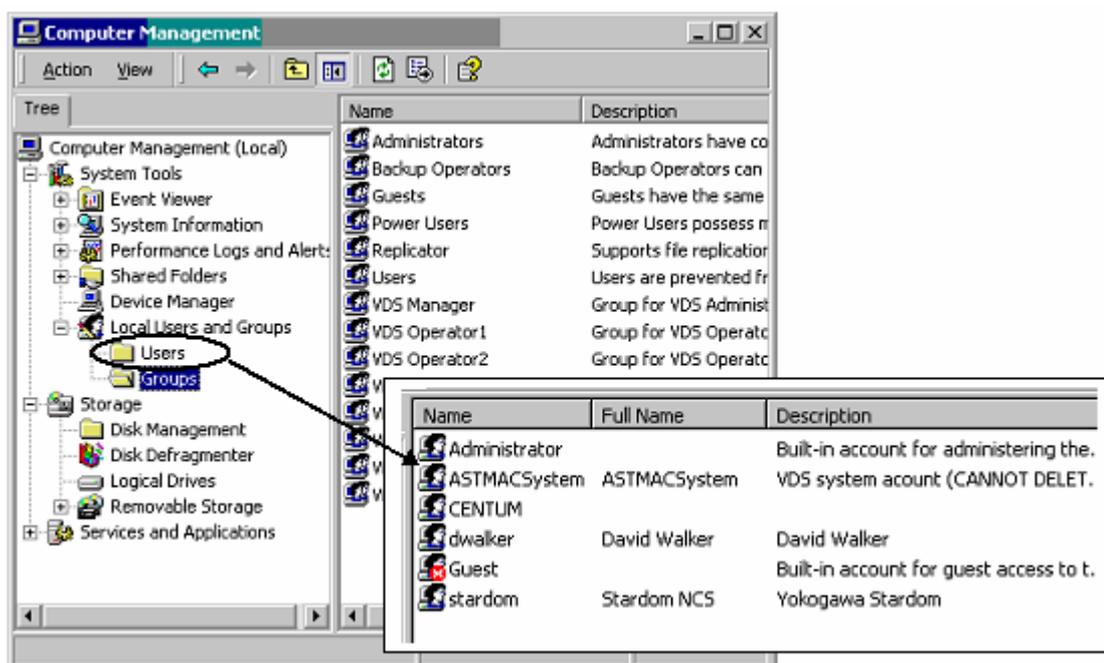
Для VDS должна быть установлена своя учётная запись. Рекомендуется, чтобы она называлась “stardom”. Эта учётная запись должна иметь привилегию уровня Administrator.

Когда программное обеспечение VDS установлено, в Windows создаются следующие группы:

- VDS Manager
- VDS Operator1 ~ 7

Их можно посмотреть в:

[Control Panel] → [Administrative Tools] → [Computer Management].



Для операторов, которые входят в систему, они могут быть присвоены как члены группы VDS Operator. Более подробно о доступе к конфигурации для членов группы VDS Operator см. раздел 3.8.

Система также создаёт пользователя с именем “ASTMSystem” для инсталляции. Он должен быть членом групп “Administrators” и “VDS Manager”. Никогда не модифицируйте этого пользователя и не используйте эту учётную запись в процессе обычной работы.

3.2.3 Пусковое системное меню VDS.

Все конфигурационные функции VDS запускаются через **System Launcher VDS**. Сам **System Launcher** запускается через кнопку “Пуск” (Start):

[Start] → [All Programs] → [Yokogawa VDS]

Или двойным щелчком по ярлыку “**Yokogawa VDS**” на рабочем столе:



Это приводит к появлению окна “**System Launcher**”:



Перечень инструментов размещённых в окне и их назначение следующие:

- **Development Builder** – содержит все функции для построения функций оперирования, таких как **Object Builder**, **Graphic Builder**;
- **Mode Change** – поддерживает переключения между режимами разработки, оперирования и выключения;
- **Support Tools** – содержит системные средства обслуживания;
- **System Configuration** – средства начального конфигурирования системы (см. раздел [3.2.6](#));
- **System Management** – для сохранения проектов и дополнительного лицензирования.

3.2.4 Подключение коммуникаций.

Установка IP адресов для PC подробно описана в разделе [2.2.4](#). Кратко, IP адресация в PC должна быть установлена следующим образом:

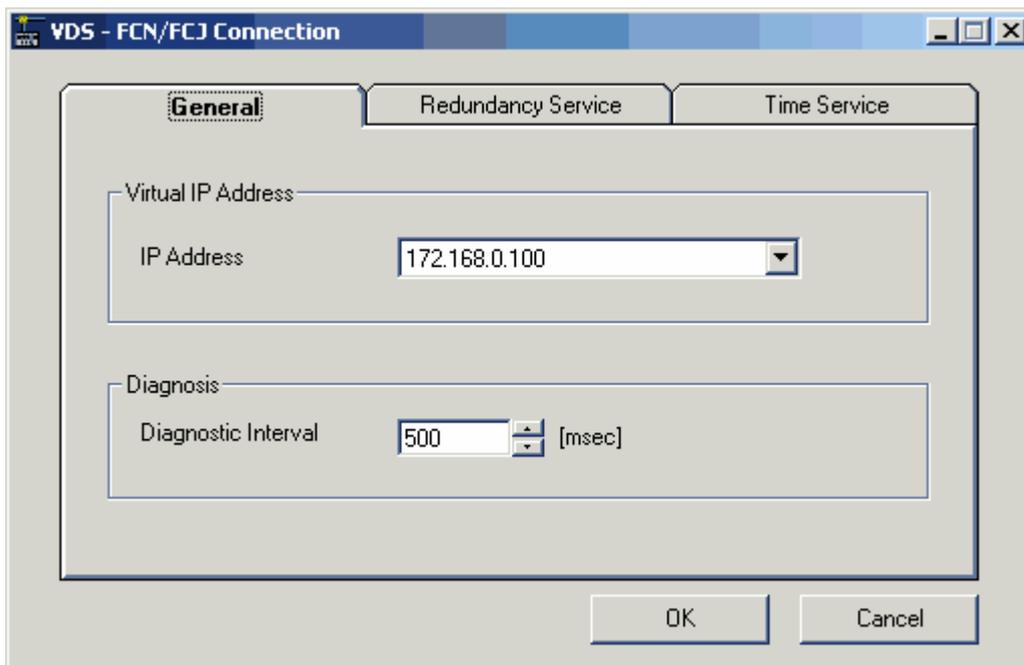
если Virtual IP адрес в FCX xxx.yyy.0.zzz, то устанавливаются следующие IP адреса для PC:

- xxx.yyy.0.nnn
- xxx.yyy.1.nnn
- xxx.yyy.2.nnn (если Ethernet дублирован)

Как только IP адресация установлена, должны быть сконфигурированы VDS – FCN **FCJ Connection Setting**. При этом для использования в VDS Virtual IP адрес устанавливается из вышеприведённых адресов.

Доступ через:

[System Launcher] → [System Configuration] → [FCN FCJ Connection Setting]



Для того чтобы сделать сеть резервированной выберите IP address to be set as a virtual IP адрес из выпадающего меню “**IP address**”. Эти установки не требуются, если сеть нерезервированна.

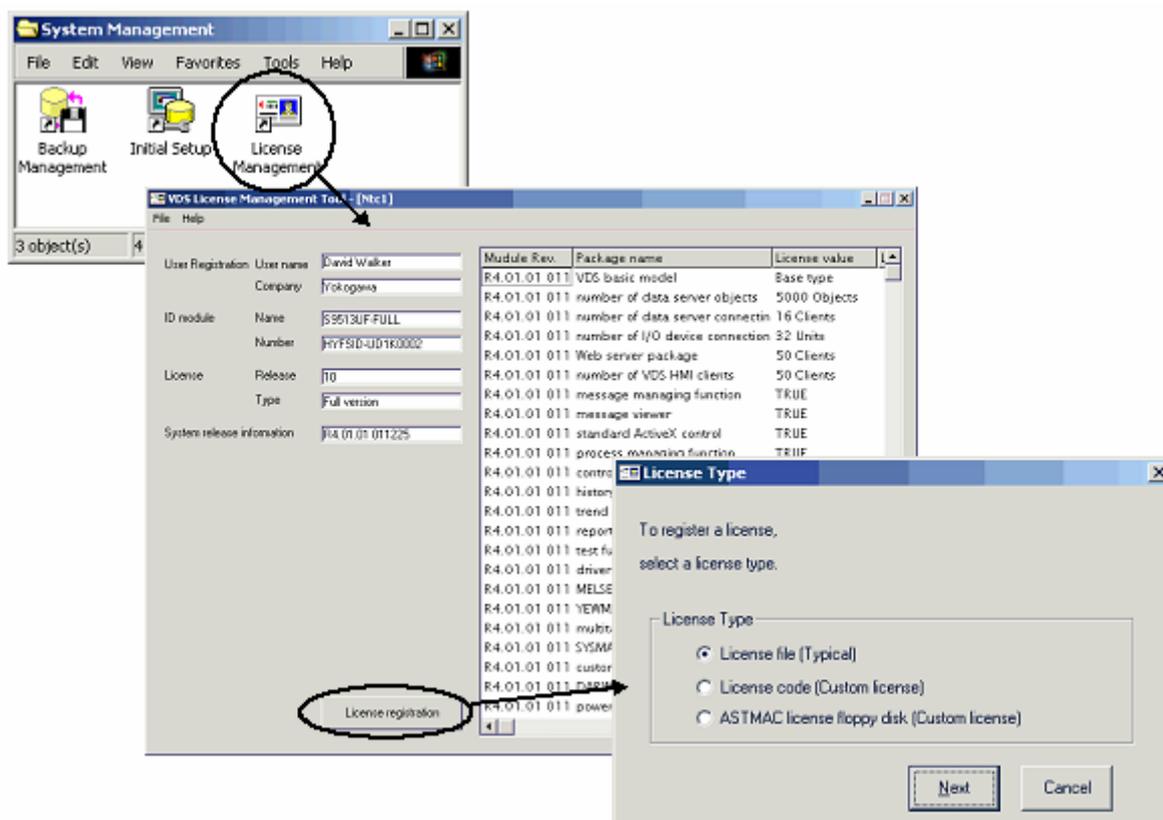
- **General** – диагностический интервал “**Diagnostic Interval**” это период проверки состояния сети. По умолчанию 500 мсек.
- **Redundancy Service** – если Ethernet резервирован откройте закладку “**Redundancy Service**” и установите разрешение обслуживания резервированной сети “**Enable Redundant Network Service**”.

-
- **Time Service** – этот VDS может быть использован как сервер временной синхронизации. В противном случае этот VDS может синхронизироваться от другого сервера.

После установки этих параметров PC должен быть перезагружен.

3.2.5 Управление лицензиями (License Management).

Инсталлированные лицензии регистрируются при помощи **License Manager**. Кроме того, он позволяет инсталлировать дополнительные лицензии при необходимости.

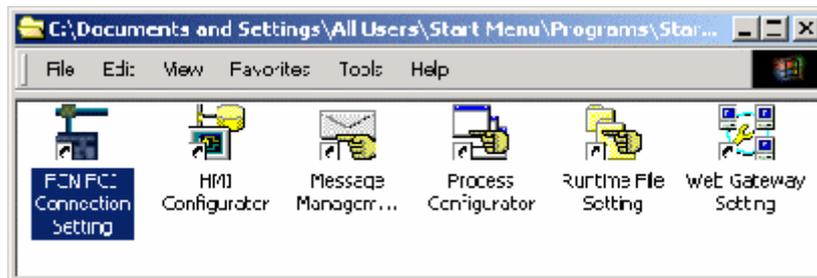


License Manager вызывается из **System Launcher** через **License Management**. При этом открывается окно “**VDS License Management Tool**”. На нём отображается текущая информация о лицензиях, включающая листинг всех загруженных программных пакетов.

Для регистрации дополнительной лицензии нажмите кнопку “**License Registration**”, при этом откроется новое окно “**License Type**”. Выберите необходимый тип лицензии (обычно “**License file**”) и нажмите кнопку “**Next**”. Выберите инсталлируемый файл.

3.2.6 Конфигурирование системы.

Для конфигурирования системы применяются следующие программы:



Runtime File Setting – используется для работы с файлами:

- **Data Server Configuration File** – выбирает исполняемый файл проекта (**Runtime Project**) (обычно Workspace.odw) и размещение рабочих файлов;
- **Message Configuration File** – выбирает исполняемый файл сообщений (**Runtime Message**) (обычно usermsg.ymg) и размещение рабочих файлов.

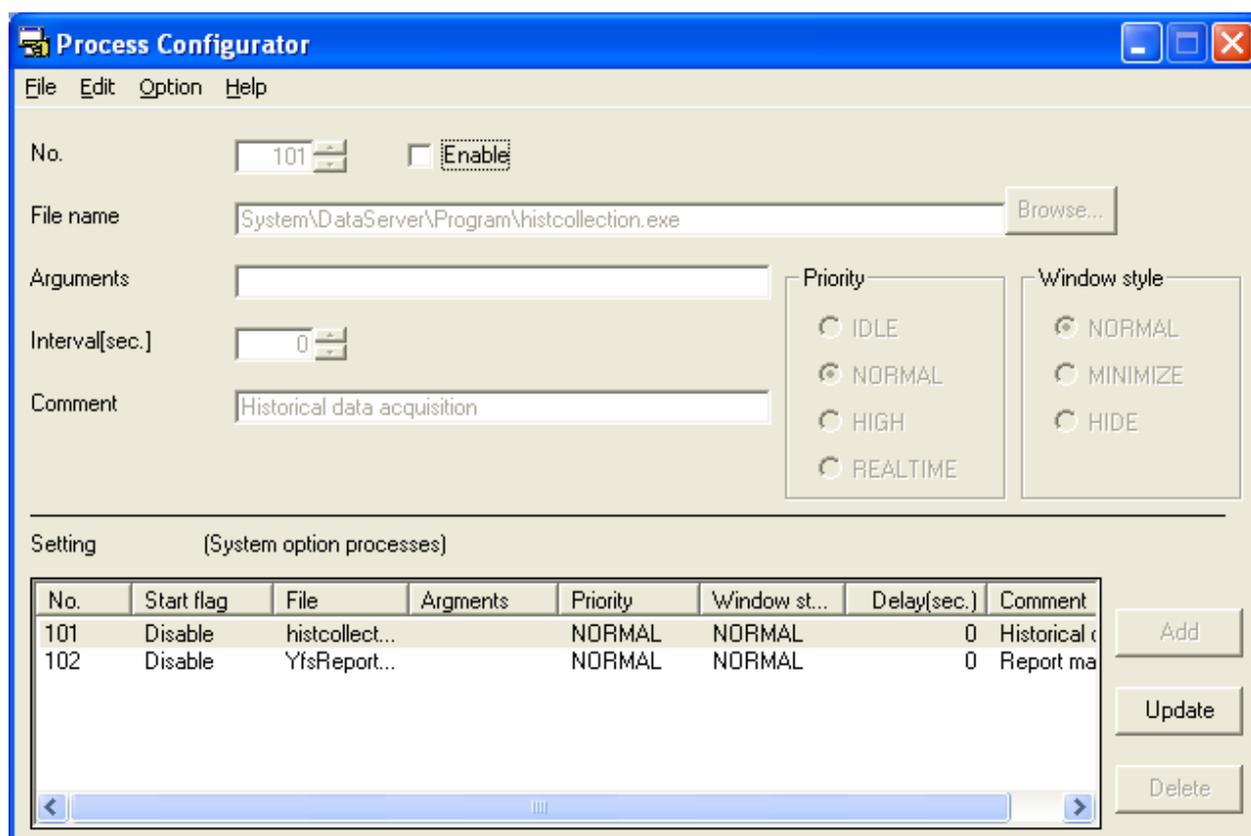
HMI Configurator – конфигуратор **HMI Server**. Более подробно см. раздел [3.9.1](#);

FCN FCJ Connection Setting – подключение коммуникаций **FCN FCJ**. Более подробно см. раздел [3.2.4](#);

Process Configurator – используется для регистрации опциональных пакетов (таких как **Trend Package**, **Report Package**) и приложений пользователя (таких как программы **Visual Basic**), которые используют базу данных **VDS**.

Как зарегистрировать функции Trend Package и Report Package:

Для того чтобы зарегистрировать функции Trend и Report, вызовите **Process Configurator**, далее выберите опцию “**System Option Processes**” в выпадающем меню “**Option**”.

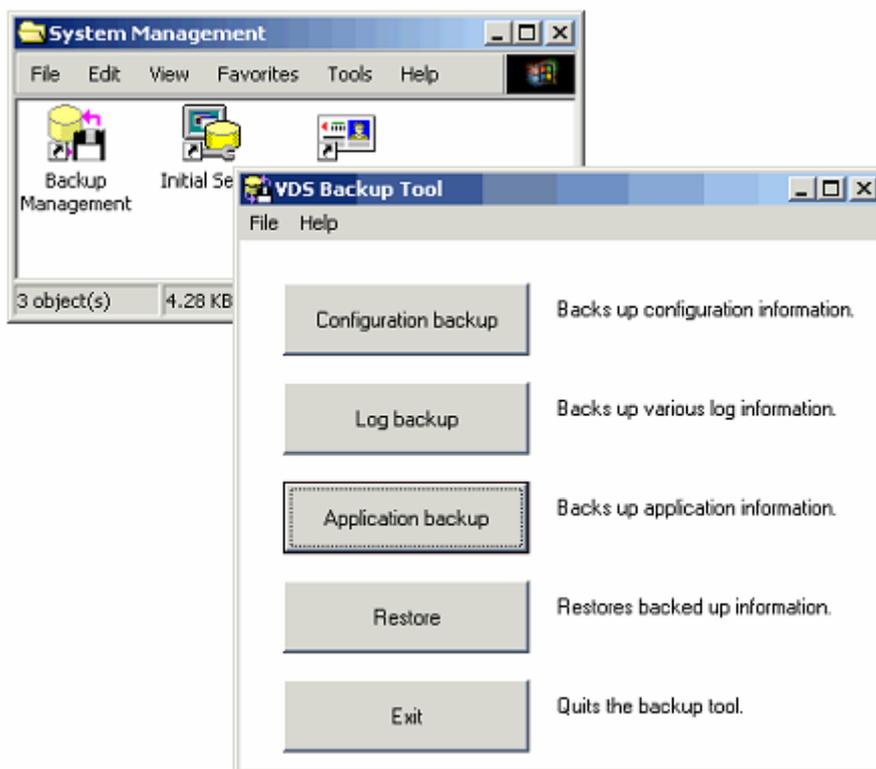


В этом окне могут быть изменены только параметры “**Start flag**”. Если необходимо их изменить с целью открытия возможности конфигурирования нажмите кнопку “**Update**”. Когда процесс регистрации закончится, произведите сохранение “**Save**” и выйдите “**Exit**”.

3.2.7 Средство сохранения конфигурации (Backup Tool).

Средство сохранения конфигурации (**Backup Tool**) позволяет извлечь конфигурационные параметры и сохранить их на диске. При этом может быть сохранена следующая информация:

- Конфигурационная информация (**Configuration Information**);
- Имя пользователя и пароль входа в систему (**Logs Information**);
- Информация о приложениях (**Application Information**).



3.3 Построитель объектов (Object Builder).

3.3.1 Использование Object Builder.

Object Builder привязывает теги в FCX, FAM3 и других устройствах (таких как: PLC сторонних производителей, OPC шлюзы и т.д.) к **Data Server**. **Object Builder** реализует следующие функции:

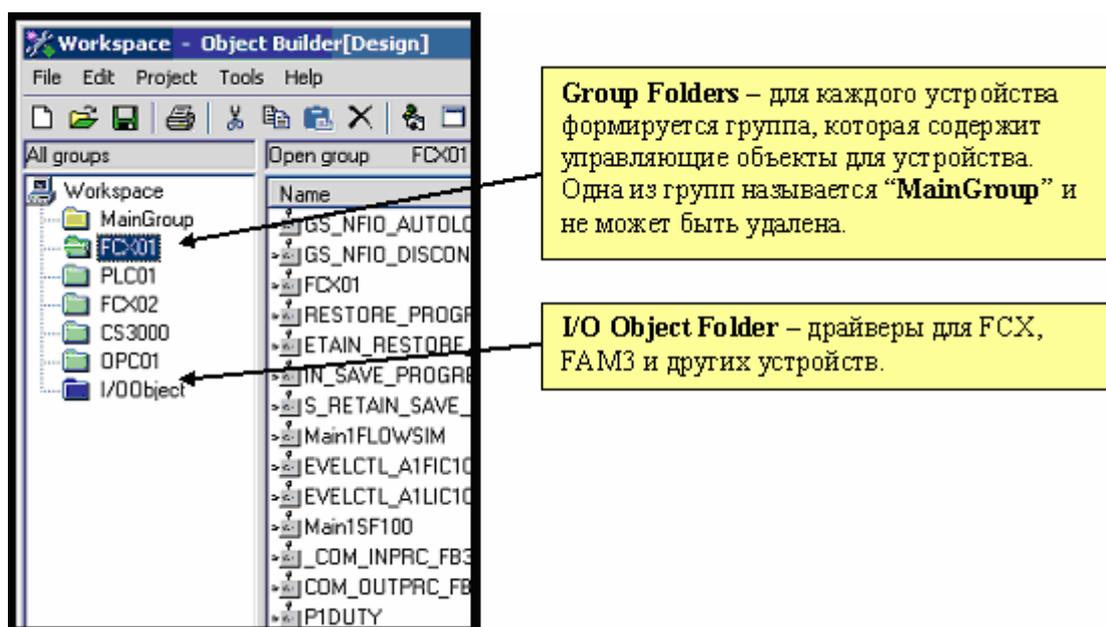
- Импорт и создание управляющих объектов (**Control Objects**) и объектов ввода/вывода (**I/O Objects**);
- Запуск и Останов (Run/Stop) VDS;
- Конфигурирование исторического архива (**History Builder**);
- Конфигурирование рапортов (**Report Builder**);
- Конфигурирование подсистемы защиты от неправомерного доступа (**Security Builder**).

Важнейшими функциями, обеспечивающими работу **Data Server**, являются:

- Привязка **Control Object** к переменной внутри FCX или другого устройства;
- **I/O Object** – это драйвер устройства, который специфицирует протокол доступа к ресурсам устройства, из которых переменные читаются или в которые они пишутся.

Для Stardom FCX **Object Builder** может автоматически генерировать **Control Objects**, используя средства импорта. Для других устройств **Control Objects** могут быть созданы вручную.

Object Builder имеет интерфейс типа explorer с папками в левой стороне окна, которые соответствуют компонентам системы:



3.3.2 Импортрование базы данных тегов (Tag Database) FCX.

3.3.2.1 Имена реализаций объектов (Instance Names).

Когда база данных FCX импортируется, **Object Builder** создаёт множество реализаций **Control Objects** из блоков PASPOU и присваивает им имена на основе имён реализаций переменных. Имя реализации **Control Object** строится по следующим правилам:

TaskNameFBNameVariableName.....

Пример 1:

Если тег “FIC100” присутствует в программе с именем задачи “Task1” (т.е. объектный адрес – Task1.FIC100), то имя **Control Object** будет:

Task1FIC100

Пример 2:

Если тег “FIC100” присутствует в функциональном блоке (**Function Block**) с именем “Level” в задаче “Task1” (т.е. объектный адрес – Task1.Level.FIC100), то имя **Control Object** будет:

Task1LevelFIC100

1. В частном случае именем переменной может быть имя функционального блока, как это было показано на примерах выше;
2. Максимальное количество поддерживаемых символов в имени объекта не превышает 16. Если оно превышено, имя урезается до 16 символов слева.
3. Если имя задачи начинается с символа подчёркивания “_”, то оно не будет включено в имя объекта. В примере 1, если бы имя задачи было бы “_Task1”, то имя объекта было бы “FIC100”.
4. Имя объекта может быть изменено в процессе импортирования, но однажды созданное, потом изменено быть не может.
5. Что может быть управляющим объектом (**Control Object**):
 - Управляющими объектами могут быть только функциональные блоки PASPOU;
 - Функциональные блоки, определённые пользователем, не могут быть управляющими объектами;
 - Переменные и переменные ввода/вывода могут быть управляющими объектами, но переменные должны создаваться вручную;
 - В то же время структуры данных (**Data Structures**) (такие как CData_Real) не могут быть управляющими объектами.

3.3.2.2 Классы объектов (Object Classes).

Control Objects могут принадлежать одному из четырёх классов:

| | |
|-----------------------------------|---|
| Device Tag Object | Вход/выход в устройстве (FCX или другом) |
| FCN/FCJ Object | Функциональный блок в FCX |
| Process Tag Object | Функциональный блок. Для подключения устройств сторонних производителей |
| Application Support Object | Объекты, связанные с ресурсами системы |

Примечание: В течение процесса импортирования класс объекта не может наблюдаться или изменяться.

Классы FCN/FCJ Object и Device Tag Object присваиваются автоматически. Однако если объект создаётся вручную, ему должен быть присвоен класс.

3.3.2.3 Функция импорта объекта.

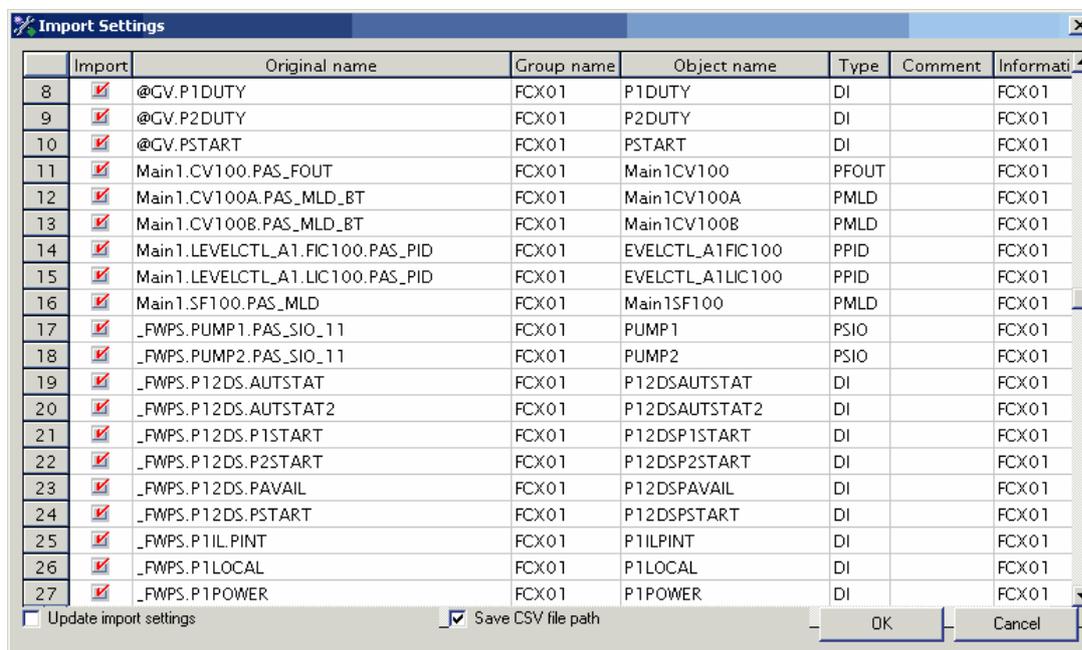
В течение процесса импорта построитель читает файлы “ADLST.csv”. Эти файлы находятся в проекте индивидуально для каждого FCX и находятся в папках:

\ProjectName\C\Configuration\R\FCXnn

Где: nn – номер FCX.

Эти файлы являются базами данных FCX и из них могут быть извлечены все данные необходимые для импортирования.

Таблица импортированных объектов:



| | Import | Original name | Group name | Object name | Type | Comment | Informati |
|----|-------------------------------------|----------------------------------|------------|------------------|-------|---------|-----------|
| 8 | <input checked="" type="checkbox"/> | @GV.P1DUTY | FCX01 | P1DUTY | DI | | FCX01 |
| 9 | <input checked="" type="checkbox"/> | @GV.P2DUTY | FCX01 | P2DUTY | DI | | FCX01 |
| 10 | <input checked="" type="checkbox"/> | @GV.PSTART | FCX01 | PSTART | DI | | FCX01 |
| 11 | <input checked="" type="checkbox"/> | Main1.CV100.PAS_FOUT | FCX01 | Main1CV100 | PFOUT | | FCX01 |
| 12 | <input checked="" type="checkbox"/> | Main1.CV100A.PAS_MLD_BT | FCX01 | Main1CV100A | PMLD | | FCX01 |
| 13 | <input checked="" type="checkbox"/> | Main1.CV100B.PAS_MLD_BT | FCX01 | Main1CV100B | PMLD | | FCX01 |
| 14 | <input checked="" type="checkbox"/> | Main1.LEVELCTL_A1.FIC100.PAS_PID | FCX01 | EVELCTL_A1FIC100 | PPID | | FCX01 |
| 15 | <input checked="" type="checkbox"/> | Main1.LEVELCTL_A1.LIC100.PAS_PID | FCX01 | EVELCTL_A1LIC100 | PPID | | FCX01 |
| 16 | <input checked="" type="checkbox"/> | Main1.SF100.PAS_MLD | FCX01 | Main1SF100 | PMLD | | FCX01 |
| 17 | <input checked="" type="checkbox"/> | _FWPS.PUMP1.PAS_SIO_11 | FCX01 | PUMP1 | PSIO | | FCX01 |
| 18 | <input checked="" type="checkbox"/> | _FWPS.PUMP2.PAS_SIO_11 | FCX01 | PUMP2 | PSIO | | FCX01 |
| 19 | <input checked="" type="checkbox"/> | _FWPS.P12DS.AUTSTAT | FCX01 | P12DSAUTSTAT | DI | | FCX01 |
| 20 | <input checked="" type="checkbox"/> | _FWPS.P12DS.AUTSTAT2 | FCX01 | P12DSAUTSTAT2 | DI | | FCX01 |
| 21 | <input checked="" type="checkbox"/> | _FWPS.P12DS.P1START | FCX01 | P12DSP1START | DI | | FCX01 |
| 22 | <input checked="" type="checkbox"/> | _FWPS.P12DS.P2START | FCX01 | P12DSP2START | DI | | FCX01 |
| 23 | <input checked="" type="checkbox"/> | _FWPS.P12DS.PAVAIL | FCX01 | P12DSPAVAIL | DI | | FCX01 |
| 24 | <input checked="" type="checkbox"/> | _FWPS.P12DS.PSTART | FCX01 | P12DSPSTART | DI | | FCX01 |
| 25 | <input checked="" type="checkbox"/> | _FWPS.P11L.PINT | FCX01 | P11LPINT | DI | | FCX01 |
| 26 | <input checked="" type="checkbox"/> | _FWPS.P1LOCAL | FCX01 | P1LOCAL | DI | | FCX01 |
| 27 | <input checked="" type="checkbox"/> | _FWPS.P1POWER | FCX01 | P1POWER | DI | | FCX01 |

Update import settings Save CSV file path

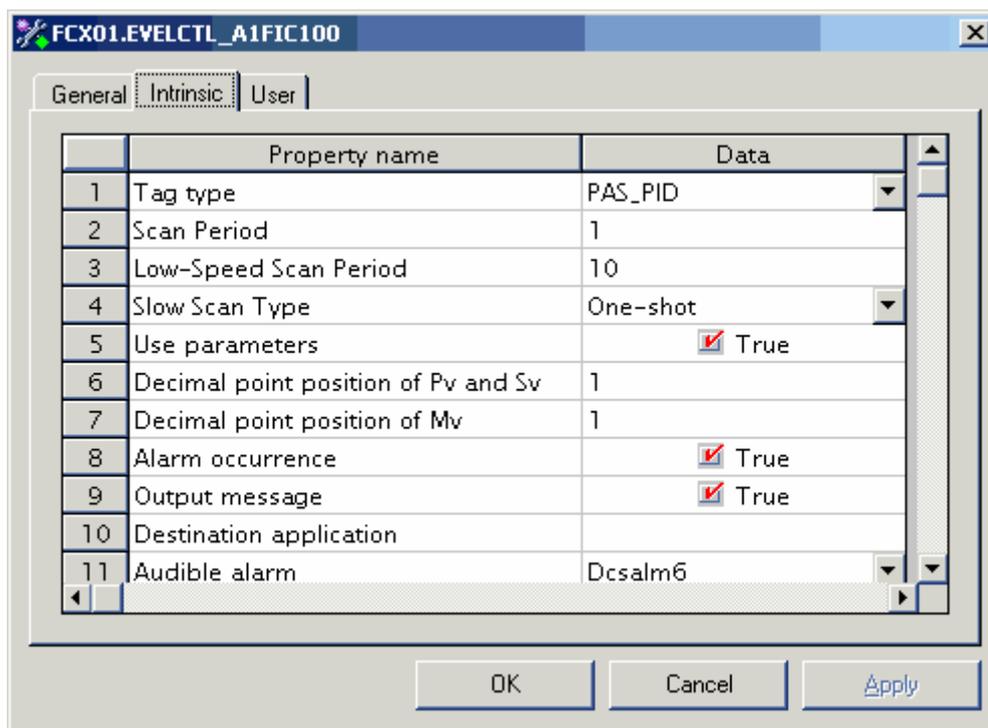
При импорте следующие параметры могут быть модифицированы:

- Import** – выбирает надо ли этот объект импортировать;
- Group Name** – выбирает в какую группу импортировать объект;

- **Object Name** – имя объекта создаётся автоматически, как было описано выше. Однако оно может быть изменено. После завершения импортирования изменить его уже нельзя;
- **Type** – тип объекта, связан с типом функционального блока или типом ввода/вывода. Более подробно см. ниже;
- **Comment** – комментарий к объекту, может быть введён. Комментарий отображается в **Object Viewer** и в модуле показаний HMI;
- **Using VBA** (на рисунке не показан) – выбирается используется ли VBA. Этот параметр распространяется на все объекты, принадлежащие одной группе, и при изменении хотя бы одного из них, автоматически изменяются все остальные.

Примечание: Только переменные, маркированные как OPC тип в Logic Designer, будут выбраны при импортировании. Это применимо к переменным, но не к функциональным блокам.

Однажды созданный объект, может быть модифицирован, для этого необходимо дважды щёлкнуть по его имени. Однако имя объекта не может быть изменено. Адрес и другие атрибуты можно изменять. Эти атрибуты могут быть найдены в закладке “**Intrinsic**” свойств **Control Object**.



3.3.2.4 Запись в переключатели (Switches).

Многие переменные в FCX (или других устройствах) являются внутренними переключателями или флагами. При импортировании они автоматически переобозначаются в DI, т.е. дискретные входы. Это означает, что для **Data Server** они доступны только для чтения и не могут управляться дистанционно с HMI.

Для разрешения записи в переключатели для каждого управляющего объекта вручную должен быть создан другой управляющий объект обозначенный как DO. Другой способ

заключается в том, что управляющий объект должен быть переобозначен как DR, который обеспечивает доступ к объекту, как по чтению, так и по записи.

Например, имя переключателя P1DUTY будет иметь два Control Objects связанных с ним:

P1DUTY – Type: DI (Class = Device Tag)

P1DUTY_DO – Type: DO (or DR) (Class = Device Tag)

В мнемосхеме разрабатываемой для HMI кнопка или другой переключающий объект должен писать в управляющий объект P1DUTY_DO.

3.3.3 Создание управляющих групп (Control Groups).

При импортировании базы данных FCX управляющая группа (**Control Group**) создаётся автоматически. Именем управляющей группы становится имя ресурса присвоенное ему в Logic Designer (обычно FCX01, FCX02 и т.д.).

В течение процесса импортирования можно перенести управляющий объект в другую управляющую группу. Кроме того, вручную созданные управляющие объекты могут быть привязаны к нескольким управляющим группам одновременно.

Управляющие группы должны создаваться до создания или импортирования объектов.

3.3.4 Создание управляющих объектов (Control Object).

В некоторых случаях, например, таких как описано в предыдущем пункте или при необходимости привязки устройств сторонних производителей, не имеющих утилиты для импортирования объектов, применяется способ ручного создания управляющих объектов.

Управляющий объект может быть создан в любой управляющей группе (папке), и связан с любым объектом ввода/вывода (драйвер), а также OPC тегами, PLC, и т.д.

Данные, которые должны быть указаны, следующие:

- **Name** – имя Control Object;
- **Class** – класс объекта;
- **Type** – тип объекта.

Информация об объекте ввода/вывода:

- **Name** – имя **I/O Object** (драйвер устройства);
- **Address** – имя переменной в устройстве источнике;
- **Scanning Method** – метод сканирования, с которым **Control Object** читает/пишет в переменную в устройстве источнике.

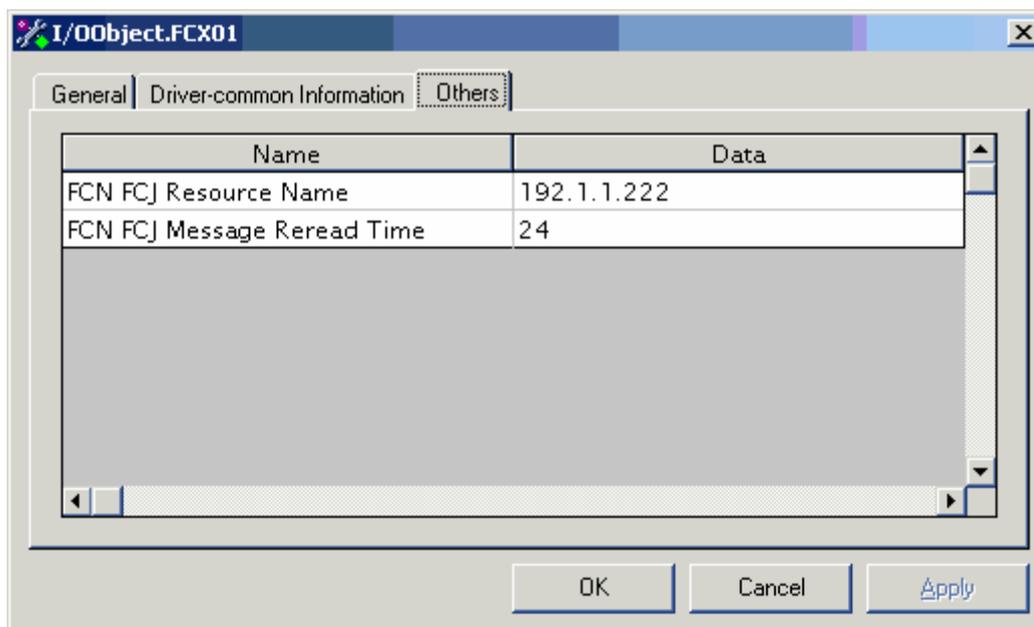
Дополнительно в закладке “**Intrinsic**” могут быть введены значения атрибутов объекта, таких как период сканирования, параметры алармов и т.д.

3.3.5 Создание объектов ввода/вывода (I/O Object).

Объект ввода/вывода (**I/O Object**) является драйвером полевого устройства. В систему включены следующие объекты:

- Stardom Controller (FCN/FCJ)
- FA-M3 PLC (Ethernet, Serial, Modem)
- OPC
- MELSEC (Ethernet and Serial)
- SYSMAC
- EZSocket
- DARWIN
- DAQSTATION
- M&C Power Monitor

Когда база данных FCX контроллера импортируется, **I/O Object** создаются автоматически. Свойства объекта могут быть просмотрены после двойного щелчка по нему.



По закладке “**Others**” убедитесь в том, что для “**FCN FCJ Resource Name**” правильно установлен IP адрес. По умолчанию именем I/O Object является имя ресурса (Resource Name) (например FCX01), но как только имя специфицируется в файле “**HOSTS**” оно должно быть заменено вместе с IP адресом.

Это необходимо по той причине что файл “**HOSTS**” в Windows является поисковой таблицей, которая связывает имя хоста и его IP адрес.

Размещен файл в папке:

Windows\System32\drivers\etc

3.3.6 Запуск функций сервера данных (Data Server)

Object Builder функционирует также, как и управляющая панель для оперирования с **Data Server**.

Data Server может работать в трёх режимах:

- Design/Development
- Debugging
- Runtime/Operation

Режим устанавливается через выпадающее меню **Object Builder** “Tools” или через ярлык Stardom в системной панели задач.

Заметим, что утилита системной панели задач поддерживает переключения между режимами Development и Operation, но не поддерживает Debugging. Это переключение можно произвести только через **Object Builder** и только тогда, когда утилита системной панели задач находится в режиме Development.

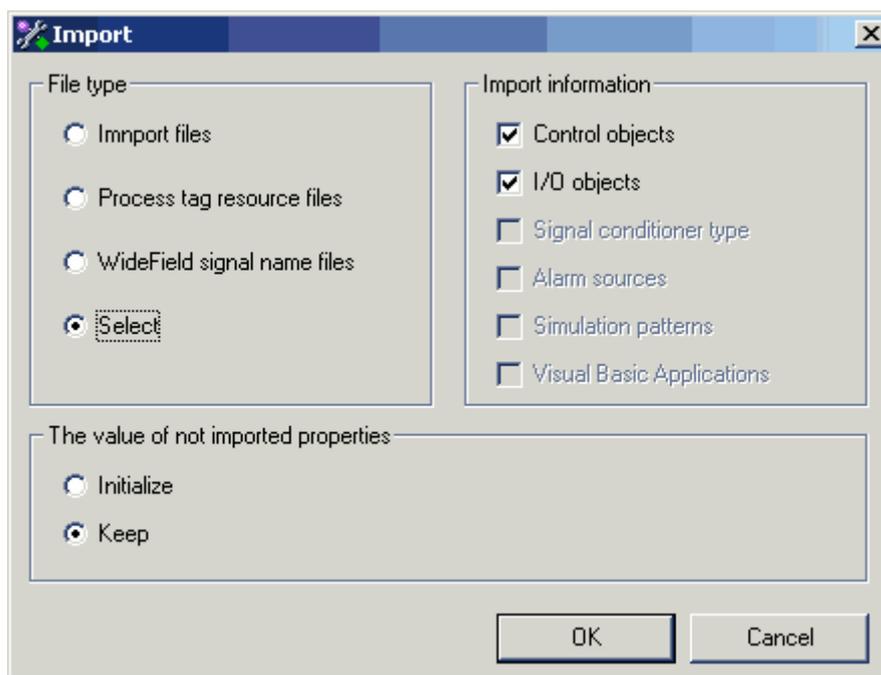
Когда режимом **Data Server** является Runtime или Debugging, он сканирует значения величин через **Control Objects**. Функции HMI сервера запускаются автоматически как Windows сервис после загрузки операционной системы.

3.3.7 Процедуры.

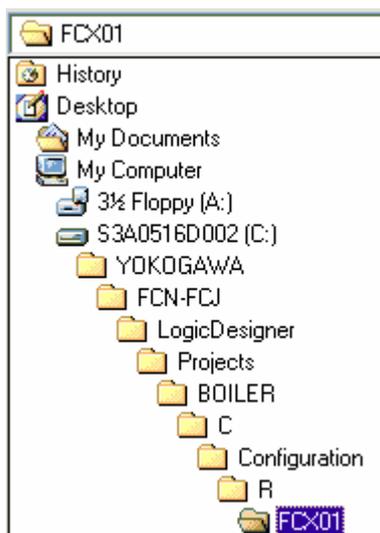
3.3.7.1 Импорт и модификация базы данных тегов (Tag Database).

Как импортировать базу данных FCX:

1. Выберите опцию **“Import Settings”** выпадающего меню **“File”**, должно появиться окно **Object Builder “Import”**:



2. В партии **“File Type”** установите маркер **“Select”**. Убедитесь, что в партии **“Import information”** отмаркированы **“Control objects”** и **“I/O objects”**. Нажмите кнопку **“OK”**;
3. В появившемся окне выберите тип файла **“CSV” (STARDOM ADLST)**, проверьте что опция **“Use Previous Settings”** не выбрана. Нажмите кнопку **“OK”**;



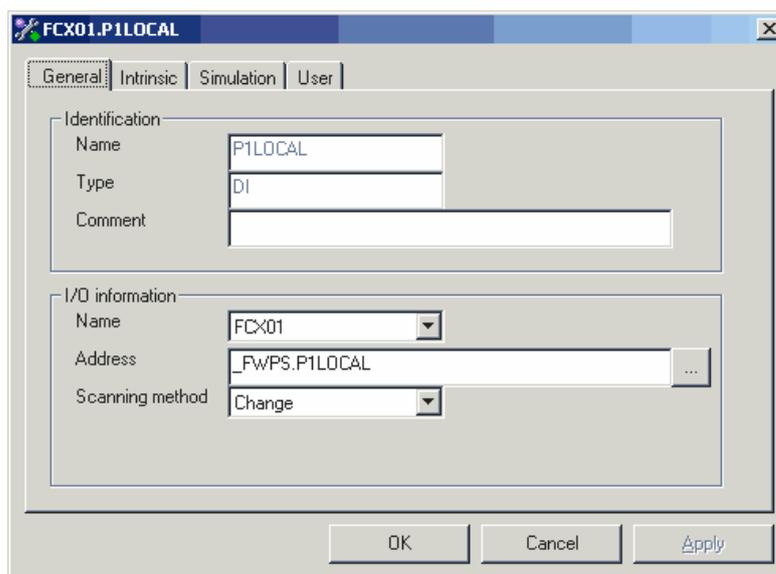
4. Укажите размещение файла, он должен быть размещён в папке “FCX...” проекта, как показано на рисунке выше. Выберите файл “ADLST” и нажмите кнопку “OK”;
5. В рабочее окно “Import Settings” выдаётся перечень импортированных объектов управления для заданного FCX, можно редактировать столбцы “Object name”, “Group name” и “Type”. Маркеры в столбце “Import” предназначены для специфицирования объектов, являются ли они импортированными или нет. По окончании редактирования нажмите кнопку “OK”, объекты импортируются в соответствующие группы.

| | Import | Original name | Group name | Object name | Type | Comment | Informati |
|----|-------------------------------------|----------------------------------|------------|------------------|-------|---------|-----------|
| 8 | <input checked="" type="checkbox"/> | @CV.P1DUTY | FCX01 | P1DUTY | DI | | FCX01 |
| 9 | <input checked="" type="checkbox"/> | @CV.P2DUTY | FCX01 | P2DUTY | DI | | FCX01 |
| 10 | <input checked="" type="checkbox"/> | @CV.PSTART | FCX01 | PSTART | DI | | FCX01 |
| 11 | <input checked="" type="checkbox"/> | Main1.CV100.PAS_FOUT | FCX01 | Main1CV100 | PFOUT | | FCX01 |
| 12 | <input checked="" type="checkbox"/> | Main1.CV100A.PAS_MLD_BT | FCX01 | Main1CV100A | PMLD | | FCX01 |
| 13 | <input checked="" type="checkbox"/> | Main1.CV100B.PAS_MLD_BT | FCX01 | Main1CV100B | PMLD | | FCX01 |
| 14 | <input checked="" type="checkbox"/> | Main1.LEVELCTL_A1.FIC100.PAS_PID | FCX01 | EVELCTL_A1FIC100 | PPID | | FCX01 |
| 15 | <input checked="" type="checkbox"/> | Main1.LEVELCTL_A1.LIC100.PAS_PID | FCX01 | EVELCTL_A1LIC100 | PPID | | FCX01 |
| 16 | <input checked="" type="checkbox"/> | Main1.SF100.PAS_MLD | FCX01 | Main1SF100 | PMLD | | FCX01 |
| 17 | <input checked="" type="checkbox"/> | _FWPS.PUMP1.PAS_SIO_11 | FCX01 | PUMP1 | PSIO | | FCX01 |
| 18 | <input checked="" type="checkbox"/> | _FWPS.PUMP2.PAS_SIO_11 | FCX01 | PUMP2 | PSIO | | FCX01 |
| 19 | <input checked="" type="checkbox"/> | _FWPS.P12DS.AUTSTAT | FCX01 | P12DSAUTSTAT | DI | | FCX01 |
| 20 | <input checked="" type="checkbox"/> | _FWPS.P12DS.AUTSTAT2 | FCX01 | P12DSAUTSTAT2 | DI | | FCX01 |
| 21 | <input checked="" type="checkbox"/> | _FWPS.P12DS.P1START | FCX01 | P12DSP1START | DI | | FCX01 |
| 22 | <input checked="" type="checkbox"/> | _FWPS.P12DS.P2START | FCX01 | P12DSP2START | DI | | FCX01 |
| 23 | <input checked="" type="checkbox"/> | _FWPS.P12DS.PAVAIL | FCX01 | P12DSPAVAIL | DI | | FCX01 |
| 24 | <input checked="" type="checkbox"/> | _FWPS.P12DS.PSTART | FCX01 | P12DSPSTART | DI | | FCX01 |
| 25 | <input checked="" type="checkbox"/> | _FWPS.P11L.PINT | FCX01 | P11LPINT | DI | | FCX01 |
| 26 | <input checked="" type="checkbox"/> | _FWPS.P1LOCAL | FCX01 | P1LOCAL | DI | | FCX01 |
| 27 | <input checked="" type="checkbox"/> | _FWPS.P1POWER | FCX01 | P1POWER | DI | | FCX01 |

Update import settings Save CSV file path [OK] [Cancel]

Как модифицировать базу данных FCX:

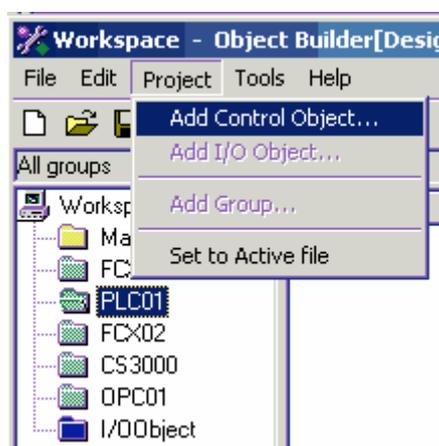
Ранее импортированные **Control Objects**, могут быть модифицированы. Для этого щёлкните дважды по объекту. После этого адрес, из которого данные читаются или по которому пишутся, может быть изменён:



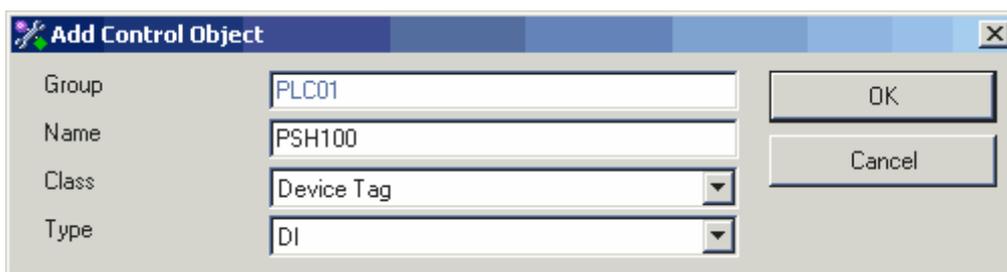
Как создать группу:

1. Щёлкните по “**Workspace**”;
2. Выберите опцию “**Add Group...**” в выпадающем меню “**Project**”;
3. Группа будет создана в виде папки в “**Workspace**”.

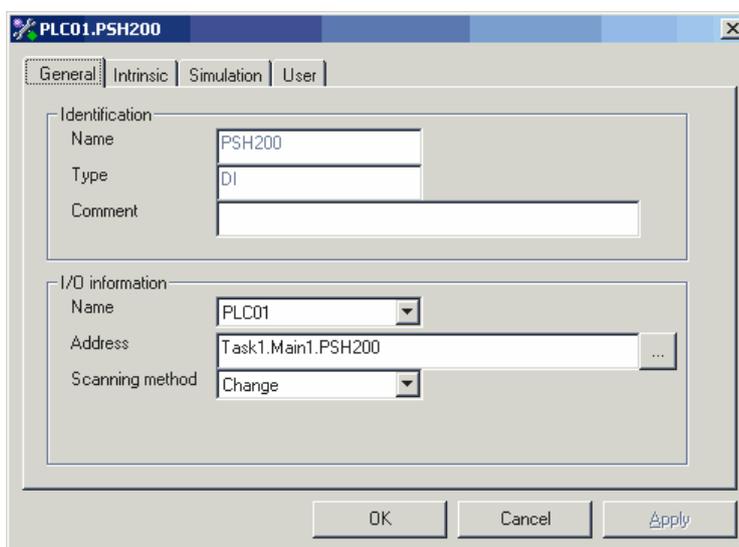
Как вручную создать управляющий объект:



1. Выберите группу, в которой объект будет размещаться;
2. Выберите опцию “**Add Control Object...**” в выпадающем меню “**Project**”;
3. Присвойте объекту имя и выберите класс и тип.



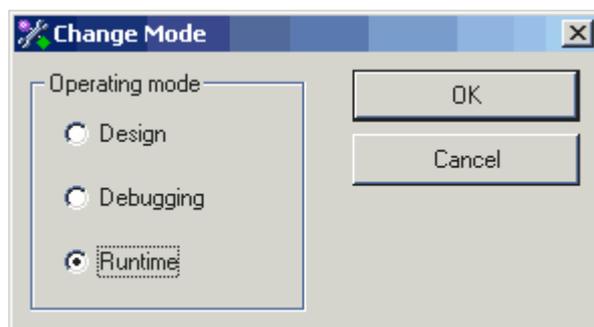
4. Появляется такое же окно, как и при модифицировании существующего объекта. В партии “**I/O information**” в опции “**Name**” выберите **IO Object** (или драйвер) через который будет производиться чтение или запись, а в опции “**Address**” адрес переменной или функционального блока в полевом устройстве.



3.3.7.2 Запуск функций VDS.

Как запустить функции VDS из Object Builder:

Выберите опцию “**Change Mode**” в выпадающем меню “**Tools**”, появляется следующее диалоговое окно:

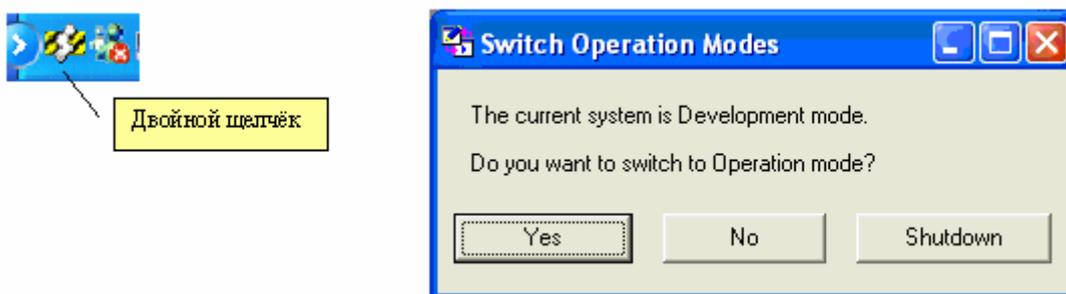


- Перед переходом в режимы “**Debugging**” или “**Runtime**” сохраните проект.

- При выборе режима отладки открывается окно отладчика Visual Basic. Если оно не нужно, закройте его. Режим отладки выбирается при необходимости проведения тестирования приложений при создании проекта.

Как запустить функции VDS через системную панель задач:

Щёлкните дважды по иконке Stardom на системной панели задач. Открывается следующее окно:



Это окно позволяет переключаться между режимами **Operation (Runtime)** и **Development (Design)**, или отключать функции **Data Server**. Режим **Debugging** может быть запущен только в режиме **Development**.

3.4 Графический редактор (Graphic Builder).

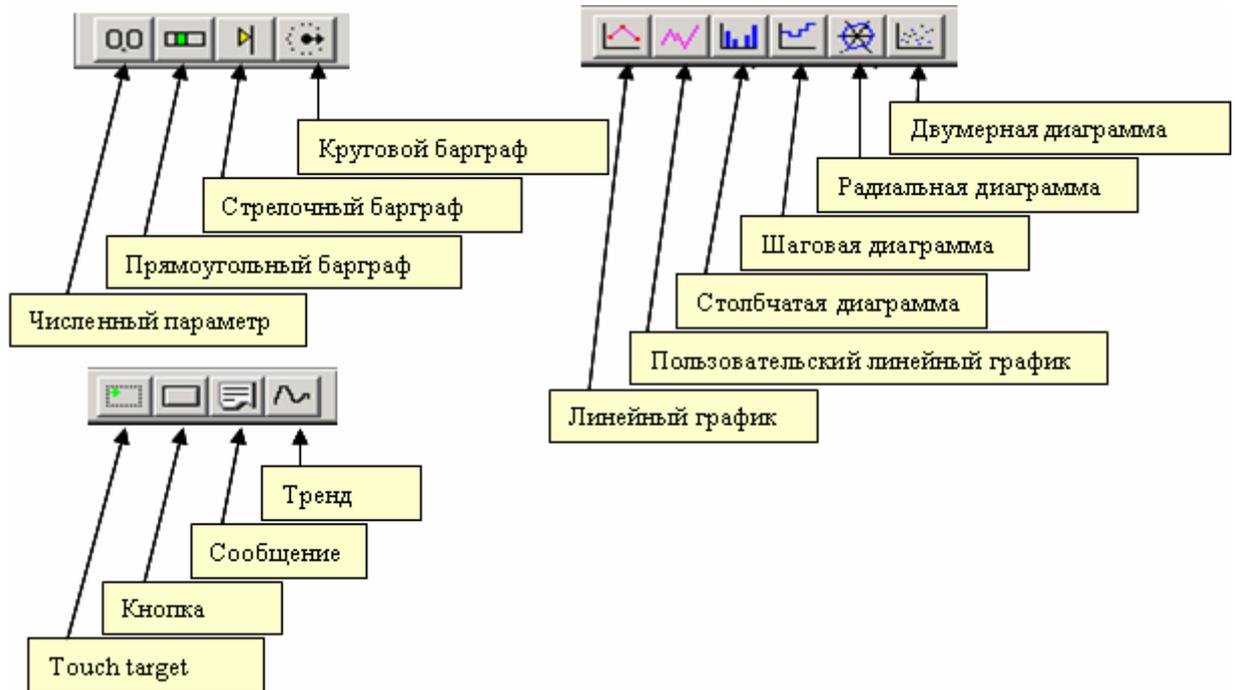
3.4.1 Основные характеристики.

Более подробно процедуры и функции графического редактора описаны в “**VDS Instruction Manual 34P02E02-01E, Section 2**”.

Графические средства встроены в Web страницы и эти Web страницы могут быть модифицированы с использованием стандартного HTML редактора.

Максимальная ёмкость одной графической страницы составляет – до 200 данных из Data Server и данных используемых для условий модифицирования.

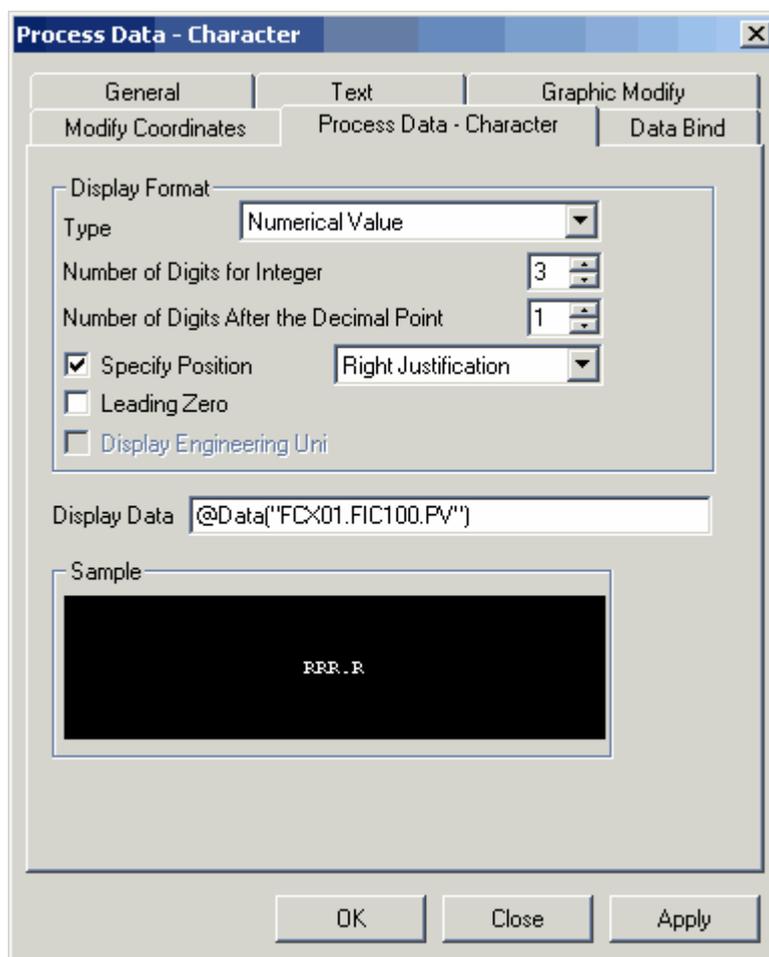
3.4.2 Прimitives (Primitives).



3.4.3 Графические объекты (Display Objects).

Графические объекты, такие как барграф или объекты отображения данных, ссылаются на **Control Object** в **Group**, как он был определен в **Object Builder**.

Как специфицировать данные в графическом объекте:



Формат привязки графического объекта к управляющему следующий:

```
@Data("Group_Name.Control_Object_Name.Variable")
```

Например, для отображения PV управляющего объекта 'FIC100' из группы FCX01:

```
@Data("FCX01.FIC100.PV")
```

3.4.4 Функциональные графические объекты (Function Objects).

К функциональным графическим объектам относятся **Touch Target** и **Pushbutton**, в которых происходят определённые действия при щелчке по ним.

Существует несколько видов действия:

- **Open URL** – вызывает другую Web страницу;
- **Open Monitoring or Logon Window** – вызывает другую графическую страницу;
- **Setting** – Записывает заранее predetermined величину в **Control Object**;
- **Call Data Input Window** – вызывает диалог для ввода величины записываемой в **Control Object**;
- **Call Menu Dialog** – вызывает меню величин записываемых в **Control Object**;
- **Print** – распечатывает графическую страницу.

Как специфицировать данные в функциональном графическом объекте:

Формат привязки к Control Object зависит от функции.

Функции Open Monitoring or Logon Window.

Существуют несколько типов окон, которые могут быть открыты:

- Graphic Index Window – вызывает перечень графических окон;
- Graphic Window – вызывает специфицированное графическое окно;
- Object View Index Window – вызывает перечень Control Objects;
- Object View Window – вызывает Control Object в виде лицевой панели (Faceplate);
- Logon Window – запароливает графическое окно;
- Logoff Window – распароливает графическое окно.

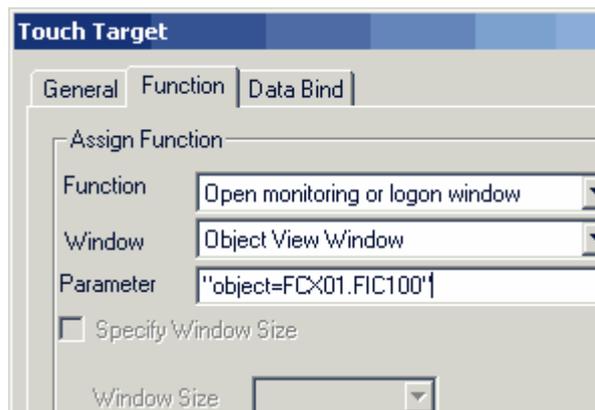
Для графических окон действует следующий синтаксис описания параметров:

`"window=Window_Name"`

Для окна обзора объектов (Object View Window), действует следующий синтаксис описания параметров:

`"object=Group_Name.Object_Name"`

Пример:



"object=FCX01.FIC100"

Вызывает лицевую панель функционального блока типа PID с именем FIC100.

Для других типов окон специфицирование имени окна не требуется.

Функции Setting.

Существуют три функции Setting:

- Setting
- Call Data Input Window
- Call Menu Dialog

В них синтаксис для ссылки на данные следующий:

"Group_Name.Control_Object_Name.Variable"

Пример:

"FCX01.PSTART_DO.CV"

Примечание: Для того чтобы узнать, какую переменную использовать в Control Object, необходимо понять внутреннюю природу его структуры данных. Для получения более подробной информации о структурах данных функциональных блоков смотри раздел [2.4.2](#).

Touch Target [X]

General | **Function** | Data Bind

Assign Function

Function: Call Data Input Window

Data: "FCX01.LIC100.SV"

Acknowledg: No Acknowledgment

Display Cursor

Cursor: 0

OK Close Apply

3.4.5 Модифицирование поведения графических объектов (Graphic Modify).

Графические объекты (примитивы) могут изменять (модифицировать) своё поведение при выполнении заранее определённых условий состояния процесса. Существуют следующие способы модификации:

- Color Change – изменение цвета;
- Alarm blinking – аларменное мигание;
- Conditional blinking – условное мигание;
- Make primitive invisible – перевод в невидимое состояние;
- Modify string (для текстовых) – изменение строки символов;
- Invert String (для текстовых) – инверсия строки.

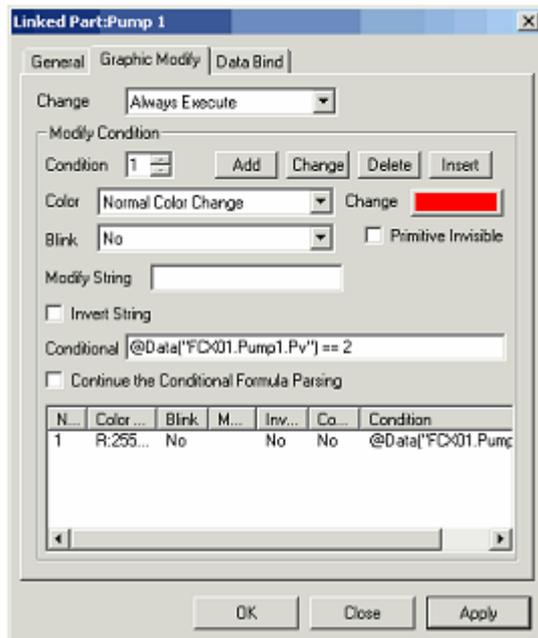
Всего может использоваться до 8 способов модификации примитивов.

Условия для модификации задаются в следующем формате:

```
@Data("Group_Name.Control_Object_Name.Variable")  
[comparator] VALUE
```

Пример:

```
@Data("FCX01.Pump1.Pv") == 2
```



Для компаратора (comparator) операторы сравнения, формирующие логические условия, следующие:

| | |
|----|------------------|
| == | равно |
| != | не равно |
| < | меньше чем |
| <= | меньше или равно |
| > | больше чем |
| >= | больше или равно |

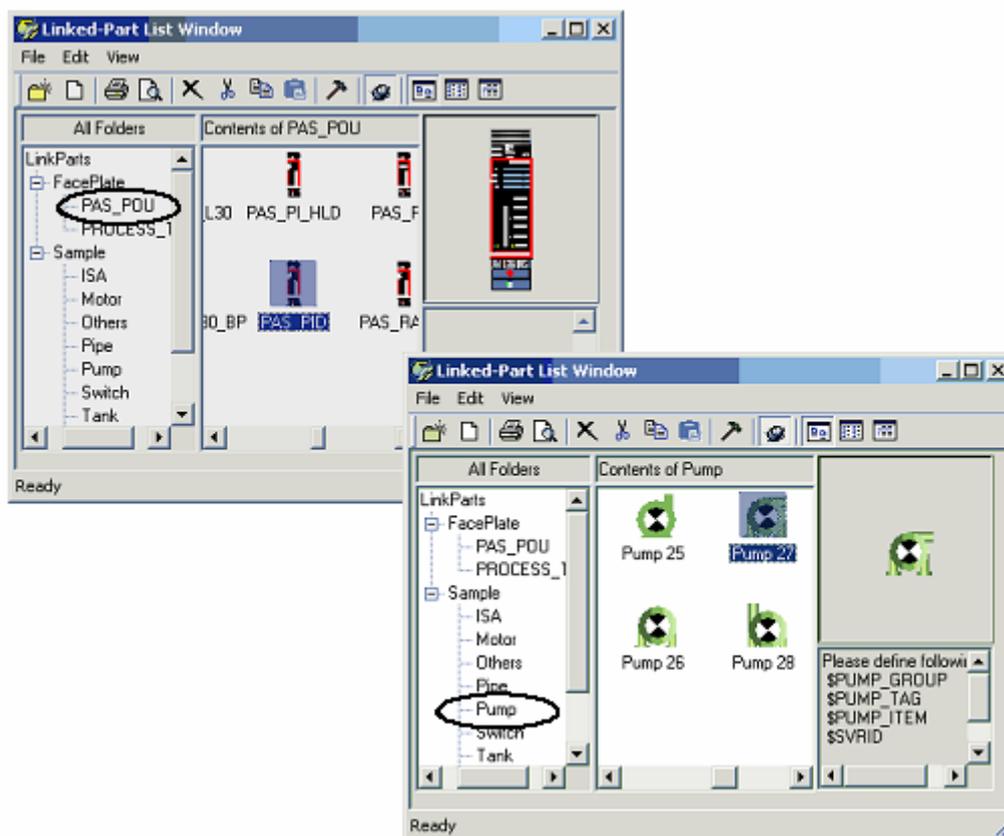
Логические операторы (|| – или, && – и) могут использоваться между логическими условиями.

3.4.6 Библиотечные примитивы (Linked Parts).

Группа примитивов может быть сгруппирована и сохранена для повторного использования. Они называются библиотечными примитивами (**Linked Part**). Кроме того, существуют уже готовые библиотеки примитивов. Библиотечный примитив содержит определение для привязки данных (**Data Bind**), в котором имена объектов, а также другие параметры должны быть связаны с наследуемыми именами (**generic names**).

Доступны следующие готовые примитивы:

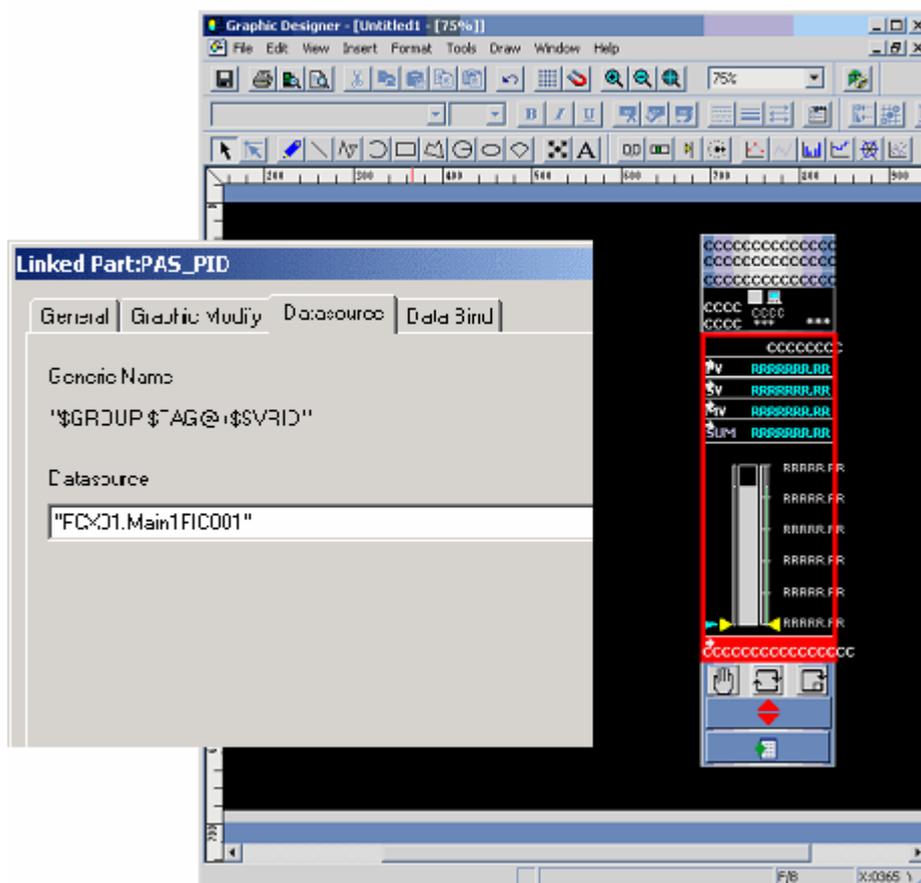
- PAS_POU – связаны непосредственно с PAS_POU блоками FCX;
- PROCESS_TAG – подобны PAS_POU, но привязаны к другим контроллерам;
- ISA – стандартные символы (ISA);
- Motor – двигатели;
- Others – прочие (конвейеры, пламя, кучи и т.д.);
- Pipe – трубы;
- Pump – насосы;
- Switch – выключатели;
- Tank – резервуары;
- Valve – клапаны.



Для того, чтобы поместить библиотечный примитив в графическое окно выберите необходимую тему, щёлкните по ней, в средней части окна откроется содержимое раздела (набор примитивов по выбранной теме). Выберите необходимый примитив и перетащите его в графическое окно. После перемещения должны быть установлены параметры привязки данных.

- Щёлкните правой кнопкой мыши по объекту и выберите в контекстном меню опцию “Properties”, откроется окно “**Linked Part:...**”;
- Выберите необходимую закладку, на ней распечатываются список имён наследуемых объектов.

Начальные значения параметров наследуемых объектов устанавливаются путём специфицирования “**Datasource**”:



- Введите имя наследуемого параметра в “**Datasource**” как показано ниже:

“Group_Name.Control_Object.Item_Name@Server_ID”

Item_Name не требуется для PAS_POU и PROCESS_TAG.

Server_ID не требуется, если **Data Server** и **HMI Server** работают на одной и той же PC.

Пример:

“FCX01.FIC001”

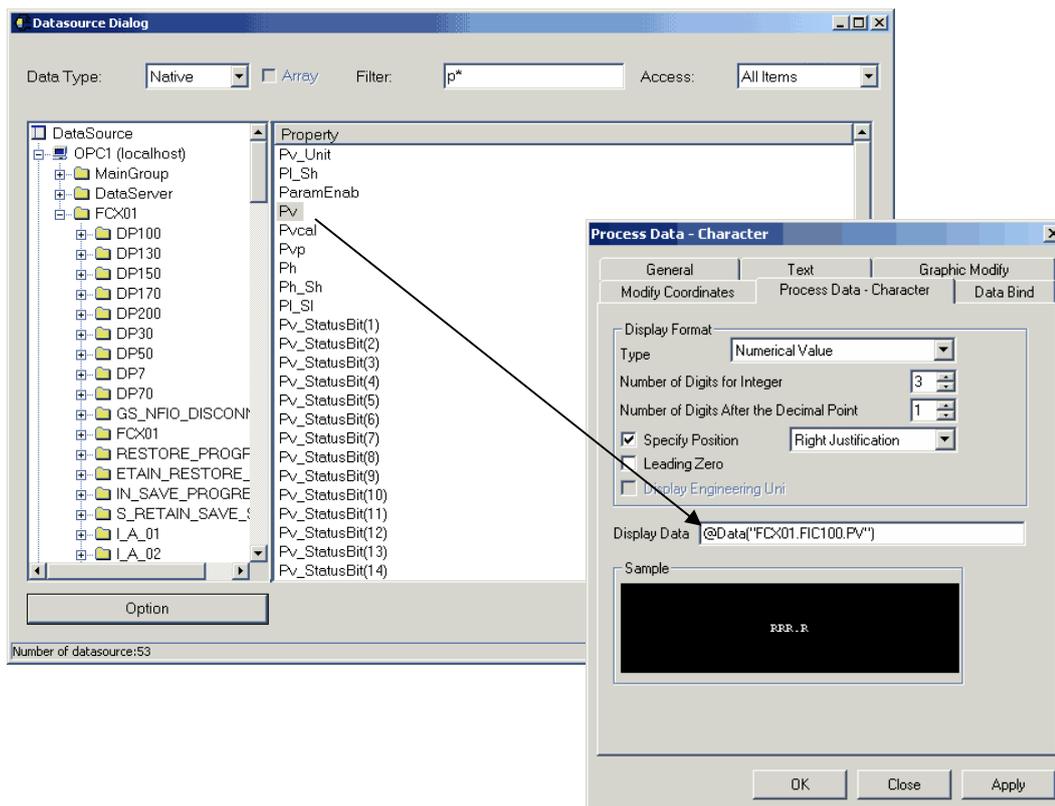
Во избежание ошибок и для упрощения работы, имя объекта рекомендуется копировать из списка имён диалога “**Datasource**” в лист закладки “**Datasource**”.

Имя наследуемого объекта в таблице привязки данных (Data Bind) tab определяется автоматически в соответствии с расширенным форматом строки источника данных.

3.4.7 Настройка источника данных (Datasource).

При настройке идентификатора в **Graphic Modifier**, **Display Object** или **Linked Part** вместо ввода идентификатора вручную удобнее использовать “**Datasource tool**” с выпадающим меню перечня идентификаторов.

На этот диалог можно выйти через опцию “**Datasource**” выпадающего меню “**Tools**” **Graphic Designer**.



Как использовать процедуру:

1. Откройте источник данных в “DataSource” (например OPC1);
2. Откройте папку FCX (например, FCX01), под ней открывается перечень объектов;
3. Щёлкните по одному из объектов, с правой стороны открывается перечень параметров.
4. Количество параметров в списке может быть сужено путём применения символического фильтра (например “P*”).
5. Щёлкните по требуемому параметру и перетащите в поле данных “Display Data”, как показано на рисунке.

3.4.8 HMI Deployment Tool.

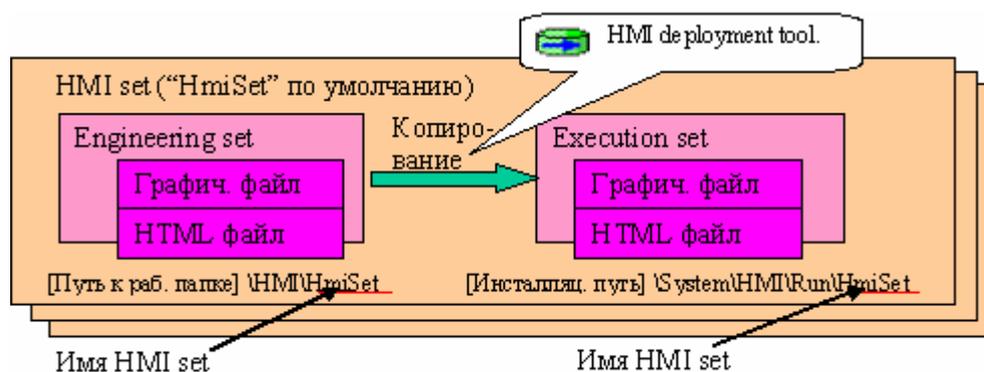
Однажды созданное графическое окно должно быть преобразовано из окна для работы в режиме редактирования в окно для работы в режиме Runtime так чтобы Web сервер мог обслуживать его как графический HMI клиент. Это преобразование осуществляется при помощи **HMI Deployment Tool**. Средство запускается через опцию “**HMI Deployment Tool**” выпадающего меню “**Tools**” **Graphic Builder**.

При исполнении **HMI Deployment Tool** извлекает графику из папки Engineering Set и помещает её в папку Runtime Set.

По умолчанию папка Engineering Set находится:

\VDS\Work\HMI\HmiSet

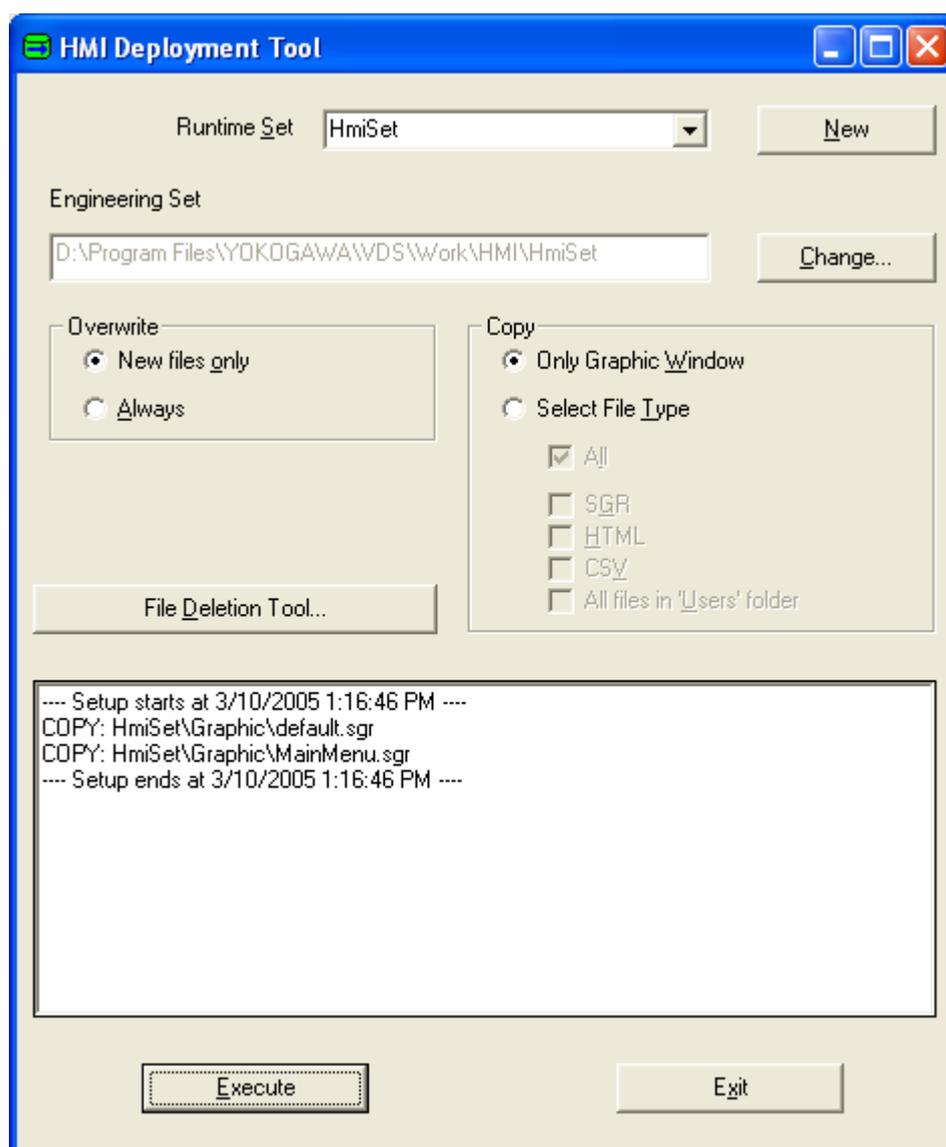
Графика содержится в подпапке “**Graphics**”, находящейся в папке Engineering Set.



Может быть создан новый графический образ, при этом его имя (**Set Name**), размещение в Engineering Set и некоторые другие параметры отличаются.

Опции, применимые в **Deployment Tool**, следующие:

- **Overwrite** (перезаписывание):
 - New Files Only – только новые файлы (по умолчанию);
 - Always – все.Если выбрана опция “**New Files Only**”, то преобразуются только файлы, которые были изменены.
- **File Type** (тип файла):
 - Only Graphic Window – только графические окна (по умолчанию);
 - Select FileType – с выбором типа файлов;
 - All – все;
 - SGR – файлы с расширением *.sgr;
 - HTML – HTML файлы;
 - CSV – файлы с расширением *.csv;
 - All files in ‘Users’ folder – все файлы из пользовательской папки “Users”.



Если выбрана опция “**Only Graphic Window**”, то преобразуются только файлы с расширением “*.sgr”.

3.5 Функция архивирования Истории (Historian) и Тренды (Trend).

Функция архивирования Истории (**Historian**) осуществляет сбор данных и их накопление в исторической базе данных VDS. Эти данные затем могут быть использованы для визуализации трендов в графической форме, быть доступны другим клиентам через OPC или быть обработаны с использованием VBA программы.

Конфигурирование архива осуществляется с помощью конфигуратора Истории (**History Builder**), запускаемого из **Object Builder**.

Предельная ёмкость архива:

- 32 блока;
- 4 группы на блок 8 перьев на группу;
- Максимум до 16 блоков могут иметь время сканирования 10 секунд или быстрее.

3.5.1 Структура архива Истории.

- **Блок (Block)** – единица накопления данных с установленным временем сканирования, типом и другими атрибутами;
- **Группа (Group)** – каждый блок состоит из 4 групп, в которых определен набор переменных **Control Object**. Всего в группе может быть определены до 8 переменных.



3.5.2 Функция архивирования Истории.

Данные, специфицированные внутри трендовых групп, собираются со скоростью определённой для блока в целом. Данные удерживаются внутри блока в течение периода накопления (**Acquisition Period**). После этого, либо удаляются, либо переписываются в долговременные архивные файлы (**Long Term History**), где они хранятся в течение заранее специфицированного периода.



Конфигурационные параметры исторического блока (History Block):

- Block Name – имя блока;
- **Sampling Period** – период сканирования (периодичность сохранения данных в блоке);
- **Acquisition Period** – период накопления (период, в течение которого, данные накапливаются в блоке).

Примечание: Моменты записи данных на диск не привязаны непосредственно к моменту запуска накопления.

- **Acquisition Method** – метод накопления (непрерывное (Continuous) или в течение заданного времени (Batch Sampling)) (более подробно см. ниже);
- **Trigger Tagname** – если тренд типа Batch Sampling эта переменная должна быть истинной для запуска процесса накопления;
- **Dead band Limit** – предельное значение изменения входной величины выраженное в процентах шкалы, при превышении которой осуществляется сбор и накопление данных;

- **Delay** – определяет момент записи на диск. Постоянное обращение к диску для сохранения данных одновременно во множестве блоков может вызвать перегрузку системы. Устанавливаемое время, выраженное в миллисекундах с шагом 100мс., задерживает момент записи на диск и позволяет избежать перегрузки;
- **Pretrigger acquisition time period** – для трендов типа Batch Sampling можно задать период времени предшествующий возникновению условия запуска, в течение которого накопленные до наступления условия данные будут сохраняться;
- **Long-term Storage** – специфицирует необходимо ли сохранение истории в долговременном архивном файле;
- **Storage filename** – имя и размещение долговременного архивного файла для данного блока.

Метод накопления.

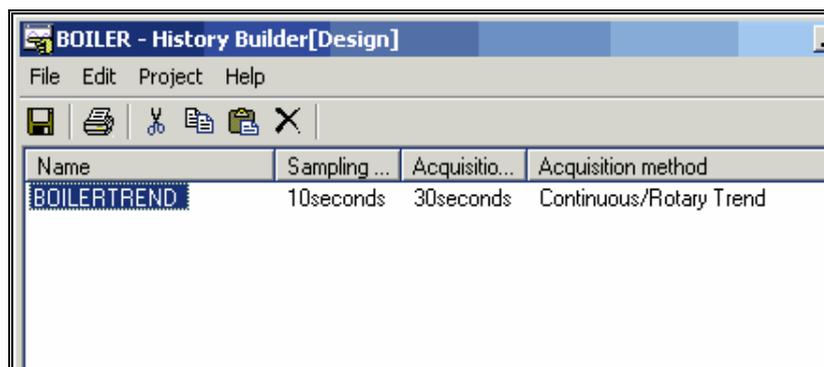
Типы методов накопления блока следующие:

- **Continuous/Rotary Trend** – непрерывное циклическое накопление данных;
- **Batch/Rotary Trend** – тренд, который может быть запущен и остановлен, но ведёт себя в запущенном состоянии подобно **Continuous/Rotary Trend**. Запускается переменной специфицированной в “**Trigger Tagname**”;
- **Batch/Non-Rotary Trend** – тренд, который может быть запущен и остановлен, но останавливается тогда когда трендовый блок заполняется. Запускается переменной специфицированной в “**Trigger Tagname**”;
- **Non-acquiring** – данных не собирает. Используется для таких ситуаций, когда данные собирает какое либо внешнее приложение и их отображение производится с использованием интерфейса VBA.

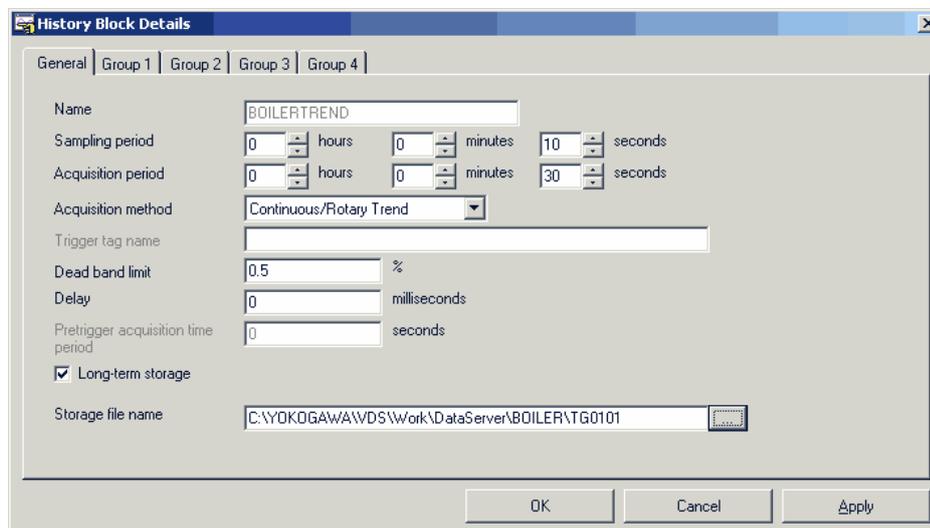
3.5.3 Процедуры.

Как установить параметры Истории:

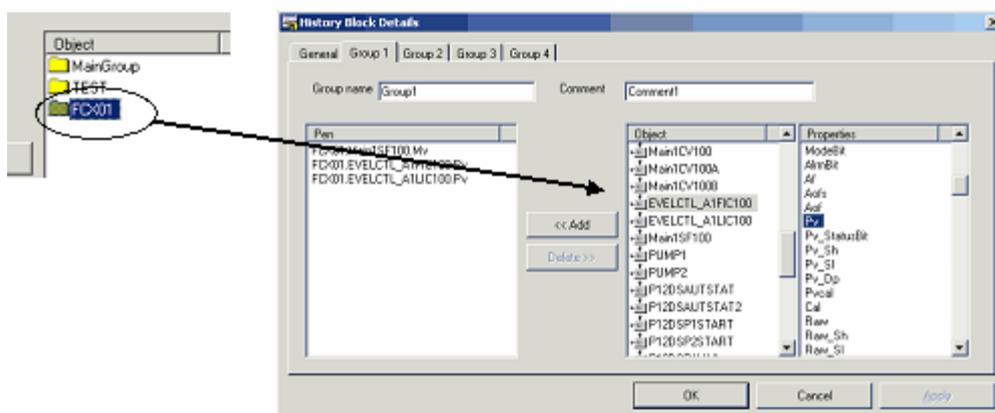
1. В **Object Builder** выберите опцию “**History Builder**” в выпадающем меню “**Tools**”. Открывается окно **History Builder** с перечнем исторических блоков “**History Blocks**”, которые были созданы ранее:



2. Для того чтобы создать новый **History Block**, выберите опцию “**Add History Block**” в выпадающем меню “**Project**”. Открывается окно “**History Block Details**”:



3. Введите конфигурационную информацию для блока в закладке “**General**”;
4. Выберите группу накопления тренда и откройте соответствующую закладку;
5. Для добавления точки двойным щелчком по имени группы в “**Object Window**” откройте перечень управляющих объектов, выберите необходимый объект, в окне “**Properties**” появится перечень переменных структуры данных объекта;

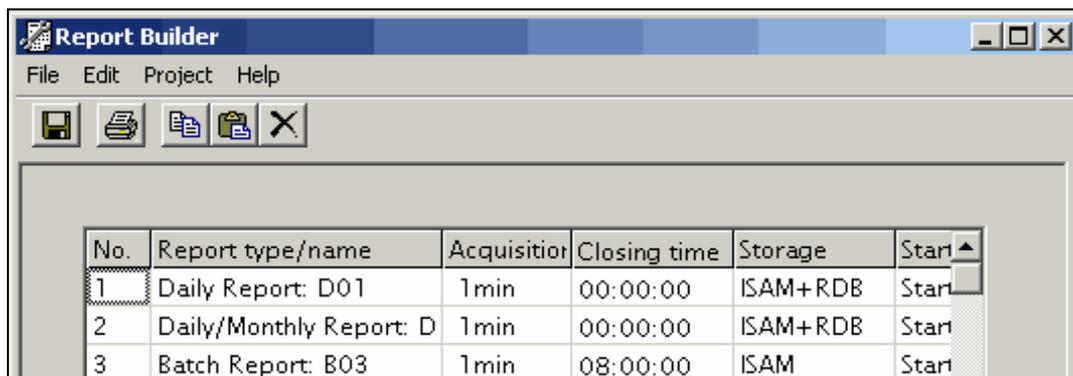


6. Выберите требуемую переменную и нажмите кнопку “<<Add”, в окне “Pen” появится имя переменной связанной с очередным пером;
7. Повторите процесс для остальных перьев группы, всего до 8 штук;
8. Нажмите кнопку “OK” чтобы завершить процесс.

3.6 Функция генерирования Рапортов (Report).

3.6.1 Общий обзор.

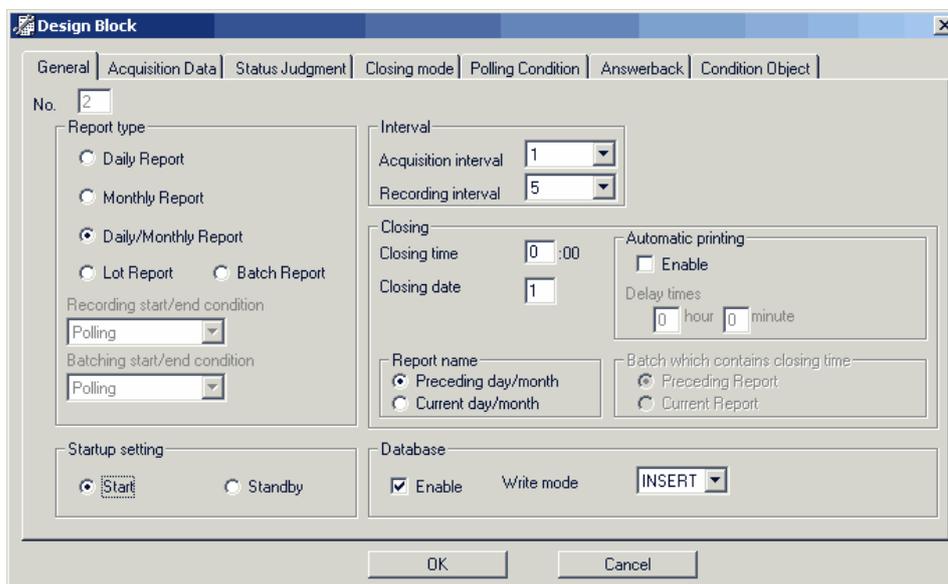
Функция генерирования Рапортов используется для сбора, обработки и печати данных в заранее определённых форматах.



Функция генерирования Рапортов исполняет следующие функции:

- **Сбор данных (Collects Data)** с заданной периодичностью накопления;
- **Расчёт (Calculates)** усреднённых, интегральных, и других закрывающихся периодически значений, сохранение их на диске в заданные интервалы времени записи;
- **Распечатка (Prints)** рапортов с заранее определённым интервалом времени распечатки или по запросу или сохранение их на диске в форме электронных Excel таблиц.

3.6.2 Функции Рапорта.



Функция генерирования Рапортов создаёт следующие типы рапортов:

- Ежедневный (Daily)** – отчёт за прошедшие сутки;
- Помесячный (Monthly)** – отчёт за прошедший месяц;
- Смешанный ежедневно-помесячный (Daily/Monthly)** – смешанный ежедневно-помесячный отчёт;
- По состоянию (Lot)** – отчёт по периоду, в течение которого заранее определённое условие истинно;
- Событийный (Batch)** – отчёт по возникновению заранее определённого события.

Процесс закрытия отчёта это расчетная функция, которая вычисляет итоговые данные, используя массивы накопленных за отчётный период данных. Результаты расчета могут быть следующими:

- Усреднённое значение (**Average**);
- Суммарное значение (**Total**);
- Максимальное (Минимальное) значение (**Maximum/Minimum**);
- Момент времени максимального (минимального) значения (**Time at Maximum/ Minimum**);
- Общее отклонение (**Difference Total**);
- Состояние накопительного процесса (**Acquisition Status**).

Расчёты могут выполняться в одном из двух режимов:

- Recording Mode** – режим записи. Расчёт производится по мере поступления данных;

-
- **Closing Mode** – режим закрытия. Расчёт производится при закрытии отчёта;

3.6.3 Сбор данных (Data Collection) и функция обработки (Processing Function).

Сбор данных осуществляется с заданной периодичностью (**Acquisition Interval**) а запись данных в ISAM историческую базу данных с периодом записи (**Recording Interval**). Расчёты могут исполняться как в моменты записи (**Recording Time**), так и по закрытию отчёта (**Closing Time**), т.е. в момент распечатки рапорта.

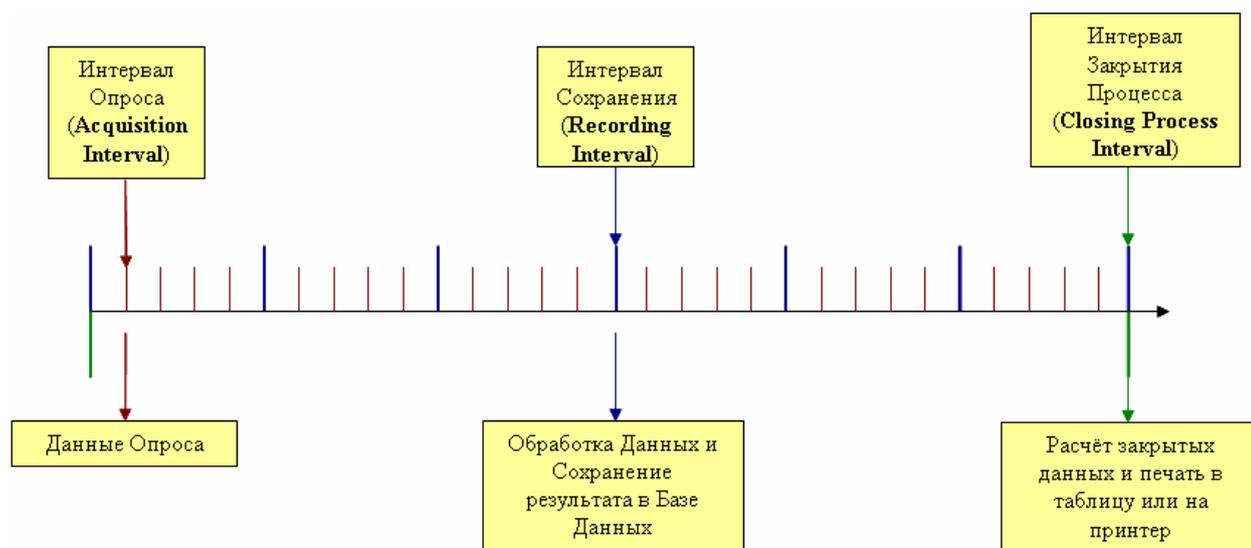
Если расчёты не выполнялись в процессе записи (т.е. накопление мгновенных значений (**Instantaneous Value**)), то в базу данных записывается только последнее значение величины. В этом случае период сбора данных и период их записи должны быть одинаковы.

Пример 1:

Если процесс записи определён как “**Instantaneous Value**” (т.е. без расчёта), и процесс закрытия (**Closing Process**) определён как усреднённое значение (**Average**), то усреднённое значение всех записанных мгновенных значений рассчитывается при печати рапорта.

Пример 2:

Если процесс записи определён как “**Average**”, то данные, записанные в базу данных с интервалом записи, уже являются усреднёнными значениями. Если процесс закрытия отчёта определён также как “**Average**”, то при его выполнении рассчитывается среднее значение всех усреднённых величин, накопленных за период закрытия.



Примечание: Одна и та же точка опроса данных может принадлежать нескольким процессам закрытия, но только один расчёт может быть приписан ей в процессе записи.

3.6.4 Функции событийных рапортов (Lot, Batch).

Если тип рапорта определён как событийный “Batch” или по состоянию “Lot”, то должны быть заданы условия запуска и останова процесса записи.

Для рапорта типа Batch задаются событийные условия запуска и останова как процесса записи, так и процесса закрытия.

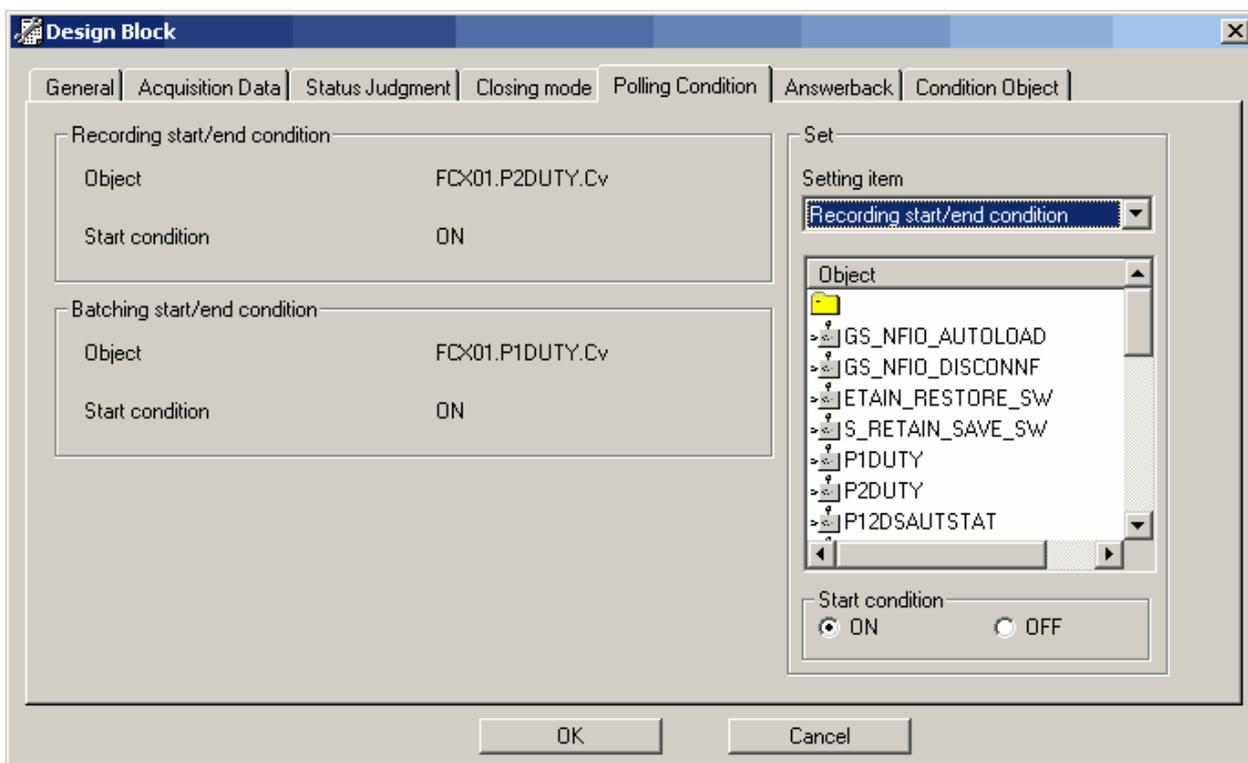
Для рапорта типа Lot только процесс записи может быть запущен и остановлен. При этом рапорт генерируется по факту останова записи автоматически.

Опции процесса записи:

- **Polling** – процесс записи запускается и останавливается в процессе сканирования объекта типа DI;
- **Event-Driven** – процесс записи запускается и останавливается приложением пользователя;
- **Everyday** (только для рапортов типа Batch) – процесс записи непрерывный с ежедневным процессом закрытия;

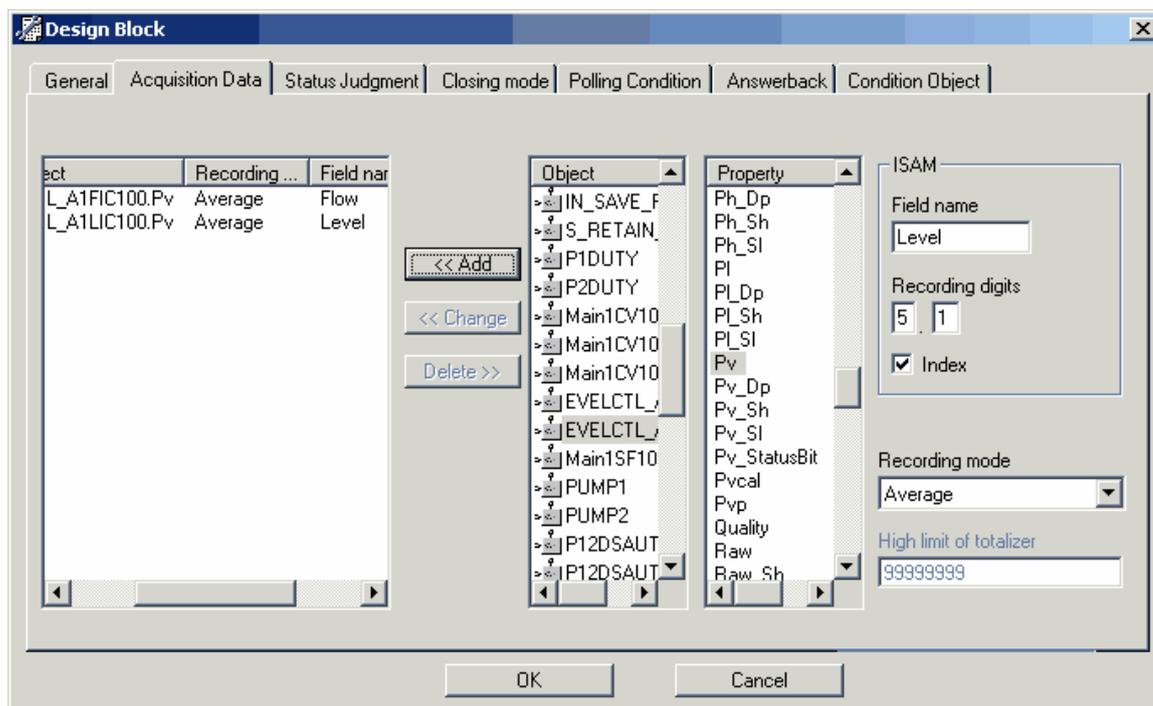
Опции процесса записи типа Batch (для рапортов типа Batch):

- **Polling** – процессы закрытия и печати отчёта выполняются по изменению состояния объекта типа DI;
- **Event-Driven** – процессы закрытия и печати отчёта управляются приложением пользователя;



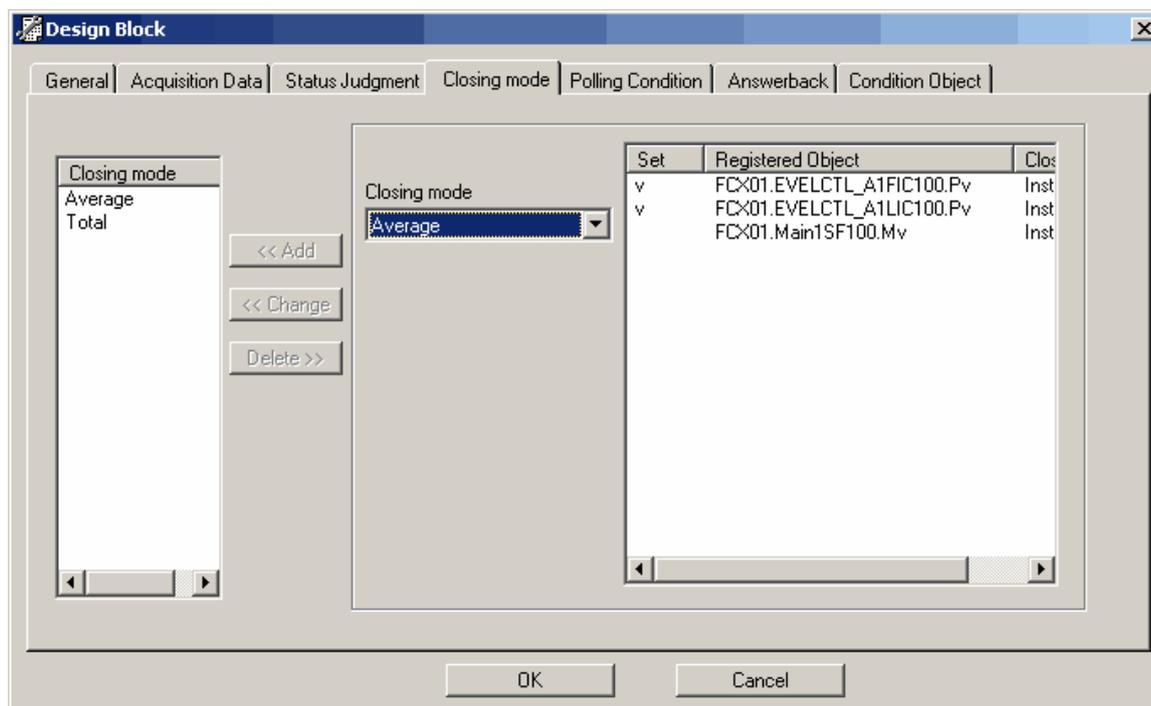
В закладке “**Polling Condition**” выбрать “**Recording**” или “**Batching**” процесс и **Control Object** управляющий процессом. Условие запуска позволяет запустить процесс, если состояние условия истинно (True) и остановить его, если состояние условия ложно (False).

3.6.5 Выбор данных для рапорта.



Закладка “**Acquisition Data**” позволяет выбрать данные для записи и генерации рапорта. В поле “**Object**” выбирается **Control Object**, в поле “**Property**” его переменная, в поле “**ISAM Field name**” вводится имя результата, в поле “**ISAM Recording digits**” его формат (общее количество десятичных знаков до десятичной точки и количество знаков после), в поле “**Recording mode**” выбирается режим записи. После завершения ввода информации нажмите кнопку “<<Add”, точка записи будет включена в перечень точек для сбора данных.

3.6.6 Конфигурирование процесса закрытия рапорта.



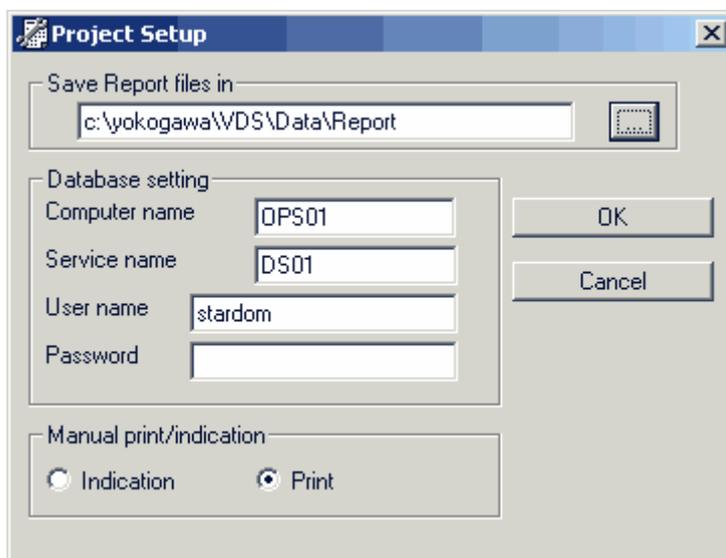
В закладке “**Closing Mode**” для каждой точки записи может быть задан режим закрытия рапорта.

Процедура:

1. Выберите “**Closing Mode**”;
2. Выберите требуемые точки, маркируя их в поле “**Set**”;
3. Нажмите кнопку “<<**Add**”;
4. Повторите процедуру для других точек.

3.6.7 Установки для функции генерирования рапортов.

Утилита установки для функции генерирования рапортов вызывается путём выбора опции “**Project Setup**” в выпадающем меню “**Project**”.

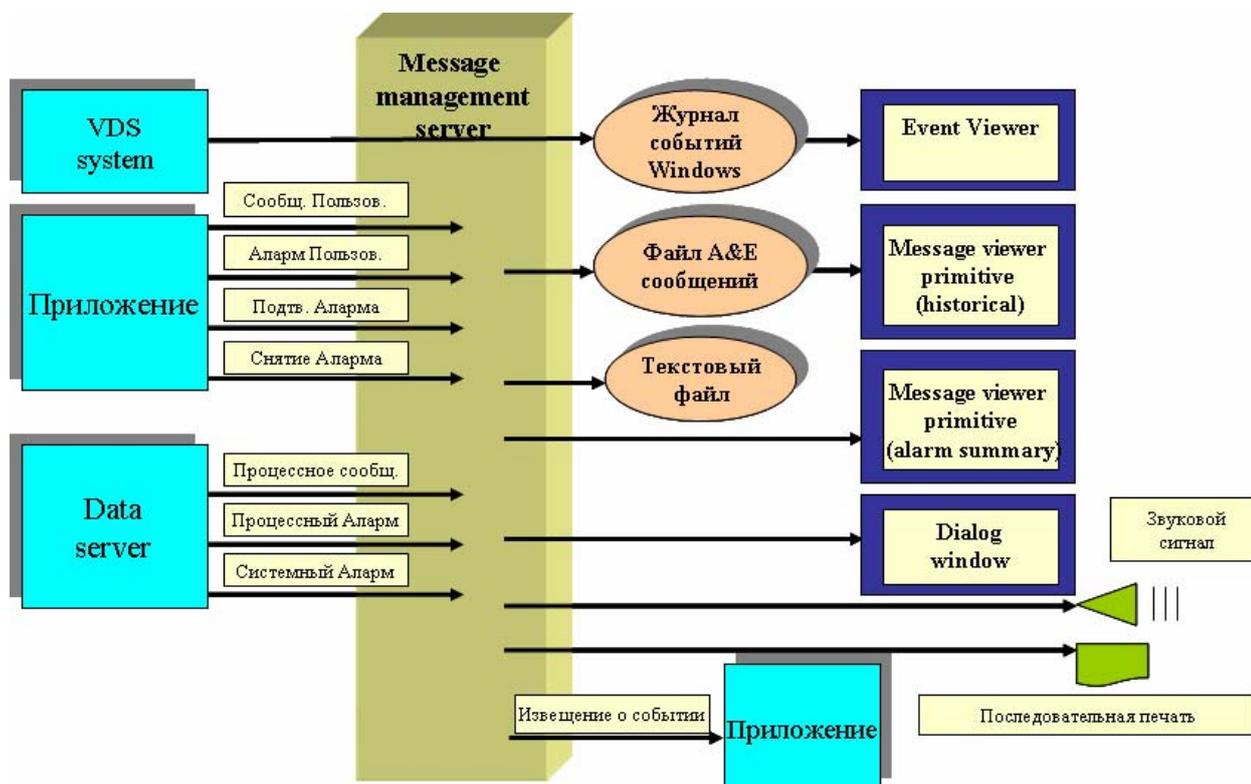


Устанавливаются следующие параметры:

- Размещение файла рапорта;
- Установки для базы данных:
 - **Computer Name** – имя компьютера в котором размещена база данных;
 - **Service Name** – имя сервиса базы данных Oracle;
 - **User name** – пароль входа в базу данных Oracle.
- Способ отображения рапорта (выдача на принтер (print), выдача на монитор (indication))

3.7 Функция сообщений (Message).

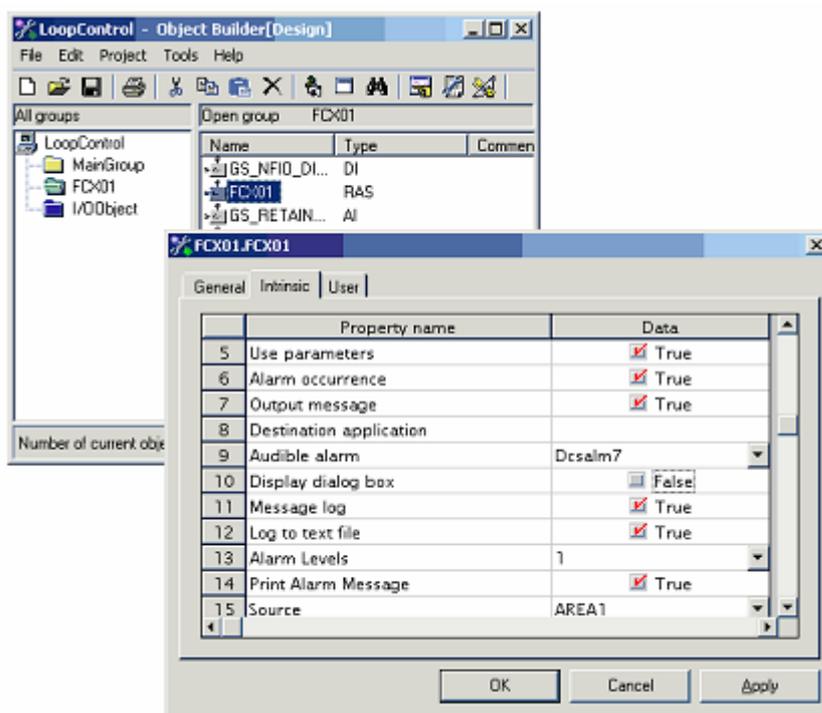
Перед тем как процессное или системное сообщение выводится на графический дисплей, оно захватывается и обрабатывается сервером управления сообщениями Message Management Server. Схематически это показано на рисунке ниже:



Функция сообщений должна быть сконфигурирована как на уровне VDS, так и на уровне FCX.

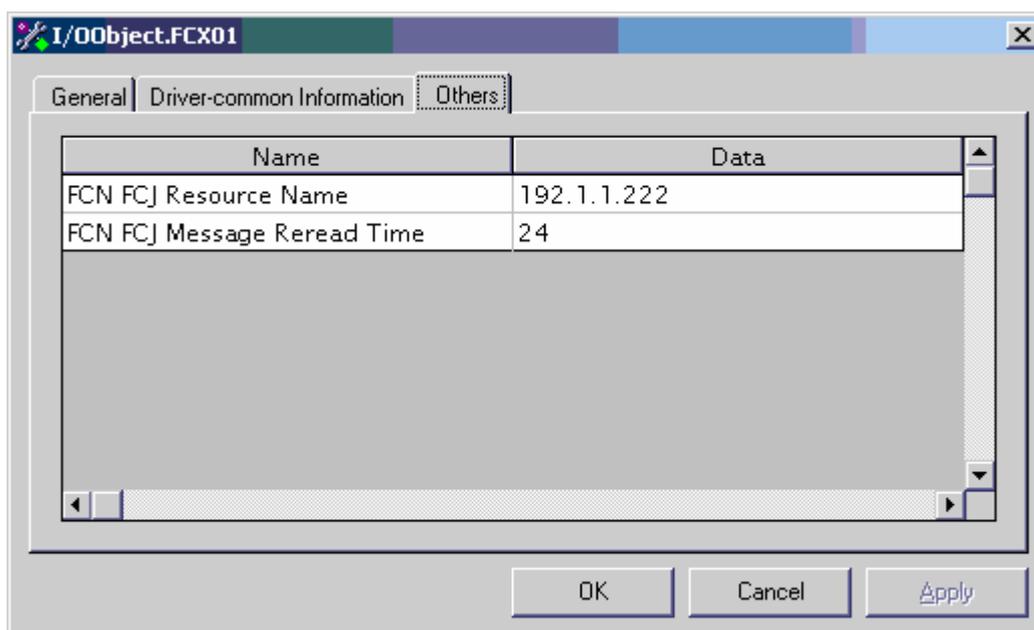
Конфигурирование функции сообщений на уровне FCX:

В **Object Builder** выберите “FCX”. Один из объектов в правой стороне экрана – FCX. Двойным щелчком по нему откройте окно свойств:



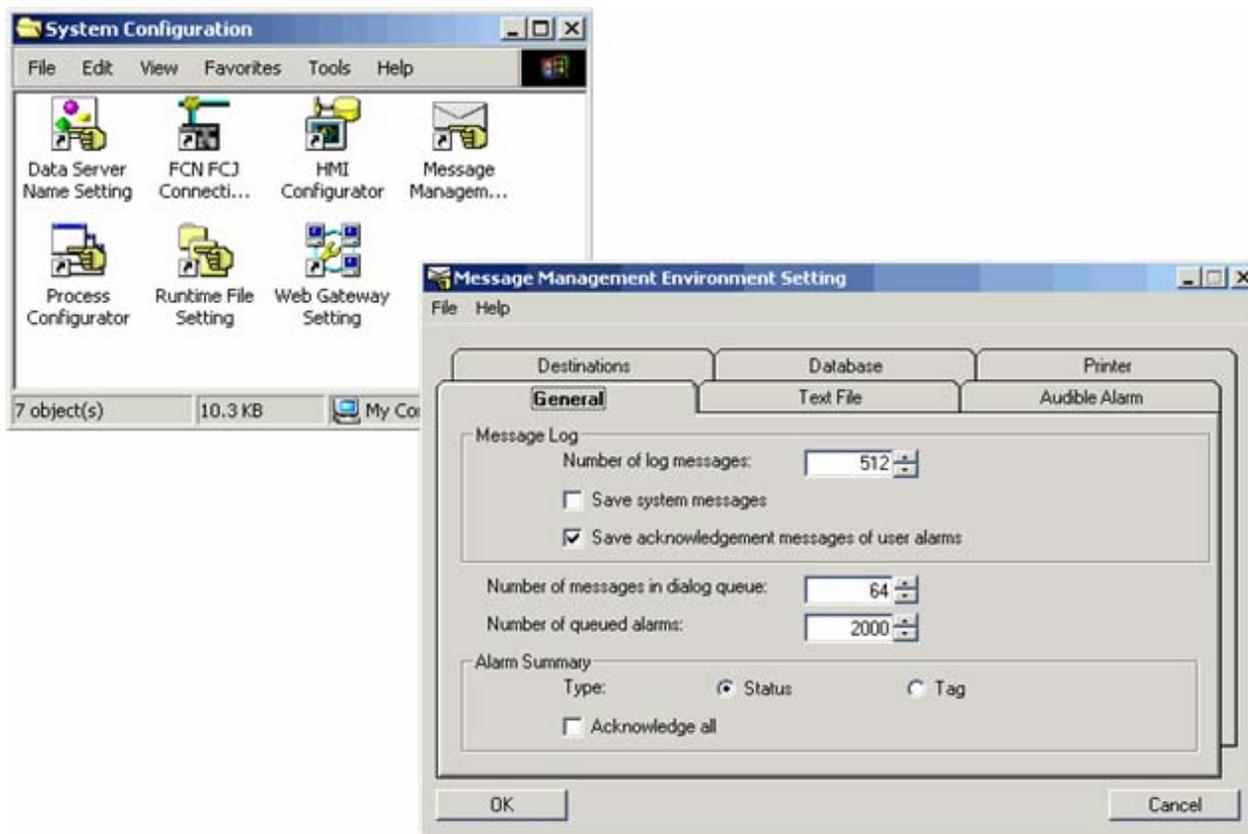
Регистрация сообщений в архиве Истории Сообщений:

Пока VDS не запущен, сообщения накапливаются внутри FCX в течении заранее специфицированного времени сохранения (**Reread Time**). Он не сохраняет данные предшествующие этому времени, даже если они присутствовали. В этом случае он выводит сообщение в журнал событий Windows сообщающее о том, что старые сообщения не сохранены.



Конфигурирование сообщений в VDS:

Конфигурация сообщений VDS устанавливает все параметры связанные с регистрацией и представлением алармов.

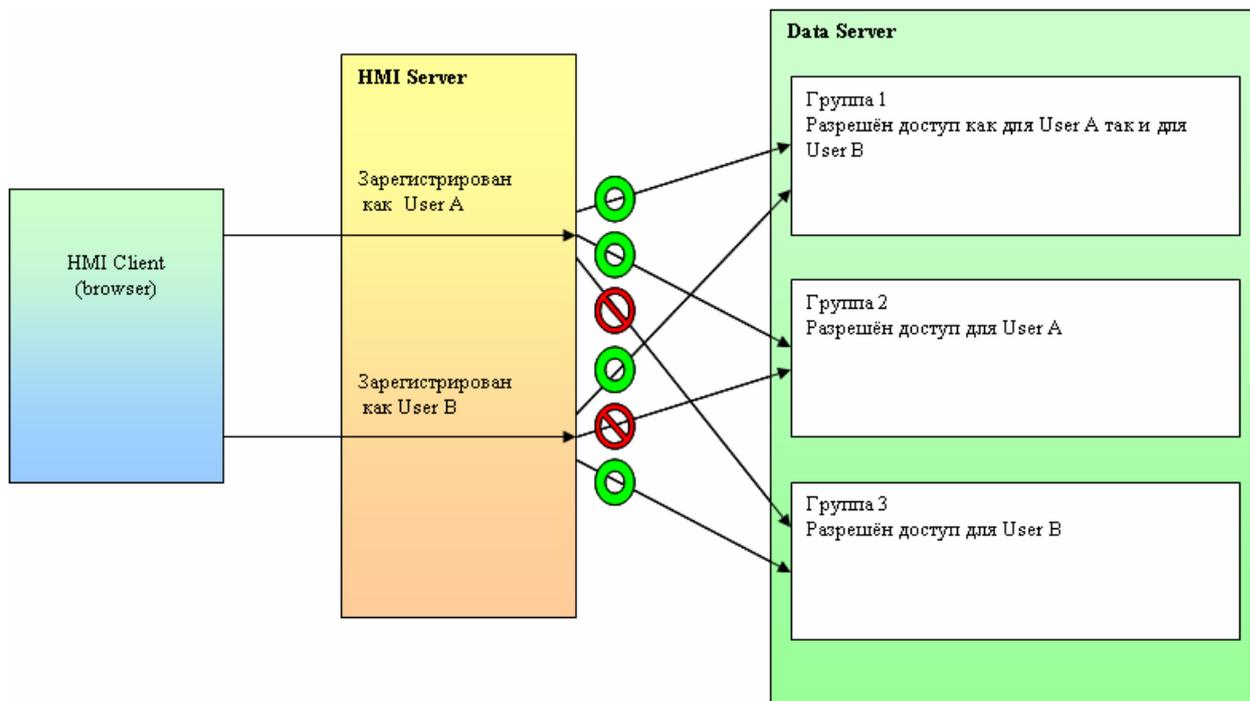


3.8 Функция защиты от неправомерного доступа.

Конфигуратор функции защиты от неправомерного доступа (**Security Builder**) определяет уровни доступа пользователей использующих функции VDS. Пользователи, являющиеся операторами, включаются в группы операторов VDS (**VDS Operator's group**) и являются членами этих групп. Существуют также группы пользователей операционной системы Windows (см. раздел [3.2.2](#)). Существует 7 **VDS Operator's group**.

Основной задачей функции защиты от неправомерного доступа является: ограничить возможности по оперированию с управляющими объектами в **Data Server** из HMI (**Human Machine Interface**) клиентов.

VDS устанавливает пределы уровней доступа, базированные на системе паролей пользователей, работающих с HMI Server. Поэтому уровни доступа даже к одинаковым графическим страницам будут отличаться в зависимости от того, с каким паролем вошёл в систему другой пользователь, использующий своё имя пользователя.

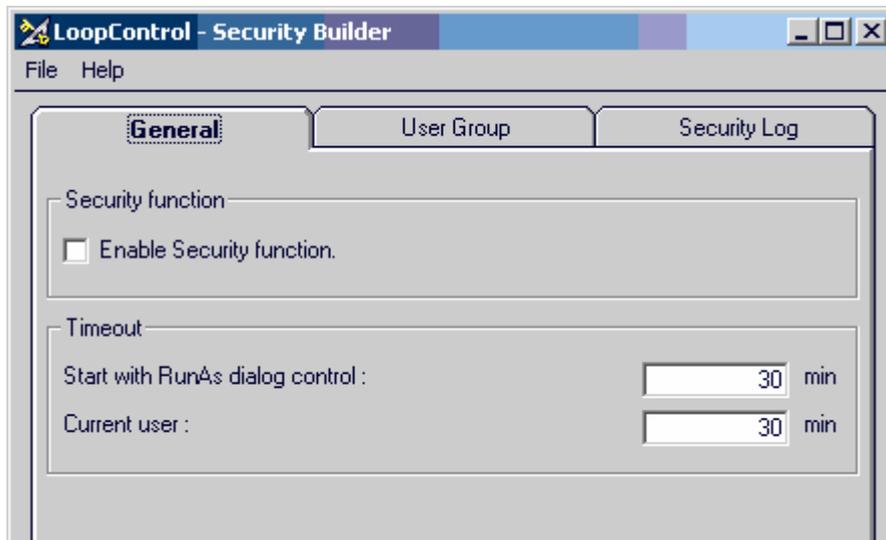


В примере на рисунке выше обозначены три группы пользователей Data Server.

- Группа 1 имеет право доступа как для User A, так и для User B. Группа 2 и Группа 3 имеют право доступа только для User A или только для User B соответственно.
- Когда пользователь входит под именем User A, он имеет право доступа к управляющим объектам в Группе 1 и Группе 2, но не имеет в Группе 3.
- Когда пользователь входит под именем User B, он имеет право доступа к управляющим объектам в Группе 1 и Группе 3, но не имеет в Группе 2.

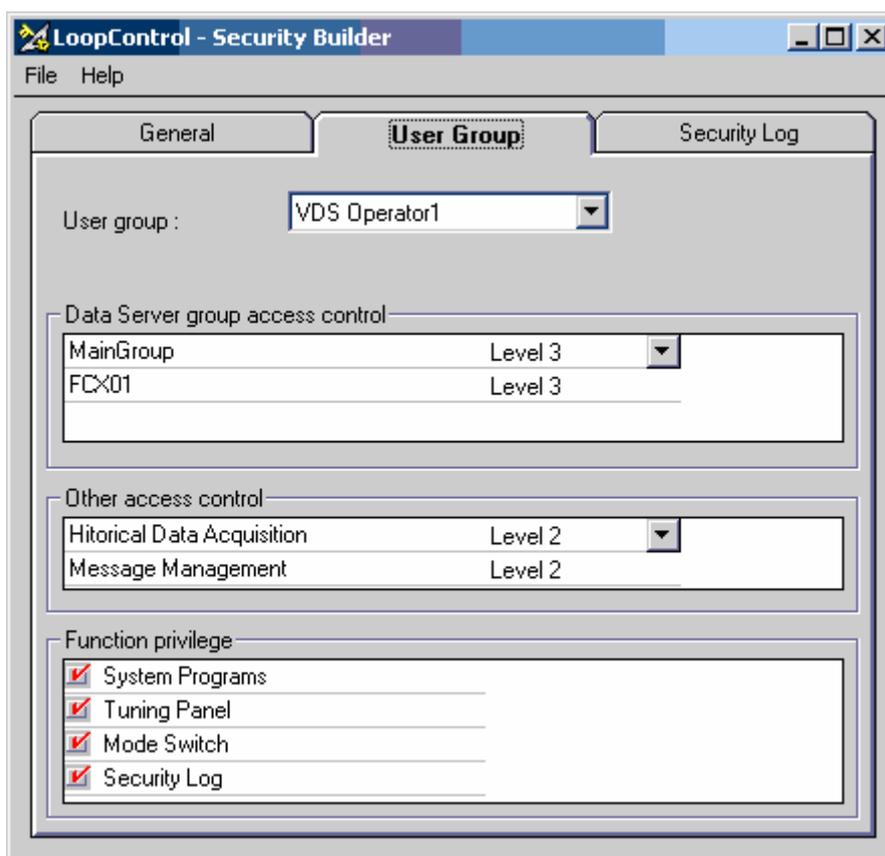
Оперирование из HMI клиента в **Data Server** ограничено сессиями. Сессии создаются для каждого браузера, и пользователь приписан к каждой сессии. Такая авторизация доступа, данная пользователю, используется для доступа к **Data Server**.

Конфигурирование защиты доступно через выбор опции “**Security Builder**” в выпадающем меню “**Tools**” **Object Builder**.

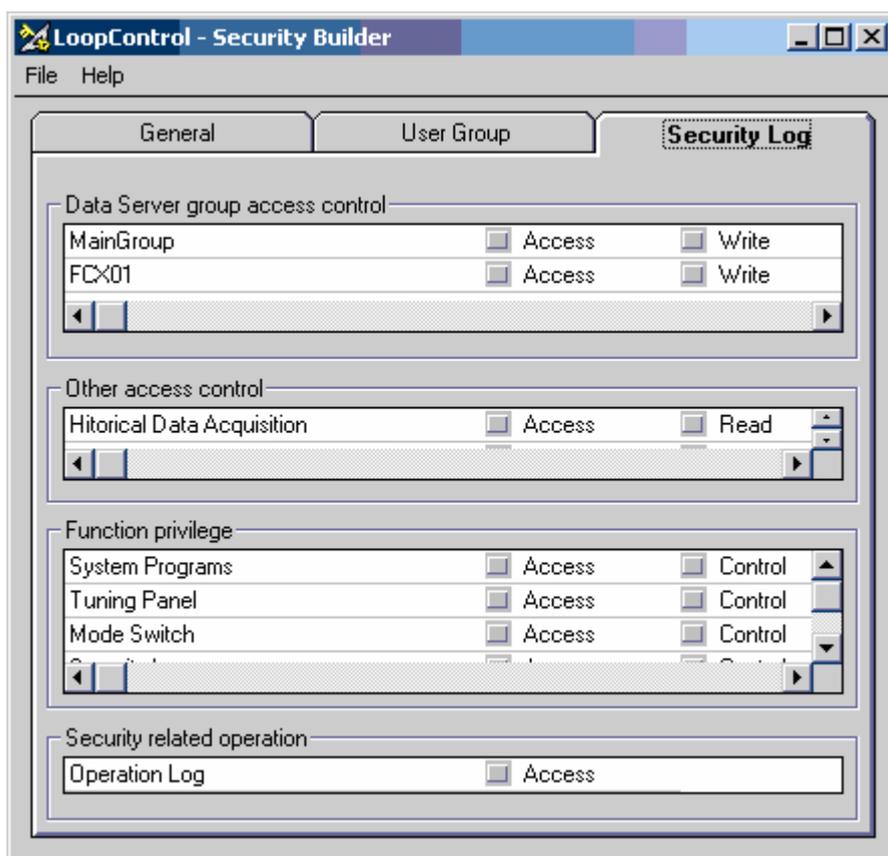


Установки защит определяются следующими категориями:

- **General** – закладка для установок общего характера для всех пользователей:
 - “Security function /Enable Security function” – разрешает функционирование системы защиты от неправомерного доступа;
 - “Timeout/Start with RunAs dialog control:” – устанавливает время ожидания ввода пароля;
 - “Timeout/Current user:” – устанавливает время ожидания для изменения текущего пользователя.



- **User Group** – закладка для конфигурирования индивидуальной защиты от неправомерного доступа:
 - **Data Server group access control** – устанавливает уровни доступа для каждого FCX и других контроллеров. Существуют 3 уровня, которые имеют следующие значения:
 1. **Level 1** – доступ только для чтения;
 2. **Level 2** – доступ по записи и чтению за исключением возможности изменения уставок алармов;
 3. **Level 3** – полный доступ как по чтению, так и по записи.
 - **Other access Control** – устанавливает уровни доступа для других функций VDS:
 1. **Historical Data Acquisition** – уровень доступа (2 уровня);
 2. **Message Management** – уровень доступа (2 уровня).
 - **Function Privilege** – устанавливает права доступа к функциям:
 1. System Programs – доступ к системным программам;
 2. Tuning Panel – доступ к настроечной панели;
 3. Mode Switch – доступ к переключению режимов;
 4. Security Log – доступ к системе защиты от неправомерного доступа.



- **Security Log** – закладка для конфигурирования доступа по чтению и по записи к регистрационным файлам каждой из вышеупомянутых категорий.

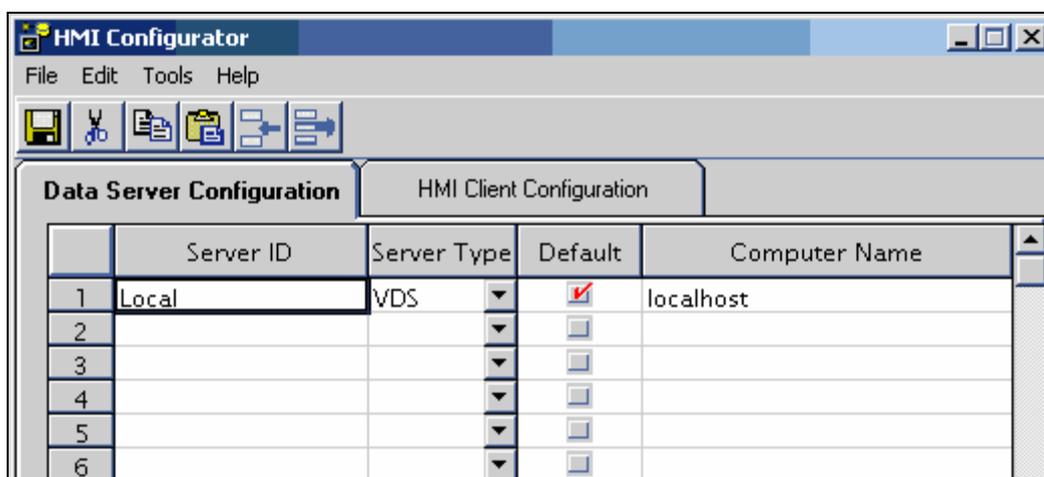
3.9 Клиент серверные функции HMI.

3.9.1 Конфигурирование HMI Server.

HMI сервер (**HMI Server**) читает данные из **Data Server** и делает их доступными HMI клиентам (**HMI Client**). Если **HMI Server** находится в той же PC что и **Data Server**, то конфигурирования не требуется. Однако, если они находятся на разных PC, то размещение **Data Server** необходимо специфицировать в **HMI Configurator**.

Конфигурирование Data Server:

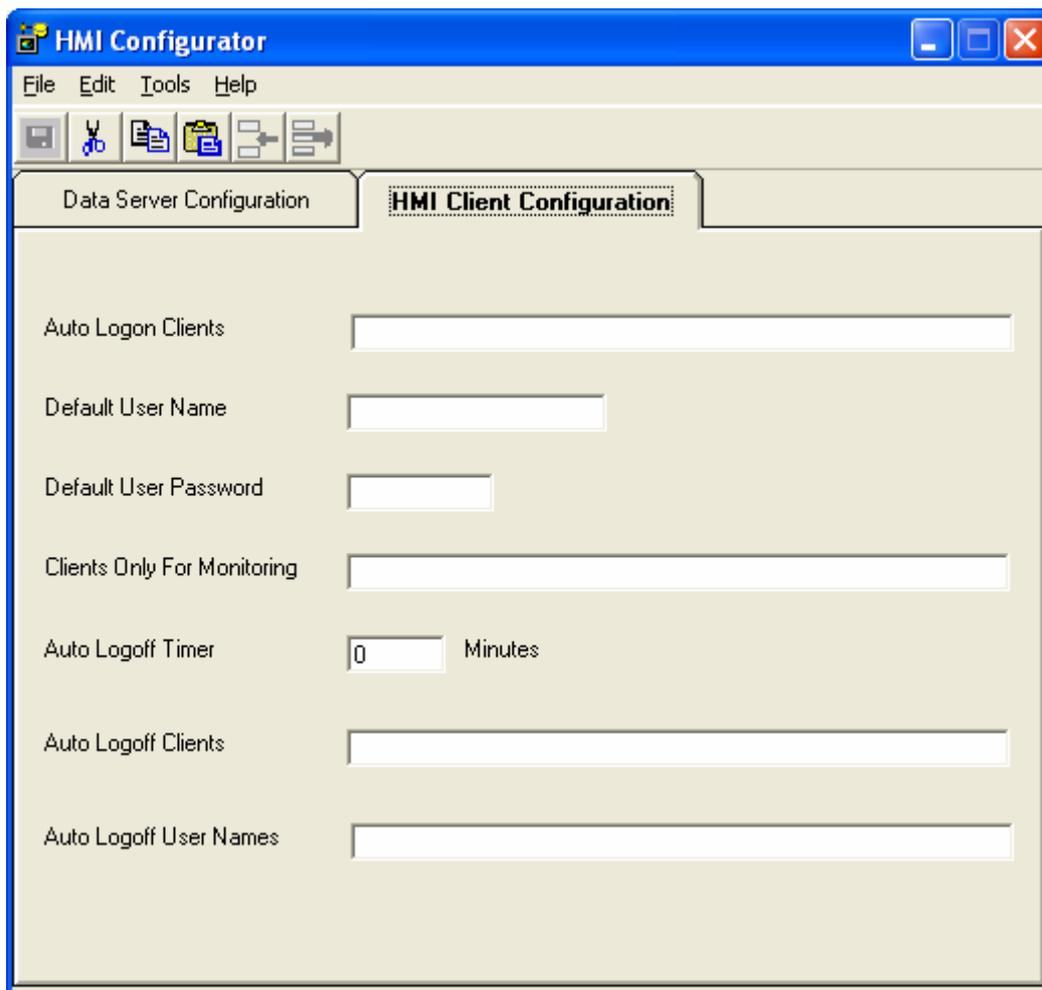
HMI Server может взаимодействовать со множеством серверов данных. Связь с этими серверами специфицируется в закладке “**Data Server Configuration**” следующим образом:



- **Server ID** – идентификатор Data Server ID (имя сервиса);
- **Server Type** – всегда “VDS”;
- **Default** – выбирает какой из Data Server является сервером по умолчанию для Web сервера;
- **Computer Name** – имя PC, в котором Data Server является резидентным.

Конфигурирование HMI Client:

Закладка для конфигурирования клиентов “**HMI Client Configuration**” устанавливает правила доступа для клиентских компьютеров следующим образом:



- **Auto Logon Clients** – специфицирует имена PC (или IP адреса), которые не требуют входа пользователя. Могут быть вписаны несколько имён, разделённых запятыми. Можно использовать групповой символ “*”. Если ввести в поле только “*”, то все HMI клиенты, подключённые к серверу, не будут требовать входа в систему (не рекомендуется). Если это поле введено, то поля “**Default User Name**” и “**Default User Password**” также должны быть введены;
- **Clients Only for Monitoring** – специфицирует имена PC (или IP адреса), которые имеют права доступа к серверу данных только по чтению, независимо от того, кто входит в систему. Если введен символ “*”, все клиенты будут иметь доступ только по чтению;
- **Auto Logoff Clients/User Names** – специфицирует имена PC (или IP адреса) или имена пользователей, которые будут выходить из системы автоматически по истечению периода времени, указанного в опции “**Auto Logoff Timer**”. Если введен символ “*”, все клиенты будут выходить из системы автоматически по истечению этого периода.

3.9.2 Функция операторского окна (Operator Display).

Операторские окна показываются как Web страницы Internet Explorer в Web клиенте.

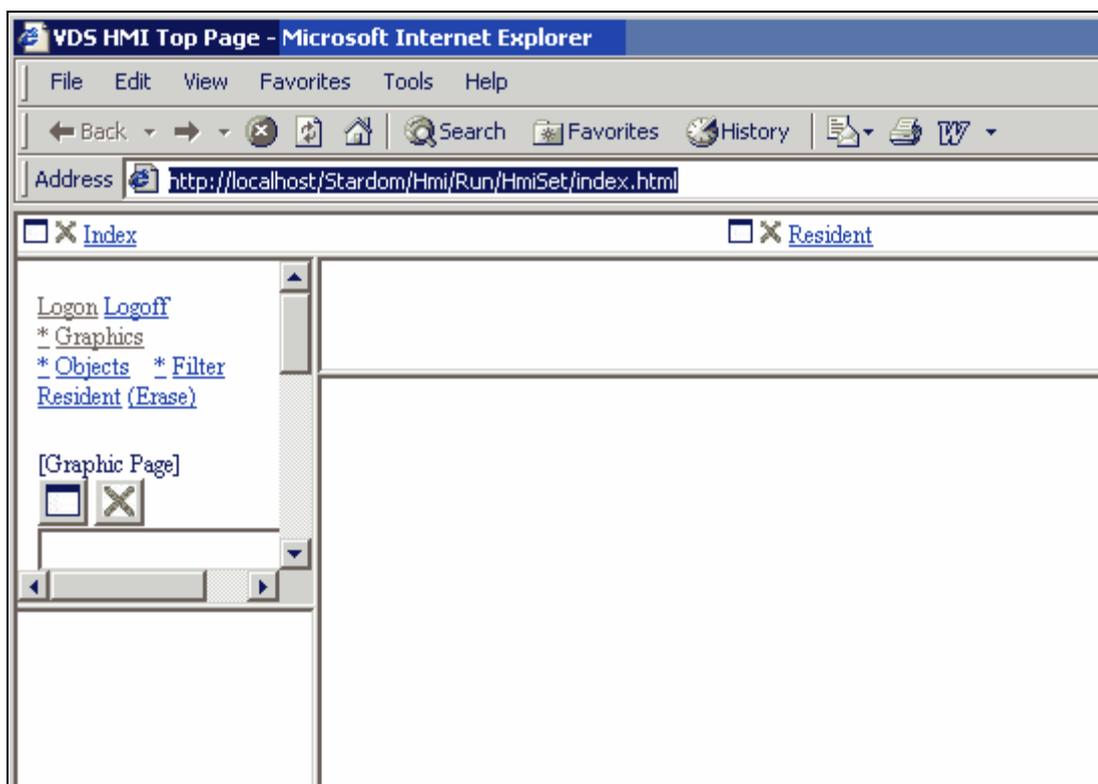
Существуют следующие типы окон:

- Logon/Logoff Windows – для защиты пользователя;
- Graphics Windows – графические окна;
- Objects Windows – отображают объекты управления в виде лицевых панелей.

Когда запускается Web клиент, используются следующие адреса для подключения к HMI Server:

`http://localhost/Stardom/Hmi/Run/HmiSet/index.html`

Где: “localhost” – имя (или IP адрес) Web сервера. Если клиент запущен на том же PC что и сервер, то используйте слово “localhost”.



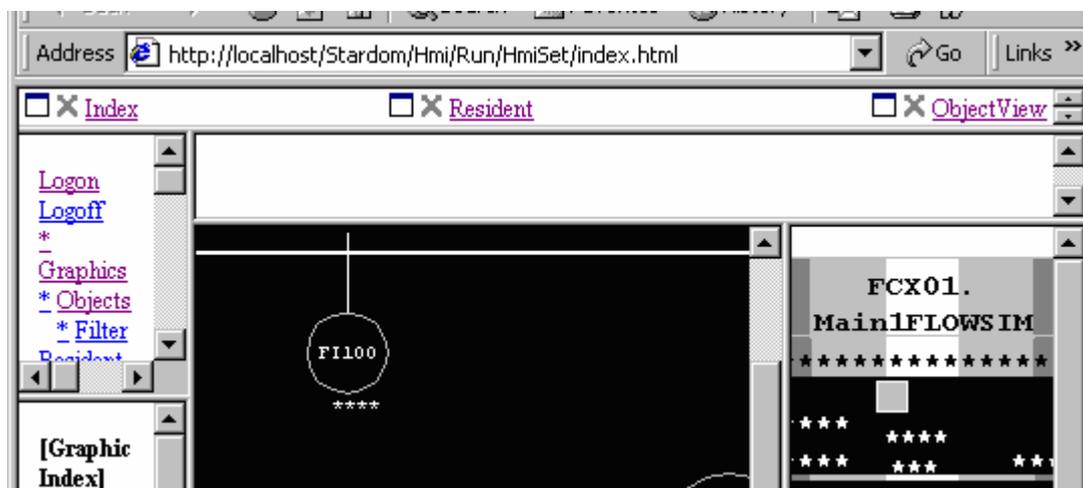
Для того, чтобы окна стали доступны, пользователь должен предварительно войти в сервер. Это выполняется щелчком по ссылке “Logon”, при этом появляется окно ввода имени пользователя и пароля:



По умолчанию поля пусты.

При желании пользователь может прятать или показывать внутри Web страницы дополнительные фреймы:

- **Index** – фрейм с левой стороны с перечнем графических окон и объектов;
- **Resident** – фрейм размещённый по центру для отображения содержимого графического окна;
- **Object View** – фрейм с правой стороны для отображения лицевых панелей.



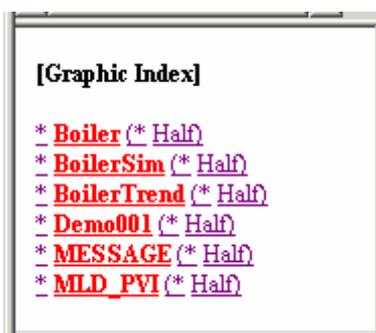
Эти фреймы могут быть закрыты нажатием кнопки “x”.

Отображение графических окон:

Для отображения графического окна щёлкните по ссылке “[Graphics](#)” для вывода перечня. Перечень графических окон отображается во фрейме расположенном ниже. При отображении графических окон используются следующие опции:

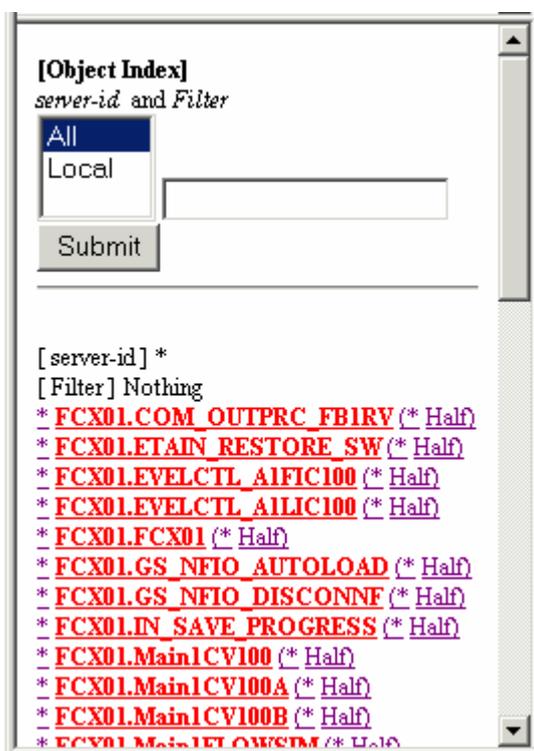
- Щелчок по имени графического окна вызывает появление полноразмерного показания графического окна во фрейме **Resident**.

- Щелчок по символу ‘*’ слева от имени графического окна вызывает появление полноразмерного показания графического окна в новом окне.
- Щелчок по ссылке “Half” вызывает появление показания графического окна половинного размера во фрейме **Resident**.
- Щелчок по символу ‘*’ слева от ссылки “Half” вызывает появление показания графического окна половинного размера в новом окне.



Отображение объектов:

Для отображения объекта щёлкните по ссылке “Objects” для вывода перечня. Перечень объектов отображается во фрейме расположенном ниже. При отображении объектов используются следующие опции:



- Щелчок по имени объекта вызывает появление лицевой панели для этого Control Object во фрейме **Object View**.

- Щелчок по символу ‘*’ слева от имени объекта вызывает появление полноразмерного показания лицевой панели для этого Control Object в новом окне.
- Щелчок по ссылке “Half” вызывает появление лицевой панели половинного размера для этого Control Object во фрейме **Object View**.
- Щелчок по символу ‘*’ слева от ссылки “Half” вызывает появление лицевой панели половинного размера для этого Control Object в новом окне.

Для облегчения поиска объекта используйте контекстный поиск по имени тега. Это необходимо, когда количество объектов в одном перечне слишком велико.