

Оглавление

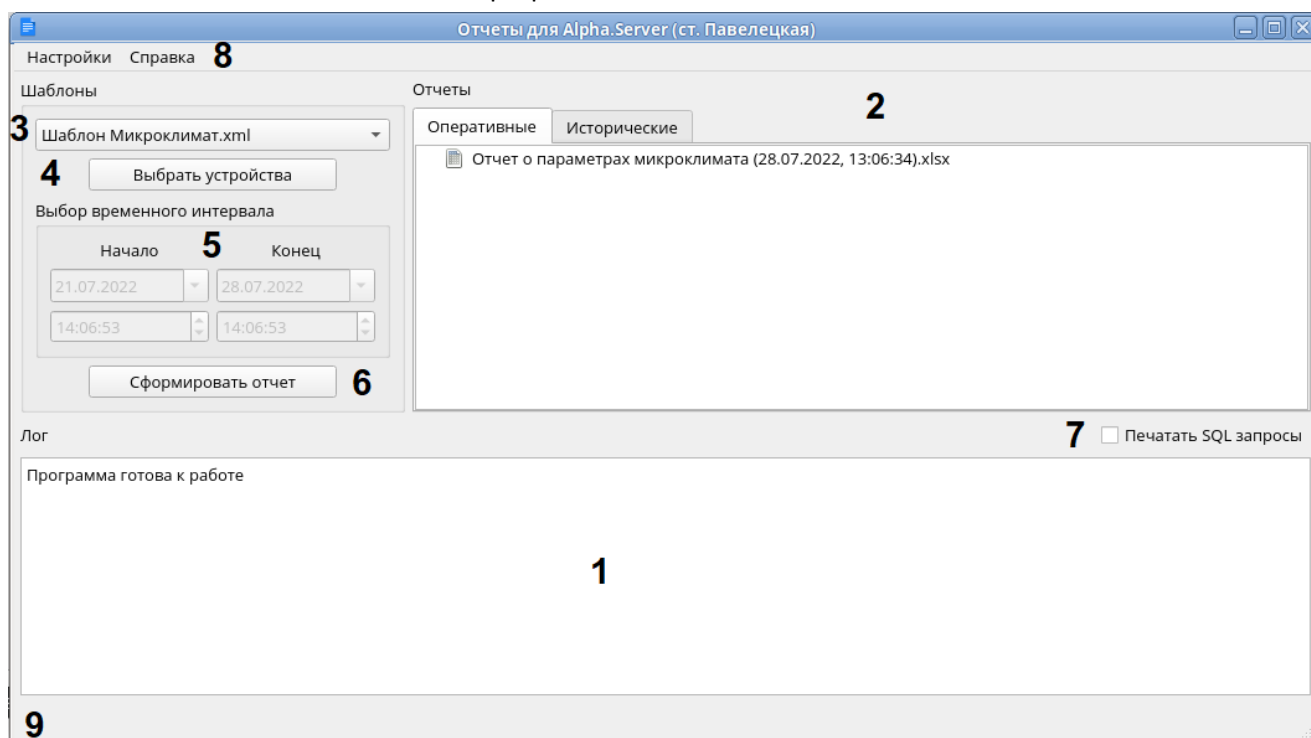
Отчеты для Alpha.Server	2
Описание интерфейса программы Отчеты для Alpha.Server	2
Запуск программы с параметрами	6
Создание шаблонов	8
Структура конфигурационного документа station.xml	8
Структура шаблона	9
Узел style.....	10
Узел structure	12
Узел types	14
Пример SQL запроса к оперативным данным	15
Пример SQL запроса к историческим данным.....	17

Отчеты для Alpha.Server

Для использования программы необходимо, чтобы в Alpha.Server был установлен модуль Alpha.RMap. Подробности по его установке и настройке есть в инструкции к модулю.

Описание интерфейса программы Отчеты для Alpha.Server

Ниже показано главное окно программы

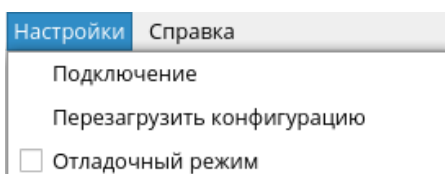


Рисунок

Цифрами обозначены следующие элементы:

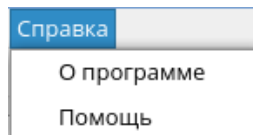
- 1 – Окно вывода информации о работе программы
- 2 – Отображение файлов и папок из директории ./reports, куда сохраняются отчеты
- 3 – Выбор шаблонов
- 4 – Выбор устройств в шаблоне, для которых будет сформирован отчет
- 5 – Для исторических отчетов выбор временного интервала
- 6 – Кнопка запуска генерации выбранного отчета
- 7 – Переключатель для отображения в логте текущих sql запросов
- 8 – Верхнее меню, дающее доступ к настройкам программы
- 9 – Статус бар со вспомогательной информацией

Меню настроек:



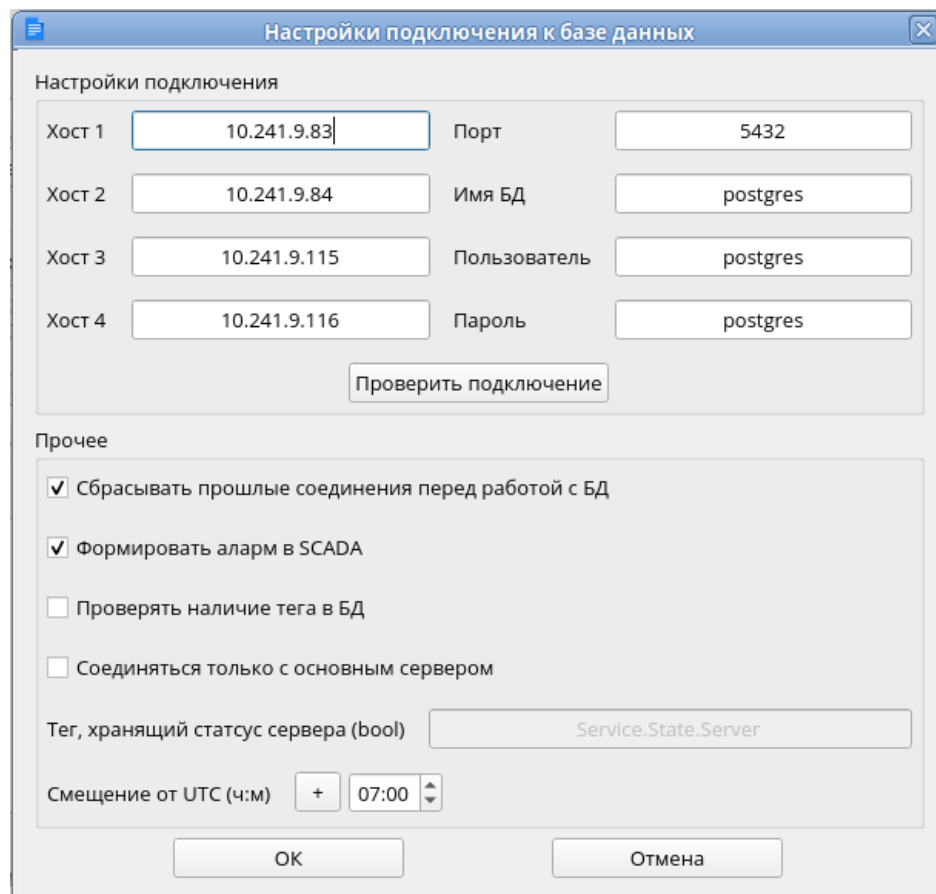
Рисунок

Меню справки:



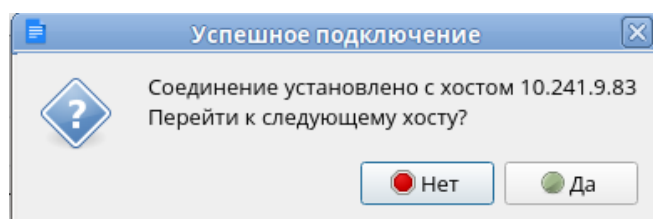
Рисунок

При нажатии на элемент меню (8) Настройки – Подключение, открывается окно настроек программы, которые позволяют изменить атрибуты узла <settings> файла station.xml, который лежит в директории templates и содержит в себе информацию о подключении к базе данных и об объектах на станции.



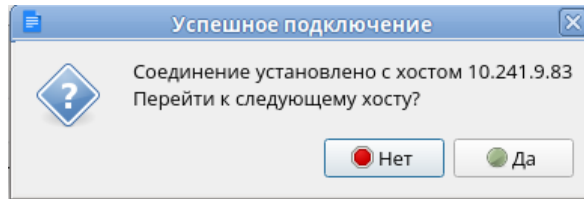
Рисунок

В открывшемся окне настраивается то, по каким ip адресам будет производиться подключение к БД, а так же информация, необходимая для непосредственного подключения, такая как порт, имя бд, пользователь бд и его пароль. С помощью кнопки Проверить подключение, можно проверить правильность настроек и подключиться к базе данных. В случае успеха, появится следующее окно:



Рисунок

В случае, если к хосту не удалось подключиться, программа выдаст ошибку подключения:



Рисунок

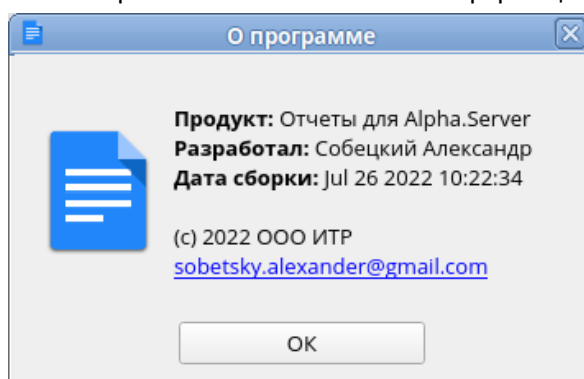
Прочие настройки:

- Сбрасывать прошлые соединения перед работой с БД – при подключении к БД программа завершает все прошлые свои сессии, отправляя команду в postgres;
- Формировать аларм в SCADA – программа записывает динамическое сообщение в сигнал Alarm.GenerateAlarm;
- Проверять наличие тега в БД – перед sql запросом сначала спрашивать базу данных, имеется ли данный тег агрегата;
- Соединяться только с основным сервером – если эта настройка отключена, то программа подключается к БД первого доступного хоста. Если включена, то помимо подключения, проверяется так же статусный сигнал сервера (обычно Service.State.Server) и только, если статусный сигнал равен true, то сервер считается основным и программа с ним работает, иначе – подключается и проверяет следующий по списку хост;
- Тег, хранящий статус сервера (bool) – путь до статусного сигнала в дереве OPC сервера;
- Смещение от UTC (ч:м) – смещение, учитываемое при формировании метки времени, которая приходит с сервера в формате UTC.

При выборе пункта настроек «Перезагрузить конфигурацию», программа заново загрузит список шаблонов и их содержимое, а так же заново прочтет конфигурацию файла station.xml.

Включение опции «Отладочный режим» позволяет построить тело отчета без соединения с базой данных.

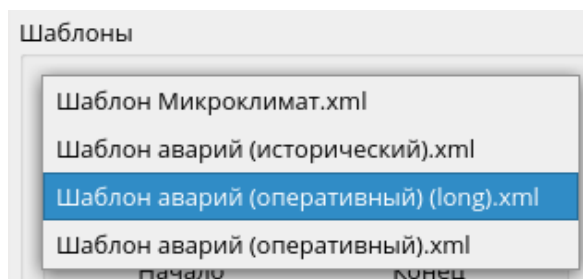
Пункт «О программе» меню Справка показывает окно с информацией о приложении:



Рисунок

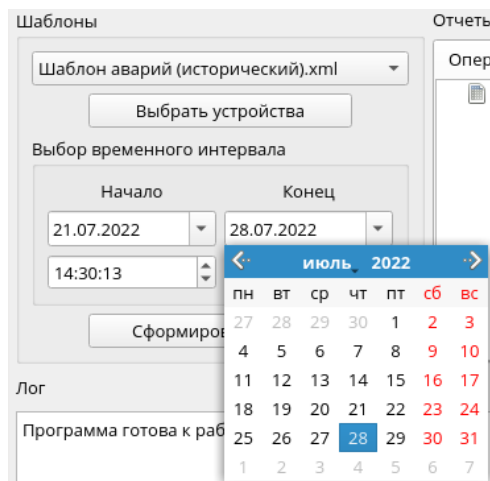
Пункт «Помощь» меню Справка открывает этот документ.

После запуска и успешной конфигурации программы пользователь должен выбрать шаблон из списка имеющихся в папке templates шаблонов:



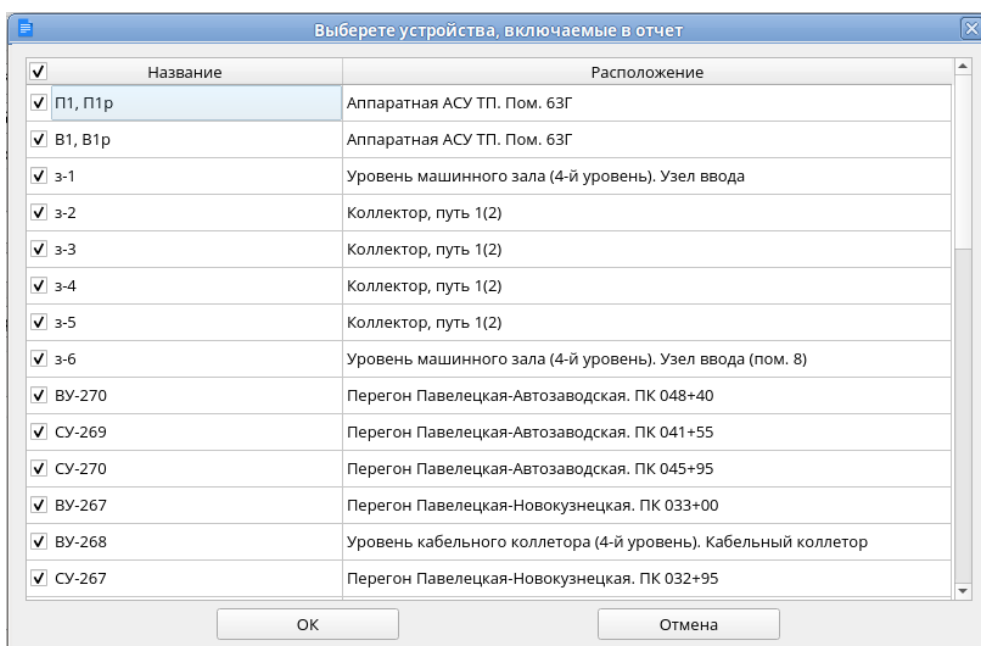
Рисунок

При выборе оперативных шаблонов элемент 5 остается в заблокированном состоянии, он разблокируется только для исторических шаблонов, позволяя выбрать временной интервал:



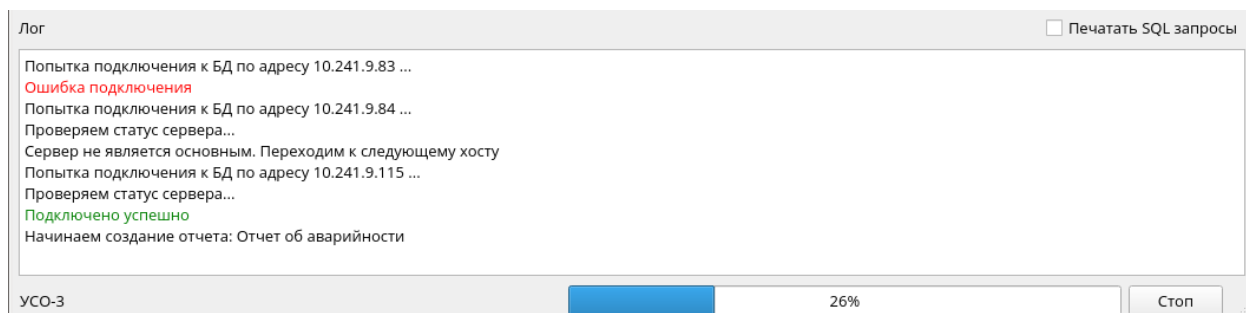
Рисунок

После выбора шаблона, можно выбрать устройства, которые будут включены в отчет, нажав на кнопку 'Выбрать устройства'. Откроется новое окно, где будет представлен список всех объектов, которые включены в этот отчет. Галочкой отмечены устройства, включенные в отчет. Лево́й клавишей мыши можно менять отображение в отчете того или иного объекта, кликая по чекбоксам.



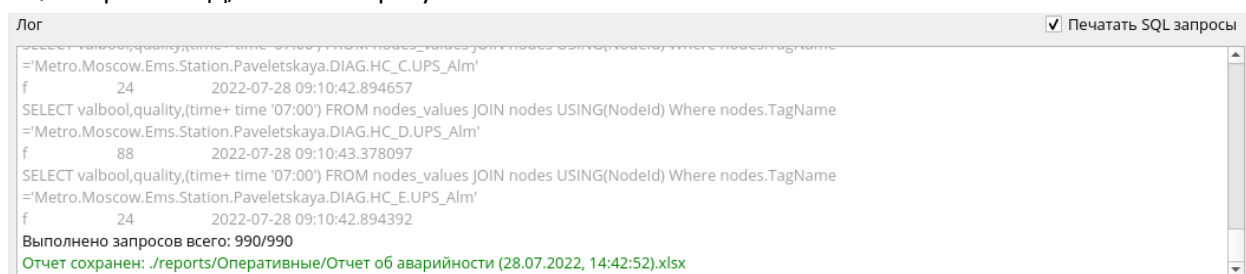
Рисунок

При нажатии кнопки Сформировать отчет, программа соединится с базой данных postgres, затем начнет формирование отчета по выбранному шаблону. В нижнем статус-баре (элемент 9) отображается информация о ходе программы, с возможностью остановки формирования отчета.



Рисунок

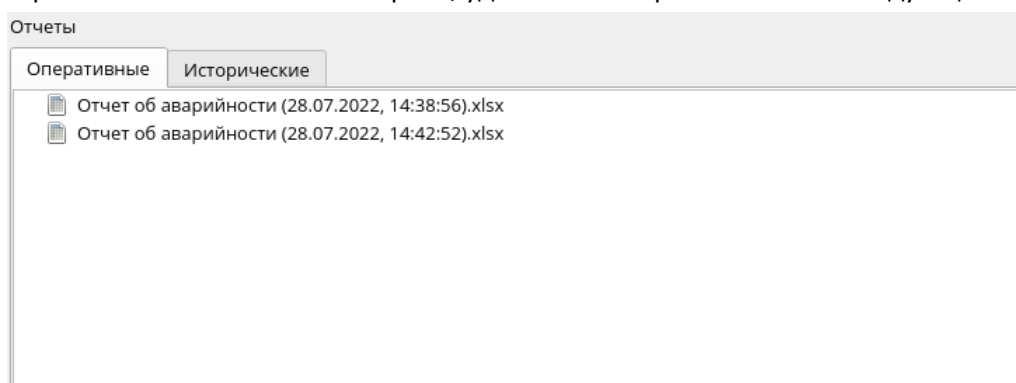
При отмеченном пункте «Печатать SQL запросы», в лог так же будут выводиться текущие SQL запросы к БД, а также их результат:



Рисунок

При сохранении отчета, программа добавляет к его имени дату и время, что бы можно было создавать несколько одинаковых отчетов без их перезаписи.

Сохраненные отчеты можно открыть, удалить или переименовать в следующем окне:



Рисунок

Запуск программы с параметрами

Программу отчетов можно запускать с параметрами, например, чтобы настроить генерацию отчетов по расписанию, и программа отчетов автономно время от времени запускалась планировщиком.

Параметры, которые можно использовать при запуске для автоматической генерации отчета:

--min	Открытие окна в свернутом состоянии
--no_fold	Убрать кнопку свертывания у окна
--modal	Окно открывается поверх других окон
--close <sec>	Указать временной интервал <sec>, после которого программа автоматически закроется
--screen <mon>	Указать номер монитора <mon> (по умолчанию открытие на 1-ом мониторе)
--template <name>	Указать имя шаблона <name>
--date <start end>	Указать временной интервал для исторических отчетов <start end> в формате <yyyy-MM-dd hh:mm:ss yyyy-MM-dd hh:mm:ss>. Или <lastX>, где X - число дней
--tag <tag>	Указать тег объекта <tag>, по которому будет сгенерирован отчет

Например, запуск отчетов следующим образом:

-template "Шаблон аварий (исторический).xml" -date "last10" -close 1 -min

Откроет программу отчеты в свернутом режиме (-min)

Загрузит для работы шаблон "Шаблон аварий (исторический).xml" (-template)

Выставит период времени по отчету – последние 10 дней (-date "last10")

И закроет программу через 1 секунду после сохранения отчета (-close 1)

А следующая строка:

-template "Шаблон аварий (исторический).xml" -tag "ORK.Masterskie.SHUP_1" -date "2022-03-10 11:50|2022-03-15 13:00" -close 20

Запустит программу в обычном несвернутом режиме

Загрузит для работы шаблон "Шаблон аварий (исторический).xml" (-template)

Но отчет будет формироваться только по одному тегу "ORK.Masterskie.SHUP_1" (-tag)

Программа будет автоматически закрыта через 20 секунд (-close)

В качестве временных интервалов (-date) используются конкретные даты и время

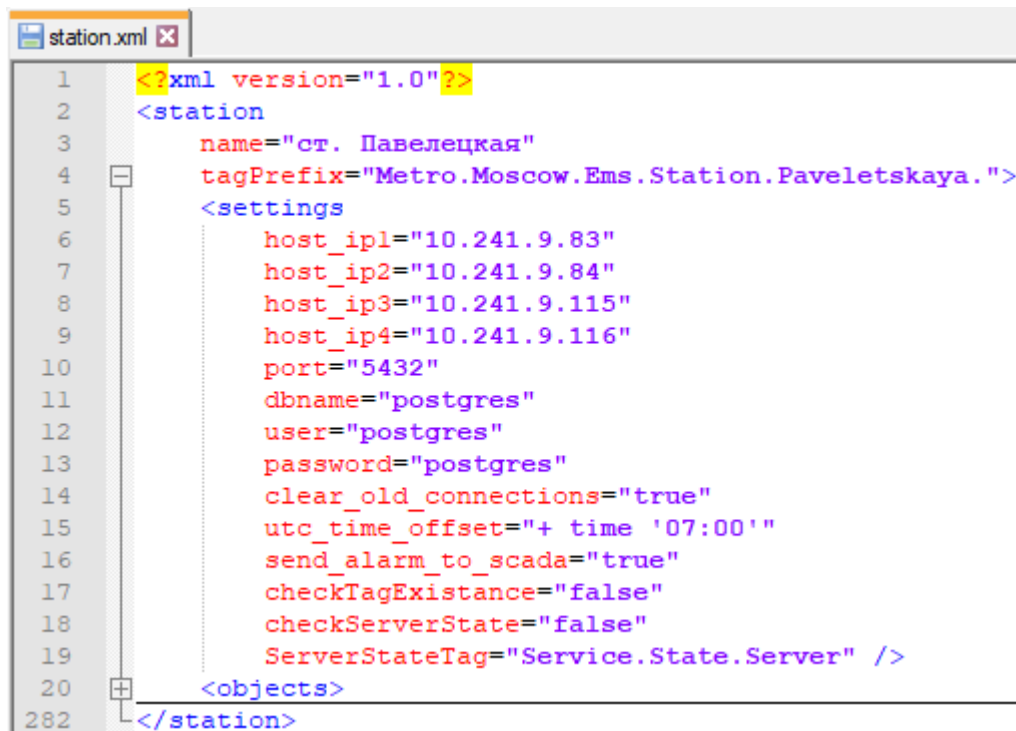
"start|end" = "2022-03-10 11:50|2022-03-15 13:00"

Значит дата начала периода "2022-03-10 11:50", а дата конца "2022-03-15 13:00"

Создание шаблонов

Структура конфигурационного документа station.xml

Все шаблоны должны располагаться в папке templates, которая располагается рядом с исполняемым файлом программы. Помимо шаблонов, в этой папке находится конфигурационный файл station.xml, который содержит информацию об объектах на станции и информацию о подключении к базе данных postgresql.



Рисунок

Узел <station> содержит в себе следующие узлы:

- <settings> - для настройки подключения к базе данных
- <objects> - содержит информацию по объектам на станции

Узел <station> имеет атрибуты:


- name – имя станции
- tagPrefix – префикс, который будет добавляться перед тегом объекта из узла <objects>

С помощью следующих атрибутов узла <settings> настраивается подключение к базе данных postgresql:

- host_ip – ip адрес для подключения к базе данных (до 10 ip узлов)
- port – порт для подключения
- dbname – имя базы данных
- user – имя пользователя БД
- password – пароль пользователя
- clear_old_connection – флаг, принимающий значения true и false, показывающий необходимость принудительного сброса всех своих прошлых соединений в БД, перед началом работы, если таковые могли остаться

- utc_time_offset – положительное или отрицательное смещение 'чч:мм' от времени utc
- send_alarm_to_scada – формировать аларм в Alpha.Server об успешном сохранении сформированного отчета
- checkTagExistance – перед SQL запросом проверять наличие тега в базе данных
- checkServerState – соединяться только с основным сервером (если сервер не в резерве, а в работе). Состояние сервера определяется по тегу статуса
- ServerStateTag – орс тег, хранящий статусный сигнал сервера

Узел <objects> содержит внутри себя узлы <obj>, каждый из которых описывает один существующий объект на станции.



```

checkTagExistance="false"
checkServerState="false"
ServerStateTag="Service.State.Server" />
<objects>
  <obj
    name="П1, П1р"
    tag="US03.P1_V1"
    location="Аппаратная АСУ ТП. Пом. 63Г"
    common_type="Вентиляция" />
  <obj
    name="В1, В1р "
    tag="US03.P1_V2"
    location="Аппаратная АСУ ТП. Пом. 63Г"
    common_type="Вентиляция" />

```

Рисунок

Каждый узел <obj> содержит следующие атрибуты:

- name – имя объекта
- tag – его тег. tagPrefix + tag = полный тег объекта
- location – расположение объекта
- common_type – тип объекта,

Узел <objects> позволяет один раз описать информацию об объектах на станции, а в шаблонах оперировать только тегами этих объектов, для их идентификации.

Структура шаблона

Как уже было сказано в предыдущей главе, все шаблоны находятся в папке templates. Рассмотрим подробнее структуру шаблона.

Каждый шаблон содержит узел <template> со следующими атрибутами:

- mode – режим оперативный (oper) или исторический (history), от этого будет зависеть возможность выбора временных интервалов в программе
- name – имя, которое будет использовано при сохранении отчета по данному шаблону
- save_path – папка, в которую будет сохранен шаблон. Настоятельно рекомендовано сохранять отчеты только в папку reports, так как программа отображает список отчетов только из этой папки. Дополнительные директории внутри папки reports должны быть созданы заранее.

- includeNullResult – флаг, принимающий значения false или true, который определяет, включать ли в отчет данные по объектам, если SQL запрос по объекту не вернул никаких данных.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    save_path если не обозначен, то отчет сохраняется рядом с exe.
    Если обозначен, например save_path="C:\\Отчеты\\",
    то убедитесь, что путь существует!
    Программа не создает сама новые папки
-->
<template
    mode="oper"
    name="Отчет об аварийности"
    save_path=".\\reports\\Оперативные\\"
    includeNullResult = "false">
    ...
</template>
```

Рисунок

В свою очередь, узел <template> содержит узлы:

- <styles> - стили, которые применимы к конкретной ячейке excel
- <types> - типы формирования отчета, т.е. для каждого объекта можно переопределить поведение выгрузки данных в отчет, создав новый тип
- <structure> - структура отчета

Узел style

Начнем с узла <style>. Вместо того, чтобы каждой ячейке в отчете задавать цвет текста, название шрифта, размер и другие параметры, можно один раз определить стиль с этими параметрами, а в нужной ячейке указать название этого стиля, чтобы применить к ней все параметры, описанные в стиле.

Каждый стиль может содержать следующие атрибуты:

- textBold – флаг жирности шрифта (true – жирный, false – нет)
- textSize – размер текста (число)
- textFontName – имя шрифта (Arial, Times New Roman, ...)
- textColor – цвет текста (в формате argb или спец. слово)
- textAlign_h – выравнивание текста по горизонтали
- textAlign_w – выравнивание текста по вертикали
- cellColor – цвет заливки ячейки (в формате argb или спец. слово)
- cellBorder – отрисовать границы у ячейки

Цвет через атрибуты textColor и cellColor можно задавать через hex значения в формате ARGB, например FF2F75B5 – голубой цвет:

- FF – alpha (альфа канал)
- 2F – red (красный)

- 75 – green (зеленый)
- B5 – blue (голубой)

Цвет так же можно задать одной из предопределенных констант:

- @black - черный
- @white - белый
- @red - красный
- @darkred – темно-красный
- @blue - голубой
- @darkblue – темно-голубой
- @green - зеленый
- @darkgreen – темно-зеленый
- @yellow - желтый
- @darkyellow – темно-желтый

Горизонтальное выравнивание textAlign_h может принимать следующие значения:

- general – по значению
- left – по левому краю
- center – по центру
- right – по правому краю
- fill – с заполнением
- justify – по ширине
- center_continuous – по центру выделения
- distributed - равномерно

Вертикальное выравнивание textAlign_w может принимать следующие значения:

- top – по верхнему краю
- center – по центру
- bottom – по нижнему краю
- justify – по высоте
- distributed - равномерно

Атрибут cellBorder может принимать следующие значения:

- none – без внешней границы
- dashdot – тире точка
- dashdotdot – тире точка точка
- dashed –тире
- dotted – точка
- double_ - двойная линия
- hair – ворсинистая граница
- medium – средняя граница
- mediumdashdot – средняя тире точка
- mediumdashdotdot – средняя тире точка точка
- mediumdashed – средняя тире
- slantdashdot – наклонная точка

- thick – толстая граница
- thin – тонкая граница

На следующем рисунке показан пример двух стилей h1 и h2

```
<styles>
  <!-- Стили шапки -->
  <h1
    textBold="true"
    textSize="22"
    textFontName="Arial"
    textColor="@white"
    textAlign_h = "center"
    textAlign_w = "center"
    cellColor="FF2F75B5"
    cellBorder="thin"
  />
  <h2
    textBold="true"
    textSize="14"
    textFontName="Arial"
    textColor="FF000000"
    textAlign_h = "center"
    textAlign_w = "center"
    cellColor="FFDDEBF7"
    cellBorder="thin"
  />
```

Рисунок

Узел structure

Этот узел описывает шапку будущего отчета и определяет объекты, которые будут включены в отчет.

Шапка строится построчно вниз слева на право. Каждый узел <row> определяет одну строку в отчете .xlsx начиная с ячейки A1:

	A	B	C	D	E
1					
2					
3					
4					
5					

Рисунок

Каждая строчка <row> может содержать любое количество ячеек <cell>, которые будут напечатаны в отчете в данной строке слева на право. Узел <row> не имеет своих атрибутов.

Узел <cell> настраивает значение и формат одной ячейки в отчете .xlsx и может иметь следующие атрибуты:

- value – текст внутри ячейки
- style – имя стиля из узла <styles>, определяющий формат текста в этой ячейке
- height – высота этой строки, в которой находится ячейка (опционально)
- width – ширина столбца, в котором находится ячейке (опционально)
- mergeCells – количество ячеек, которые необходимо объединить слева на право, начиная с текущей ячейки (опционально)

Ниже показана одна из возможных шапок отчета:

```
<structure>
  <!-- Шапка -->
  <row>
    <cell value = "Отчет об аварийности оборудования" style="h1" mergeCells="7" height="55" />
  </row>
  <row>
    <cell value = "Оперативный" style="h2" mergeCells="7" height="30" />
  </row>
  <row>
    <cell value = "Время генерации отчета: %report_time%" style="h2" mergeCells="7" height="30" />
  </row>
  <row>
    <cell value="Наименование" style="h3" width="50" />
    <cell value="Объект" style="h3" width="20" />
    <cell value="Тип объекта" style="h3" width="20" />
    <cell value="Расположение" style="h3" width="60" />
    <cell value="Значение" style="h3" width="20" />
    <cell value="Качество" style="h3" width="20" />
    <cell value="Метка времени" style="h3" width="40" />
  </row>
  <!-- Данные -->
</structure>
```

Рисунок

Каждая из первых трех строк будет состоять из одной ячейки, объединенной с семью другими в строке, со своим стилем и высотой.

Четвертая строка является шапкой нашей таблицы и состоит из семи ячеек, каждая из которых задает ширину своего столбца.

Вот, как выглядит итоговая шапка в отчете:

A	B	C	D	E	F	G
Отчет об аварийности оборудования						
Оперативный						
Время генерации отчета: 25.03.2022, 15:31:31						
Наименование	Объект	Тип объекта	Расположение	Значение	Качество	Метка времени

Рисунок

Вместо текста у атрибута value ячейки cell можно вставлять управляющие последовательности вида %код%.

Управляющие последовательности, которые можно использовать в шапке отчета:

- %report_time% - время создания отчета
- %report_dt_start% - начальный интервал времени для исторического отчета
- %report_dt_end% - конечный интервал времени для исторического отчета

Ниже шапки отчета (но внутри узла <structure>) должен располагаться узел <report_data>, в котором перечислены теги объектов на станции, которые будут включены в отчет. Для каждого объекта необходимо указать тег и тип представления объекта в отчете.

```
<!-- Данные -->
<report_data>
  <!-- Вентиляция -->
  <data tag="US03.P1_V1" type="type_VENT"/>
  <data tag="US03.P1_V2" type="type_VENT"/>
</report_data>
```

Рисунок

Каждый из узлов <data> должен содержать два обязательных атрибута:

- tag – тег объекта из документа station.xml
- type – имя типа из узла <types>

Иначе говоря, узел report_data определяет какие объекты будут включены в отчет и какая информация для каждого объекта будет отображаться (в зависимости от атрибута type).

Узел types

Данный узел содержит типы, определяющие то, как информация по конкретному объекту будет отображена в отчете.

Каждый тип имеет свое уникальное имя и два атрибута:

- single_mode – режим одиночного отображения результатов sql запроса (true/false)
- sql_query – строка SQL запроса

Так же каждый тип должен обязательно содержать следующие узлы:

- <tags> - список тегов объекта, по которым будет сделан отчет
- <row> - строка, которая печатается в отчет по результату sql запроса из sql_query

```
<types>
  <type_VENT      single_mode="true" sql_query="SELECT valbool,quality,%time% FROM nodes_values
    <tags>
      <tag name="Неисправность" value=".State.State6" />
    </tags>
    <row>
      <cell value="%obj_name%. %tag_name%" style="cell_Result_left" />
      <cell value="%obj_name%" style="cell_Result_center" />
      <cell value="%obj_type%" style="cell_Result_center" />
      <cell value="%station_name%. %obj_location%" style="cell_Result_left" />
      <cell value="%sql_result_1%" style="cell_Result_center" />
      <cell value="%sql_result_2%" style="cell_Result_center" />
      <cell value="%sql_result_3%" style="cell_Result_center" />
    </row>
  </type_VENT>
  <type_Z      single mode="true" sql query="SELECT valbool,quality,%time% FROM nodes values
  <type_VOU      single mode="true" sql query="SELECT valbool,quality,%time% FROM nodes values
  <type_Vsh      single mode="true" sql query="SELECT valbool,quality,%time% FROM nodes values
  <type_USO      single mode="true" sql query="SELECT valbool,quality,%time% FROM nodes values
  <type_AVR      single mode="true" sql query="SELECT valbool,quality,%time% FROM nodes values
  <type_HC      single mode="true" sql query="SELECT valbool,quality,%time% FROM nodes values
</types>
```

Рисунок

Каждый узел <tag> содержит следующие атрибуты

- name – название тега
- value – тег относительно объекта

Узел <row> состоит из узлов <cell>, атрибуты которых были описаны в предыдущей главе. В атрибут value узла <cell> так же можно печатать управляющие последовательности.

Управляющие последовательности, которые можно использовать внутри каждого типа в ячейках cell и в строке sql_query:

- %obj_name% - имя объекта из файла station.xml (атрибут name узла obj)
- %obj_location% - расположение объекта из файла station.xml (атрибут location узла obj)
- %obj_type% - тип объекта из файла station.xml (атрибут common_type узла obj)
- %tag_name% - название тега (атрибут name узла tag)
- %station_name% - имя станции из файла station.xml (атрибут name узла station)
- %full_tag% - полный тег, состоящий из префикса станции, тега объекта и тега типа:
- %report_dt_start% - начальный интервал времени для исторического отчета
- %report_dt_end% - конечный интервал времени для исторического отчета

- %sql_result_i% - i-й результат SQL запроса

Дополнительные управляющие последовательности для строки sql_query:

- %time% - обращение к элементу БД time с учетом смещения utc:
time + time'чч::мм'
- %actualtime% - обращение к элементу БД actualtime time с учетом смещения utc:
actualtime + time'чч::мм'
- %utc_offset% - значение смещения utc без знака + или -.

Кроме прочего, внутри cell можно писать и EXCEL формулы. Например, следующая формула (пример для LibreOffice) печатает ненулевое значение складывая его предварительно с числом 500:

value="=IF(%sql_result_1%<>0, %sql_result_1%+500, 0)"

Пример SQL запроса к оперативным данным

Покажем на примере, каким образом программа делает отчет.

Пусть узел <report_data> состоит из всего одного объекта с типом type_V:

```
<report_data>
  <data tag="ORK.Masterskie.SHUV_1.V1" type="type_V"/>
</report_data>
```

Рисунок

Информация об объекте в файле station.xml представлена следующим образом:

```
<obj
  name="ШУВ1. В1"
  tag="ORK.Masterskie.SHUV_1.V1"
  location="ОРК1. Здание мастерских"
  common_type="Вентиляция В" />
```

Рисунок

Сам тип type_V выглядит так:

```
--
<type_V single_mode="true" sql_query="SELECT valbool,quality,%time% FROM nodes_values JOIN n
  <tags>
    <tag name="Авария" value=".V.Alarm" />
    <tag name="Пожар" value=".FireAlarm" />
  </tags>
  <row>
    <cell value="%obj_name%. %tag_name%" style="cell_Result_left" />
    <cell value="%obj_name%" style="cell_Result_center" />
    <cell value="%obj_type%" style="cell_Result_center" />
    <cell value="%station_name%. %obj_location%" style="cell_Result_left" />
    <cell value="%sql_result_1%" style="cell_Result_center" />
    <cell value="%sql_result_2%" style="cell_Result_center" />
    <cell value="%sql_result_3%" style="cell_Result_center" />
  </row>
</type_V>
```

Строка sql_query выглядит так:

```
SELECT valbool,quality,%time% FROM nodes_values JOIN nodes USING(NodeId) Where
nodes.TagName = '%full_tag%'
```

Тогда программа, подставив все данные вместо %кодовых слов% получит следующее:

```
<row>
  <cell value="ШУВ1. В1. Авария"
  <cell value="ШУВ1. В1"
  <cell value="Вентиляция В"
  <cell value="Электродепо Сокол. ОРК1. Здание мастерских"
  <cell value="%sql_result_1%"
  <cell value="%sql_result_2%"
  <cell value="%sql_result_3%"
</row>
```

Затем программа возьмет содержимое строки sql_query и для каждого узла <tag> в типе <type_V> сформирует SQL запрос, заменив управляющее слово %full_tag% на полный тег, который складывается из префикса станции, тега самого объекта и тега, относительно объекта, по которому будет сделан отчет. А так же заменит управляющее слово %time% на выражение, которое необходимо, что бы получить информацию со смещением относительно UTC.

В итоге, будет сформировано 2 запроса:

```
SELECT valbool,quality, (time+ time '07:00') FROM nodes_values JOIN nodes USING(NodeId) Where
nodes.TagName ='Metro.Moscow.Depo.Sokol.ORK.Masterskie.SHUV_1.V1.V.Alarm'
```

```
SELECT valbool,quality, (time+ time '07:00') FROM nodes_values JOIN nodes USING(NodeId) Where
nodes.TagName ='Metro.Moscow.Depo.Sokol.ORK.Masterskie.SHUV_1.V1.FireAlarm'
```

База данных по каждому запросу вернет строку, состоящую из трех запрашиваемых колонок значение valbool, качество quality и метка времени timestamp.

Вывод данных			
	valbool boolean	quality integer	?column? timestamp without time zone
1	f	24	2022-03-15 13:56:31.342132

Вывод данных			
	valbool boolean	quality integer	?column? timestamp without time zone
1	f	24	2022-03-15 13:56:31.342132

Рисунок

Если бы мы вместо управляющего слова %time% мы написали бы просто time, т.е. запрос был бы вида:

```
SELECT valbool,quality, time FROM nodes_values JOIN nodes USING(NodeId) Where nodes.TagName
='Metro.Moscow.Depo.Sokol.ORK.Masterskie.SHUV_1.V1.V.Alarm'
```

То и ответ от базы данных был бы следующий (обратите внимание на разницу во времени)

	valbool boolean	quality integer	time timestamp without time zone
1	f	24	2022-03-15 06:56:31.342132

Рисунок

Подробнее о том, как строить SQL запросы, а так же о том, какие данные можно запрашивать у БД сервера, написано в документации к модулю RMap.

Что бы забрать колонки из строки результата, используются кодовые слова %sql_result_i%, где i – номер результата, начиная с 1.

Т.е. в данном случае, %sql_result_1% это строка valbool, %sql_result_2% - качество, а %sql_result_3% - метка времени.

А вот так визуально выглядит результирующий отчет в формате .xlsx:

	A	B	C	D	E	F	G
1	Отчет об аварийности оборудования						
2	Оперативный						
3	Время генерации отчета: 28.03.2022, 11:02:57						
4	Наименование	Объект	Тип объекта	Расположение	Значение	Качество	Метка времени
5	ШУВ1. В1. Авария	ШУВ1. В1	Вентиляция В	Электродепо Сокол. ОРК1. Здание мастерских	0	24	2022-03-15 13:56:31.342132
6	ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродепо Сокол. ОРК1. Здание мастерских	0	24	2022-03-15 13:56:31.342132

Рисунок

Пример SQL запроса к историческим данным

Рассмотрим пример запроса к историческим данным того же объекта. Для этого немного преобразуем шапку отчета, в нее добавится информация о временном периоде, за который будут запрашиваться данные. Для этого используются управляющие символы %report_dt_start% и %report_dt_end%. Так же уберем столбец качества сигнала, который выводился в оперативном отчете.

```

<!-- Шапка -->
<row>
  <cell value = "Отчет об аварийности оборудования" style="h1" mergeCells="6" height="55" />
</row>
<row>
  <cell value = "За период: %report_dt_start% - %report_dt_end% " style="h2" mergeCells="6" height="30" />
</row>
<row>
  <cell value = "Время генерации отчета: %report_time%" style="h2" mergeCells="6" height="30" />
</row>
<row>
  <cell value="Наименование" style="h3" width="50" />
  <cell value="Объект" style="h3" width="20" />
  <cell value="Тип объекта" style="h3" width="20" />
  <cell value="Расположение" style="h3" width="60" />
  <cell value="Значение" style="h3" width="20" />
  <cell value="Метка времени" style="h3" width="40" />
</row>

```

Рисунок

Тип будет выглядеть следующим образом:

```

<type_V          single_mode="false" sql_query="SELECT valbool,%time% FROM nodes_history h .
<tags>
  <tag name="Авария" value=".V.Alarm" />
  <tag name="Пожар" value=".FireAlarm" />
</tags>
<row>
  <cell value="%obj_name%. %tag_name%" style="cell_Result_left" />
  <cell value="%obj_name%" style="cell_Result_center" />
  <cell value="%obj_type%" style="cell_Result_center" />
  <cell value="%station_name%. %obj_location%" style="cell_Result_left" />
  <cell value="%sql_result_1%" style="cell_Result_center" />
  <cell value="%sql_result_2%" style="cell_Result_center" />
</row>
</type_V>

```

Рисунок

Обратите внимание, что так как столбец качество мы решили не выводить, то и количество управляющих слов %sql_result_i% сократилось.

Атрибут single_mode равен false, чуть позже покажем, на что он влияет. Строка sql_query выглядит так:

```
SELECT valbool,%time% FROM nodes_history h JOIN nodes n USING(NodeId) Where (n.TagName = '%full_tag%' AND (%time% BETWEEN '%report_dt_start%' AND '%report_dt_end%'))"
```

Здесь управляющее слово %time% используется два раза, потому что нам нужно как отобразить метку времени, так и соблюсти условие нахождения метки времени в выбранном временном промежутке.

Выберем в программе временной промежуток следующим образом:

Рисунок

Тогда строки sql_query для каждого тега после замены всех управляющих слов будут выглядеть так:

```

SELECT valbool,(time+ time '07:00') FROM nodes_history h JOIN nodes n USING(NodeId) Where
(n.TagName ='Metro.Moscow.Depo.Sokol.ORK.Masterskie.SHUV_1.V1.V.Alarm' AND ((time+ time
'07:00') BETWEEN '2022-02-26 11:25:30' AND '2022-03-28 11:30:30'))

SELECT valbool,(time+ time '07:00') FROM nodes_history h JOIN nodes n USING(NodeId) Where
(n.TagName ='Metro.Moscow.Depo.Sokol.ORK.Masterskie.SHUV_1.V1.FireAlarm' AND ((time+ time
'07:00') BETWEEN '2022-02-26 11:25:30' AND '2022-03-28 11:30:30'))

```

Результат запроса для тега V.Alarm:

	valbool boolean	?column? timestamp without time zone
1	t	2022-03-28 11:26:02.771216
2	f	2022-03-28 11:26:06.426885
3	t	2022-03-28 11:26:12.348422
4	f	2022-03-28 11:26:14.083648

Рисунок

Результат запроса для тега .FireAlarm:

	valbool boolean	?column? timestamp without time zone
1	t	2022-03-28 11:25:58.20403
2	f	2022-03-28 11:26:04.567191
3	t	2022-03-28 11:26:08.879713
4	f	2022-03-28 11:26:10.271019
5	t	2022-03-28 11:26:15.208288

Рисунок

Как мы видим, SQL запрос может вернуть нам несколько строк, но несмотря на то, что в типе мы описали одну строку <row>, в режиме single_mode="false", каждая из результирующих строк попадет в отчет:

Отчет об аварийности оборудования					
За период: 2022-02-26 11:25:30 - 2022-03-28 11:30:30					
Время генерации отчета: 28.03.2022, 11:31:56					
Наименование	Объект	Тип объекта	Расположение	Значение	Метка времени
ШУВ1. В1. Авария	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:26:02.771216
ШУВ1. В1. Авария	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	0	2022-03-28 11:26:06.426885
ШУВ1. В1. Авария	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:26:12.348422
ШУВ1. В1. Авария	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	0	2022-03-28 11:26:14.083648
ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:25:58.20403
ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	0	2022-03-28 11:26:04.567191
ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:26:08.879713
ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	0	2022-03-28 11:26:10.271019
ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:26:15.208288

Рисунок

Но, если бы мы сделали отчет, указав флаг single_mode = "true", тогда в отчет бы попала только первая строка результата, все остальные бы не были учтены:

Отчет об аварийности оборудования					
За период: 2022-02-26 11:25:30 - 2022-03-28 11:30:30					
Время генерации отчета: 28.03.2022, 11:35:32					
Наименование	Объект	Тип объекта	Расположение	Значение	Метка времени
ШУВ1. В1. Авария	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:26:02.771216
ШУВ1. В1. Пожар	ШУВ1. В1	Вентиляция В	Электродело Сокол. ОРК1. Здание мастерских	1	2022-03-28 11:25:58.20403

Рисунок