

ТОPAZ DB APPLICATION SERVICE

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Москва 2023

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	3
1 ВВЕДЕНИЕ.....	4
2 КОНФИГУРИРОВАНИЕ ПРОГРАММЫ	5
2.1 Конфигурирование программы в ОС Windows.....	5
2.2 Конфигурирование программы в ОС Linux	6
2.2.1 Создание и настройка магистрали	6
2.2.2 Загрузка конфигурации и расширения	10
3 ЗАПУСК ПРОГРАММЫ	13
3.1 Запуск в ОС Windows	13
3.2 Запуск в ОС Linux	14

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АПТС	–	аварийно-предупредительная телесигнализация
БД	–	база данных
ОС	–	операционная система
ПАМИ	–	подсистема автоматического мониторинга измерений
ПКМ	–	правая кнопка мыши
ТМ-модель	–	телемеханическая модель

1 Введение

Программное обеспечение TOPAZ DB APPLICATION SERVICE (далее – программа) предназначено для запуска на исполнение расчетных модулей и предоставления им доступа к данным БД. Модули реализуют следующие алгоритмы:

- контроль порогов;
- расчет оперативных блокировок;
- декодер команд ФСК ПАМИ;
- управление светодиодами (в зависимости от АПТС по различным уровням тревожности устанавливается соответствующая цветовая индикация);
- RastrWin (анализ и оптимизация режимов работы, проведение расчетов для электрических сетей и систем);
- имитация управления (режим тренажер);
- окрашивание топологической схемы по базовым напряжениям Notificator.

2 Конфигурирование программы

2.1 Конфигурирование программы в ОС Windows

Для корректной работы приложения необходимо настроить конфигурационный файл (Рис. 2.1.1), который находится в папке установки ПО:

- файл topazdbal.json для консольного приложения topazdbal.exe;
- файл TOPAZDBApplicationService.json для службы TOPAZDBApplication.exe.

```

{
  "common" : {
    "maxEventId" : 1, // номер события для резервирования topazdbal(команды и-примы, 1-примы, 2-второй элемент) (по умолчанию: 1)
    "startEventId" : 10000, // по умолчанию: 1
    "startEventId" : 1001, // по умолчанию: 1
    "logLevel" : "all", // уровень логирования(файлы логов создается в папке log). Возможные значения: "all" - логируется все, "warning" - логируется только предупреждения(все кроме информационных сообщений),
    "maxLogFileMB" : 10, // максимальный размер файла логов в МБ (по умолчанию: 10)
    "startEventScheduler" : true, // запуск агрегирования событий (по умолчанию true)
    "delayBeforeAveraging" : 120000, // задержка в миллисекундах между окончанием события и началом подсчета среднего значения для задания в периодический график (по умолчанию: 120000)
    "startEventScheduler" : true // выполнять события вместо планировщика ОС Windows (по умолчанию true)
  },
  // список компонентов для резервирования(MAIN/STANDBY). Для каждого компонента startEventId должен быть кастомный, можно определить более 2-х компонентов
  // логика необязательная, если не нет режим работы только MAIN
  "reserveUnits" : [
    {
      "id" : 1, // номер компонента для резервирования (по умолчанию: 1)
      "hostAddress" : "192.168.100.9,192.168.200.9", // IP-адреса для отправки и приема пакетов для выбора режима MAIN/STANDBY и стартага
      "port" : 5001, // TCP порт для прослушивания и установки соединения (по умолчанию: 5001)
      "clientConnectTimeout" : 3000, // таймアウト подключения клиента в данном компоненте резервирования (по умолчанию: 3000)
      "clientResponseTimeout" : 1000, // таймアウト выполнения запроса клиента в данном компоненте резервирования (по умолчанию: 1000)
    },
    {
      "id" : 2,
      "hostAddress" : "192.168.100.10,192.168.200.10",
      "port" : 5001,
      "clientConnectTimeout" : 3000,
      "clientResponseTimeout" : 1000
    }
  ],
  "primaryConnection" : { // настройки основного подключения к БД
    "hostAddress" : "127.0.0.1",
    "dbName" : "DB_SVOLOKHOVO",
    "login" : "root",
    "password" : "root",
    "port" : 3306
  },
  "secondaryConnection" : { // настройки резервного подключения к БД
    "hostAddress" : "127.0.0.1",
    "dbName" : "DB_SVOLOKHOVO",
    "login" : "root",
    "password" : "root",
    "port" : 3306
  }
}

```

Рис. 2.1.1. Пример конфигурационного файла

Пояснения после символа «//» до конца строки считаются комментариями.

Параметр **maxLogFileSizeMb** (в МБ) указывает максимальный размер лог-файла (лог-файл создается для каждого компонента в папке log), по достижении которого файл переименовывается в имя_компонента_0.log. Если файл с таким названием уже есть, то он заменяется. На диске хранится для каждого компонента maxLogFileSizeMb * 2 Мбайта логов.

Для вывода отладочной информации в консольное приложение topazdbal.exe, для параметра logLevel нужно задать необходимый уровень логирования, отличный от «none».

Параметр **startEventScheduler** активирует внутренний планировщик TOPAZDBAL, отвечающий за вызов различных процедур (events) БД через заданные промежутки времени. Таким образом, снижается нагрузка на БД. После изменения какого-либо конфигурационного параметра требуется произвести перезапуск программного обеспечения.

2.2 Конфигурирование программы в ОС Linux

Для запуска программы в ОС Linux необходимо добавить в конфигурацию соответствующую магистраль, а также установить расширение topazdbal.

2.2.1 Создание и настройка магистрали

Добавление и настройка магистрали осуществляется в программе TOPAZ ModelCreator. В окне «Телемеханическая модель» нужно выбрать режим отображения иерархии «Б» ТМ-модели, с помощью ПКМ вызвать контекстное меню нужного процесса, в котором необходимо выбрать пункт «Добавить магистраль» (Рис. 2.2.1).

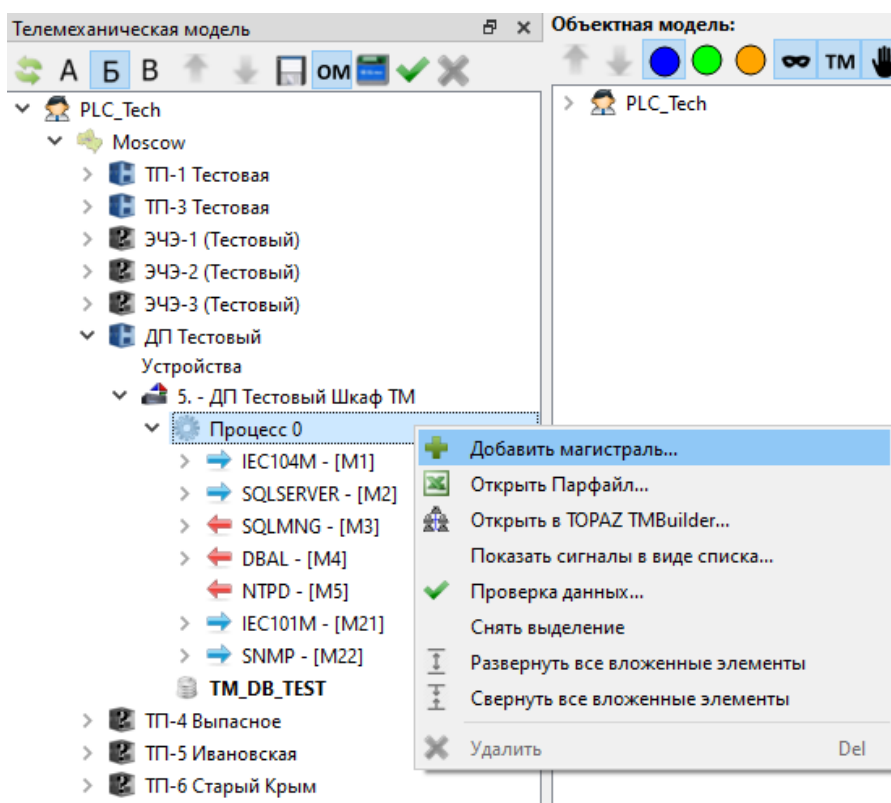
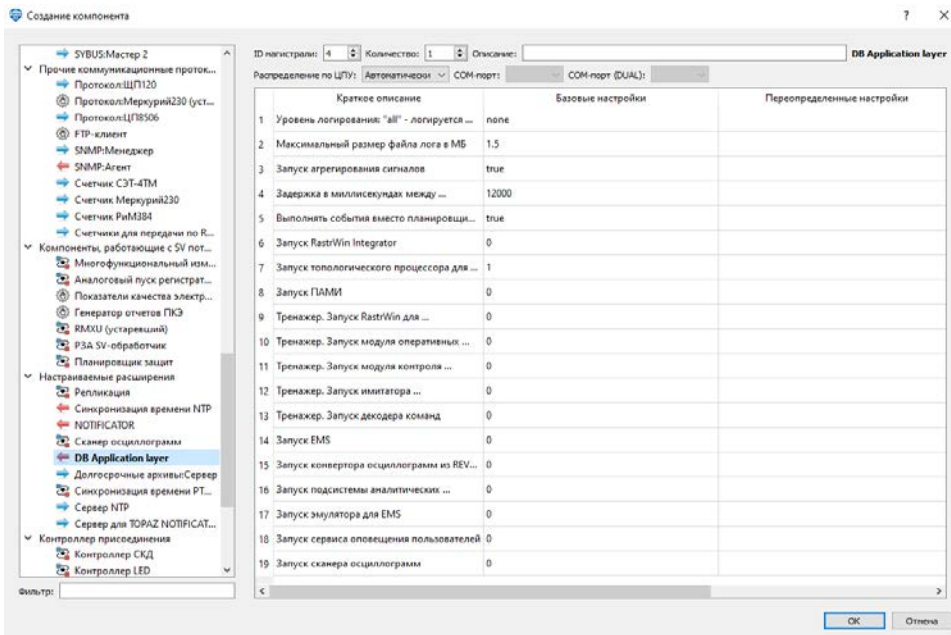
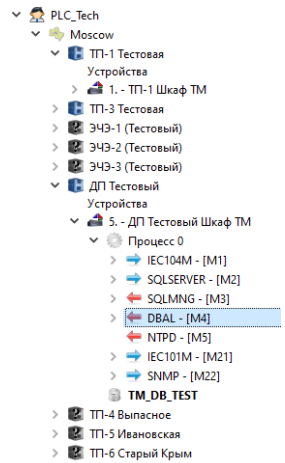


Рис. 2.2.1. Добавление магистрали

В открывшемся окне «Создание компонента» двойным кликом выбрать DB Application layer (Рис. 2.2.2, а), указать номер магистрали (ID) и количество магистралей. Также можно добавить словесное описание магистрали и изменить настройки по умолчанию. В дереве ТМ-модели появится новая магистраль (Рис. 2.2.2, б).



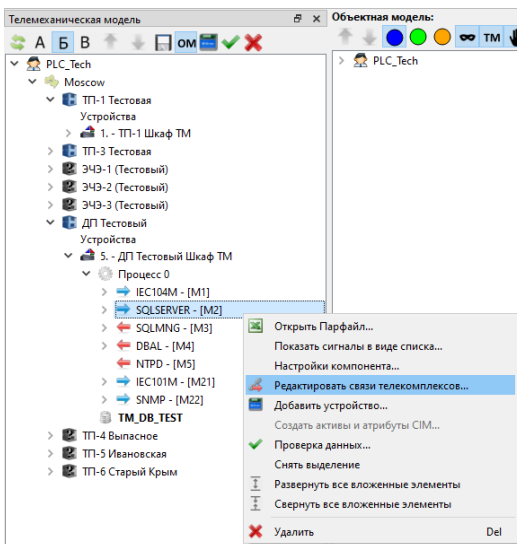
а) Окно «Создание компонента»



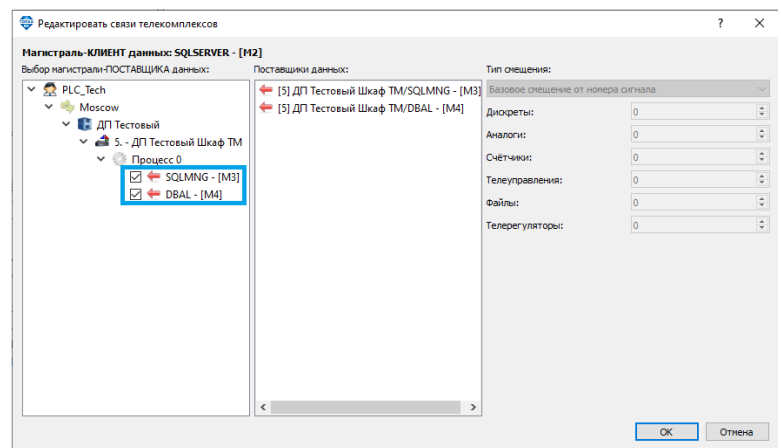
б) Дерево ТМ-модели с новой магистралью

Рис. 2.2.2

Далее следует связать добавленную магистраль с компонентом SQL-сервера, настроенного на БД. Для этого нужно в контекстном меню элемента SQLSERVER выбрать пункт «Редактировать связи телекомплексов...» (Рис. 2.2.3, а). В появившемся окне указать связь созданной нами магистрали с базой данных, установив галочку рядом с названием магистрали (Рис. 2.2.3, б).



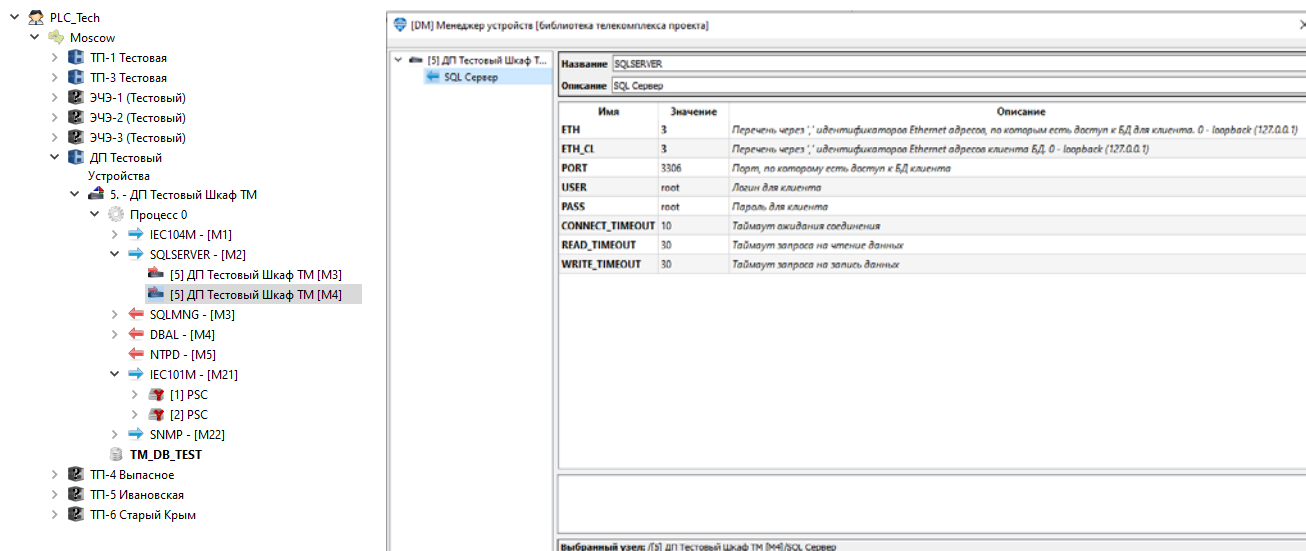
а) Вызов окна «Редактировать связи телекомплексов»



б) Окно «Редактировать связи телекомплексов»

Рис. 2.2.3

Для созданной магистрали необходимо задать значение идентификаторов Ethernet. Для этого следует навести курсор на нужную магистраль во вкладке SQLSERVER дерева ТМ-модели и двойным кликом вызвать окно «Менеджер устройств [библиотека телекомплекса проекта]». В данном окне необходимо прописать значения полей «ETH» и «ETH_CL». Также при необходимости можно задать остальные параметры.

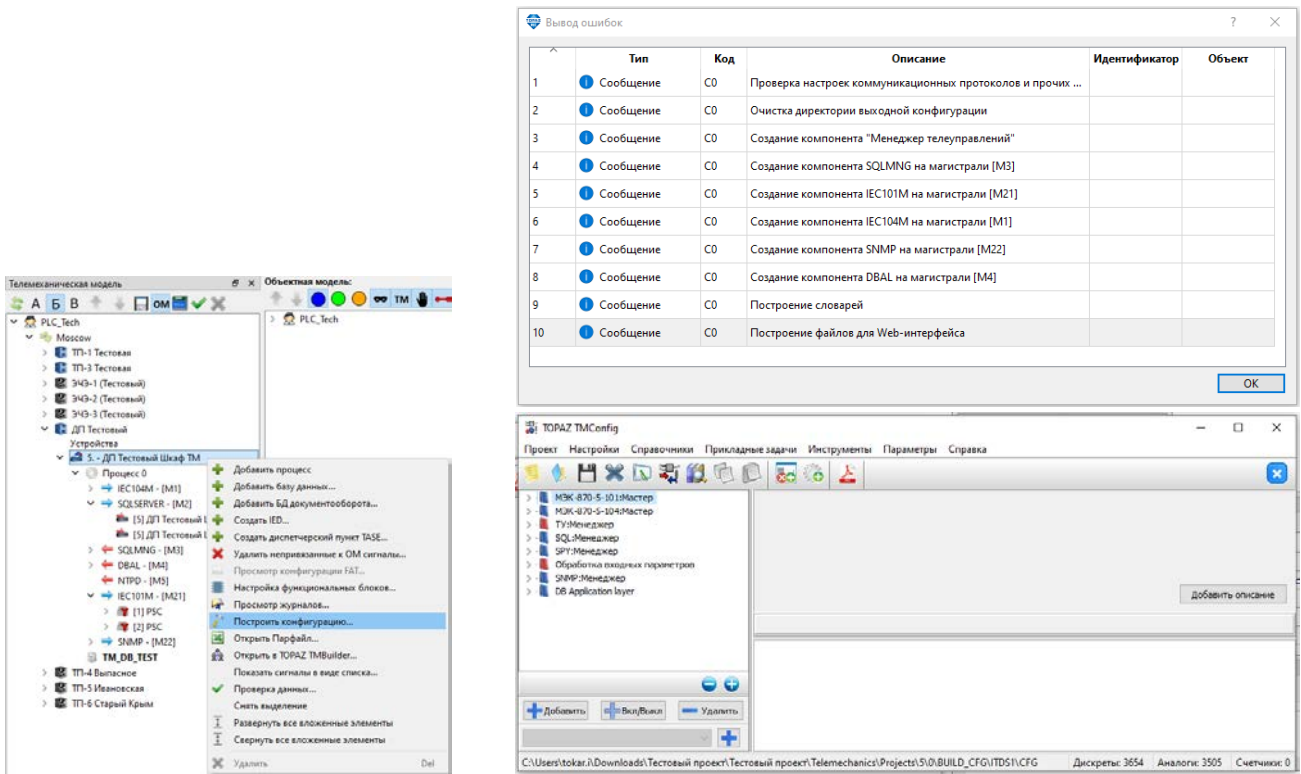


а) Дерево ТМ-модели с выбранной магистралью

б) Окно «Менеджер устройств [библиотека телекомплекса проекта]»

Рис. 2.2.4

Далее следует построить конфигурацию. Для этого нужно с помощью ПКМ вызвать контекстное меню нужного устройства (Рис. 2.2.5, а), выбрать «Построить конфигурацию...», вызвав окна «Вывод ошибок» и «ТОPAZ ТМConfig» (Рис. 2.2.5, б).



а) Запуск построения конфигурации

б) Окна «Выход ошибок» и «ТОPAZ TMConfig»

Рис. 2.2.5

Интерфейс окна «ТОPAZ TMConfig» позволяет просмотреть файл конфигурации. Для этого нужно развернуть дерево компонента DB Application layer и выбрать «Конфигурация», тогда в правой части окна отобразится содержание файла (Рис. 2.2.6).

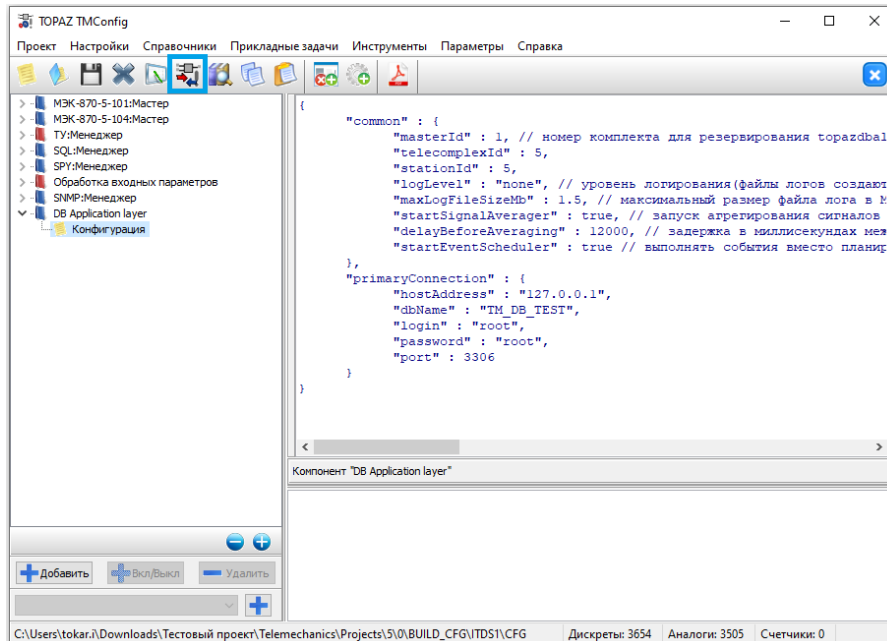



Рис. 2.2.6

2.2.2 Загрузка конфигурации и расширения

Загрузка конфигурации на устройство осуществляется в программе «ТОPAZ TMLoader», для вызова которой нужно нажать кнопку  на панели управления окна «ТОPAZ TMConfig». Окно «ТОPAZ TMLoader» представлено на Рис. 2.2.7.

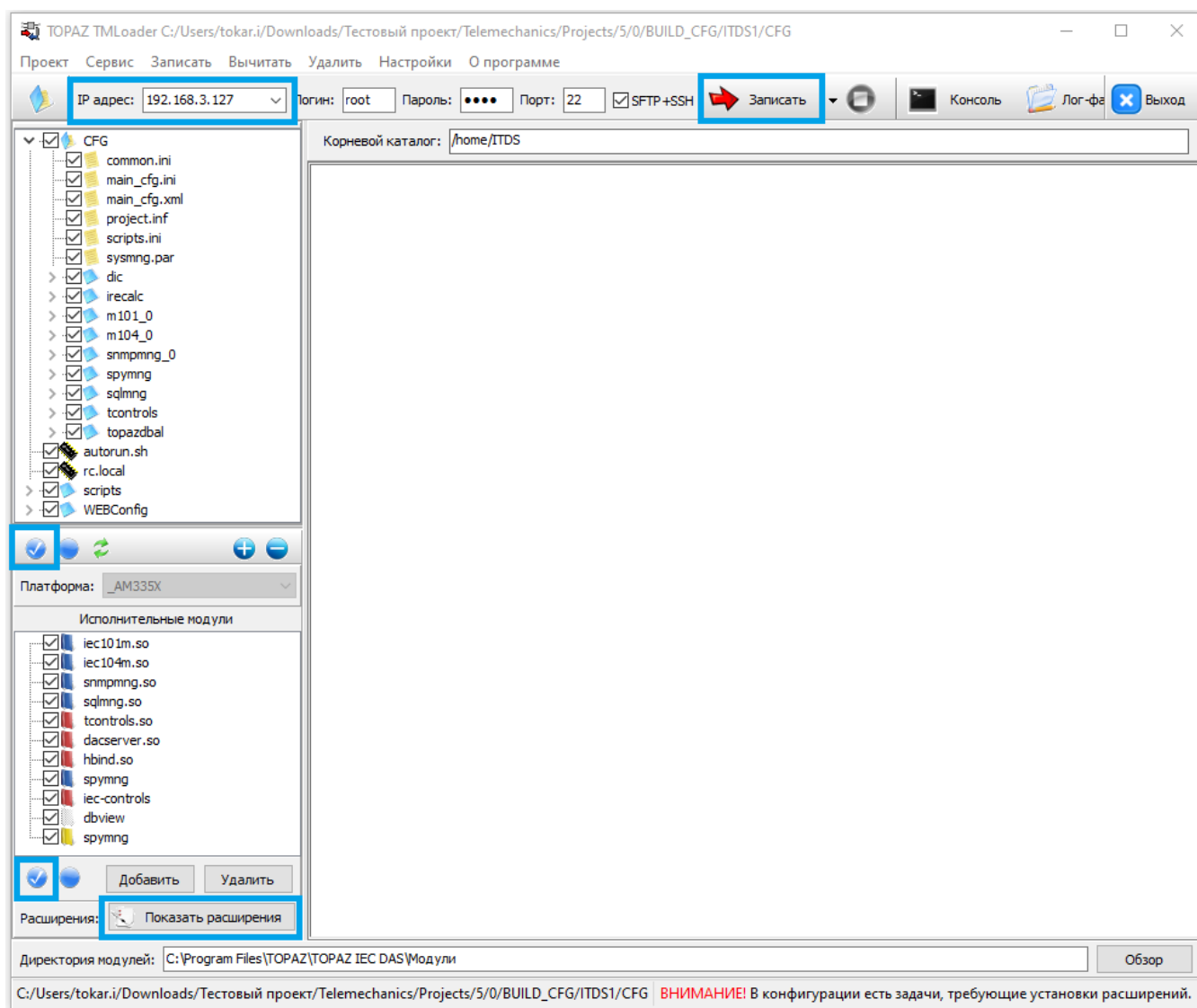

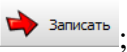



Рис. 2.2.7. Окно «TMLoader», обязательные действия обведены

Для загрузки конфигурации на устройство нужно выполнить следующие действия в окне «ТОPAZ TMLoader»:

- 1) нажать обе кнопки , выбрав для загрузки все файлы конфигурации и все исполнительные модули;
- 2) прописать IP-адрес устройства;
- 3) нажать кнопку ;

4) нажать кнопку , после чего появится окно «Расширения» (Рис. 2.2.8);

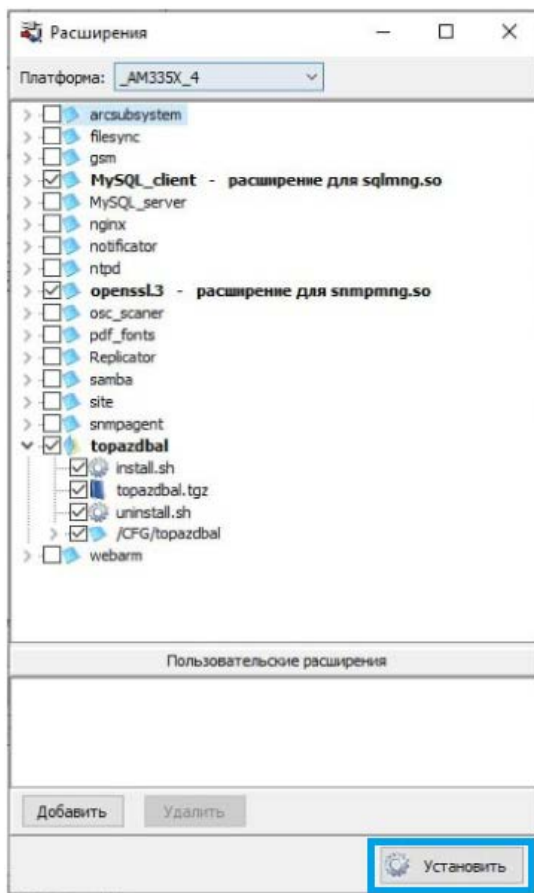



Рис. 2.2.8

5) в окне «Расширения» нужно раскрыть дерево «topazdbal» и установить галочки в соответствии с Рис. 2.2.8;

6) в окне «Расширения» нажать кнопку , предварительно проверив настройки компонента (Рис. 2.2.2);

7) в окне «ТОPAZ TMLoader» развернуть вкладку «Сервис» и выбрать «Перезапустить процессы» (Рис. 2.2.9).

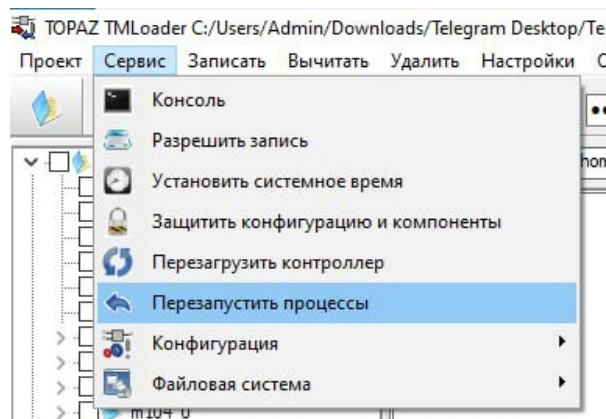


Рис. 2.2.9

Конфигурация topazdbal располагается в директории /home/ITDS/topazdbal/topazdbal.json. Конфигурация, загружаемая из TMBuilder, располагается в директории /home/ITDS/CFG/topazdbal/topazdbal.json.

Подмена файлов конфигурации происходит при перезапуске процессов (или старте контроллера) с помощью скрипта /home/ITDS/scripts/dbal_conf.sh.

3 Запуск программы

3.1 Запуск в ОС Windows

В ОС Windows программа может запускаться как служба «TOPAZ DB Application Service» и как консольное приложение `topazdbal.exe`. Запуск службы осуществляется либо с помощью стандартной оснастки `services.msc` (Рис. 3.1.1), либо с помощью вкладки «Службы» диспетчера задач (Рис. 3.1.2).

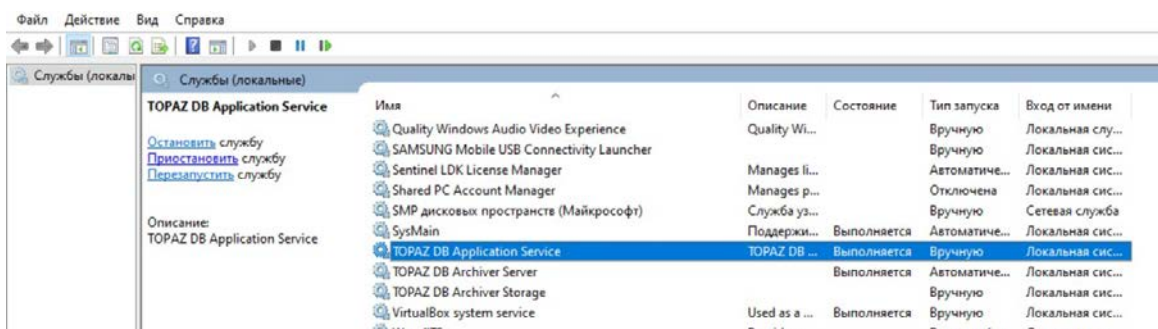


Рис. 3.1.1

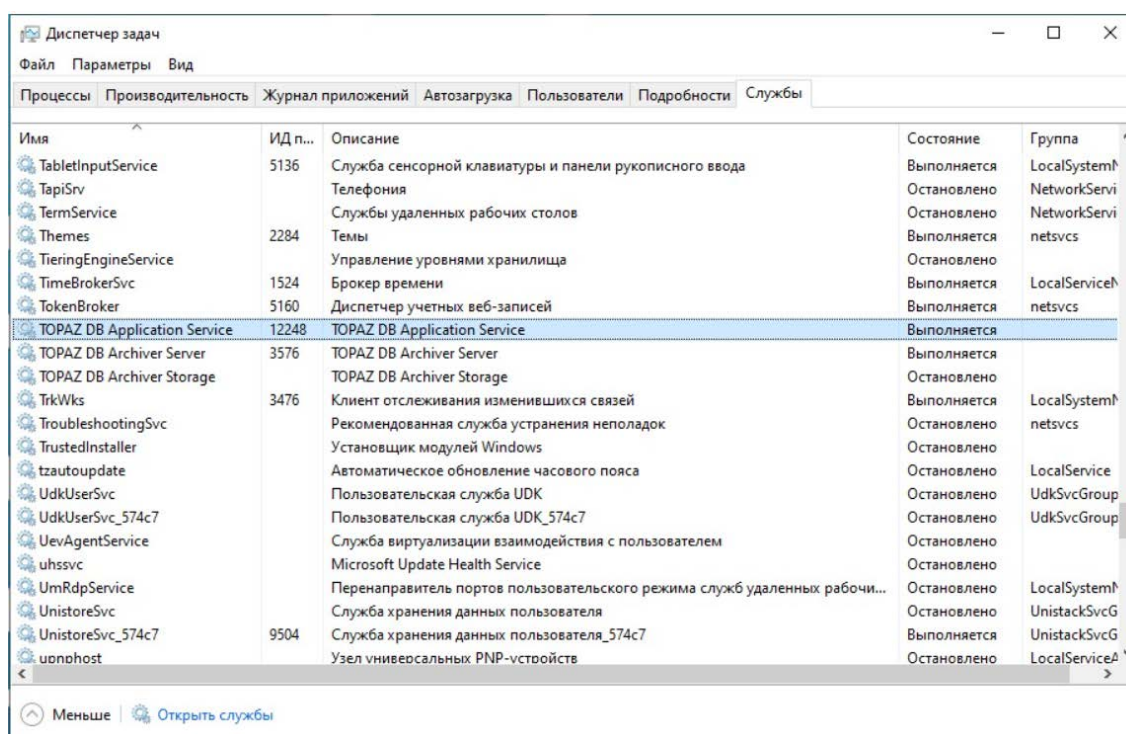


Рис. 3.1.2

В окне программы `TOPAZDBApplication.exe` можно проверить состояние службы, а также выполнить запуск/останов. Консольное приложение `topazdbal.exe` запускается из папки установки ПО и представляет собой отладочное средство (для записи логов в файл необходимо запустить от Администратора).

3.2 Запуск в ОС Linux

Компонент устанавливается в директорию `/home/ITDS/topazdbal` и автоматически запускается.

Управление процессом осуществляется с помощью следующих команд:

- Команда запуска: **`/etc/init.d/topazdbal start`**
- Команда остановки: **`/etc/init.d/topazdbal stop`**
- Команда перезапуска: **`/etc/init.d/topazdbal restart`**

Запуск по умолчанию осуществляется в режиме псевдодемона. Терминал освобождается при запуске пользователем, сообщения в консоль не выводятся, лог-файлы помещаются в директорию `/home/ITDS/topazdbal/log`. При запуске стартует второй рабочий процесс, при этом первый – «монитор» следит за рабочим процессом, а в случае падения по ошибке вновь его запускает (причина ошибок и стек вызовов указываются в лог-файлах).