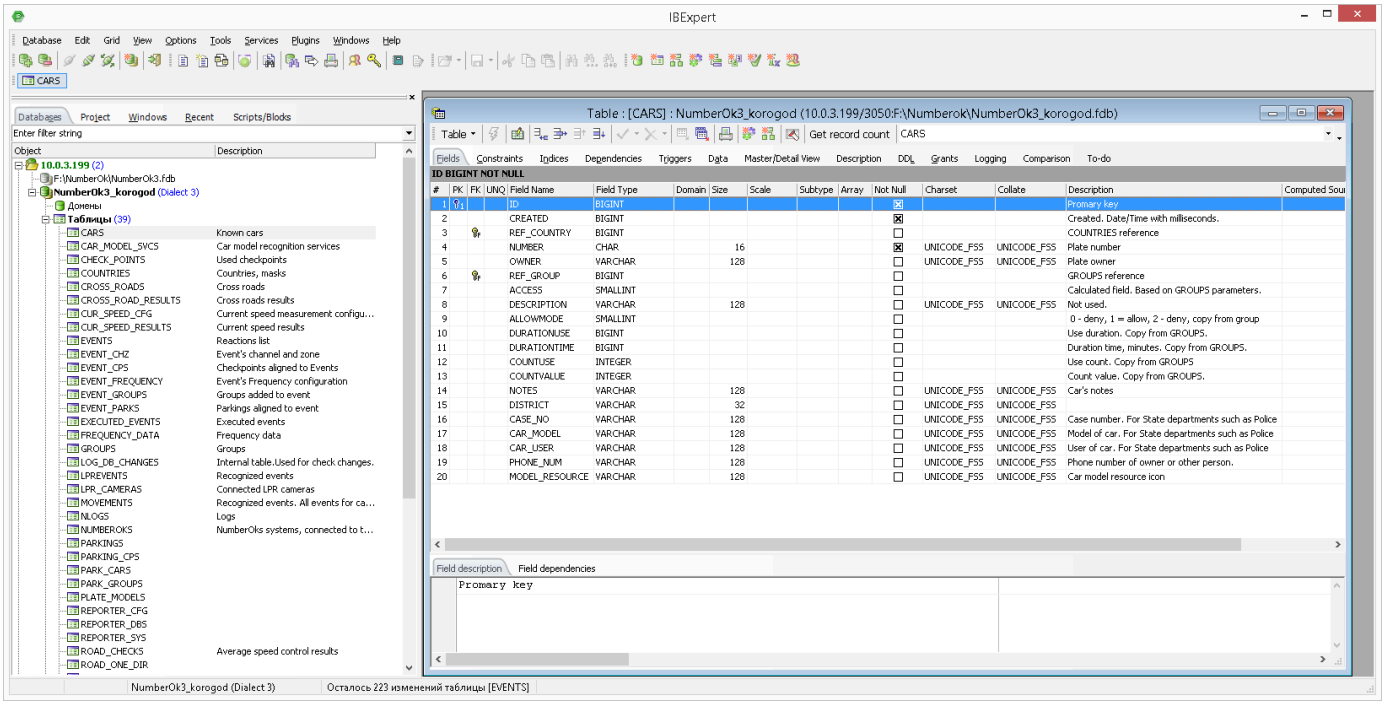
Managing NumberOK via DataBase

Updated:	**2018-02-15**
Latest version always at:	https://goo.gl/KHfmWx *

Intro



Managing NumberOK via FireBird database is considered an unsupported non-documented method.

The database is accessible from outside NumberOK and access to the database should be restricted on OS and file system levels.

NumberOk vendor doesn't take any responsibility for the software features malfunctioning due to improper data manipulations. NumberOK employs transactions to keep data consistent.

Updating Data

Whenever it is required to update database with a plenty of information, employing NumberOK native features (e.g. data import) is a recommended method.

Set up required groups via Car database > Groups and ACS.

The general recommended workflow to update license plates lists is as follows (under Car database > Vehicles):

1. Export to XLS
2. Amend licence plate lists as required
3. Import to XLS with configuration settings Standard structure + Update database.

Please, note that Replace database option will change groups identificators that will affect reactions and other access related features.

Manipulating data directly

There are a couple of relatively safe operations.
The database is located at `C:\ProgramData\FF\NumberOk3\NumberOk3.fdb` unless different location was indicated during NumberOK installation.
You will find DB access credentials in `C:\ProgramData\FF\NumberOk3\NumberOk.ini` file (`Database.FB_Password`, `Database.FB_Path`, `Database.FB_User`, `Database.FB_host`, `Database.FB_port`).
You may want to use one of officially recommended tools to explore the database.

Getting recognition events (including vehicle type)

To get all recognition events for today including next fields: timestamp, license plate, vehicle type and the path to the image used following SQL query:

```
SELECT timestamp_, number, car_type, framejpg
 FROM lprevents  WHERE
 timestamp_ >= DATEDIFF(SECOND, TIMESTAMP'1970-01-01 00:00', CURRENT_DATE) * 1000 AND
 timestamp_ < ( DATEDIFF(SECOND, TIMESTAMP'1970-01-01 00:00', CURRENT_DATE) + 86400 ) *
1000 ORDER BY 1;
```

Description
DATEDIFF(SECOND, TIMESTAMP'1970-01-01 00:00', CURRENT_DATE) - diff in seconds between UTC start date (1970/01/01 00:00) and current date 00:00. 86400 = 24 * 60 * 60 - offset to the next day.
Car type:
CAR=0x0001,BUS=0x0002,SUV=0x0004,VAN=0x0008,LCV=0x0010,TRUCK=0x0020,0=unrecognized

Adding and removing particular licence plates

Manage groups via NumberOK Car database > Groups and ACS.

For each license plate to add take a note of `GROUPS.ID` and `COUNTRIES.ID`.

To add license plate **ABC12D** belonging to **Mr Whyte** (**UK**, id is 23) associated with group id **3** :

```
INSERT INTO CARS (CREATED, REF_COUNTRY, NUMBER, OWNER, REF_GROUP, ACCESS, DESCRIPTION, ALLOWMODE, DURATIONUSE, DURATIONTIME, COUNTUSE, COUNTVALUE, NOTES, DISTRICT, CASE_NO, CAR_MODEL, CAR_USER, PHONE_NUM) VALUES (UTCTIMES() * 1000, 23, 'ABC12D', 'Mr Whyte', 3, 0, '', 1, 0, 1, 0, 1, '', '', '', '', '', '');
```

To remove a license plate **ABC12D**:

```
DELETE FROM CARS WHERE NUMBER = 'ABC12D';
```

Single time access

The example below assumes the following:

1) the car plate number will be removed after access expiration;
2) any single plate number has a single scheduled access grant at any given moment.

Adding a scheduled entry

Example task allow **ABC12D** (owned by **Mr Whyte**) to access the site once between 09:00 and 12:30 on **February 15, 2018**. 09:00 is 9*3600+0*60=**32400**, 12:30 is 12*3600+30*60=**45000**.

The idea behind is to create an ad hoc group with access schedule details and associate a particular license plate with it.

NB! The below is in isql notation.

```
SET TERM #;

EXECUTE BLOCK AS

  DECLARE groupid BIGINT;

BEGIN

  INSERT INTO GROUPS (GROUPNAME, ALLOWMODE, TIMETABLE_ENABLED, DFROM, TFROM, DTO, TTO, DDAYS, DURATIONUSE, DURATIONTIME, COUNTUSE, COUNTVALUE, DRANGE_ENABLED, TRANGE_ENABLED)

    VALUES ('WHYTE180215', 1, 1, 20180215, 32400, 20180215, 45000, 127, 0, 1, 1, 1, 1, 1) RETURNING ID INTO :groupid;

  INSERT INTO CARS (CREATED, REF_COUNTRY, NUMBER, OWNER, REF_GROUP, ACCESS, DESCRIPTION, ALLOWMODE, DURATIONUSE, DURATIONTIME, COUNTUSE, COUNTVALUE, NOTES, DISTRICT, CASE_NO, CAR_MODEL, CAR_USER, PHONE_NUM)
```

```
    VALUES (UTCTIMES() * 1000, 23, 'ABC12D', 'Mr Whyte', :groupid, 0, '', 1, 0, 1, 0, 1,
'', '', '', '', '', '');

END

#

SET TERM ;#
```

Removing the expired entry

```
DELETE FROM GROUPS WHERE ID = (SELECT REF_GROUP FROM CARS WHERE NUMBER = 'ABC12D');
```

All license plates associated with the group will also be removed.

The above SQL instruction can be improved to remove all groups where `` `GROUPS.DTO` `` is less than current date.

General note

Please, take the above and any other data manipulations at your own risk.