

**«АСК БЕТОН»**  
**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  |
| 2.2. Установка дистрибутива системы.....             | 8  |
| 2.3. Установка сервера .....                         | 9  |
| 2.4. Установка службы .....                          | 12 |
| 3. Установка и настройка сервера телеметрии .....    | 14 |
| 3.1. Установим mc и sudo .....                       | 14 |
| 3.2. Установка Java .....                            | 14 |
| 3.3. Установка СУБД PostgreSQL .....                 | 14 |
| 3.4. Настройка СУБД PostgreSQL .....                 | 14 |
| 3.4.1. Дистрибутив .....                             | 15 |
| 3.4.2. Настройка сервиса.....                        | 15 |
| 3.4.3. Запуск сервиса .....                          | 17 |
| 3.4.4. Резервное копирование.....                    | 17 |
| 4. Запуск и проверка работоспособности Системы ..... | 18 |
| 4.1. Запуск программного обеспечения.....            | 18 |
| 4.2. 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).

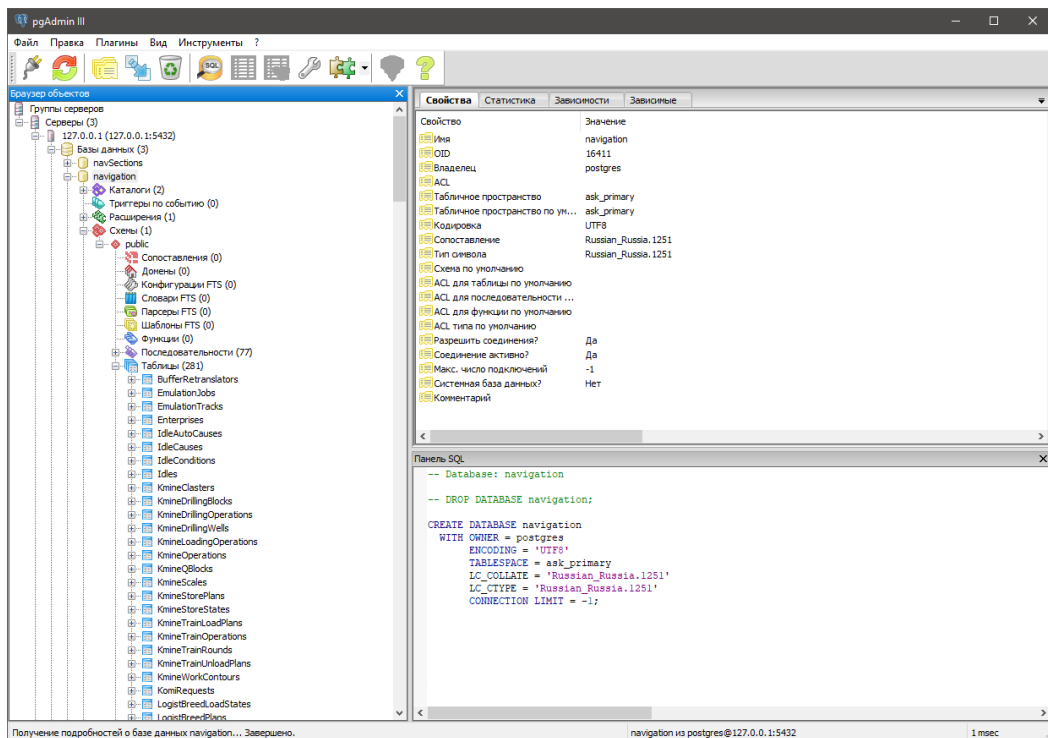
Сервер приложений использует СУБД PostgreSQL 10.10 или новее.

### 2.1. Установка и настройка базы данных

#### 2.1.1. Настройка СУБД PostgreSQL

Установка СУБД выполняется со стандартными параметрами.

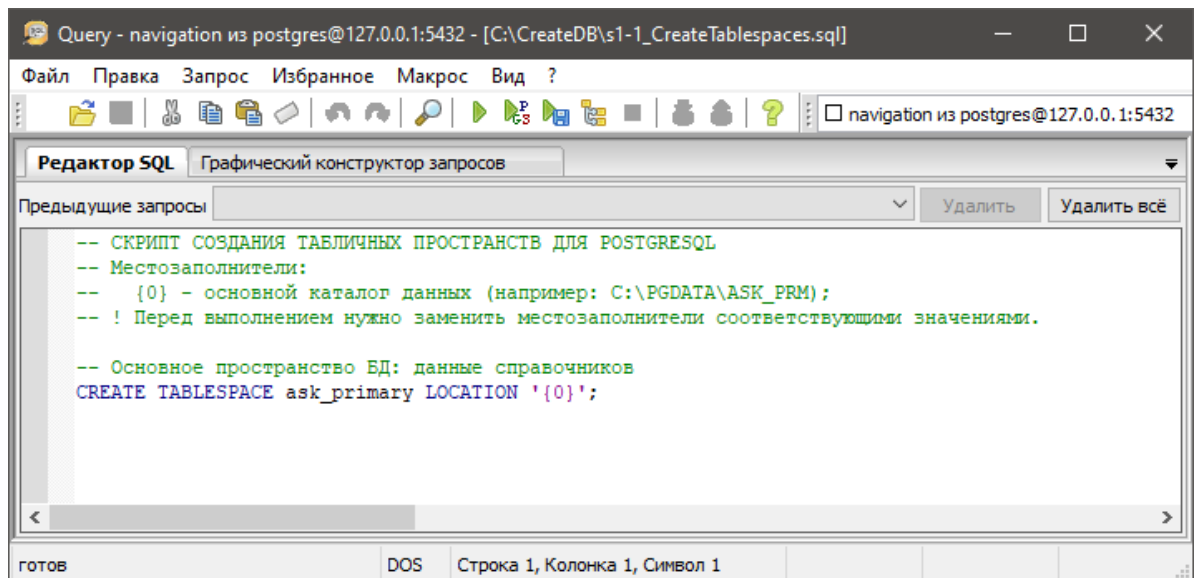
Создание базы выполняется пользователем postgres.



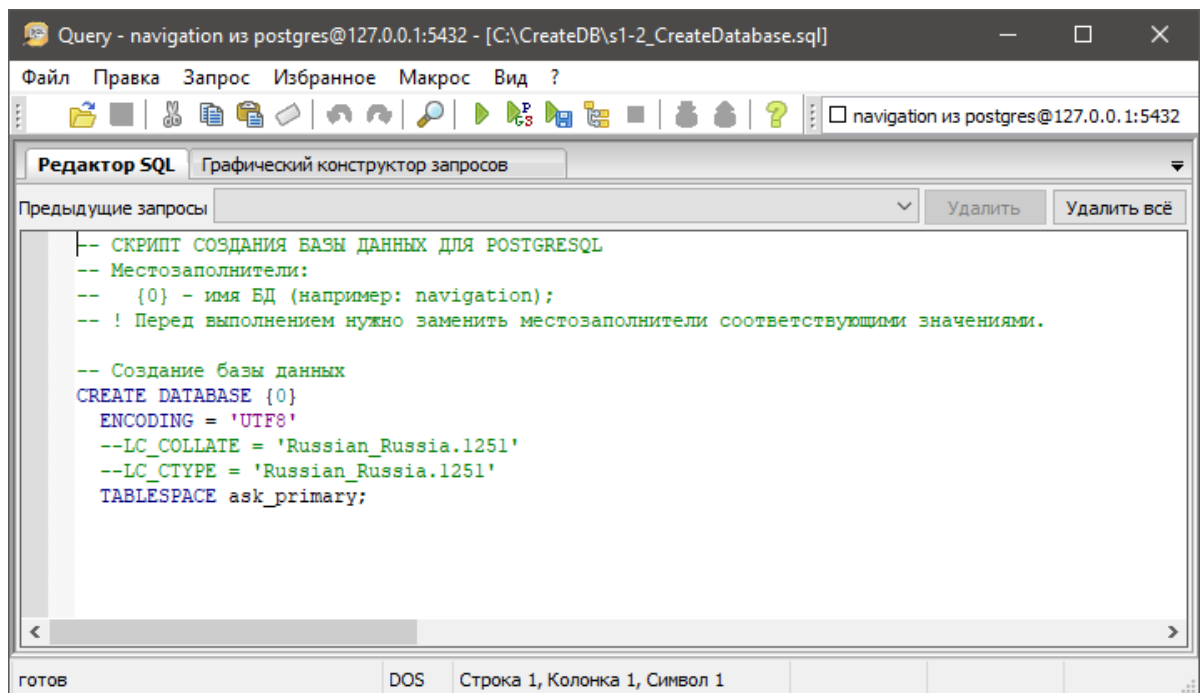
#### 2.1.2. Создание базы PostgreSQL

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

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



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



«s2\_CreateTables.sql» - создание базы данных.

The screenshot shows a PostgreSQL SQL Editor window titled "Query - navigation из postgres@127.0.0.1:5432 - [C:\CreateDB\s2\_CreateTables.sql]". The window contains the following SQL script:

```
-- СКРИПТ СОЗДАНИЯ ТАБЛИЦ ДЛЯ POSTGRESQL
-- Местоаполнители: 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,
```

The status bar at the bottom indicates "готов" (ready), "DOS", and "Строка 1, Колонка 1, Символ 1" (Line 1, Column 1, Symbol 1).

«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_PFM);
-- {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.

```

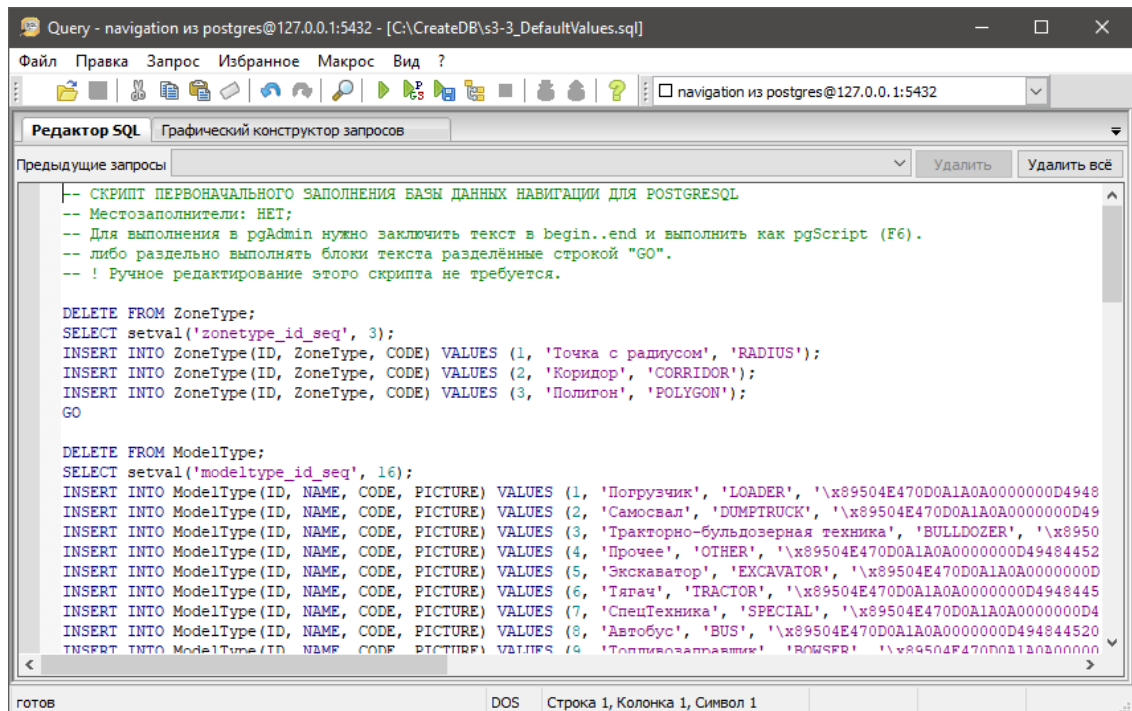
-- СКРИПТ ПЕРВОНАЧАЛЬНОГО ЗАПОЛНЕНИЯ ТАБЛИЦ РОЛЕЙ И ПОЛЬЗОВАТЕЛЕЙ ДЛЯ POSTGRESQL
-- Местозаполнители:
-- {0} - UUID администратора;
-- {1} - UUID системного пользователя;
-- Для выполнения в pgAdmin нужно заключить текст в begin..end и выполнить как pgScript (F6).
-- либо раздельно выполнять блоки текста 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, Trails) VALUES (2, 'Администратор', 'ADMIN', TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE);
INSERT INTO Roles(ID, NAME, CODE, Disp, Adm, Map, RepairCauses, Repairs, Reserves, Fuel, Workers, Waysheets, Trails) VALUES (3, 'Система', 'SYSTEM', TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE);
INSERT INTO Roles(ID, NAME, CODE, Disp, Map, UsersRead, Workers, Reserves, Repairs, RepairCauses, Fuel, Waysheets, Trails) VALUES (4, 'Пользователь', 'USER', TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE);
INSERT INTO Roles(ID, NAME, CODE) VALUES (5, 'Гость', 'VIEWER');
GO

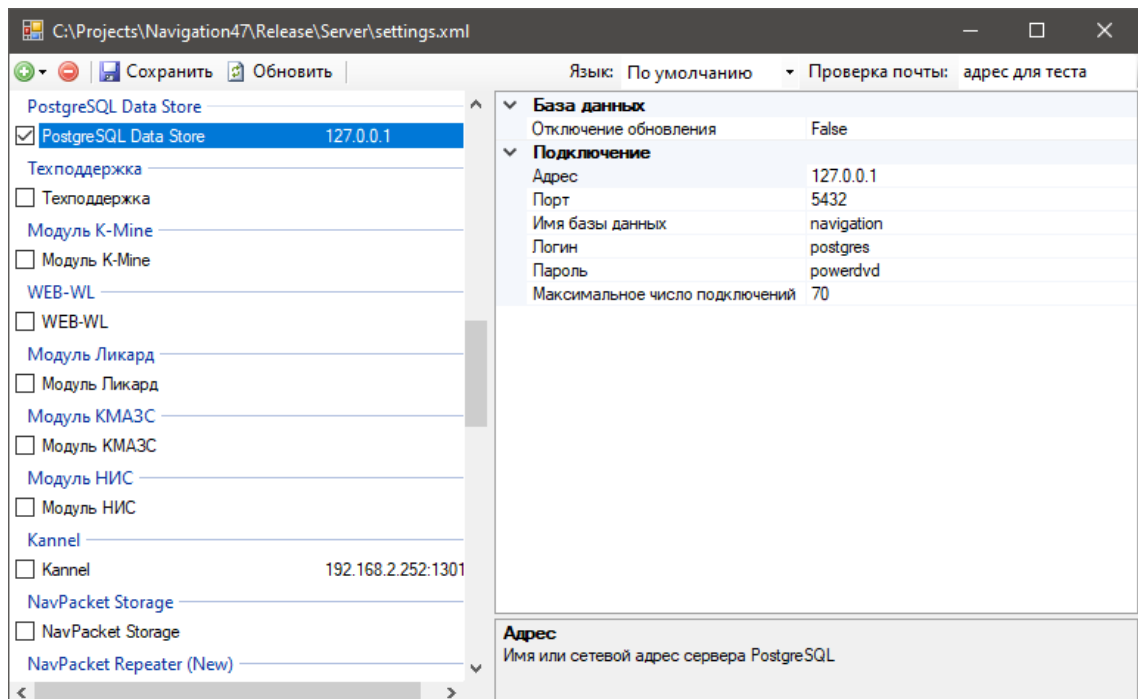
-- ПЕРЕД ВЫПОЛНЕНИЕМ ВСТАВКИ НУЖНО ЗАДАТЬ UUID ДЛЯ ПОЛЬЗОВАТЕЛЕЙ
DELETE FROM Users;
INSERT INTO Users(ID, Name, Login, Role, Password, Email, Phone, StartTime) VALUES ({0}, 'Администратор', 'admin', 'ADMIN', 'admin', 'admin@navSections.ru', '880000000000', '2000-01-01 00:00:00');
INSERT INTO Users(ID, Name, Role, Password, Email, Phone, StartTime) VALUES ({1}, 'Система', 1, ('md5' || md5('{0}')), 'system@navSections.ru', '880000000000', '2000-01-01 00:00:00');
GO

```

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



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.

## 2.2. Установка дистрибутива системы

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



### 2.3. Установка сервера

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

Добавление модулей происходит выбором из выпадающего списка при нажатии кнопки:



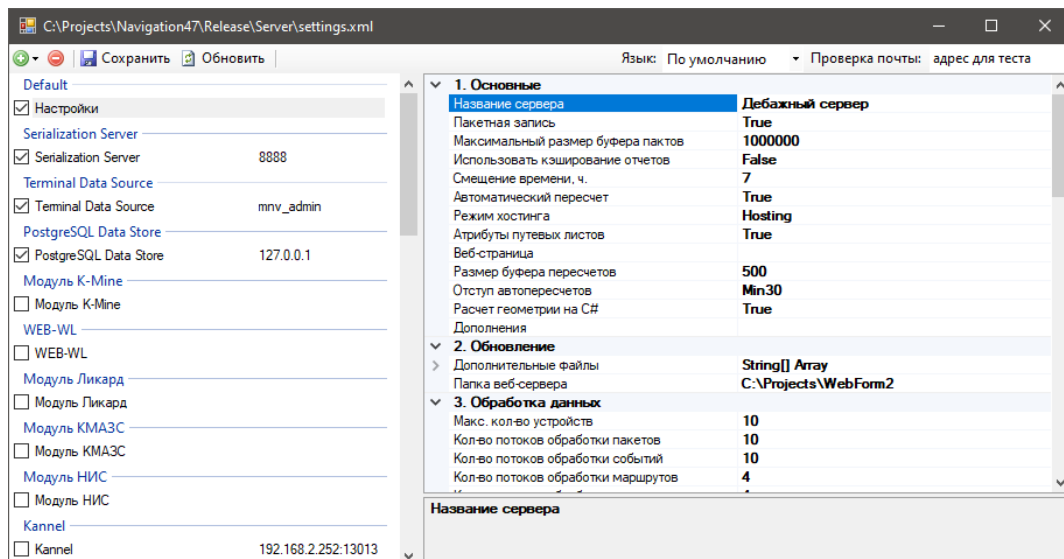
Сервер приложений может работать с несколькими серверами телеметрии, в этом случае добавляется несколько одинаковых модулей и выполняется их конфигурация.

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.

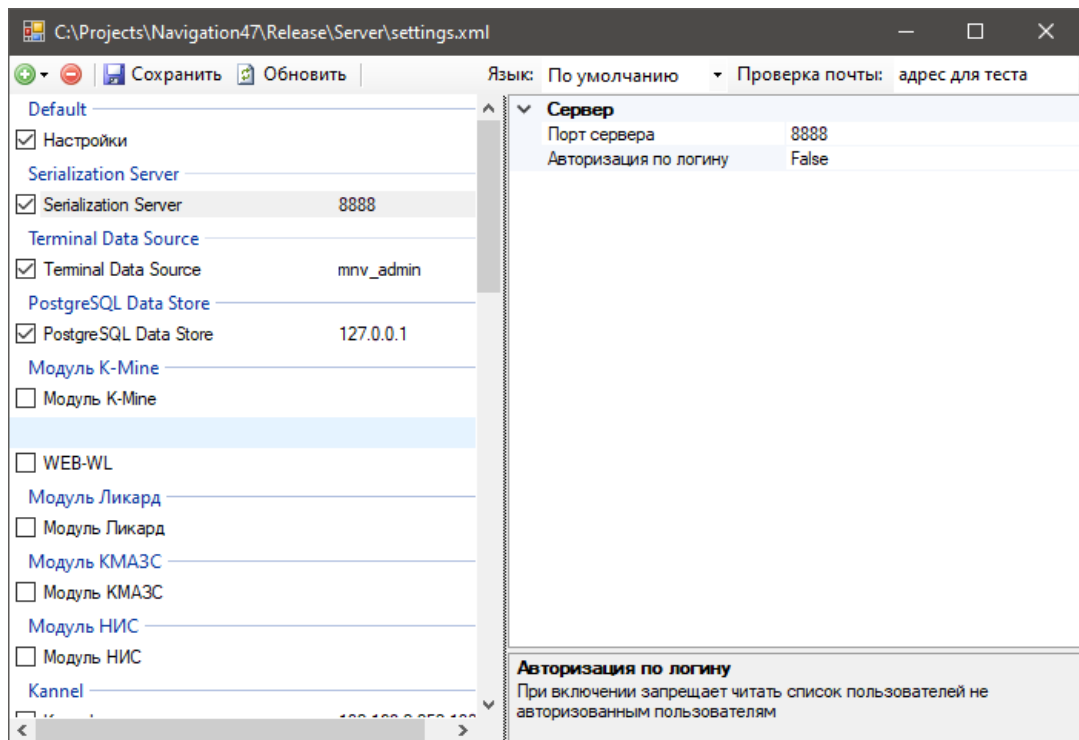
Укажите каталог клиентского приложения.

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

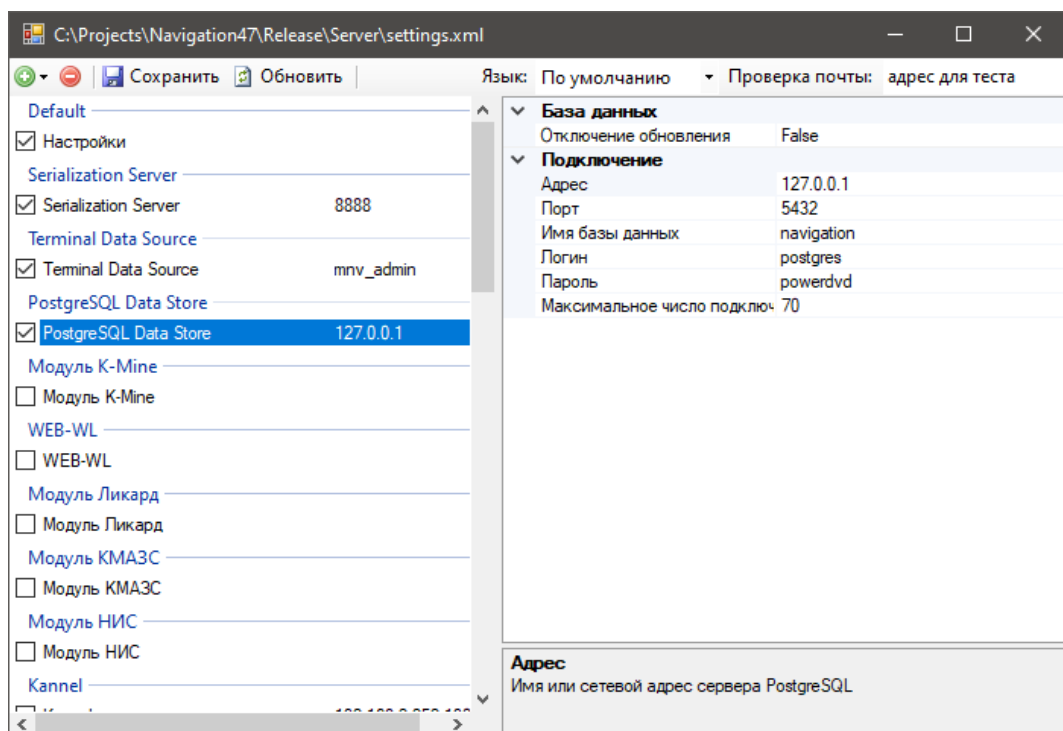


- На вкладке «Serialization Server» укажите TCP-порт по которому клиенты будут подключаться к навигационному серверу.

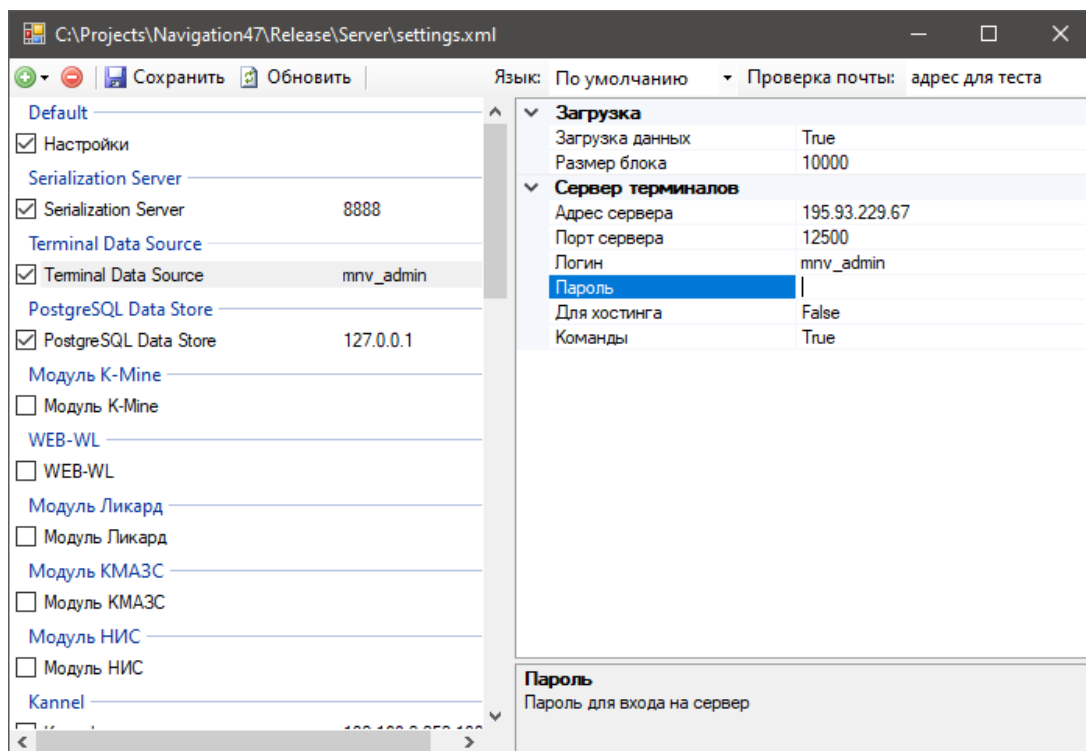
Определите как клиенты будут авторизоваться на сервере: выбирая свою учетную запись из выпадающего списка (авторизация по логину – False), либо вводя логин в форму авторизации (авторизация по логину – True).



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



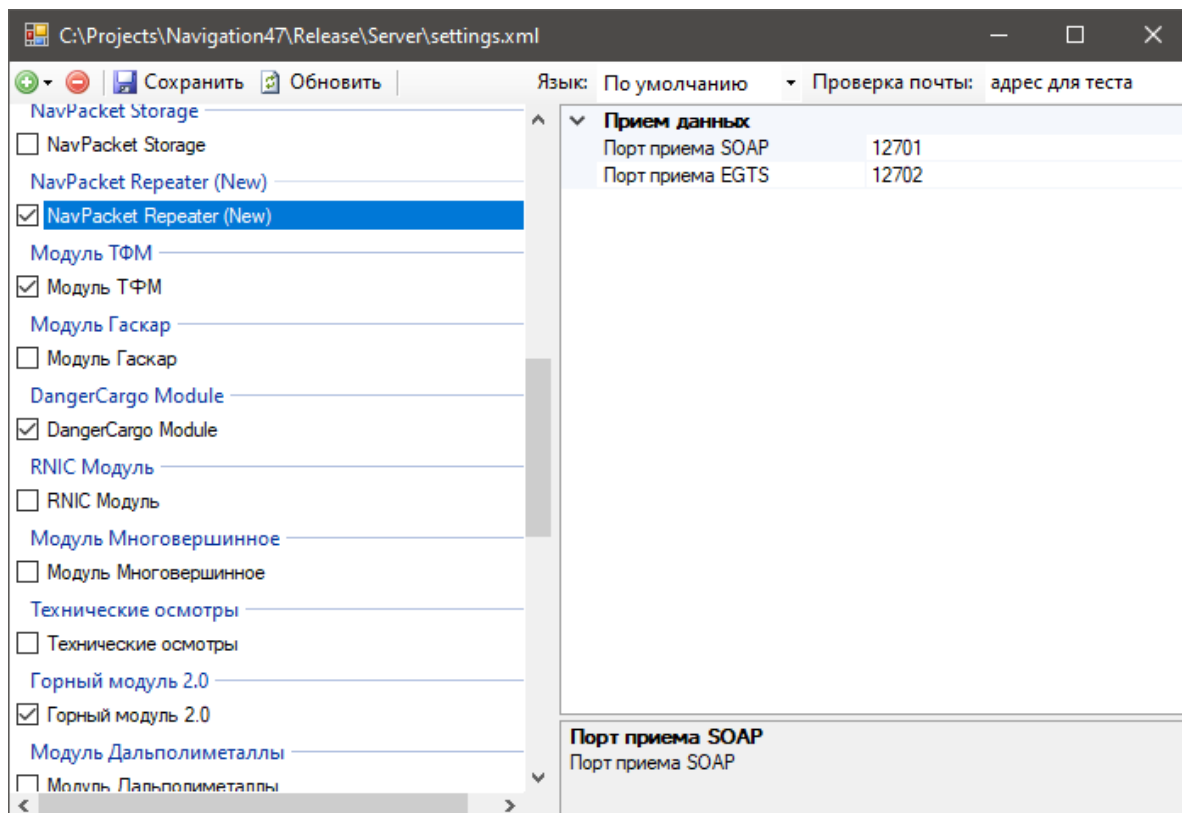
- Модуль «Terminal Data Source» позволяет сконфигурировать подключение к серверу телеметрии. Укажите адрес сервера, порт, логин и пароль для подключения. Для загрузки данных выберите параметр «True».



- 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.

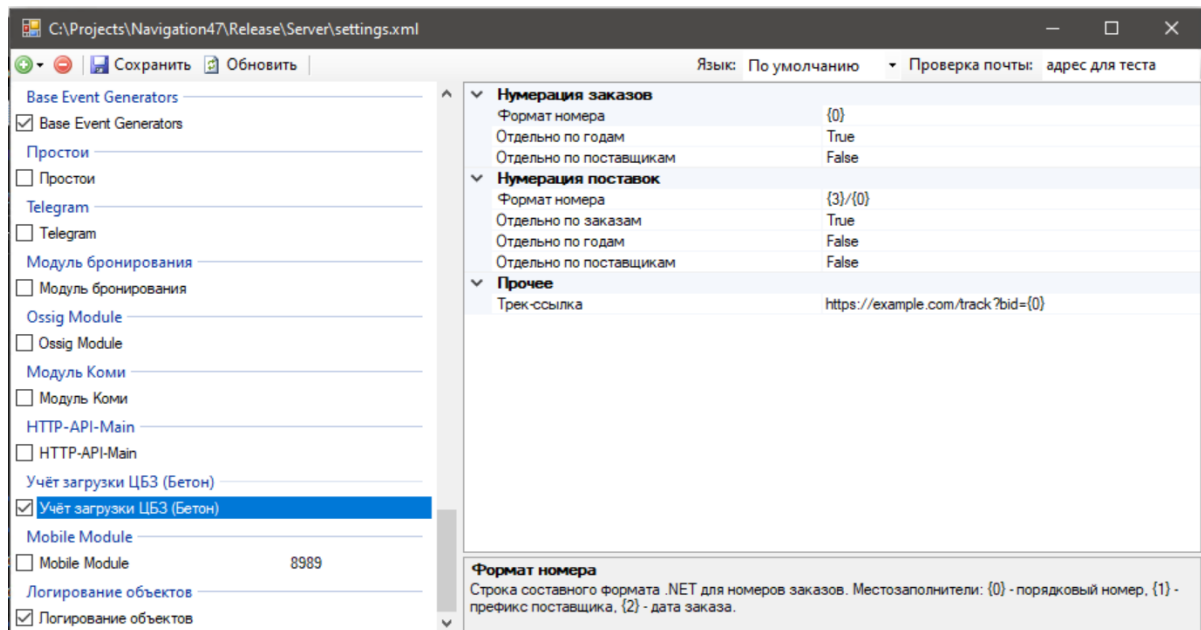
В протоколе SOAP по умолчанию порт 12700.

В протоколе EGTS по умолчанию порт 12701.



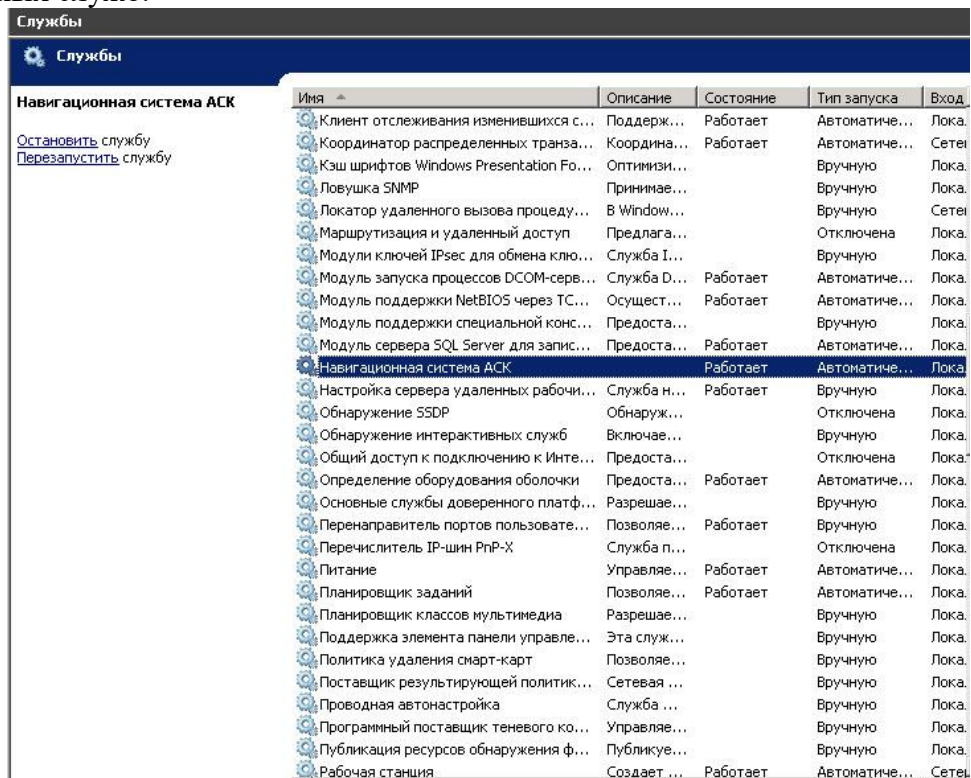
- Модуль «Учет загрузки» не требует специальных настроек для работы.

В нем можно явно указать формат номеров заказов и поставок, а так же формат трек-ссылки для отслеживания состояния заказов на веб-ресурсе (не входит в состав данного ПО).



## 2.4. 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 АСК» в списке системных служб.



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.

The image shows a Windows-style dialog box titled "Свойства: Навигационная система ASK (WIN-HB5765NMQM)". It has four tabs: "Общие", "Вход в систему", "Восстановление" (selected), and "Зависимости".

Under the "Восстановление" tab, there is a text label: "Действие компьютера, выполняемое при сбое службы. [Помощь при настройке действий по восстановлению!](#)".

Below this are three dropdown menus for recovery actions:

- Первый сбой: [Перезапуск службы]
- Второй сбой: [Перезапуск службы]
- Последующие сбои: [Перезапуск службы]

There are two time input fields:

- Сброс счетчика ошибок через: 0 дн.
- Перезапуск службы через: 1 мин.

A checkbox is present: ☐ Включить действия для остановок с ошибками. To its right is a button labeled "Параметры перезагрузки...".

Below these is a section titled "Выполнение программы" containing:

- A text field labeled "Программа:" followed by an "Обзор..." button.
- A text field labeled "Параметры командной строки:".
- A checkbox labeled "Дописать в командную строку счетчик ошибок (/fail=%1%)".

At the bottom of the dialog are three buttons: "ОК", "Отмена", and "Применить".

### 3. 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.

#### 3.1. Установим mc и sudo

```
apt install mc
apt install sudo
```

#### 3.2. Установка Java

Пример для Debian Linux:

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

Убедиться что Java установлена:

```
# java -version
```

#### 3.3. Установка СУБД PostgreSQL

Пример для Debian Linux:

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

Убедиться что СУБД установлена:

*Заходим от пользователя postgres*

*su – postgres (если «-» лишний ) попробуем su postgres*

*Откроем командный интерфейс*

*psql*

*(# psql –version) –*

#### 3.4. Настройка СУБД PostgreSQL

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

*psql#*

*CREATE USER navigation WITH password '12345';*

**Если база будет храниться 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 дампы 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:

заменяем 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 рута выполняем команду:

```
chmod 77 (777)
```

### 3.4.1. Дистрибутив

Распакуйте дистрибутив с программой в рабочую директорию.

Заходим в сам архив, с правой стороны тс 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.

### 3.4.2. 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.

### 3.4.3. 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.

### 3.4.4. 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. Развернутую 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
```

## 4. Запуск и проверка работоспособности Системы

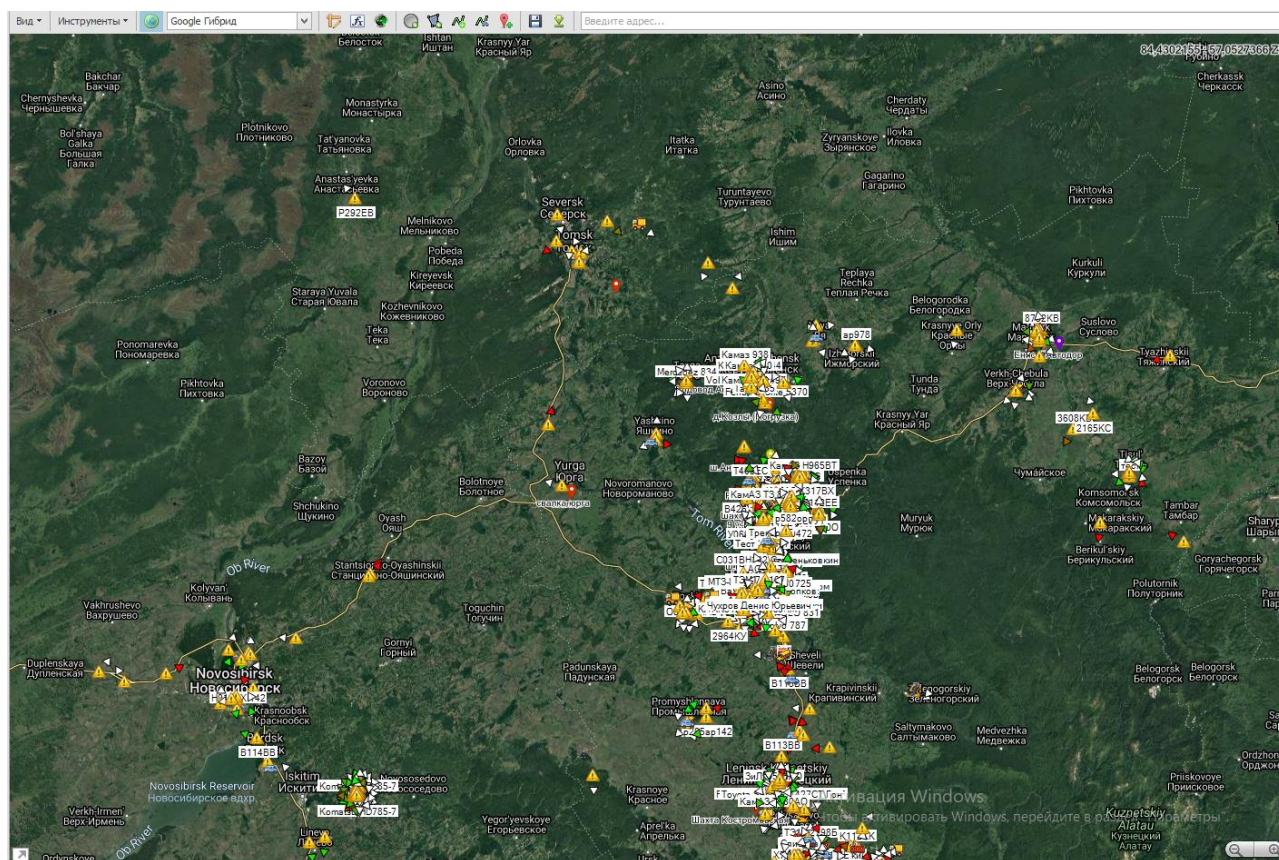
### 4.1. Запуск программного обеспечения

Для начала работы с Программным обеспечением необходимо произвести следующие операции:

- откройте клиентское приложение «АСК Бетон»;
- дождитесь загрузки окна авторизации;
- введите логин и пароль в соответствующие поля:
  - логин – admin;
  - пароль – admin;
- нажмите кнопку «Войти».

### 4.2. Проверка работоспособности Системы

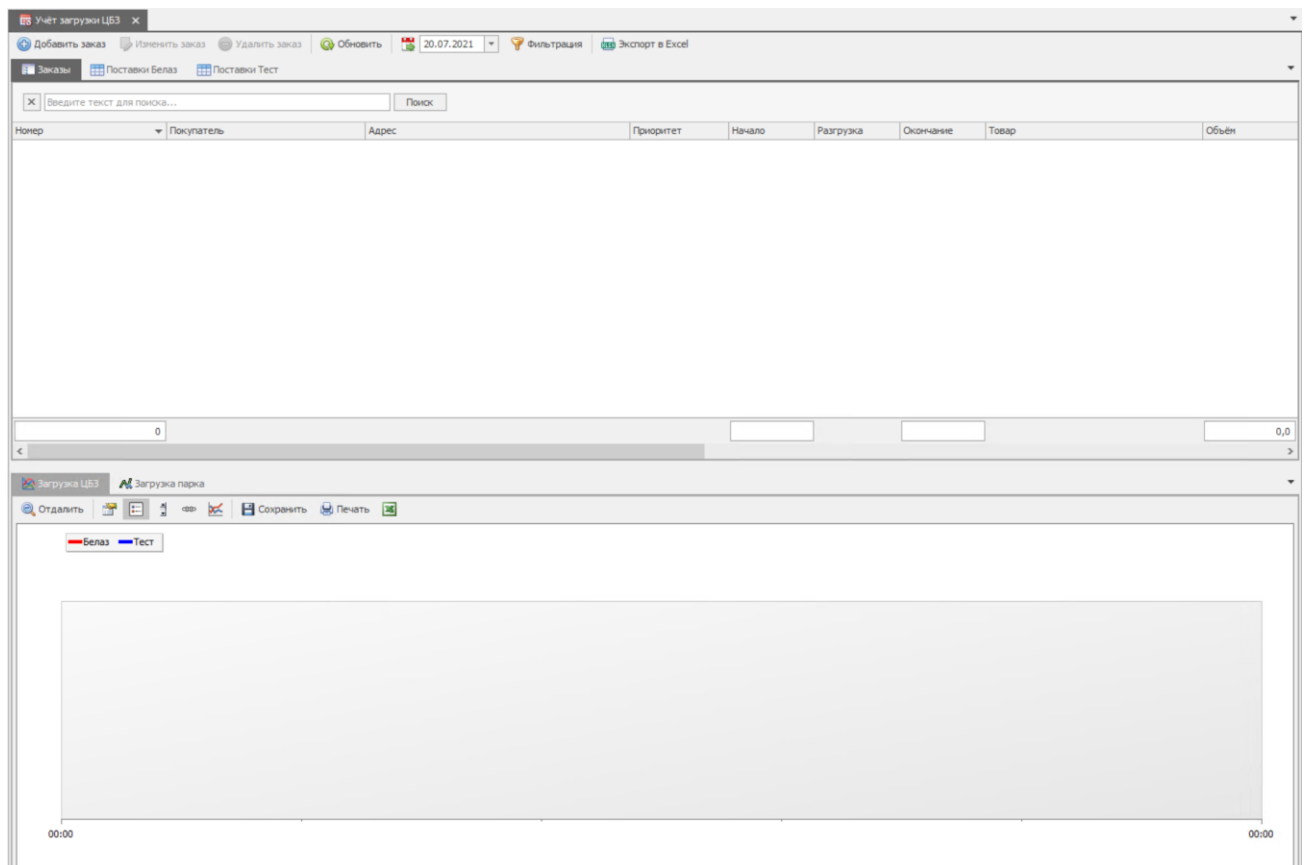
После авторизации и входа в Программное обеспечение подключаемся к серверу и выбираем раздел «Мониторинг», далее «Текущее положение» (Рисунок 1)



(Рисунок 1)

Добавив единицы техники, эксперт также может протестировать некоторые функциональные возможности Системы.

Для проверки работоспособности модуля «Бетон» открыть раздел «Инструменты» и выбрать «Учет загрузки».



(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.