

«АСК - ПАССАЖИРСКИЙ ТРАНСПОРТ»
ИНСТРУКЦИЯ ПО УСТАНОВКЕ СИСТЕМЫ

Содержание

1. Программные требования.....	3
2. Установка и настройка сервера приложений.....	4
2.1. Установка и настройка базы данных.....	4
2.1.1. Настройка СУБД PostgreSQL.....	4
2.1.2. Создание базы PostgreSQL.....	4
2.1.3. Установка дистрибутива системы.....	8
2.1.4. Установка сервера.....	9
2.1.4. Установка службы.....	12
2.1.5. Установка и настройка сервера телеметрии.....	13
3. Запуск и проверка работоспособности Системы.....	18
3.1. Запуск программного обеспечения.....	18
3.2. Проверка работоспособности Системы.....	18

1. Программные требования

Минимальные системные требования для клиентского ПО:

Операционная система под управлением ОС Windows 7 или новее.

Установленный .Net Framework 4.7

Процессор 2-ядерный 1,6 Ггц.

Объем ОЗУ 2 Гб

1 Гб свободного места на жёстком диске (HDD)

Скорость Интернет-соединения — 256 килобит в секунду.

Рекомендованные системные требования для клиентского ПО:

Операционная система под управлением ОС Windows 7 или новее.

Установленный .Net Framework 4.7

Процессор 4-ядерный 2,2 Ггц.

Объем ОЗУ 8 Гб

1 Гб свободного места на жёстком диске (HDD)

Скорость Интернет-соединения — 1 мегабит в секунду.

2. Установка и настройка сервера приложений

Сервер приложений служит для конечной обработки и хранения данных, поступающих с сервера телеметрии и последующей их передачи клиентскому приложению.

Рекомендации по составу аппаратных средств формируются разработчиком системы согласно предполагаемому количеству объектов мониторинга.

Для корректной работы сервера приложений рекомендуется использовать серверную операционную систему семейства Windows (Microsoft Windows Server 2008 R2 или новее).

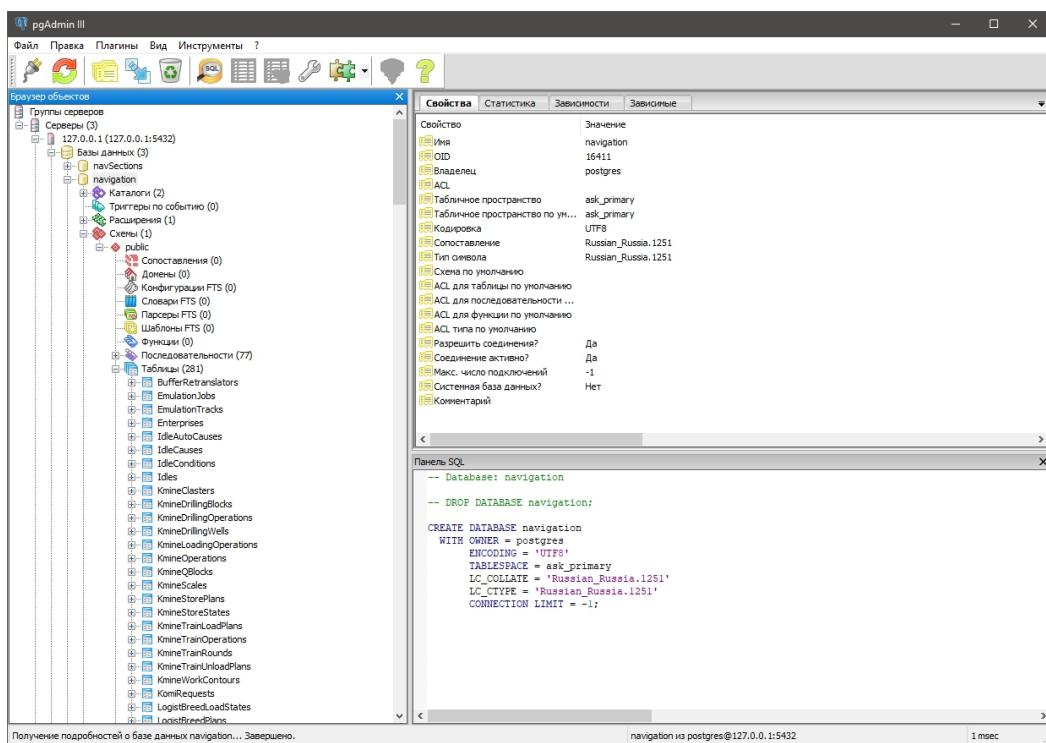
Сервер приложений использует СУБД PostgreSQL 10.10 или новее.

2.1. Установка и настройка базы данных

2.1.1. Настройка СУБД PostgreSQL

Установка СУБД выполняется со стандартными параметрами.

Создание базы выполняется пользователем `postgres`.



2.1.2. Создание базы PostgreSQL

Создание базы данных производится путем последовательного исполнения шести sql-запросов в окне pgAdmin III или pgAdmin 4:

«s1-1_CreateTablespaces.sql» - создание табличного пространства.

```
-- СКРИПТ СОЗДАНИЯ ТАБЛИЧНЫХ ПРОСТРАНСТВ ДЛЯ POSTGRESQL
-- Местозаполнители:
-- {0} - основной каталог данных (например: C:\PGDATA\ASK_PRM);
-- ! Перед выполнением нужно заменить местозаполнители соответствующими значениями.

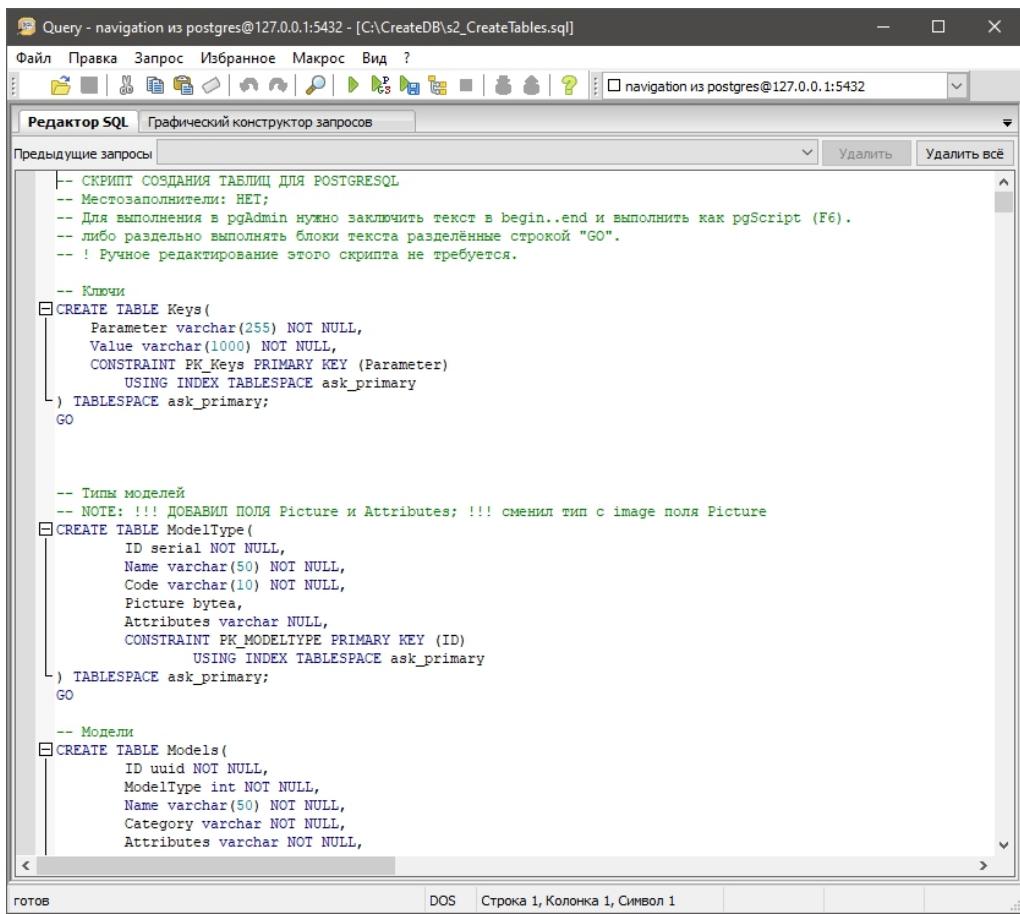
-- Основное пространство БД: данные справочников
CREATE TABLESPACE ask_primary LOCATION '{0}';
```

«s1-2_CreateDatabase.sql» - создание базы данных.

```
-- СКРИПТ СОЗДАНИЯ БАЗЫ ДАННЫХ ДЛЯ POSTGRESQL
-- Местозаполнители:
-- {0} - имя БД (например: navigation);
-- ! Перед выполнением нужно заменить местозаполнители соответствующими значениями.

-- Создание базы данных
CREATE DATABASE {0}
    ENCODING = 'UTF8'
    --LC_COLLATE = 'Russian_Russia.1251'
    --LC_CTYPE = 'Russian_Russia.1251'
    TABLESPACE ask_primary;
```

«s2_CreateTables.sql» - создание базы данных.



The screenshot shows the pgAdmin III application window. The title bar reads "Query - navigation из postgres@127.0.0.1:5432 - [C:\CreateDB\s2_CreateTables.sql]". The menu bar includes "Файл", "Правка", "Запрос", "Избранное", "Макрос", "Вид", and "?". Below the menu is a toolbar with various icons. The main area is titled "Редактор SQL | Графический конструктор запросов". A dropdown menu "Предыдущие запросы" is open, showing "navigation из postgres@127.0.0.1:5432" and buttons for "Удалить" and "Удалить все". The SQL code in the editor is as follows:

```

-- СКРИПТ СОЗДАНИЯ ТАБЛИЦ ДЛЯ POSTGRESQL
-- Местозаполнители: нет;
-- Для выполнения в pgAdmin нужно заключить текст в begin..end и выполнить как pgScript (F6).
-- либо раздельно выполнять блоки текста разделённые строкой "GO".
-- ! Ручное редактирование этого скрипта не требуется.

-- Ключи
CREATE TABLE Keys(
    Parameter varchar(255) NOT NULL,
    Value varchar(1000) NOT NULL,
    CONSTRAINT PK_Keys PRIMARY KEY (Parameter)
        USING INDEX TABLESPACE ask_primary
) TABLESPACE ask_primary;
GO

-- Типы моделей
-- NOTE: !!! ДОБАВИЛ ПОЛЯ Picture и Attributes; !!! сменил тип с image поля Picture
CREATE TABLE ModelType(
    ID serial NOT NULL,
    Name varchar(50) NOT NULL,
    Code varchar(10) NOT NULL,
    Picture bytea,
    Attributes varchar NULL,
    CONSTRAINT PK_MODELTYPE PRIMARY KEY (ID)
        USING INDEX TABLESPACE ask_primary
) TABLESPACE ask_primary;
GO

-- Модели
CREATE TABLE Models(
    ID uuid NOT NULL,
    ModelType int NOT NULL,
    Name varchar(50) NOT NULL,
    Category varchar NOT NULL,
    Attributes varchar NOT NULL,
)

```

«s3-1_DefaultSection.sql» - создание базовой секции данных.

```
-- СКРИПТ ПЕРВОНАЧАЛЬНОГО ЗАПОЛНЕНИЯ ТАБЛИЦЫ СЕКЦИЙ ДЛЯ POSTGRESQL
-- Местозаполнители:
-- {0} - адрес СУБД (например: localhost);
-- {1} - порт СУБД (например: 5432);
-- {2} - логин (например: postgres);
-- {3} - пароль;
-- {4} - имя БД (например: navSections);
-- {5} - макс. кол-во соединений (например: 20);
-- {6} - каталог BULK (например: C:\PGDATA\COPY\ASK_NAV);
-- {7} - удалённый каталог BULK (например: C:\PGDATA\COPY\ASK_NAV);
-- {8} - каталог основного табличного пространства (например: C:\PGDATA\ASK_PRM);
-- {9} - каталог табличного пространства навигационных данных (например: C:\PGDATA\ASK_NAV);
-- {10} - каталог табличного пространства логов (например: C:\PGDATA\ASK_LOG);
-- ! Перед выполнением нужно заменить местозаполнители соответствующими значениями.

-- ПЕРЕД ВЫПОЛНЕНИЕМ ВСТАВКИ НУЖНО СКОРРЕКТИРОВАТЬ НАСТРОЙКИ СЕКЦИИ!
DELETE FROM Sections;
SELECT setval('sections_id_seq', 1);
INSERT INTO Sections(ID, BEGIN_DATE, FACTORY, PARAMETERS)
VALUES (1, '00010101', '16E22B48-8B97-4E2B-BF57-BABB45826103',
'<section>
<settings>
<value key="address">{0}</value>
<value key="port">{1}</value>
<value key="login">{2}</value>
<value key="password">{3}</value>
<value key="db">{4}</value>
<value key="suffix">.base</value>
<value key="max_connections">{5}</value>
<value key="bulk_dir">{6}</value>
<value key="remote_bulk_dir">{7}</value>
<value key="ts_primary">{8}</value>
<value key="ts_navsens">{9}</value>
<value key="ts_log">{10}</value>
</settings>
</section>');

```

«s3-2_DefaultUsers.sql» - добавление дефолтных значений в справочники ролей и пользователей.

```
-- СКРИПТ ПЕРВОНАЧАЛЬНОГО ЗАПОЛНЕНИЯ ТАБЛИЦ РОЛЕЙ И ПОЛЬЗОВАТЕЛЕЙ ДЛЯ POSTGRESQL
-- Местозаполнители:
-- {0} - UUID администратора;
-- {1} - UUID системного пользователя;
-- Для выполнения в pgAdmin нужно заключить текст в begin..end и выполнить как pgScript (F6).
-- либо раздельно выполнять блоки текста разделённые строкой "GO".
-- ! Перед выполнением нужно заменить местозаполнители соответствующими значениями.
-- P.S.: Онлайн генератор UUID: https://www.uuidgenerator.net/

DELETE FROM Roles;
SELECT setval('roles_id_seq', 5);
INSERT INTO Roles(ID, NAME, CODE, Syst, Fixed) VALUES (1, 'Системная учетная запись', 'SYSTEM', TRUE, TRUE);
INSERT INTO Roles(ID, NAME, CODE, Fixed, Disp, Adm, Dev, Map, RepairCauses, Repairs, Reserves, Fuel, Workers, Waysheets, Trailers) VALUES (2, 'Администратор', 'ADMINISTRATOR', FALSE, TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE);
INSERT INTO Roles(ID, NAME, CODE, Disp, Map, RepairCauses, Repairs, Reserves, Fuel, Workers, Waysheets, Trailers) VALUES (3, 'Система', 'SYSTEM', FALSE, FALSE);
INSERT INTO Roles(ID, NAME, CODE, Disp, Map, UsersRead, Workers, Reserves, Repairs, RepairCauses, Fuel, Waysheets, Trailers) VALUES (4, 'Гость', 'GUEST', FALSE, FALSE);

-- ПЕРЕД ВЫПОЛНЕНИЕМ ВСТАВКИ НУЖНО ЗАДАТЬ UUID ДЛЯ ПОЛЬЗОВАТЕЛЕЙ
DELETE FROM Users;
INSERT INTO USERS(ID, Name, Login, Role, Password, Email, Phone, StartTime) VALUES ('{0}', 'Администратор', 'admin', '1234567890', 'admin@system.ru', '+79851234567', '2023-01-01 00:00:00');
INSERT INTO Users(ID, Name, Role, Password, Email, Phone, StartTime) VALUES ('{1}', 'Система', 1, 'md5({1})', 'system@system.ru', '+79851234567', '2023-01-01 00:00:00');
GO
```

«s3-3_DefaultValues.sql» - добавление дефолтных значений в остальные справочники.

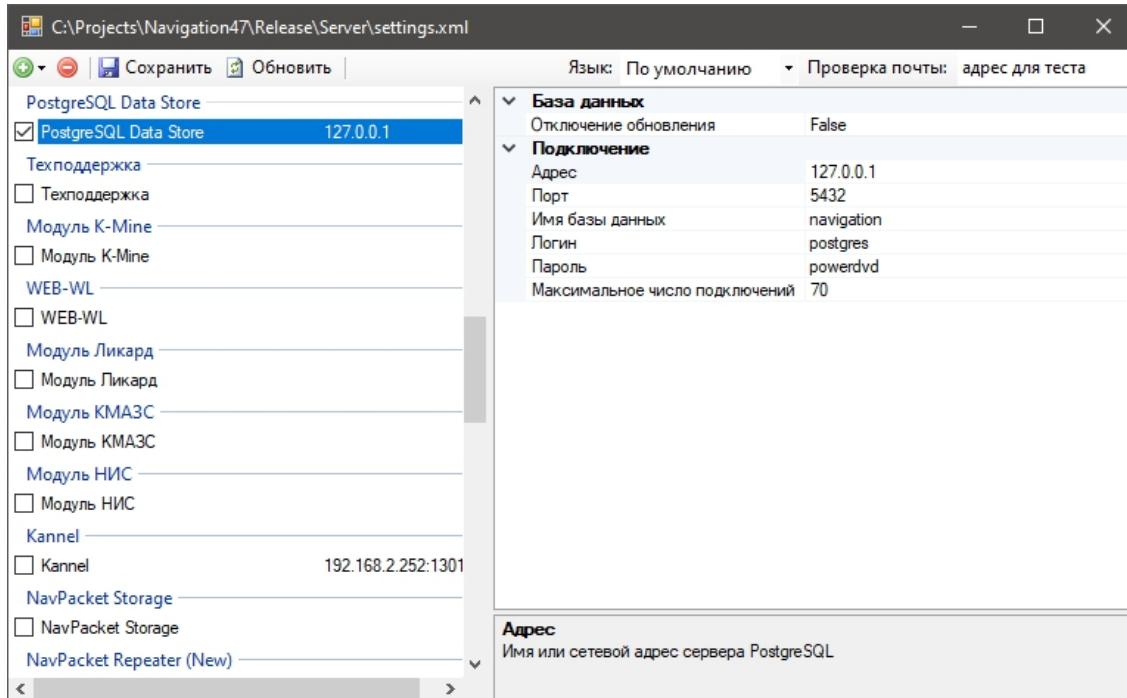
```
-- СКРИПТ ПЕРВОНАЧАЛЬНОГО ЗАПОЛНЕНИЯ БАЗЫ ДАННЫХ НАВИГАЦИИ ДЛЯ POSTGRESQL
-- Местозаполнители: NET;
-- Для выполнения в pgAdmin нужно заключить текст в begin..end и выполнить как pgScript (F6).
-- либо раздельно выполнять блоки текста разделённые строкой "GO".
-- ! Ручное редактирование этого скрипта не требуется.

DELETE FROM ZoneType;
SELECT setval('zonetype_id_seq', 3);
INSERT INTO ZoneType(ID, ZoneType, CODE) VALUES (1, 'Точка с радиусом', 'RADIUS');
INSERT INTO ZoneType(ID, ZoneType, CODE) VALUES (2, 'Коридор', 'CORRIDOR');
INSERT INTO ZoneType(ID, ZoneType, CODE) VALUES (3, 'Полигон', 'POLYGON');
GO

DELETE FROM ModelType;
SELECT setval('modeltype_id_seq', 16);
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (1, 'Погрузчик', 'LOADER', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (2, 'Самосвал', 'DUMPTRUCK', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (3, 'Тракторно-бульдозерная техника', 'BULLDOZER', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (4, 'Прочее', 'OTHER', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (5, 'Экскаватор', 'EXCAVATOR', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (6, 'Тягач', 'TRACTOR', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (7, 'СпецТехника', 'SPECIAL', '\x89504E470D0A1A0A0000000D4948');
INSERT INTO ModelType(ID, NAME, CODE, PICTURE) VALUES (8, 'Автобус', 'BUS', '\x89504E470D0A1A0A0000000D4948');

TNSPRT TNSPRT ModelType/TD NAME CODE PICTURE VATNPS /G 'Топливозаправщик' 'ROWSFR' '\x89504E470D0A1A0A0000000D494844520
```

В конфигураторе сервера должен быть включен модуль PostgreSQL Data Store.



Сервер после обновлений сам выполняет требуемые изменения в таблицах. Поэтому выполнение данных скриптов необходимо только при первичном развертывании системы.

2.1.3. Установка дистрибутива системы

Распакуйте дистрибутив с программой в рабочую директорию (рекомендуется корень диска). Убедитесь, что пользователь имеет полный доступ к рабочему каталогу.

2.1.4. Установка сервера

Запустите файл «NSConfigurator.exe» из рабочей директории программы.

Добавление модулей происходит выбором из выпадающего списка при нажатии кнопки  .

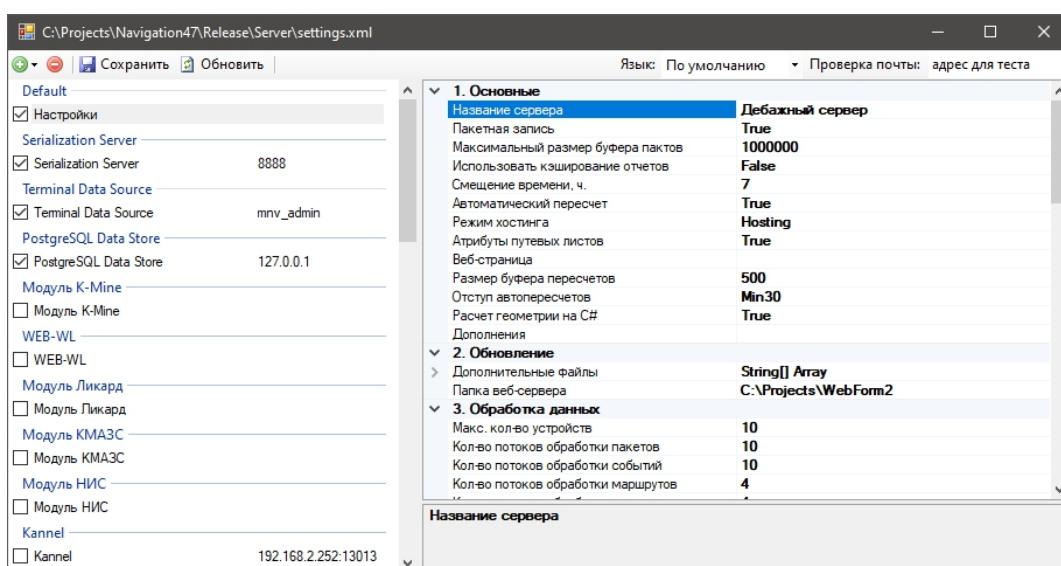
Сервер приложений может работать с несколькими серверами телеметрии, в этом случае добавляется несколько одинаковых модулей и выполняется их конфигурация.

После настройки всех модулей нажмите кнопку «Сохранить».

- На вкладке «Настройки» укажите SMTP-сервер, порт, логин и пароль почтового ящика, который будет использоваться для системных уведомлений.

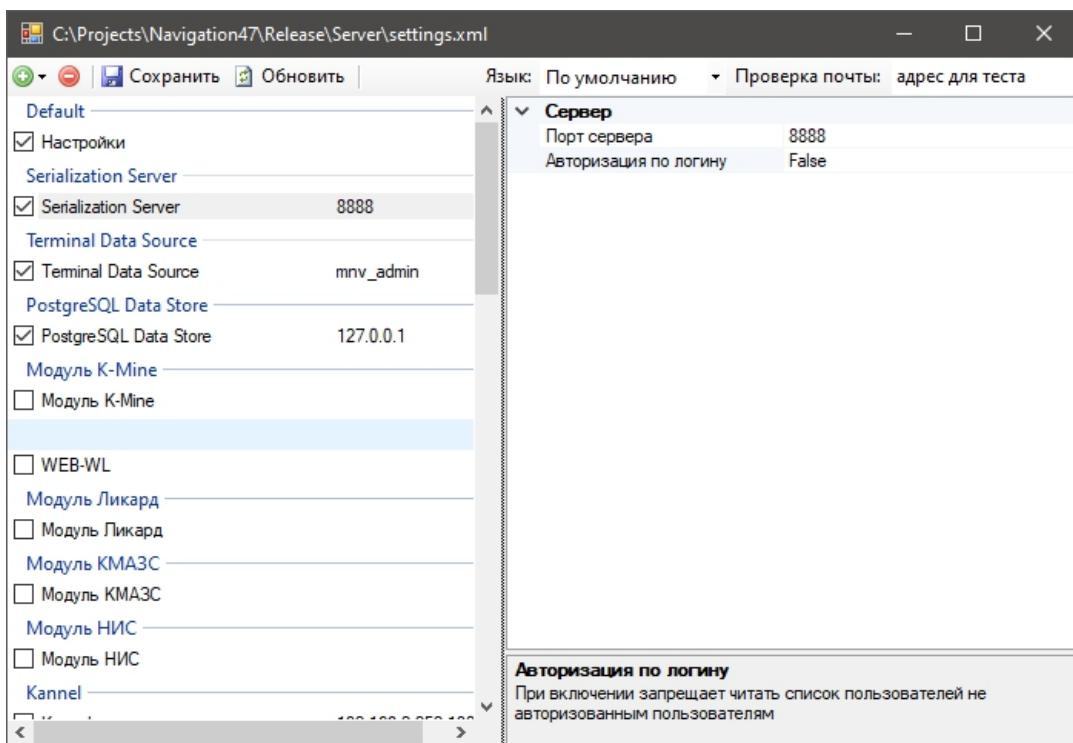
Укажите каталог клиентского приложения.

Укажите название навигационного сервера и смещение времени (часовой пояс относительно Гринвича), остальные параметры рекомендуется оставить по умолчанию.

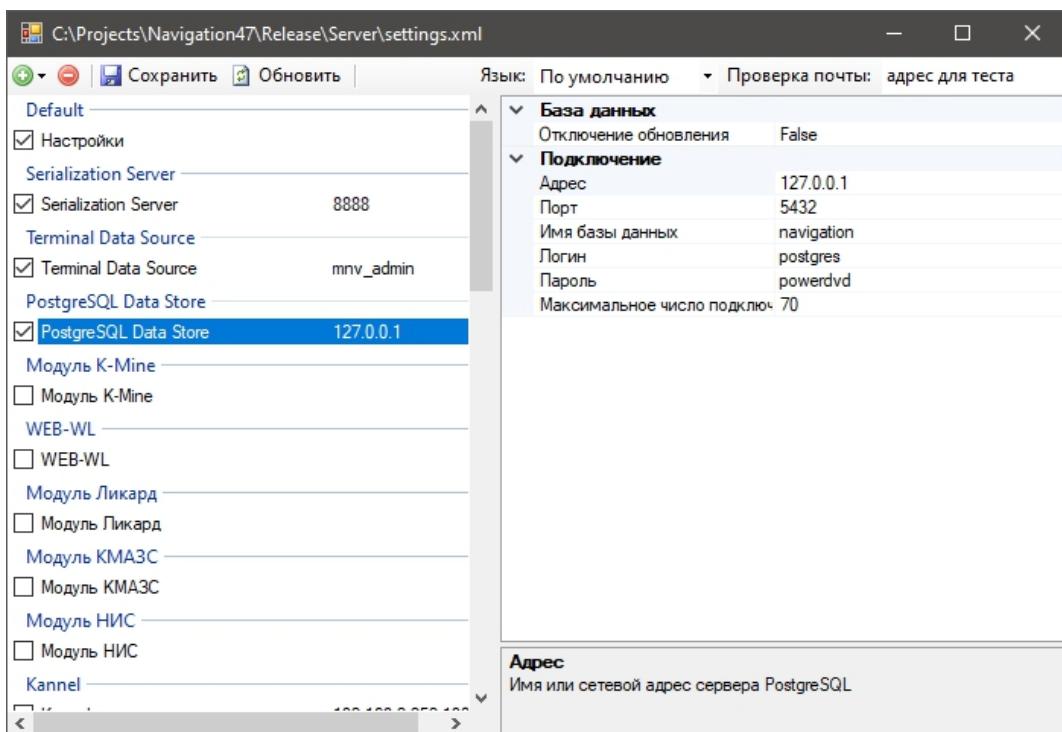


- На вкладке «Serialization Server» укажите TCP-порт по которому клиенты будут подключаться к навигационному серверу.

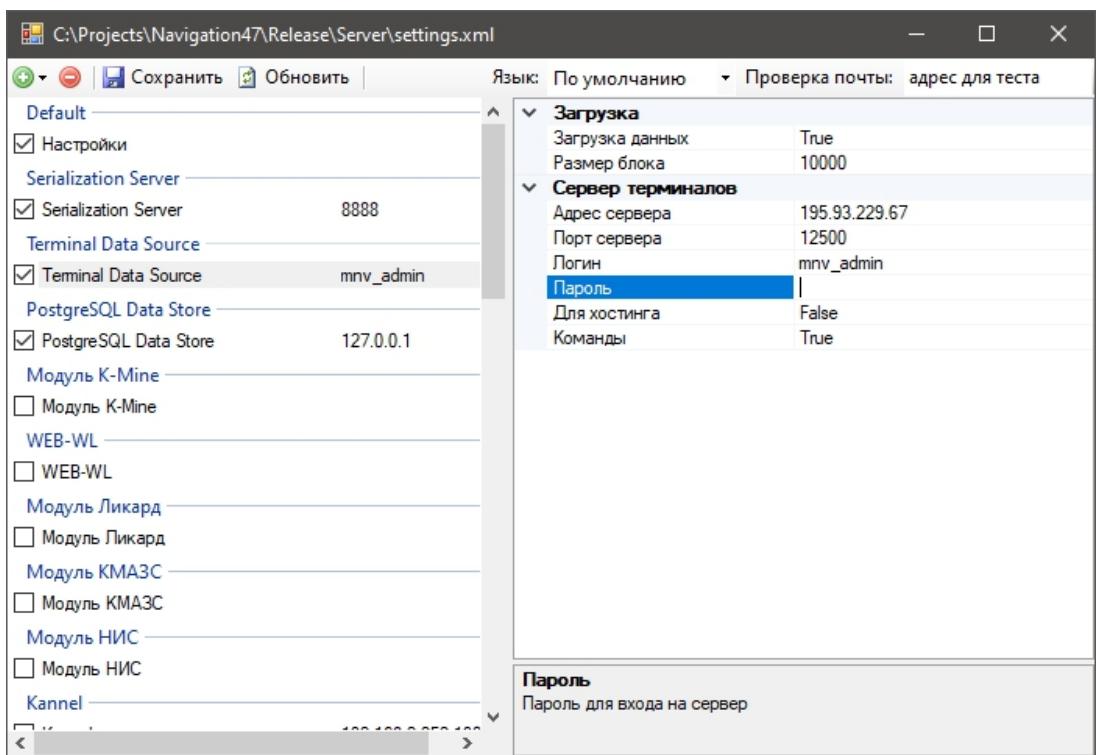
Определите как клиенты будут авторизоваться на сервере: выбирая свою учетную запись из выпадающего списка(авторизация по логину – False), либо вводя логин в форму авторизации (авторизация по логину – True).



- Модуль «PostgreSQL Data Store» настраивает подключение к базе данных. Подключение – указывается адрес базы данных, логин, пароль и название базы, максимальное количество подключений.



- Модуль «Terminal Data Source» позволяет сконфигурировать подключение к серверу телеметрии.
Укажите адрес сервера, порт, логин и пароль для подключения.
Для загрузки данных выберите параметр «True».

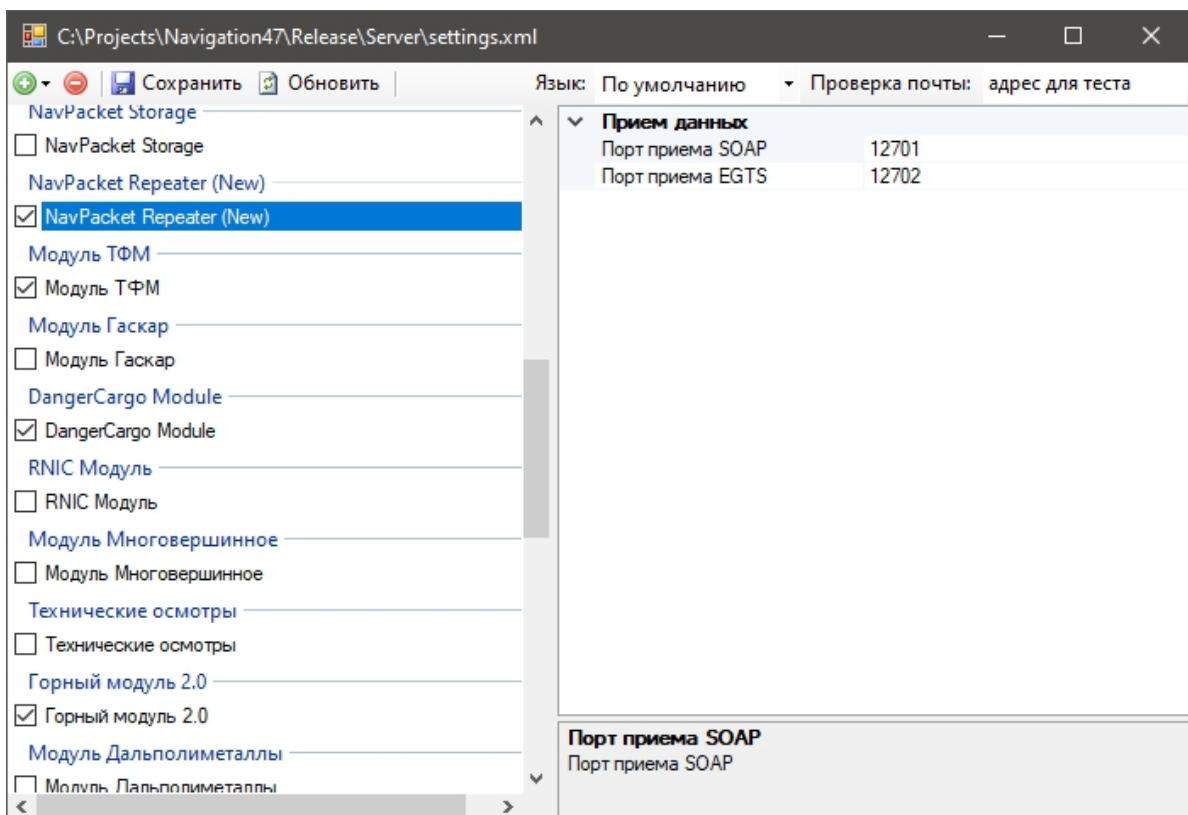


- Модуль «Retranslation Module» не имеет параметров для конфигурации. Все настройки производятся в «Мониторе работы сервера» из клиентского приложения администратором.

- В модуле «NavPacket Repeater» указываются порты для работы сервиса ретрансляции.

В протоколе SOAP по умолчанию порт 12700.

В протоколе EGTS по умолчанию порт 12701.



2.1.4. Установка службы

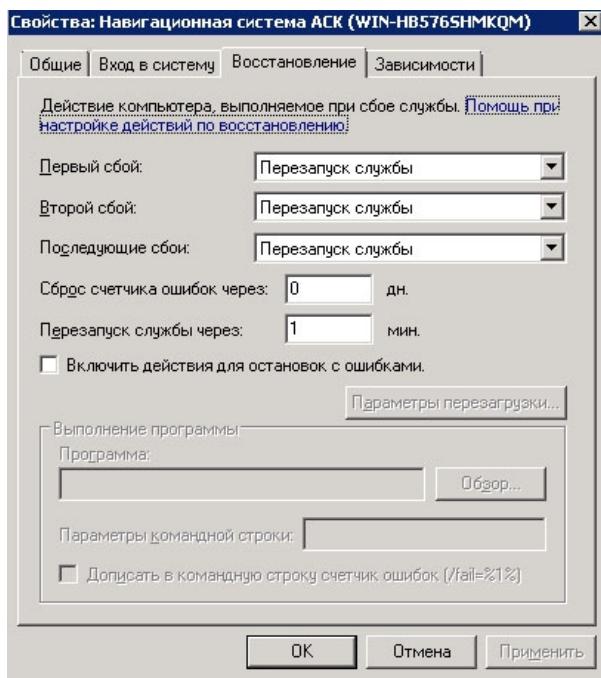
Для установки службы запустите файл «Install.bat» из рабочего каталога программы.

По завершению установки командная строка Windows выведет сообщение о статусе операции. Результатом успешной операции будет появление службы «Навигационная система ACK» в списке системных служб.

Службы					
Службы					
Имя	Описание	Состояние	Тип запуска	Вход	
Клиент отслеживания изменившихся с...	Поддерж...	Работает	Автоматиче...	Лока...	
Координатор распределенных транзак...	Координа...	Работает	Автоматиче...	Сете...	
Кеш шрифтов Windows Presentation Fo...	Оптимизи...		Бручную	Лока...	
Ловушка SNMP	Принимае...		Бручную	Лока...	
Локатор удаленного вызова процеду...	В Window...		Бручную	Сете...	
Маршрутизация и удаленный доступ	Предлага...		Отключена	Лока...	
Модули ключей IPsec для обмена клю...	Служба I...		Бручную	Лока...	
Модуль запуска процессов DCOM-серв...	Служба D...	Работает	Автоматиче...	Лока...	
Модуль поддержки NetBIOS через TC...	Осущест...	Работает	Автоматиче...	Лока...	
Модуль поддержки специальной конс...	Предоста...		Бручную	Лока...	
Модуль сервера SQL Server для запис...	Предоста...	Работает	Автоматиче...	Лока...	
Навигационная система ACK	Работает		Автоматиче...	Лока...	
Настройка сервера удаленных рабочи...	Служба н...	Работает	Бручную	Лока...	
Обнаружение SSDP	Обнаруж...		Отключена	Лока...	
Обнаружение интерактивных служб	Включает...		Бручную	Лока...	
Общий доступ к подключению к Инте...	Предоста...		Отключена	Лока...	
Определение оборудования оболочки	Предоста...	Работает	Автоматиче...	Лока...	
Основные службы доверенного платфо...	Разрешае...		Бручную	Лока...	
Перенаправитель портов пользователе...	Позволяе...	Работает	Бручную	Лока...	
Перечислитель IP-шин PnP-X	Служба п...		Отключена	Лока...	
Питание	Управляет...	Работает	Автоматиче...	Лока...	
Планировщик заданий	Позволяе...	Работает	Автоматиче...	Лока...	
Планировщик классов мультимедиа	Разрешае...		Бручную	Лока...	
Поддержка элемента панели управле...	Эта служ...		Бручную	Лока...	
Политика удаления смарт-карт	Позволяе...		Бручную	Лока...	
Поставщик результирующей политик...	Сетевая ...		Бручную	Лока...	
Проводная автонастройка	Служба ...		Бручную	Лока...	
Программный поставщик теневого ко...	Управляет...		Бручную	Лока...	
Публикация ресурсов обнаружения ф...	Публикуе...		Бручную	Лока...	
Рабочая станция	Создает ...	Работает	Автоматиче...	Сете...	

Убедитесь, что служба запущена, порты настроенные на прием данных доступны с внешних ip-адресов.

В свойствах службы на вкладке «Восстановление» укажите поведение при сбое в работе.



2.1.5. Установка и настройка сервера телеметрии

Сервер телеметрии служит для первичной обработки и хранения данных поступающих с абонентских терминалов и последующей передачи их на сервера приложений.

Рекомендации по составу аппаратных средств формируются разработчиком системы согласно предполагаемому количеству объектов мониторинга.

Для корректной работы сервера рекомендуется использовать операционную систему семейства Linux (Debian, Ubuntu актуальных версий).

Сервер телеметрии использует СУБД PostgreSQL 8 или выше и виртуальную машину Java версии 8 или выше.

1. Установим mc и sudo.

```
apt install mc
```

Apt	install	sudo
-----	---------	------

2. Установка Java.

Пример для Debian Linux:

```
# apt-get install openjdk-7-jdk
```

Убедиться что Java установлена:

```
# java -version
```

3. Установка СУБД PostgreSQL.

Пример для Debian Linux:

```
# apt-get install postgresql
```

Убедитесь что СУБД установлена:
 Заходим от пользователя postgres поробуем su postgres
 su – postgres (если «-» лииний) поробуем su postgres
 Откроем командный интерфейс psql

(# *psql -version*) –

4. Настройка СУБД

В базе данных создаем пользователя, например, «navigation» с паролем «12345»
psql#

CREATE USER navigation WITH password '12345';

Если база будет хранится на другом жёстком диске с PGSql , необходимо в конфиге ПГ изменить инстанс БД на нужный.
 Создаем базу данных «db_navigation»

psql#

CREATE DATABASE db_navigation OWNER navigation;

Даем пользователю права «superuser»:

psql#

ALTER USER navigation SUPERUSER;

\q

Из предоставленного дистрибутива восстанавливаем дамп базы данных.
 Выполняем скрипт createdb.sql из под пользователя *postgres@telemetry* , для этого:
 1) Копируем файл со скриптом на сервер
 2)Запускаем скрипт из свежескопированного файла

#*psql db_navigation <*директория файла* createdb.sql*

Убедитесь, что сервер PostgreSQL доступен с сервера приложений и рабочих мест администраторов системы (порт TCP по умолчанию 5432). Если порт не доступен внесите необходимые настройки в СУБД и firewall.

Для этого нужно отредактировать 2 файла, а именно (выделить файл в тс и нажать F4):

1) /etc/postgresql/postgresql.conf

находим раздел # CONNECTIONS AND AUTHENTICATION в параметре #listen_addresses убираем знак # и заменяем «localhost» на «*» должно получится listen_addresses = *
сохраняем клавишей F2

2) /etc/postgresql/pg_hba.conf
в самом конце файла находим параметр # IPv4 local connections:
зменяем в ADDRESS 127.0.0.1/32 на 0.0.0.0/0
и в METHOD trust на md5
сохраняем файл, выходим (из окна)

#(/var/lib/postgresql/9.4/main) – либо тут (имеет более низкий приоритет)

После внесения этих изменений, перезапускаем службу из под ruta выполняем команду: chmod 77 (777)

5. Дистрибутив

Распакуйте дистрибутив с программой в рабочую директорию. Заходим в сам архив, с правой стороны мы командера выбираем каталог куда будем разархивировать, выделяем по 1му файлы и жмём F5 (копировать).

Убедитесь, что пользователь имеет полный доступ к рабочему каталогу, файл «ask» - исполняемые.

6. Настройка сервиса

Откройте для редактирования файл server.conf.

Укажите в нем следующие параметры:

Путь к основному журналу

log= / путь к рабочей директории/log/CtrlServer.log

Как правило /home/ask/CtrlServer/log/CtrlServer.log

Путь к журналу ошибок

err_log=/путь к рабочей директории/log/Errors.log

Как правило /home/ask/CtrlServer/log/Errors.log

Путь к прошивкам

firmwares=/ путь к рабочей директории/firmwares/

Настройка TCP и UDP портов для обработчиков входящей информации

по типам приборов,

```

# DETECT – для приборов, на которые не выделяется индивидуальный порт
receiver1_port=12300
receiver1_type=DETECT
receiver2_port=12301
receiver2_type=NAVIXY
receiver3_port=12302
receiver3_type=APLICOM
receiver4_port=12303
receiver4_type=MIRCOM
# Locarus
udpreceiver1_port=12300
udpreceiver1_type=LOCARUS
# Autosat
udpreceiver2_port=50000
udpreceiver2_type=AUTOSAT

# Порт терминала ServerControl
receiver5_port=12500
receiver5_type=DISP

# Настройки базы данных
# Код типа СУБД. (PG – PostgreSQL)
ds=PG
# ip – адрес сервера СУБД
ds_server=127.0.0.1
# порт сервера
ds_port=5432
# Название базы данных
ds_db=db_navigation
# пользователь СУБД
ds_user=navigation
# пароль
ds_pass=12345

# Терминал
# имя пользователя терминала ServerControl
tm_user=user
# пароль для доступа к терминалу
tm_pass=password

```

Откройте для редактирования файл «ask» (без расширения)
Отредактируйте пути к рабочим директориям в начале файла.

7. Запуск

сервиса

Подложить файл лицензии в рабочий каталог

В случае если файл ask (без расширения, который) не исполняемый (при попытке ввода следующей команды выдаст: bash: ./ask: Отказано в доступе) следует: Выполнить в директории с сервером следующую команду:

```
chmod +x ask
```

Запустите сервис, выполнив из рабочей директории

```
# ./ask start
```

Система вернет идентификатор процесса.

Убедитесь, что сервис работает, просмотрев лог, или выполнив команду:

```
# ./ask log
```

Для удобства внесите путь к рабочему каталогу в системную переменную PATH и настройте автозапуск сервиса в файл .bashrc (в домашней директории пользователя) добавить следующие строки:

```
export PATH="$PATH:/рабочий каталог сервиса"
```

```
ASK="/home/home/CtrlServer/"
```

```
export PATH="$PATH:$ASK"
```

```
ask start
```

Убедитесь, что сервис запущен, порты, настроенные на прием данных доступны с внешних ip-адресов, порт терминала и порт СУБД доступны с рабочих мест администраторов и с сервера приложений.

Используйте руководство по ServerControl для заведения абонентских терминалов в систему.

8. Резервное копирование.

Остановите сервис телеметрии командой.

```
# ask stop
```

Для резервирования данных выполните дамп базы данных командой

```
# pg_dump db_navigation > /путь к каталогу/имя_файла.sql
```

Для восстановления данных выполните команду:

```
#psql db_navigation < /путь к каталогу/имя_файла.sql
```

Таким образом, резервное копирование данных происходит средствами СУБД. Разворнутую информацию по восстановлению и резервированию данных в своей версии PostgreSQL вы можете получить из справочной документации, например для версии 9.1.

По окончании процедуры резервирования или восстановления запустите сервис командой:

```
# ask start
```

3. Запуск и проверка работоспособности Системы

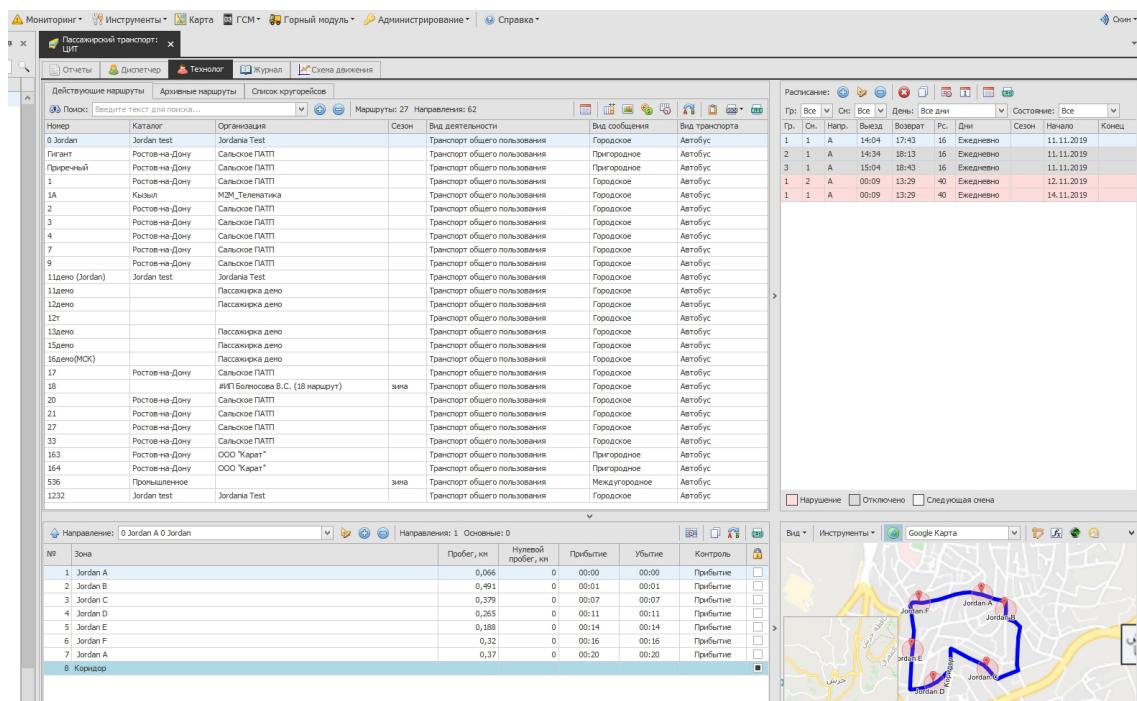
3.1. Запуск программного обеспечения

Для начала работы с Программным обеспечением необходимо произвести следующие операции:

- откройте десктопную версию АСК-Пассажирский транспорт;
- дождитесь загрузки окна авторизации;
- введите логин и пароль в соответствующие поля:
 - логин – admin;
 - пароль – admin;
- нажмите кнопку «Войти».

3.2. Проверка работоспособности Системы

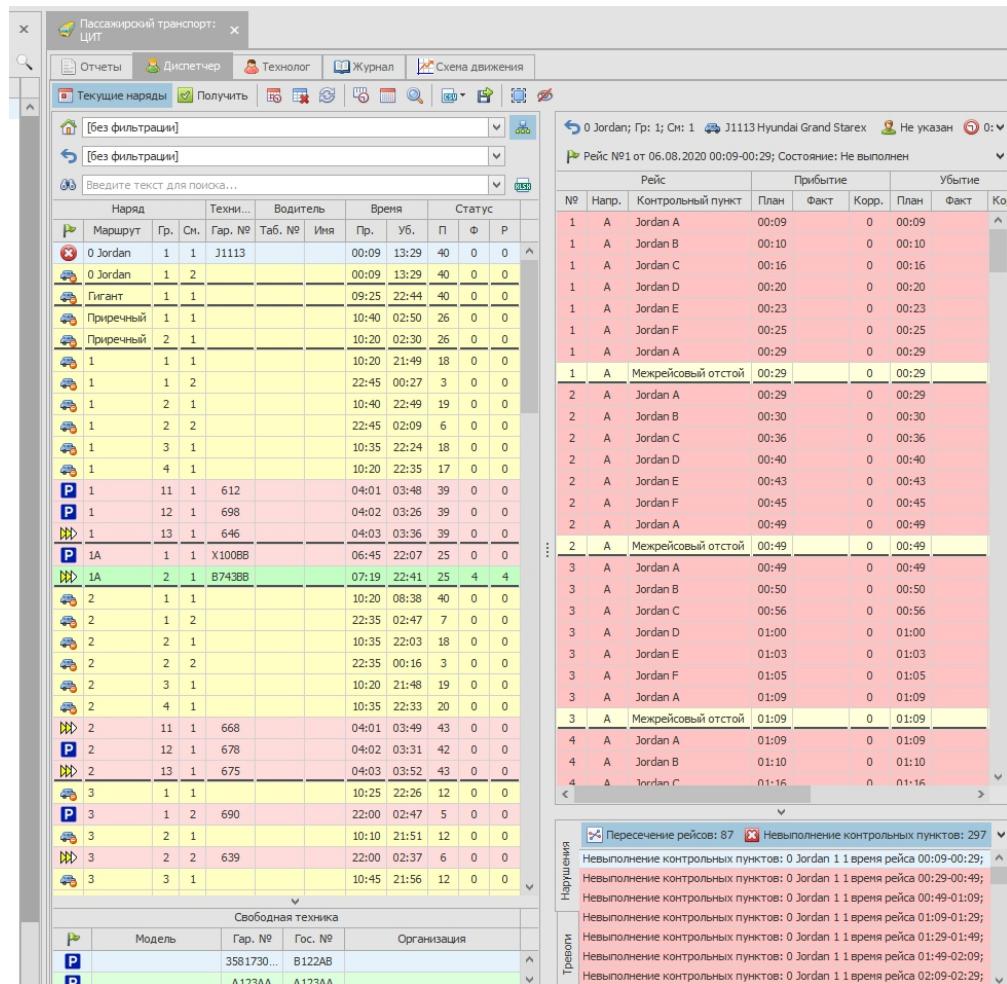
После авторизации и входа в Программное обеспечение на экране монитора отобразился раздел «Технолог» (Рисунок 1)



(Рисунок 1)

Используя вертикальное меню, эксперт также может протестировать некоторые функциональные возможности Системы.

Раздел «Диспетчер», посмотреть работу данного раздела (Рисунок 2)



(Рисунок 2)

Полное тестирование Программного обеспечения проводится Заказчиком после установки согласно документу «Описание установки Программного обеспечения» и документу «Программа и методика испытаний», входящим в комплект поставки Программного обеспечения.